# TW2880C2

Multi-Channel Surveillance Camera Controller with VGA/HDTV Display Capability

FN7883
Rev. 2.00
June 4, 2013

## Summary

## Display Controller Features List

- Video Input from Decoder Interface
  - Supports 27 MHz BT.656 input
  - Supports 54,108 MHz BT.656 byte-interleaved input
  - Supports 4 BT.656 play back input in frame interleaved or field interleaved format
  - Supports 4 BT.1120 play back input with FPGA
  - Motion Box (MD) on all display channels
- Supports 1080p HDTV and VGA Display
  - Horizontal size: 640 ~ 1920, Vertical size: 480 ~ 1080
  - Max. pixel clock: 174 MHz (if MCLK > 166 MHz)
  - Vertical Refresh frequency: Flexible, can be programmed as 50 or 60 Hz
  - Flexible output interface, Analog VGA (HD15), Digital VGA (DVI) and HDMI (V1.2)
  - Capable of displaying up to 33 Channels in one screen, logo, live video or playback combined
  - User selectable Weave/2D/3D de-interlacing method for video quality enhancement
  - Up / down scaler for arbitrary windows size
  - Three layers of OSG overlay with High-color (16 bit) bit map
  - Play back channel input can be used as an external OSG layer
  - OSD display layer for title and channel ID
  - Single Box Display for highlighting and Mouse / Cursor Overlay
- System clock at 108MHz, Memory clock 166 - 180 MHz
- I²C and parallel MCU host interface

## Recording Controller Features List

- High flexibility record output function
  - Real time / non real time field interleaved, frame interleaved output
  - Priority recording
  - Multi-mode output controlled by user
- High flexibility SPOT output function
  - Four individual output
  - Full / Half / CIF display with OSD
  - Automatic switching
- Random sequence recording (Table recording) function
- Motion Detector & Channel ID Encoding / Decoding
- Embedded font-RAM for color OSD
- Split border and background color setting
- Four privacy windows on every channel
- Four SPOT with flexible channel arrangement

# Ordering Information

| PART NUMBER | PART MARKING | PACKAGE (Pb-free) |
|---|---|---|
| TW2880P-BC2-GR (Note 1) | TW2880P PKBC2-GR | 676 Pin BGA |
| TW2880N-BC2-GR (Note 1) | TW2880N PKBC2-GR | 676 Pin BGA |

NOTE:

1. These Intersil Pb-free WLCSP and BGA packaged products employ special Pb-free material sets; molding compounds/die attach materials and SnAgCu - e1 solder ball terminals, which are RoHS compliant and compatible with both SnPb and Pb-free soldering operations. Intersil Pb-free WLCSP and BGA packaged products are MSL classified at Pb-free peak reflow temperatures that meet or exceed the Pb-free requirements of IPC/JEDEC J STD-020.

# Table of Contents

Summary ..... 1
Display Controller Features List ..... 1
Recording Controller Features List ..... 1
Ordering Information ..... 2
Table of Contents ..... 3
Display Format ..... 36
System Diagram ..... 37
Ball Description ..... 38
Ball Diagrams ..... 50
AC/DC Electrical Parameters ..... 55
Absolute Maximum Ratings ..... 55
Recommended Operating Conditions ..... 55
Supply Current and Power Dissipation ..... 55
PLL AC Parameters ..... 56
DC Characteristics ..... 56
Analog Performance Parameter ..... 57
3.3V CMOS Output I/O, M-type V-I Characteristics ..... 57
Chip Package ..... 58
Different Versions of TW2880 ..... 58
Video Interface and Down Scaler ..... 59
Digital Video Input ..... 59
Byte-interleaved Format ..... 60
Down Scaler for Live Inputs ..... 60
AC Timing for Live and PB Input ..... 60
Register Table ..... 61
Registers ..... 70
Live Video Channel 1 Horizontal Scaler Low Byte Register – 0x300 ..... 70
Live Video Channel 1 Horizontal Scaler High Byte Register – 0x301 ..... 70
Live Video Channel 1 Vertical Scaler Low Byte Register – 0x320 ..... 70
Live Video Channel 1 Vertical Scaler High Byte Register – 0x321 ..... 70
Live Video Vertical Scale Mode Register – 0x340 ..... 71
Live Video Horizontal Scale Mode Register – 0x348 ..... 71
Live Video PAL Delay Mode Register – 0x350 ..... 71
Live Video Odd Field Skew Register – 0x352 ..... 71
Live Video Vertical Scaler LPF Register – 0x354 ..... 72
Live Video Vertical Scale VAV Test Register – 0x356 ..... 72
Live Video Channel 1 Horizontal pixel adjustment Register – 0x375 ..... 72
Live Video Channel 1 Horizontal pixel adjustment Register – 0x385 ..... 72
Live Video Channel 1 Vertical pixel adjustment Register – 0x395 ..... 72
Live Video Channel 1 Vertical pixel adjustment Register – 0x3A5 ..... 73
Live Video Channel 1~4 No Video Register – 0x3BD ..... 73
Live Video Channel 1~4 Non Standard Register – 0x3C1 ..... 73
Live Video Channel Control Register – 0x3C5 ..... 74
Live Video Channel System Select Register – 0x3C6 ..... 74
Live/Record Channel Select Register – 0x3D2 ..... 74
Live/Record Set Select Register – 0x3DA ..... 75
Play Back Interface ..... 76
Introduction ..... 76
Features ..... 76
Description ..... 76
Play Back Architecture ..... 76
Functions ..... 79
Input Formatter ..... 79

Registers Table ..... 80
Registers Description ..... 84
Toggle Field Signal Enable Register – 0x357 ..... 84
PB1 Vertical Target Size LSB Register – 0x358 ..... 84
PB1 and PB2 Vertical Target Size MSB Register – 0x359 ..... 84
PB2 Vertical Target Size LSB Register – 0x35A ..... 84
PB1 Vertical Source Size LSB Register – 0x35E ..... 85
PB1 and PB2 Vertical Source Size MSB Register – 0x35F ..... 85
PB2 Vertical Source Size LSB Register – 0x360 ..... 85
PB1 Horizontal Scaler Target LSB Register – 0x364 ..... 85
PB1 and PB2 Horizontal Target Size MSB Register – 0x365 ..... 85
PB2 Horizontal Scaler Target LSB Register – 0x366 ..... 86
PB1 Horizontal Scaler Source LSB Register – 0x36A ..... 86
PB1 and PB2 Horizontal Source Size MSB Register – 0x36B ..... 86
PB2 Horizontal Scaler Source LSB Register – 0x36C ..... 86
PB Auto Downscale and Bypass Register – 0x370 ..... 87
PB MISC Control Register – 0x371 ..... 87
Horizontal Delay for Playback Port 1 Register – 0x3B5 ..... 88
Vertical Delay for Playback Port 1 Register – 0x3B6 ..... 88
PB Path Selection Register – 0x3DB ..... 88
PB In_PB_Test Register – 0x3DC ..... 88
PB In_PB_Test Pattern Generator Color Bar Shift Volum – 0x3DD ..... 89
PB Channel ID Dec PB1 Ctrl – 0x3DE ..... 89
PB Channel ID Dec PB1 VOS – 0x3DF ..... 90
PB Channel ID Dec PB1 FOS – 0x3E0 ..... 90
PB Channel ID Dec PB1 HOS – 0x3E1 ..... 90
PB Channel ID Dec PB1 MID_VAL – 0x3E2 ..... 90
PB Frame/Field Control Register – 0x3F2 ..... 90
PB Field Mode Register – 0x3F3 ..... 91
PB Digital Channel ID Enable Register – 0x3F4 ..... 91
Non-standard Video Upper Limit - NTSC Register – 0x3F8 ..... 91
Non-standard Video Lower Limit - NTSC Register – 0x3F9 ..... 91
Non-standard Video Upper Limit - PAL Register – 0x3FA ..... 91
Non-standard Video Lower Limit - PAL Register – 0x3FB ..... 91
PB CHID Decoder Control Register – 0x3FC ..... 92
Analog Channel ID Line Number Register 1 – 0x3FD ..... 92
Analog Channel ID Line Number Register 2 – 0x3FE ..... 92
MISC. Control Register – 0x3FF ..... 92
PB Channel ID Dec Read Bus PB1 Byte 1– 0x372 ..... 93
PB Channel ID Dec Read Bus PB1 Byte 2 – 0x373 ..... 94
PB Channel ID Dec Read Bus PB1 Byte 3 – 0x374 ..... 95
PB Channel ID Dec Read Bus PB1 Byte 4 – 0x3C7 ..... 95
PB Channel ID Dec Read Bus PB1 Byte 5 – 0x3C8 ..... 96
Video Write Buffer (RGBW) ..... 97
Introduction ..... 97
Window Control ..... 97
Display Map Control ..... 97
Play Back Channels ..... 98
Register Table ..... 99
Register ..... 106
Display Write Buffer Control Register – 0x600 ..... 106
Display Write Buffer Vertical Position Register A – 0x614 ..... 106
Display Write Buffer Vertical Position Register B – 0x615 ..... 106
Display Write Buffer Vertical Size Register A – 0x63C ..... 106
Display Write Buffer Vertical Size Register B – 0x63D ..... 106
Display Write Buffer Horizontal Position Register A – 0x664 ..... 107

Display Write Buffer Horizontal Position Register B – 0x665 ..... 107
Display Write Buffer Horizontal Size Register A – 0x68C ..... 107
Display Write Buffer Horizontal Size Register B – 0x68D ..... 107
Display Write Buffer Control Register 1 – 0x6B4 ..... 108
Test Pattern Timing Replace Register 1 – 0x6B5 ..... 108
Misc Control Register – 0x6B6 ..... 108
PB First Channel Number Register 1 – 0x6B7 ..... 109
PB First Channel Number Register 2 – 0x6B8 ..... 109
PB First Channel Number Enable Register – 0x6B9 ..... 109
Test Pattern Timing Replace Register 2 – 0x6BA ..... 109
PB Channel Freeze Register 1 – 0x6BB ..... 110
PB Channel Freeze Register 2 – 0x6BC ..... 110
Protection Address Register 1 – 0x6BD ..... 110
Protection Address Register 2 – 0x6BE ..... 110
Protection Address Register 3 – 0x6BF ..... 110
Display Write Buffer Horizontal Start – 0x6C0 ..... 110
Display Write Buffer Vertical Start – 0x6D4 ..... 110
Live Channel Frame Rate Correction Control Register 1 – 0x6E8 ..... 111
Live Channel Frame Rate Correction Control Register 2 – 0x6E9 ..... 111
Live Channel Frame Rate Correction Control Register 3 – 0x6EA ..... 111
Live Channel Frame Rate Correction Control Register 4 – 0x6EB ..... 111
Live Channel Frame Rate Correction Control Register 1 – 0x6EC ..... 111
Live Channel Frame Rate Correction Control Register 2 – 0x6ED ..... 111
Live Channel Frame Rate Correction Control Register 7 – 0x6EE ..... 112
Live Channel Frame Rate Correction Control Register 8 – 0x6EF ..... 112
PB0 Line Number Status Register 1 – 0x6F0 (Read Only) ..... 112
PB0 Line Number Status Register 2 – 0x6F1 (Read Only) ..... 112
PB Channel Frame Rate Correction Control Register 1 – 0x6F0 ..... 112
PB Channel Frame Rate Correction Control Register 2 – 0x6F1 ..... 112
PB Valid Status Register – 0x6F8 ..... 113
Live channel External Correction Period Register – 0x6F9 ..... 113
PB Channel Ignore and Free Running Register – 0x6FA ..... 113
FIFO Count Limit Register – 0x6FB ..... 113
Vertical Inverter Method Selection – 0x6FC ..... 114
Auto Mode PB Channel Enable Register 1 – 0x6FD ..... 114
Auto Mode PB Channel Enable Register 2 – 0x6FE ..... 114
Live channel Internal Correction Period Register – 0x6FF ..... 114
DRAM Control Unit (DCU) ..... 115
Introduction ..... 115
Memory Organization ..... 115
Display Plane and DRAM Arrangement ..... 116
DCU Handshake ..... 117
TW2880 DRAM Configuration Options ..... 117
Display Path, 64 bit, 256 Mbit, 8Mx32x2 (suggested) ..... 118
Display Path, 64 bit, 128 Mbit, 4Mx32x2 (suggested) ..... 118
Display Path, 64 bit, 64 Mbit, 2Mx32x2 ..... 118
Display Path, 32 bit, 128 Mbit, 4Mx32x1 ..... 118
Display Path, 32 bit, 64 Mbit, 2Mx32x1 (not recommended) ..... 119
Recording Path, 64 bit, 256 Mbit, 8Mx32x2 (suggested) ..... 119
Recording Path, 64 bit, 128 Mbit, 4Mx32x2 ..... 119
Recording Path, 64 bit, 64 Mbit, 2Mx32x2 ..... 119
Recording Path, 32 bit, 256 Mbit, 8Mx32x1 ..... 119
Recording Path, 32 bit, 128 Mbit, 4Mx32x1 ..... 120
Recording Path, 32 bit, 64 Mbit, 2Mx32x1 ..... 120
TW2880 DRAM Parameter Adjustments ..... 121
Command and address timing ..... 121

Data read / write timing ..... 123
Refresh options ..... 124
Frame Rate Control Unit ..... 124
Introduction ..... 124
SDRAM Allocation ..... 124
Read Buffer ID Calculation ..... 124
Write Buffer ID Calculation ..... 125
VGA Controller and Video interface ..... 126
Introduction ..... 126
Features ..... 127
Frame Buffer Controller ..... 127
Output Interface ..... 127
VGA Interface ..... 128
Block Diagram ..... 129
Display Layers ..... 129
Display Resolutions ..... 130
De-interlacing ..... 130
Upscale ..... 130
Mouse Interface ..... 131
Motion Box ..... 132
Single Box ..... 133
Peak Function ..... 133
OSG Layer ..... 133
Simple OSD ..... 133
Background and Channel Boundary ..... 133
Digital Gain ..... 134
Register Table ..... 135
Register ..... 147
De-interlace Mode Selection Register 1 – 0x400 ..... 147
Error Buffer Base Low Byte Register – 0x401 ..... 147
Error Buffer Base Middle Byte Register – 0x402 ..... 147
Error Buffer Base High Byte Register – 0x403 ..... 148
Height of one Field in Buffer Register low byte – 0x404 ..... 148
Height of one Field in Buffer Register high byte – 0x405 ..... 148
Width of one Field in Buffer Register low byte – 0x406 ..... 148
Width of one Field in Buffer Register high byte – 0x407 ..... 148
3D De-interlace ditb0 Register – 0x408 ..... 148
3D De-interlace ditb4 Register – 0x409 ..... 148
3D De-interlace ditb8 Register – 0x40A ..... 149
3D De-interlace ditb12 Register – 0x40B ..... 149
3D De-interlace ditb16 Register – 0x40C ..... 149
3D De-interlace ditb24 Register – 0x40D ..... 149
3D De-interlace ditb32 Register – 0x40E ..... 149
3D De-interlace ditb48 Register – 0x40F ..... 149
3D De-interlace afm_yf02_thd Register – 0x410 ..... 149
3D De-interlace afm_yf02_h_thd Register – 0x411 ..... 149
3D De-interlace afm_yf01_thd Register – 0x412 ..... 149
3D De-interlace afm_yf01_h_thd Register – 0x413 ..... 149
3D De-interlace 32 Caption Threshold Register – 0x414 ..... 150
Up-scale Control Register – 0x415 ..... 150
Up-scale Background Color – 0x416 ..... 150
Up-scale Horizontal Factor low byte Register – 0x417 ..... 150
Up-scale Horizontal Factor high byte Register – 0x418 ..... 151
Up-scale Vertical Factor low byte Register – 0x419 ..... 151
Up-scale Vertical Factor high byte Register – 0x41A ..... 151
Up-scale Horizontal Start low byte Register – 0x41B ..... 151

Up-scale Horizontal Start high byte Register – 0x41C ..... 151
Up-scale Vertical Start low byte Register – 0x41D ..... 152
Up-scale Vertical Start high byte Register – 0x41E ..... 152
Up-scale Output Width low byte Register – 0x41F ..... 152
Up-scale Output Width high byte Register – 0x420 ..... 152
Up-scale Output Height low byte Register – 0x421 ..... 152
Up-scale Output Height high byte Register – 0x422 ..... 152
Up-scale X Boundary Width Register – 0x423 ..... 153
Up-scale Y Boundary Width Register – 0x424 ..... 153
Up-scale Panorama / Water Glass Control Register – 0x425 ..... 153
Up-scale Horizontal Center Scale Factor low byte Register – 0x426 ..... 153
Up-scale Horizontal Center Scale Factor high byte Register – 0x427 ..... 154
2D De-interlace Control Register – 0x428 ..... 154
2D De-interlace Threshold 1 Register – 0x429 ..... 154
2D De-interlace Threshold 2 Register – 0x42A ..... 154
2D De-interlace Threshold 3 Register – 0x42B ..... 154
2D De-interlace Threshold 4 Register – 0x42C ..... 154
2D De-interlace Threshold 5 Register – 0x42D ..... 154
2D De-interlace Threshold 6 Register – 0x42E ..... 154
3D De-interlace Control Register – 0x42F ..... 154
3D De-interlace Threshold 1 Register – 0x430 ..... 155
3D De-interlace Threshold 2 Register – 0x431 ..... 155
Upscaler Option Register – 0x432 ..... 155
Single Box 4 to 7 Register – 0x440 to 0x46B ..... 155
Mouse Base Address Low Byte Register – 0x46C ..... 155
Mouse Base Address Middle Byte Register – 0x46D ..... 155
Mouse Base Address High Byte Register – 0x46E ..... 155
Mouse Update Enable Register – 0x46F ..... 156
Mouse Index Register – 0x470 ..... 156
Window 32 Vertical Position Register Low Byte – 0x472 ..... 156
Window 32 Horizontal Position Register Low Byte – 0x473 ..... 156
Window 32 Vertical Size Register Low Byte – 0x474 ..... 156
Window 32 Horizontal Size Register Low Byte – 0x475 ..... 157
Window 32 Vertical and Horizontal Position Register High Byte – 0x476 ..... 157
Window 32 Vertical and Horizontal Size Register High Byte – 0x477 ..... 157
Window 32 Enable Register – 0x47E ..... 157
Display MISC. Control Register – 0x480 ..... 158
VerTical Sync Delay and Boundary Width Register – 0x481 ..... 158
Boundrary Red Color Intensity Register – 0x482 ..... 158
Boundrary Green Color Intensity Register – 0x483 ..... 159
Boundrary Blue Color Intensity Register – 0x484 ..... 159
Horizontal Upscale Factor Low Byte Register – 0x485 ..... 159
Horizontal Upscale Factor High Byte Register – 0x486 ..... 159
Vertical Upscale Factor Low Byte Register – 0x487 ..... 159
Vertical Upscale Factor High Byte Register – 0x488 ..... 159
Position Upscale Enable Register – 0x489 ..... 160
OSD Blink Rate Register – 0x48A ..... 160
Motion Box Cursor Vertical Position Register – 0x48B ..... 160
Motion Box Out Boundary Red Color Register – 0x493 ..... 160
Motion Box Out Boundary Green Color Register – 0x494 ..... 160
Motion Box Out Boundary Blue Color Register – 0x495 ..... 160
Motion Box Inner Boundary Red Color Register – 0x496 ..... 161
Motion Box Inner Boundary Green Color Register – 0x497 ..... 161
Motion Box Inner Boundary Blue Color Register – 0x498 ..... 161
Motion Box Mask Red Color Register – 0x499 ..... 161
Motion Box Mask Green Color Register – 0x49A ..... 161
Motion Box Mask Blue Color Register – 0x49B ..... 161

Motion Box Plane Red Color Register – 0x49C ..... 161
Motion Box Plane Green Color Register – 0x49D ..... 162
Motion Box Plane Blue Color Register – 0x49E ..... 162
Horizontal Start Low Byte Register – 0x4A0 ..... 162
Horizontal Start High Byte Register – 0x4A1 ..... 162
Vertical Start Low Byte Register – 0x4A2 ..... 162
Vertical Start High Byte Register – 0x4A3 ..... 162
rti Wndow Register – 0x4A4 ..... 162
Vertical Total Adjust Mode Register – 0x4A5 ..... 163
Vertical Total from Input Video Low Byte Register – 0x4A6 ..... 163
Vertical Total from Input Video High Byte Register – 0x4A7 ..... 163
Vertical Total after Adjustment Byte Register – 0x4A8 ..... 163
Vertical Total after Adjustment High Register – 0x4A9 ..... 163
Boundary Enable Register – 0x4AA ..... 164
PS2 Mouse Control Register – 0x4B4 ..... 164
PS2 Mouse Write Data Register – 0x4B5 ..... 164
PS2 Mouse Write Data Register 1– 0x4B6 ..... 164
PS2 Mouse Write Data Register 2– 0x4B7 ..... 164
PS2 Mouse Write Data Register 3– 0x4B8 ..... 164
PS2 Mouse Write Data Register 4– 0x4B9 ..... 165
PS2 Interrupt Register– 0x4BA ..... 165
External OSD Control Register– 0x4BB ..... 165
External OSD Alpha Register– 0x4BC ..... 165
External OSD Color R Register– 0x4BD ..... 165
External OSD Color G Register– 0x4BE ..... 166
External OSD Color B Register– 0x4BF ..... 166
Interrupt Mask Register– 0x4C0 ..... 166
Interrupt Status Register– 0x4C1 ..... 166
Frame Rate Control Debug Register– 0x4C2 ..... 167
Misc Control Register – 0x4C3 ..... 167
Boundary Enable Register 2 – 0x4C4 ..... 168
Image control Register – 0x4CE ..... 168
Peak Control Register – 0x4D0 ..... 168
Digital R Gain Register – 0x4DA ..... 169
Digital G Gain Register – 0x4DB ..... 169
Digital B Gain Register – 0x4DC ..... 169
R Offset Register – 0x4DD ..... 169
G Offset Register – 0x4DE ..... 169
B Offset Register – 0x4DF ..... 169
Boundary Channel Enable Register 1 – 0x4F0 ..... 170
Boundary Channel Enable Register 2 – 0x4F1 ..... 170
Boundary Channel Enable Register 3 – 0x4F2 ..... 170
Boundary Channel Enable Register 4 – 0x4F3 ..... 170
Frame Rate Control Enable Register 1 – 0x4F4 ..... 170
Frame Rate Control Enable Register 2 – 0x4F5 ..... 170
Frame Rate Control Enable Register 3 – 0x4F6 ..... 170
Frame Rate Control Enable Register 4 – 0x4F7 ..... 171
Red Color for No Video Channel – 0x4F8 ..... 171
Green Color for No Video Channel – 0x4F9 ..... 171
Blue Color for No Video Channel – 0x4FA ..... 171
Horizontal Total Register 1 – 0x500 ..... 171
Horizontal Total Register 2 – 0x501 ..... 171
Vertical Total Register 1 – 0x502 ..... 172
Vertical Total Register 2 – 0x503 ..... 172
Horizontal Display End Register 1 – 0x504 ..... 172
Horizontal Display End Register 2 – 0x505 ..... 172
Vertical Display End Register 1 – 0x506 ..... 172

Vertical Display End Register 2 – 0x507 ..... 172
Horizontal Sync Starting position Register – 0x508 ..... 173
Vertical Sync Starting position Register – 0x509 ..... 173
Horizontal Sync Width Register – 0x50A ..... 173
Vertical Sync Width Register – 0x50B ..... 173
SBOX0 Control Register – 0x50C ..... 173
SBOX Line Width Control Register 1 – 0x511 ..... 174
SBOX Line Width Control Register 2 – 0x512 ..... 174
SBOX0 HL Control Register 1 – 0x513 ..... 175
SBOX0 HL Control Register 2 – 0x514 ..... 175
SBOX0 HR Control Register 1 – 0x51B ..... 175
SBOX0 HR Control Register 2 – 0x51C ..... 175
SBOX0 VT Control Register 1 – 0x523 ..... 175
SBOX0 VT Control Register 2 – 0x524 ..... 175
SBOX0 VB Control Register 1 – 0x52B ..... 175
SBOX0 VB Control Register 2 – 0x52C ..... 175
BOX Boundary Color R byte – 0x533 ..... 175
BOX Boundary Color G byte – 0x534 ..... 175
BOX Boundary Color B byte – 0x535 ..... 176
BOX Plane Color R byte – 0x536 ..... 176
BOX Plane Color G byte – 0x537 ..... 176
BOX Plane Color B byte – 0x538 ..... 176
Screen Background Color R byte – 0x539 ..... 176
Screen Background Color G byte – 0x53A ..... 176
Screen Background Color B byte – 0x53B ..... 176
MOUSE0 Horizontal Position Register 1 – 0x53C ..... 176
MOUSE0 Horizontal Position Register 2 – 0x53D ..... 176
MOUSE0 Vertical Position Register 1 – 0x53E ..... 176
MOUSE0 Vertical Position Register 2 – 0x53F ..... 177
MOUSE1 Horizontal Position Register 1 – 0x540 ..... 177
MOUSE1 Horizontal Position Register 2 – 0x541 ..... 177
MOUSE1 Vertical Position Register 1 – 0x542 ..... 177
MOUSE1 Vertical Position Register 2 – 0x543 ..... 177
MOUSE Control Register – 0x544 ..... 177
Mouse Background Color R byte – 0x545 ..... 178
Mouse Background Color G byte – 0x546 ..... 178
Mouse Background Color B byte – 0x547 ..... 178
Mouse Foreground Color R byte – 0x548 ..... 178
Mouse Foreground Color G byte – 0x549 ..... 178
Mouse Foreground Color B byte – 0x54A ..... 178
Display Mode Control Register – 0x54B ..... 178
Mouse RAM Address Register – 0x54C ..... 179
Mouse RAM Data Register – 0x54D ..... 179
Mouse Register Update Enable Register – 0x54E ..... 179
Mouse RAM Write Enable Register – 0x54F ..... 179
MD BOX Control Register 0 – 0x550 ..... 179
MD BOX Line Width Control Register 0 – 0x560 ..... 180
MD BOX 0 Horizontal Left Position Register 1 – 0x568 ..... 180
MD BOX 0 Horizontal Left Position Register 2 – 0x569 ..... 181
MD BOX 0 Vertical Top Position Register 1 – 0x588 ..... 181
MD BOX 0 Vertical Top Position Register 2 – 0x589 ..... 181
MD BOX 0 Horizontal Size Register 1 – 0x5A8 ..... 181
MD BOX 0 Horizontal Size Register 2 – 0x5A9 ..... 181
MD BOX 0 Vertical Size Register 1 – 0x5C8 ..... 181
MD BOX 0 Vertical Size Register 2 – 0x5C9 ..... 182
MD BOX HCELL Register 0 – 0x5E8 ..... 182
MD BOX VCELL Register 0 – 0x5F0 ..... 182

MD BOX Cursor Horizontal Position Register 0 – 0x5F8 ..... 182
HDMI Controller ..... 183
Introduction ..... 183
Physical Port ..... 184
Control and Configuration ..... 184
Registers/Configuration Logic ..... 184
Slave I²C Interface ..... 184
Audio Data Capture Logic ..... 185
S/PDIF ..... 185
I²S ..... 185
TMDS Digital Core ..... 185
DDC Master I²C Interface ..... 185
Register Grouping ..... 187
Usages and Conventions ..... 187
Base Register Set ..... 188
Vendor ID Register ..... 188
Device ID Register ..... 188
Device Revision Register ..... 188
Software Reset Register ..... 188
System Control Register #1 ..... 189
System Status Register ..... 189
Legacy Registers ..... 189
Data Control Register ..... 190
Automatic Ri Check ..... 190
Ri Status Register ..... 190
Ri Command Register ..... 191
Ri Line Start Register ..... 191
Ri From RX Register ..... 191
DE Generator Register Set ..... 191
Video DE Delay Register ..... 192
Video DE Control Register ..... 193
Video DE Top Register ..... 193
Video DE Count Register ..... 193
Video DE Line Register ..... 194
Video H Resolution Register ..... 194
Video V Refresh Register ..... 194
Video Embedded Sync Decoding Registers ..... 195
Video Interface Adjustment Register ..... 195
Video SYNC Polarity Detection Register ..... 195
Video Hbit to HSYNC Register ..... 196
Video Field2 HSYNC Offset Register ..... 196
Video HSYNC Length Register ..... 196
Video Vbit to VSYNC Register ..... 196
Video VSYNC Length Register ..... 197
Video Control Register ..... 197
Video Mode Register ..... 198
Interrupt Registers ..... 199
Interrupt State Register ..... 199
Interrupt Source Register ..... 199
Interrupt Unmask Register ..... 201
Interrupt Control Register ..... 201
TMDS Control Register Set ..... 202
TMDS Control Register ..... 202
DDC Master Registers ..... 205
DDC I²C Manual Register ..... 205
DDC I²C Target Slave Address Register ..... 205

DDC I²C Target Segment Address Register ..... 205
DDC I²C Target Offset Address Register ..... 205
DDC I²C Data Count Register ..... 206
DDC I²C Status Register ..... 206
DDC I²C Command Register ..... 206
DDC I²C Data Register ..... 207
DDC I²C FIFO Count Register ..... 207
Audio Registers ..... 208
ACR Control Register ..... 208
ACR Audio Frequency Register ..... 208
ACR N Software Value Register ..... 208
ACR CTS Software Value Register ..... 209
ACR CTS Hardware Value Registers ..... 209
Audio In Mode Register ..... 209
Audio In S/PDIF Control Register ..... 210
Audio In S/PDIF Extracted Fs and Length Register ..... 210
Audio In I²S Channel Swap Register ..... 210
Audio Error Threshold Register ..... 211
Audio In I²S Data In Map Register ..... 211
Audio In I²S Control Register ..... 212
Audio In I²S Channel Status Registers ..... 212
Audio Sample Rate Conversion Register ..... 213
Audio I²S Input Length Register ..... 214
HDMI Control Register ..... 214
Audio Path Status Register ..... 215
Diagnostic Power Down Register ..... 215
Packet Registers ..... 216
Packet Buffer Control #1 Register ..... 217
Packet Buffer Control #2 Register ..... 217
Packet Registers ..... 218
SPD InfoFrame Registers ..... 218
Audio InfoFrame Registers ..... 219
MPEG InfoFrame Registers ..... 220
Generic Packet Registers ..... 221
General Control Packet Register ..... 222
Generic Packet #2 Registers ..... 222
Appendices ..... 223
Handling Audio ..... 224
Enabling Audio Inputs ..... 224
Encoding Audio on HDMI ..... 225
Handling DVD Audio ..... 226
Sink Support for Audio Content Protection ..... 226
Transmitting ACP Packets ..... 226
Handling Video ..... 228
Programming Video Input Mode and Video Output Mode ..... 228
Handling Interlaced Video ..... 231
Summary of Video Processing Path Options ..... 232
480i Input ..... 233
480i Muxed YCbCr 4:2:2 Embedded Sync Input ..... 233
480i Muxed YCbCr 4:2:2 Embedded Sync Input – Fixing for CEA-861B ..... 234
480i YCbCr 4:2:2 Mux YC Separate Sync Input ..... 235
576i Input ..... 236
576i YCbCr 4:2:2 Embedded Sync Input ..... 236
576i YCbCr 4:2:2 Embedded Sync Input – Fixing for CEA-861B ..... 237
780p Input ..... 238
720p RGB Input ..... 238
720p YCbCr 4:4:4 Separate Sync Input ..... 239

720p YCbCr 4:2:2 Separate Sync Input ..... 240
720p YCbCr 4:2:2 Embedded Sync Input ..... 241
720p YCbCr 4:2:2 Embedded Sync Input – Fixing for CEA-861B ..... 242
1080i Input ..... 243
1080i RGB Input ..... 243
1080i YCbCr 4:4:4 Separate Sync Input ..... 244
1080i YCbCr 4:2:2 Separate Sync Input ..... 245
1080i YCbCr 4:2:2 Embedded Sync Input ..... 246
1080i YCbCr 4:2:2 Embedded Sync Input – Fixing for CEA-861B ..... 247
1080p YCbCr 4:4:4 Separate Sync Input ..... 248
1080p YCbCr 4:2:2 Separate Sync Input ..... 249
1080p YCbCr 4:2:2 Embedded Sync Input ..... 250
1080p YCbCr 4:2:2 Embedded Sync Input – Fixing for CEA-861B ..... 251
Handling InforFrames ..... 252
AVI InforFrames ..... 252
General Control Packets ..... 254
Handling Audio Content Protection (ACP) Packets ..... 254
Operating DDC Master ..... 255
Device Addresses ..... 255
DDC Read Operation ..... 256
Write Command ..... 258
Abort Command ..... 258
Power Down Control ..... 259
Logic Blocks Affected by Power Down ..... 259
Registers Affected by Power Down ..... 259
OSG for Display Path ..... 260
Introduction ..... 260
Features ..... 260
OSG Graphic Buffer ..... 260
Memory Diagram ..... 260
Bitmap Buffer ..... 261
Write Memroy by Host ..... 262
Block Transfer ..... 262
Block Fill ..... 262
BitBlt ..... 262
De-compression ..... 263
Pixel Bit Extension ..... 264
Color Conversion ..... 264
Selective Overwrite ..... 264
Pixel Operation Sequence ..... 264
Bitmap Buffer Display ..... 264
Eight Sub-windows ..... 265
Transparent ..... 266
Alpha Blending ..... 266
Blinking ..... 266
RGB Format ..... 266
Upscale ..... 267
Host write buffer threshold setting ..... 268
Host Write Data Sequence ..... 269
Registers Table ..... 270
Register Descriptions ..... 274
OSG Control Register – 0x100 ..... 274
OSG Status Register 2 – 0x101 ..... 274
OSG Mode Register 1 – 0x102 ..... 274
OSG Mode Register 2 – 0x103 ..... 275
OSG BitBlt Operation Register – 0x104 ..... 276

OSG Display Window 1 Control Register – 0x105....276
OSG Display Window 2 Control Register – 0x106....276
OSG Alpha Blending Register – 0x107....277
OSG Alpha Blending Color Low Byte Register – 0x108....277
OSG Alpha Blending Color High Byte Register – 0x109....277
OSG Alpha Blending Color Low Byte Register – 0x10A....277
OSG Alpha Blending Color High Byte Register – 0x10B....278
OSG Alpha Blending Color Low Byte Register – 0x10C....278
OSG Alpha Blending Color High Byte Register – 0x10D....278
OSG Alpha Blending Color Low Byte Register – 0x10E....278
OSG Alpha Blending Color High Byte Register – 0x10F....278
OSG Blinking Foregound Color Low Byte Register – 0x110....278
OSG Blinking Foreground Color High Byte Register – 0x111....278
OSG Blinking Foregound Color Low Byte Register – 0x112....279
OSG Blinking Foreground Color High Byte Register – 0x113....279
OSG Fill Color Low Byte – 0x114....279
OSG Fill Color High Byte – 0x115....279
OSG Overwrite Color Low Byte – 0x116....279
OSG Overwrite Color High Byte – 0x117....279
OSG Color Conversion Source Color 1 Low Byte – 0x118....279
OSG Color Conversion Source Color 1 High Byte – 0x119....280
OSG Color Conversion Target Color 1 Low Byte – 0x11A....280
OSG Color Conversion Target Color 1 High Byte – 0x11B....280
OSG Color Conversion Source Color 2 Low Byte – 0x11C....280
OSG Color Conversion Source Color 2 High Byte – 0x11D....280
OSG Color Conversion Target Color 2 Low Byte – 0x11E....280
OSG Color Conversion Target Color 2 High Byte – 0x11F....280
OSG Color Conversion Source Color 3 Low Byte – 0x120....281
OSG Color Conversion Source Color 3 High Byte – 0x121....281
OSG Color Conversion Target Color 3 Low Byte – 0x122....281
OSG Color Conversion Target Color 3 High Byte – 0x123....281
OSG Color Conversion Source Color 4 Low Byte – 0x124....281
OSG Color Conversion Source Color 4 High Byte – 0x125....281
OSG Color Conversion Target Color 4 Low Byte – 0x126....281
OSG Color Conversion Target Color 4 High Byte – 0x127....282
OSG Bitmap Buffer Horizontal Size Low Byte – 0x128....282
OSG Bitmap Buffer Horizontal Size High Byte – 0x129....282
OSG Bitmap Buffer Source Window 1 Vertical Start Low Byte – 0x12A....282
OSG Bitmap Buffer Source Window 1 Vertical Start High Byte – 0x12B....282
OSG Bitmap Buffer Source Window 1 Horizontal Start Low Byte – 0x12C....282
OSG Bitmap Buffer Source Window 1 Horizontal Start High Byte – 0x12D....283
OSG Bitmap Buffer Source Window 1 Vertical Size Low Byte – 0x12E....283
OSG Bitmap Buffer Source Window 1 Vertical Size High Byte – 0x12F....283
OSG Bitmap Buffer Source Window 1 Horizontal Size Low Byte – 0x130....283
OSG Bitmap Buffer Source Window 1 Horizontal Size High Byte – 0x131....283
OSG Bitmap Buffer Target Window Vertical Start Low Byte – 0x132....283
OSG Bitmap Buffer Target Window Vertical Start High Byte – 0x133....284
OSG Bitmap Buffer Target Window Horizontal Start Low Byte – 0x134....284
OSG Bitmap Buffer Target Window Horizontal Start High Byte – 0x135....284
OSG Bitmap Buffer Source Window 2 Vertical Start Low Byte – 0x136....284
OSG Bitmap Buffer Source Window 2 Vertical Start High Byte – 0x137....284
OSG Bitmap Buffer Source Window 2 Horizontal Start Low Byte – 0x138....284
Bitmap Buffer Source Window 2 Horizontal Start High Byte – 0x139....285
OSG Host Write Data Port – 0x13A....285
OSG Bitmap Buffer Start Address Low Byte – 0x13B....285
OSG Bitmap Buffer Start Address Middle Byte – 0x13C....285
OSG Bitmap Buffer Start Address High Byte – 0x13D....285

OSG FORMAT Control – 0x17E ..... 286
OSG Bitmap Buffer Start Address High Byte – 0x13D ..... 286
OSG Window 1 Vertical Start in Bitmap Buffer Low Byte – 0x144 ..... 286
OSG Window 1 Vertical Start in Bitmap Buffer High Byte – 0x145 ..... 286
OSG Window 1 Horizontal Start in Bitmap Buffer Low Byte – 0x146 ..... 286
OSG Window 1 Horizontal Start in Bitmap Buffer High Byte – 0x147 ..... 287
OSG Window 1 Vertical Size Low Byte – 0x148 ..... 287
OSG Window 1 Vertical Size High Byte – 0x149 ..... 287
OSG Window 1 Horizontal Size Low Byte – 0x14A ..... 287
OSG Window 1 Horizontal Size High Byte – 0x14B ..... 287
OSG Window 1 Vertical Start On Screen Low Byte – 0x14C ..... 287
OSG Window 1 Vertical Start On Screen High Byte – 0x14D ..... 287
OSG Window 1 Horizontal Start On Screen Low Byte – 0x14E ..... 288
OSG Window 1 Horizontal Start On Screen High Byte – 0x14F ..... 288
OSG Window 2 Vertical Start in Bitmap Buffer Low Byte – 0x150 ..... 288
OSG Window 2 Vertical Start in Bitmap Buffer High Byte – 0x151 ..... 288
OSG Window 2 Horizontal Start in Bitmap Buffer Low Byte – 0x152 ..... 288
OSG Window 2 Horizontal Start in Bitmap Buffer High Byte – 0x153 ..... 288
OSG Window 2 Vertical Size Low Byte – 0x154 ..... 289
OSG Window 2 Vertical Size High Byte – 0x155 ..... 289
OSG Window 2 Horizontal Size Low Byte – 0x156 ..... 289
OSG Window 2 Horizontal Size High Byte – 0x157 ..... 289
OSG Window 2 Vertical Start On Screen Low Byte – 0x158 ..... 289
OSG Window 2 Vertical Start On Screen High Byte – 0x159 ..... 289
OSG Window 2 Horizontal Start On Screen Low Byte – 0x15A ..... 290
OSG Window 2 Horizontal Start On Screen High Byte – 0x15B ..... 290
OSG Overwrite Color2 Low Byte – 0x15C ..... 290
OSG Overwrite Color2 High Byte – 0x15D ..... 290
OSG Overwrite Color3 Low Byte – 0x15E ..... 290
OSG Overwrite Color3 High Byte – 0x15F ..... 290
OSG Overwrite Color4 Low Byte – 0x160 ..... 291
OSG Overwrite Color4 High Byte – 0x161 ..... 291
OSG Vertical Blank Register – 0x162 ..... 291
OSG Display Window 3 Control Register – 0x163 ..... 291
OSG Alpha Blending Register – 0x164 ..... 292
OSG Alpha Blending Color Low Byte Register – 0x165 ..... 292
OSG Alpha Blending Color High Byte Register – 0x166 ..... 292
OSG Alpha Blending Color Low Byte Register – 0x167 ..... 292
OSG Alpha Blending Color High Byte Register – 0x168 ..... 293
OSG Blinking Foregound Color Low Byte Register – 0x169 ..... 293
OSG Blinking Foreground Color High Byte Register – 0x16A ..... 293
OSG Display Window 1 Enable Register – 0x16B ..... 293
OSG Display Window 2 Enable Register – 0x16C ..... 293
OSG Display Window 3 Enable Register – 0x16D ..... 293
OSG Display Window 1 and 2 Select Register – 0x16E ..... 294
OSG Display Window 3 Select Register – 0x16F ..... 294
OSG Window 3 Vertical Start in Bitmap Buffer Low Byte – 0x170 ..... 294
OSG Window 3 Vertical Start in Bitmap Buffer High Byte – 0x171 ..... 294
OSG Window 3 Horizontal Start in Bitmap Buffer Low Byte – 0x172 ..... 294
OSG Window 3 Horizontal Start in Bitmap Buffer High Byte – 0x173 ..... 295
OSG Window 3 Vertical Size Low Byte – 0x174 ..... 295
OSG Window 3 Vertical Size High Byte – 0x175 ..... 295
OSG Window 3 Horizontal Size Low Byte – 0x176 ..... 295
OSG Window 3 Horizontal Size High Byte – 0x177 ..... 295
OSG Window 3 Vertical Start On Screen Low Byte – 0x178 ..... 295
OSG Window 3 Vertical Start On Screen High Byte – 0x179 ..... 295
OSG Window 3 Horizontal Start On Screen Low Byte – 0x17A ..... 296

OSG Window 3 Horizontal Start On Screen High Byte – 0x17B ..... 296
OSG Threshold 1 – 0x17C ..... 296
OSG Threshold 2 – 0x17D ..... 296
OSG FORMAT Control – 0x17E ..... 296
OSG FORMAT Control – 0x17F ..... 297
OSG Original Vertical Size for Window 1 Low Byte– 0x1C0 ..... 297
OSG Original Vertical Size for Window 1 High Byte – 0x1C1 ..... 297
OSG Original Horizontal Size for Window 1 Low Byte– 0x1C2 ..... 297
OSG Original Horizontal Size for Window 1 High Byte – 0x1C3 ..... 297
OSG Bitmap Buffer Horizontal Size Low Byte – 0x1CC ..... 297
OSG Bitmap Buffer Horizontal Size High Byte – 0x1CD ..... 298
OSG Bitmap Buffer Start Address Low Byte – 0x1CE ..... 298
OSG Bitmap Buffer Start Address Middle Byte – 0x1CF ..... 298
OSG Bitmap Buffer Start Address High Byte – 0x1D0 ..... 298
OSD for Display Path ..... 299
Introduction ..... 299
Features ..... 299
Font and Picture Library ..... 299
Font ..... 299
Picture ..... 300
Display Information ..... 301
Channel Number ..... 301
Channel Picture ..... 301
Date and Time ..... 302
Title ..... 302
SRAM Allocation ..... 302
Font and Picture SRAM ..... 302
Display SRAM ..... 303
Registers Table ..... 305
Registers Description ..... 307
OSD Control Register – 0x180 ..... 307
OSD Mode Register – 0x181 ..... 307
OSD Font Size Register – 0x182 ..... 308
OSD Channel Number Horizontal Position Low Byte Register – 0x183 ..... 309
OSD Channel Number Horizontal Position High Byte Register – 0x184 ..... 309
OSD Channel Number Vertical Position Low Byte Register – 0x185 ..... 309
OSD Channel Number Horizontal Position High Byte Register – 0x186 ..... 309
OSD Channel Picture Horizontal Position Low Byte Register – 0x187 ..... 309
OSD Channel Picture Horizontal Position High Byte Register – 0x188 ..... 309
OSD Channel Picture Vertical Position Low Byte Register – 0x189 ..... 309
OSD Channel Picture Horizontal Position High Byte Register – 0x18A ..... 310
OSD Time and Date Horizontal Position Low Byte Register – 0x18B ..... 310
OSD Time and Date Horizontal Position High Byte Register – 0x18C ..... 310
OSD Time and Date Vertical Position Low Byte Register – 0x18D ..... 310
OSD Time and Date Horizontal Position High Byte Register – 0x18E ..... 310
OSD Title Horizontal Position Low Byte Register – 0x18F ..... 310
OSD Title Horizontal Position High Byte Register – 0x190 ..... 310
OSD Title Vertical Position Low Byte Register – 0x191 ..... 311
OSD Title Horizontal Position High Byte Register – 0x192 ..... 311
OSD Font Red Color 1 Register – 0x193 ..... 311
OSD Font Green Color 1 Register – 0x194 ..... 311
OSD Font Blue Color 1 Register – 0x195 ..... 311
OSD Font Red Color 2 Register – 0x196 ..... 311
OSD Font Green Color 2 Register – 0x197 ..... 311
OSD Font Blue Color 2 Register – 0x198 ..... 312
OSD Font Red Color 3 Register – 0x199 ..... 312

OSD Font Green Color 3 Register – 0x19A ..... 312
OSD Font Blue Color 3 Register – 0x19B ..... 312
OSD Picture Red Color 0 Register – 0x19C ..... 312
OSD Picture Green Color 0 Register – 0x19D ..... 312
OSD Picture Blue Color 0 Register – 0x19E ..... 312
OSD Picture Red Color 1 Register – 0x19F ..... 313
OSD Picture Green Color 1 Register – 0x1A0 ..... 313
OSD Picture Blue Color 1 Register – 0x1A1 ..... 313
OSD Picture Red Color 2 Register – 0x1A2 ..... 313
OSD Picture Green Color 2 Register – 0x1A3 ..... 313
OSD Picture Blue Color 2 Register – 0x1A4 ..... 313
OSD Picture Red Color 3 Register – 0x1A5 ..... 313
OSD Picture Green Color 3 Register – 0x1A6 ..... 314
OSD Picture Blue Color 3 Register – 0x1A7 ..... 314
OSD Font and Picture RAM Address Low Byte Register – 0x1A8 ..... 314
OSD Font and Picture RAM Address High Byte Register – 0x1A9 ..... 314
OSD Font and Picture RAM Data Low Byte Register – 0x1AA ..... 314
OSD Font and Picture RAM Data High Byte Register – 0x1AB ..... 314
OSD Display RAM Address Register – 0x1AC ..... 315
OSD Display RAM Address Register – 0x1AD ..... 315
OSD Display RAM Data Register – 0x1AE ..... 315
Registers Description ..... 316
OSD Control Register – 0x180 ..... 316
OSD Mode Register – 0x181 ..... 316
OSD Font Size Register – 0x182 ..... 317
OSD Channel Number Horizontal Position Low Byte Register – 0x183 ..... 318
OSD Channel Number Horizontal Position High Byte Register – 0x184 ..... 318
Osd channel Number Vertical Position Low Byte Register – 0x185 ..... 318
Osd channel Number Horizontal Position High Byte Register – 0x186 ..... 318
Osd channel Picture Horizontal Position Low Byte Register – 0x187 ..... 318
Osd channel Picture Horizontal Position High Byte Register – 0x188 ..... 318
Osd channel Picture Vertical Position Low Byte Register – 0x189 ..... 319
Osd channel Picture Horizontal Position High Byte Register – 0x18A ..... 319
OSD Time and Date Horizontal Position Low Byte Register – 0x18B ..... 319
OSD Time and Date Horizontal Position High Byte Register – 0x18C ..... 319
OSD Time and Date Vertical Position Low Byte Register – 0x18D ..... 319
OSD Time and Date Horizontal Position High Byte Register – 0x18E ..... 319
OSD Title Horizontal Position Low Byte Register – 0x18F ..... 319
OSD Title Horizontal Position High Byte Register – 0x190 ..... 320
OSD Title Vertical Position Low Byte Register – 0x191 ..... 320
OSD Title Horizontal Position High Byte Register – 0x192 ..... 320
OSD Font Red Color 1 Register – 0x193 ..... 320
OSD Font Green Color 1 Register – 0x194 ..... 320
OSD Font Blue Color 1 Register – 0x195 ..... 320
OSD Font Red Color 2 Register – 0x196 ..... 320
OSD Font Green Color 2 Register – 0x197 ..... 321
OSD Font Blue Color 2 Register – 0x198 ..... 321
OSD Font Red Color 3 Register – 0x199 ..... 321
OSD Font Green Color 3 Register – 0x19A ..... 321
OSD Font Blue Color 3 Register – 0x19B ..... 321
OSD Picture Red Color 0 Register – 0x19C ..... 321
OSD Picture Green Color 0 Register – 0x19D ..... 321
OSD Picture Blue Color 0 Register – 0x19E ..... 322
OSD Picture Red Color 1 Register – 0x19F ..... 322
OSD Picture Green Color 1 Register – 0x1A0 ..... 322
OSD Picture Blue Color 1 Register – 0x1A1 ..... 322
OSD Picture Red Color 2 Register – 0x1A2 ..... 322

OSD Picture Green Color 2 Register – 0x1A3 ..... 322
OSD Picture Blue Color 2 Register – 0x1A4 ..... 322
OSD Picture Red Color 3 Register – 0x1A5 ..... 323
OSD Picture Green Color 3 Register – 0x1A6 ..... 323
OSD Picture Blue Color 3 Register – 0x1A7 ..... 323
OSD Font and Picture RAM Address Low Byte Register – 0x1A8 ..... 323
OSD Font and Picture RAM Address High Byte Register – 0x1A9 ..... 323
OSD Font and Picture RAM Data Low Byte Register – 0x1AA ..... 323
OSD Font and Picture RAM Data High Byte Register – 0x1AB ..... 324
OSD Display RAM Address Register – 0x1AC ..... 324
OSD Display RAM Address Register – 0x1AD ..... 324
OSD Display RAM Data Register – 0x1AE ..... 324
Dualview Display Controller Unit ..... 325
Introduction ..... 325
Features ..... 325
Dual Monitor Controller Architecture ..... 325
Memory Diagram ..... 325
Frame Data Synchonization ..... 326
Dual Monitor Controller Block Diagram ..... 327
Memory Interface ..... 327
FIFO Interface ..... 327
HV Timing Control ..... 327
Down Scalar ..... 328
Line Buffer Control ..... 328
TV Encoder ..... 329
OSD Control ..... 329
Clock setting for dual monitor ..... 330
Theory of Operation ..... 330
TV Encoder Setting ..... 331
Register Table ..... 332
Registers ..... 338
Dual Monitor Horizontal Totaol Low Byte Register – 0x700 ..... 338
Dual Monitor Horizontal Total High Byte Register – 0x701 ..... 338
Dual Monitor Vertical Total Low Byte Register – 0x702 ..... 338
Dual Monitor Vertical Total High Byte Register – 0x703 ..... 338
Dual Monitor Horizontal Width Low Byte Register – 0x704 ..... 338
Dual Monitor Horizontal Width High Byte Register – 0x705 ..... 338
Dual Monitor Vertical Height Low Byte Register – 0x706 ..... 339
Dual Monitor Vertical Height High Byte Register – 0x707 ..... 339
Dual Monitor Hsync Front Porch Low Byte Register – 0x708 ..... 339
Dual Monitor Hsync Front Porch High Byte Register – 0x709 ..... 339
Dual Monitor Vsync Front Porch Register – 0x70A ..... 339
Dual Monitor Hsync Width Register – 0x70B ..... 339
Dual Monitor Vsync Width Register – 0x70C ..... 339
Dual Monitor Control Register – 0x70D ..... 340
Dual Monitor Target Width for Down Scaler Low Byte Register – 0x70E ..... 340
Dual Monitor Target Width for Down Scaler High Byte Register – 0x70F ..... 340
Dual Monitor Target Height for Down Scaler Low Byte Register – 0x710 ..... 340
Dual Monitor Target Height for Down Scaler High Byte Register – 0x711 ..... 340
Dual Monitor Mode Register – 0x712 ..... 341
Dual Monitor Horizontal Start Low Byte Register – 0x713 ..... 341
Dual Monitor Vertical Start Low Byte Register – 0x714 ..... 341
Dual Monitor HSTART and VSTART High Byte Register – 0x715 ..... 341
Dual Monitor Down Scaler Source Width Low Byte Register – 0x716 ..... 341
Dual Monitor Down Scaler Source Width High Byte Register – 0x717 ..... 342
Dual Monitor Down Scaler Source Height Low Byte Register – 0x718 ..... 342

Dual Monitor Down Scaler Source Height High Byte Register – 0x719....342
Dual Monitor TV Encoder Control Register 1 – 0x71A....342
Dual Monitor TV Encoder Control Register 2 – 0x71B....343
Dual Monitor TV Encoder Control Register 3 – 0x71C....343
Dual Monitor TV Encoder Pixel Delay Low Byte Register – 0x71D....344
Dual Monitor TV Encoder Pixel Delay High Byte Register – 0x71E....344
Dual Monitor TV Encoder Line Delay Register – 0x71F....344
Dual Monitor TV Encoder Line Offset Register – 0x720....344
Dual Monitor TV Encoder Pixel Offset Register – 0x721....344
Dual Monitor TV Encoder Control Register 4 – 0x722....344
Dual Monitor TV Encoder Control Register 5 – 0x723....345
Dual Monitor Control Register – 0x724....345
Dual Monitor DAC Power Down Register – 0x725....345
Dual Monitor Mouse Update Enable Register – 0x726....345
Dual Monitor Down Scaler Horizontal Blank Register – 0x727....345
Dual Monitor Boundary Channel Enable Register 1 – 0x728....346
Dual Monitor Boundary Channel Enable Register 2 – 0x729....346
Dual Monitor Boundary Channel Enable Register 3 – 0x72A....346
Dual Monitor Boundary Channel Enable Register 4 – 0x72B....346
Dual Monitor Red Color for No Video Channel – 0x72C....346
Dual Monitor Green Color for No Video Channel – 0x72D....346
Dual Monitor Blue Color for No Video Channel – 0x72E....346
Dual Monitor Mouse Index Register – 0x72F....347
Dual Monitor Simple OSD Register – 0x730 to 0x75E....347
Dual Monitor Horizontal Display Enable Control Register – 0x75F....347
Dual Monitor Background Red Color Register – 0x760....347
Dual Monitor Background Green Color Register – 0x761....347
Dual Monitor Background Blue Color Register– 0x762....347
Dual Monitor Control Register– 0x763....347
Dual Monitor Boundary Red Color Register– 0x764....348
Dual Monitor Boundary Green Color Register– 0x765....348
Dual Monitor Boundary Blue Color Register – 0x766....348
Dual Monitor Vertical Total Window Register – 0x767....348
Dual Monitor Vertical Total Adjust Mode Register – 0x768....348
Dual Monitor Vertical Total from Input Video Low Byte Register – 0x769....348
Dual Monitor Vertical Total from Input Video High Byte Register – 0x76A....348
Dual Monitor Vertical Total after Adjustment Byte Register – 0x76B....348
Dual Monitor Vertical Total after Adjustment High Register – 0x76C....349
Dual Monitor Single Box Register – 0x76D to 0x7AB....349
Dual Monitor Mouse Register – 0x7AC to 0x7BD....349
Horizontal Down Scaler Factor Low Byte Register – 0x7BE....349
Horizontal Down Scale Factor High Byte Register – 0x7BF....349
Vertical Down Scale Factor Low Byte Register – 0x7C0....349
Vertical Down Scaler Factor High Byte Register – 0x7C1....349
Position Down Scaler Enable Control Register – 0x7C2....349
OSD Blink Rate Register – 0x7C3....350
Dual Monitor OSG Register – 0x7C4 to 0x7F5....350
Dual Monitor Digital R Gain Register – 0x7F6....350
Dual Monitor Digital G Gain Register – 0x7F7....350
Dual Monitor Digital B Gain Register – 0x7F8....350
Dual Monitor R Offset Register – 0x7F9....350
Dual Monitor G Offset Register – 0x7FA....350
Dual Monitor B Offset Register – 0x7FB....351
DMA Controller (DMAC)....352
Overview....352
Features....352

Theory of Operation ..... 352
DMA Engine ..... 352
DRAM interface ..... 353
EXTERNAL DMA DREQ/DACK PROTOCOL ..... 353
Basic DMA Timing ..... 353
Examples ..... 355
Register Table ..... 357
Register Definition ..... 358
Display DRAM Source Vertical Position LSB Register – 0x230 ..... 358
Display DRAM Source Vertical Position MSB Register – 0x231 ..... 358
Display DRAM Destination Vertical Position LSB Register – 0x232 ..... 358
Display DRAM Destination Vertical Position MSB Register – 0x233 ..... 358
Display DRAM Vertical Transfer Size LSB Register – 0x234 ..... 358
Display DRAM Vertical Transfer Size MSB Register – 0x235 ..... 358
Display DRAM Source Horizontal Position LSB Register – 0x236 ..... 358
Display DRAM Source Horizontal Position MSB Register – 0x237 ..... 358
Display DRAM Destination Horizontal Position LSB Register – 0x238 ..... 359
Display DRAM Destination Horizontal Position MSB Register – 0x239 ..... 359
Display DRAM Horizontal Transfer Size LSB Register – 0x23a ..... 359
Display DRAM Horizontal Transfer Size MSB Register – 0x23b ..... 359
Display DRAM DMA Enable Register – 0x23c ..... 359
Host DMA Destination Vertical Position LSB Register – 0x240 ..... 359
Host DMA Destination Vertical Position MSB Register – 0x241 ..... 359
Host DMA Vertical Transfer Size LSB Register – 0x242 ..... 359
Host DMA Vertical Transfer Size MSB Register – 0x243 ..... 359
Host DMA Destination Horizontal Position LSB Register – 0x244 ..... 359
Host DMA Destination Horizontal Position MSB Register – 0x245 ..... 360
Host DMA Horizontal Transfer Size Register LSB – 0x246 ..... 360
Host DMA Control Register – 0x247 ..... 360
Host DMA Enable Register – 0x248 ..... 360
Host DMA Transfer Mode Control Register – 0x249 ..... 360
Host DMA Status Register – 0x24a ..... 361
OSG Total Transfer Size LSB Register 1 – 0x24b ..... 361
OSG Total Transfer Size LSB Register 2 – 0x24c ..... 361
OSG Total Transfer Size LSB Register 3 – 0x24d ..... 361
OSG Total Transfer Size MSB Register – 0x24e ..... 361
Host DMA Write Data – 0x25e, 0x260, 0x262, 0x264 ..... 361
Host DMA Display Pitch LSB Register – 0x268 ..... 362
Host DMA Display Pitch MSB Register – 0x269 ..... 362
Host DMA Start Address LSB Register 1 – 0x26a ..... 362
Host DMA Start Address LSB Register 2 – 0x26b ..... 362
Host DMA Start Address MSB Register – 0x26c ..... 362
Record DRAM Source Vertical Position LSB Register – 0x250 ..... 362
Record DRAM Source Vertical Position MSB Register – 0x251 ..... 362
Record DRAM Destination Vertical Position LSB Register – 0x252 ..... 362
Record DRAM Destination Vertical Position MSB Register – 0x253 ..... 363
Record DRAM Vertical Transfer Size LSB Register – 0x254 ..... 363
Record DRAM Vertical Transfer Size MSB Register – 0x255 ..... 363
Record DRAM Source Horizontal Position LSB Register – 0x256 ..... 363
Record DRAM Source Horizontal Position MSB Register – 0x257 ..... 363
Record DRAM Destination Horizontal Position LSB Register – 0x258 ..... 363
Record DRAM Destination Horizontal Position MSB Register – 0x259 ..... 363
Record DRAM Horizontal Transfer Size LSB Register – 0x25a ..... 363
Record DRAM Horizontal Transfer Size MSB Register – 0x25b ..... 363
Record DRAM DMA Enable Register – 0x25c ..... 364
Recording and Output unit ..... 365

Introduction ..... 365
Functional Description ..... 365
Picture Format Control ..... 366
Frame Rate Control Unit ..... 366
Record Channels Arrangement Example ..... 366
8 D1 channel frame interleaved real time output ..... 366
16 D1 channel field interleaved real time output ..... 367
16 channel mixed format real time output ..... 367
16 D1 channel frame interleaved real time output ..... 367
SPOT monitor memory space ..... 368
Record Postions in detail ..... 369
SPOT write buffer ..... 369
Recording Unit ..... 370
Introduction ..... 370
Record Out Format Control ..... 370
D1 Video Frame Data Arrange ..... 371
CIF Video Frame Data arrange ..... 371
16-Bit Video Frame Data arrange ..... 371
Record output Method ..... 371
Multi-mode Format Control ..... 371
Recording format example ..... 372
Record out channel output example ..... 373
Record port misc. items ..... 374
Special Output Format ..... 375
Record Port Capability Chart ..... 376
Multi-mode Record Port Programming Example ..... 376
Output Port ..... 377
Introduction ..... 377
8-bit mode ..... 379
16-bit mode ..... 380
16-bit component mode ..... 380
Field Switching mode ..... 382
32 buffer mode ..... 382
Output Pin Source Control ..... 382
Field Switching Quad mode ..... 383
Synchronization mode ..... 383
Multi-mode ..... 383
Conclusion ..... 384
Record out AC timing ..... 384
Register Table ..... 385
Registers Description ..... 391
Recording Buffer 1 Control Register A – 0xC00 ..... 391
Recording Buffer 1 Control Register B – 0xC10 ..... 391
Record Port 1 Control Register – 0xC20 ..... 392
Record Port 1 Source Control Register A – 0xC21 ..... 392
Record Port 1 Source Control Register B – 0xC22 ..... 392
Record Port 1 Custom Horizontal Control Register – 0xC23 ..... 392
Record Port 1 Custom Vertical Control Register – 0xC24 ..... 392
Record Port 2 Control Register – 0xC25 ..... 393
Record Port 2 Source Control Register A – 0xC26 ..... 393
Record Port 2 Source Control Register B – 0xC27 ..... 393
Record Port 2 Custom Horizontal Control Register – 0xC28 ..... 393
Record Port 2 Custom Vertical Control Register – 0xC29 ..... 393
Record Port 3 Control Register – 0xC2A ..... 394
Record Port 3 Source Control Register A – 0xC2B ..... 394
Record Port 3 Source Control Register B – 0xC2C ..... 394
Record Port 3 Custom Horizontal Control Register – 0xC2D ..... 394

Record Port 3 Custom Vertical Control Register – 0xC2E....394
Record Port 4 Control Register – 0xC2F....395
Record Port 4 Source Control Register A – 0xC30....395
Record Port 4 Source Control Register B – 0xC31....395
Record Port 4 Custom Horizontal Control Register – 0xC32....395
Record Port 4 Custom Vertical Control Register – 0xC33....395
Record Port 5 Control Register – 0xC34....396
Record Port 5 Source Control Address Register – 0xC35....396
Record Port 5 Source Control Data Register – 0xC36....396
Record Port 5 Source Number Register – 0xC37....396
Record Port 5 Custom Horizontal Control Register – 0xC38....396
Record Port 5 Custom Vertical Control Register – 0xC39....396
Record Port 6 Control Register – 0xC3A....397
Record Port 6 Source Control Address Register – 0xC3B....397
Record Port 6 Source Control Data Register – 0xC3C....397
Record Port 6 Source Number Register – 0xC3D....397
Record Port 6 Custom Horizontal Control Register – 0xC3E....397
Record Port 6 Custom Vertical Control Register – 0xC3F....397
Record Port 7 Control Register – 0xC40....398
Record Port 7 Source Control Address Register – 0xC41....398
Record Port 7 Source Control Data Register – 0xC42....398
Record Port 7 Source Number Register – 0xC43....398
Record Port 7 Custom Horizontal Control Register – 0xC44....398
Record Port 7 Custom Vertical Control Register – 0xC45....398
Record Port 8 Control Register – 0xC46....399
Record Port 8 Source Control Address Register – 0xC47....399
Record Port 8 Source Control Data Register – 0xC48....399
Record Port 8 Source Number Register – 0xC49....399
Record Port 8 Custom Horizontal Control Register – 0xC4A....399
Record Port 8 Custom Vertical Control Register – 0xC4B....399
Record Port 1-4 Source Control Register – 0xC4C....400
Record Port Operating Mode Select Register – 0xC4D....400
Output Pin Set 1 Control Register – 0xC4E....401
Output Pin Set 2 Control Register – 0xC4F....401
Output Pin Set 3 Control Register – 0xC50....402
Output Pin Set 4 Control Register – 0xC51....403
Network Port Control Register – 0xC52....403
Network Port Control Address Register – 0xC53....404
Network Port Control Data Register – 0xC54....404
Network Port Source Number Register – 0xC55....404
MISC. Control Register 1 – 0xC56....404
4D1 and QUAD 2nd Write Horizontal Margin Register – 0xC57....404
MISC. Control Register 2 – 0xC58....405
MISC. Control Register 3 – 0xC59....405
Record HDE Control Register – 0xC5A....405
Record VDE Control Register – 0xC5B....405
Record Buffer Freeze Control Register A – 0xC64....405
Record Buffer Freeze Control Register B – 0xC65....405
SPOT Buffer Freeze Control Register A – 0xC66....406
SPOT Buffer Freeze Control Register B – 0xC67....406
Custom Clock Control Register – 0xC68....406
Interlaced Frame Output Mode Control Register – 0xC69....406
'wr_page' Reference Selection Control Register A(Write buffer Index) – 0xC6A....406
'wr_page' Reference Selection Control Register B(Read Port Index) – 0xC6B....407
New FRSC Control Register B – 0xC6C....407
PB Ignore Bit Control Register – 0xC6D....408
PB Ignore Value Register 1 – 0xC6E....408

PB Ignore Value Register 2 – 0xC6F....408
Background Color for Cb – 0xC70....408
Background Color for Cr – 0xC71....409
Background Color for Y – 0xC72....409
Background On/off Control 1 for Record Write Buffer – 0xC73....409
Background On/off Control 2 for Record Write Buffer – 0xC74....409
Background On/off Control 1 for SPOT Write Buffer – 0xC75....409
Background On/off Control 2 for SPOT Write Buffer – 0xC76....409
New Non-real Time Field Switching Mode for Port 5 ~ Port 9 – 0xC77....409
New Field Switching Mode Field Select for Port 5 ~ Port 9 – 0xC78....410
Field Interleaved Read Mode with 4 Frames – 0xC79....410
New Non-real Time Field Switching Mode for Port 5 ~ Port 9 – 0xC7A....410
SPOT Recording Buffer 1 Control Register A – 0xC90....410
SPOT Recording Buffer 1 Control Register B – 0xCA0....411
Background Buffer Indication Register 1 – 0xCC8....411
Background Buffer Indication Register 2 – 0xCC9....411
Second Record Buffer DRAM Control Register 1 – 0xCCA....411
Second Record Buffer DRAM Control Register 2 – 0xCCB....412
Second SPOT DRAM Buffer Control Register 1 – 0xCCC....412
Second SPOT DRAM Buffer Control Register 2 – 0xCCD....412
Record Buffer Alternate Source Control Register 1 – 0xCCE....412
Record Buffer Alternate Source Control Register 2 – 0xCCF....412
Record Buffer Horizontal Offset Control Register – 0xCD0-DF....412
Record Buffer Vertical Offset Control Register – 0xCE0-EF....413
Special Component Record Mode Register – 0xCF0....413
Record Buffer Field Switching Record Mode Register I – 0xCF1....413
Record Buffer Field Switching Record Mode Register 2 – 0xCF2....413
Record Buffer Field Switching Record Mode Field Select Register I – 0xCF3....414
Record Buffer Field Switching Record Mode Field Select Register 2 – 0xCF4....414
Output Pin Set Source 2 Control Register A – 0xCF5....414
Output Pin Set Source 2 Control Register B – 0xCF6....415
Frame Rate Controller Update Register – 0xCF7....415
SPOT Buffer Field Switching Record Mode Register I – 0xCF8....415
SPOT Buffer Field Switching Record Mode Register 2 – 0xCF9....416
SPOT Buffer Field Switching Record Mode Field Select Register I – 0xCFA....416
SPOT Buffer Field Switching Record Mode Field Select Register 2 – 0xCFB....416
Write Buffer Control Register – 0xCFC....416
New FRSC Control Register A – 0xCFD....417
New Write Buffer Mapping Control Register– 0xCFE....417
Motion Dection Unit....418
Introduction....418
Motion Data Extraction....418
Mask and Detection Region Selection....419
Sensitivity Control....420
Level Sensitivity....420
Spatial Sensitivity....420
Temporal Sensitivity....420
Velocity Control....420
Blind Detection....422
Night Detection....422
Register Table....424
Registers Description....426
Motion Detection Cell Line 0 Mask Register 1 – 0x800....426
Motion Detection Cell Line 0 Mask Register 2 – 0x801....426
Motion Detection Cell Line 1 Mask Register 1 – 0x802....426
Motion Detection Cell Line 1 Mask Register 2 – 0x803....426

Motion Detection Cell Line 2 Mask Register 1 – 0x804 ..... 426
Motion Detection Cell Line 2 Mask Register 2 – 0x805 ..... 426
Motion Detection Cell Line 3 Mask Register 1 – 0x806 ..... 426
Motion Detection Cell Line 3 Mask Register 2 – 0x807 ..... 426
Motion Detection Cell Line 4 Mask Register 1 – 0x808 ..... 426
Motion Detection Cell Line 4 Mask Register 2 – 0x809 ..... 427
Motion Detection Cell Line 5 Mask Register 1 – 0x80A ..... 427
Motion Detection Cell Line 5 Mask Register 2 – 0x80B ..... 427
Motion Detection Cell Line 6 Mask Register 1 – 0x80C ..... 427
Motion Detection Cell Line 6 Mask Register 2 – 0x80D ..... 427
Motion Detection Cell Line 7 Mask Register 1 – 0x80E ..... 427
Motion Detection Cell Line 7 Mask Register 2 – 0x80F ..... 427
Motion Detection Cell Line 8 Mask Register 1 – 0x810 ..... 427
Motion Detection Cell Line 8 Mask Register 2 – 0x811 ..... 427
Motion Detection Cell Line 9 Mask Register 1 – 0x812 ..... 427
Motion Detection Cell Line 9 Mask Register 2 – 0x813 ..... 428
Motion Detection Cell Line 10 Mask Register 1 – 0x814 ..... 428
Motion Detection Cell Line 10 Mask Register 2 – 0x815 ..... 428
Motion Detection Cell Line 11 Mask Register 1 – 0x816 ..... 428
Motion Detection Cell Line 11 Mask Register 2 – 0x817 ..... 428
Motion Detection Control Register 1 – 0x818 ..... 428
Motion Detection Control Register 2 – 0x819 ..... 428
Motion Detection Control Register 3 – 0x81A ..... 428
Motion Detection Control Register 4 – 0x81B ..... 429
Motion Detection Control Register 5 – 0x81E ..... 429
Motion Detection Control Register 6 – 0x81F ..... 429
Motion Detection Cell Line 0 Result Register 1 – 0x820 ..... 429
Motion Detection Cell Line 0 Result Register 2 – 0x821 ..... 429
Motion Detection Cell Line 1 Result Register 1 – 0x822 ..... 430
Motion Detection Cell Line 1 Result Register 2 – 0x823 ..... 430
Motion Detection Cell Line 2 Result Register 1 – 0x824 ..... 430
Motion Detection Cell Line 2 Result Register 2 – 0x825 ..... 430
Motion Detection Cell Line 3 Result Register 1 – 0x826 ..... 430
Motion Detection Cell Line 3 Result Register 2 – 0x827 ..... 430
Motion Detection Cell Line 4 Result Register 1 – 0x828 ..... 430
Motion Detection Cell Line 4 Result Register 2 – 0x829 ..... 430
Motion Detection Cell Line 5 Result Register 1 – 0x82A ..... 430
Motion Detection Cell Line 5 Result Register 2 – 0x82B ..... 430
Motion Detection Cell Line 6 Result Register 1 – 0x82C ..... 431
Motion Detection Cell Line 6 Result Register 2 – 0x82D ..... 431
Motion Detection Cell Line 7 Result Register 1 – 0x82E ..... 431
Motion Detection Cell Line 7 Result Register 2 – 0x82F ..... 431
Motion Detection Cell Line 8 Result Register 1 – 0x830 ..... 431
Motion Detection Cell Line 8 Result Register 2 – 0x831 ..... 431
Motion Detection Cell Line 9 Result Register 1 – 0x832 ..... 431
Motion Detection Cell Line 9 Result Register 2 – 0x833 ..... 431
Motion Detection Cell Line 10 Result Register 1 – 0x834 ..... 431
Motion Detection Cell Line 10 Result Register 2 – 0x835 ..... 431
Motion Detection Cell Line 11 Result Register 1 – 0x836 ..... 432
Motion Detection Cell Line 11 Result Register 2 – 0x837 ..... 432
Channel ID Encoder Unit ..... 433
Channel ID Encoder ..... 433
Channel ID Definition ..... 433
Auto Channel ID ..... 433
Detection Channel ID ..... 434
User Channel ID ..... 435

Channel ID Output Method ..... 435
Digital Channel ID ..... 436
Analog Channel ID ..... 438
Channel ID in Active Region ..... 438
Registers Table ..... 440
Registers Description ..... 446
PB1 Channel ID Enc Misc Register – 0xD00 ..... 446
PB1 Channel ID Enc Enable Register – 0xD01 ..... 446
PB1 Channel ID Enc Horizontal Offset Register – 0xD02 ..... 447
PB1 Channel ID Enc High Value Register – 0xD03 ..... 447
PB1 Channel ID Enc Low Value Register – 0xD04 ..... 447
PB1 Odd Field Vertical Line Offset Register – 0xD05 ..... 447
PB1 Channel ID Enc Man0 LSB Register – 0xD06 ..... 447
PB1 Channel ID Enc Man0 MSB Register – 0xD07 ..... 448
PB1 Channel ID Enc Man1 LSB Register – 0xD08 ..... 448
PB1 Channel ID Enc Man1 SB Register – 0xD09 ..... 448
PB1 Channel ID Enc Man2 LSB Register – 0xD0A ..... 448
PB1 Channel ID Enc Man2 MSB Register – 0xD0B ..... 448
PB1 Channel ID Enc Man3 LSB Register – 0xD0C ..... 448
PB1 Channel ID Enc Man3 MSB Register – 0xD0D ..... 448
PB1 Channel ID Enc Man4 LSB Register – 0xD0E ..... 449
PB1 Channel ID Enc Man4 MSB Register – 0xD0F ..... 449
PB1 Channel ID Enc Man5 LSB Register – 0xD10 ..... 449
PB1 Channel ID Enc Man5 MSB Register – 0xD11 ..... 449
PB1 Even Field Vertical Offset Register – 0xD90 ..... 449
Active Area Channel ID Enable Register 1 – 0xDAB ..... 450
Active Area Channel ID Enable Register 2 – 0xDAC ..... 450
Active Area Channel ID in First Frame Register 1 – 0xDAD ..... 450
Active Area Channel ID in First Frame Register 2 – 0xDAE ..... 450
Active Area Channel ID Data Register – 0xDAF ..... 450
Motion Channel Selection Register – 0xDB0 ..... 451
Field selection Register – 0xDB5 ..... 451
Field selection Register – 0xDB6 ..... 451
OSD for Recording Path ..... 452
Introduction ..... 452
Features ..... 452
Description ..... 452
Layers ..... 452
Resolutions ..... 452
Timing Diagram ..... 454
Vertical timing ..... 454
Horizontal timing ..... 454
Font Composition ..... 455
Display Channel Information ..... 456
Date and Time ..... 456
Title ..... 456
Status ..... 457
SRAM Allocation ..... 457
Font SRAM ..... 457
Display SRAM ..... 458
Privacy Windows ..... 458
Introduction ..... 458
Masking Effect ..... 458
Mosaic Effect ..... 459
Registers Table ..... 460
Registers Description ..... 464

ROSD Enable Register – 0xE00 ..... 464
ROSD Time and Date Enable Register – 0xE01 ..... 464
ROSD Title Enable Register – 0xE02 ..... 464
ROSD Status Enable Register – 0xE03 ..... 465
ROSD Boundary Enable Register – 0xE04 ..... 465
ROSD Background Enable Register – 0xE05 ..... 465
ROSD Mix Control Enable Register – 0xE06 ..... 466
ROSD Boundary Width Register – 0xE07 ..... 466
ROSD Font Size Register – 0xE08 ..... 466
ROSD Time Horizontal Postion for Full Mode Low Byte Register – 0xE09 ..... 467
ROSD Time Horizontal Postion for Full Mode High Byte Register – 0xE0A ..... 467
ROSD Time Horizontal Left Postion for Half Mode Low Byte Register – 0xE0B ..... 467
ROSD Time Horizontal Left Postion for Half Mode High Byte Register – 0xE0C ..... 467
ROSD Time Horizontal Right Postion for Half Mode Low Byte Register – 0xE0D ..... 467
ROSD Time Horizontal Right Postion for Half Mode High Byte Register – 0xE0E ..... 468
ROSD Time Vertical Postion for Full Mode Low Byte Register – 0xE0F ..... 468
ROSD Time Vertical Postion for Full Mode High Byte Register – 0xE10 ..... 468
ROSD Time Vertical Top Postion for Half Mode Low Byte Register – 0xE11 ..... 468
ROSD Time Vertical Top Postion for Half Mode High Byte Register – 0xE12 ..... 468
ROSD Time Vertical Bottom Postion for Half Mode Low Byte Register – 0xE13 ..... 468
ROSD Time Vertical Bottom Postion for Half Mode High Byte Register – 0xE14 ..... 468
ROSD Title Horizontal Postion for Full Mode Low Byte Register – 0xE15 ..... 469
ROSD Title Horizontal Postion for Full Mode High Byte Register – 0xE16 ..... 469
ROSD Title Horizontal Left Postion for Half Mode Low Byte Register – 0xE17 ..... 469
ROSD Title Horizontal Left Postion for Half Mode High Byte Register – 0xE18 ..... 469
ROSD Title Horizontal Right Postion for Half Mode Low Byte Register – 0xE19 ..... 469
ROSD Title Horizontal Right Postion for Half Mode High Byte Register – 0xE1A ..... 469
ROSD Title Vertical Postion for Full Mode Low Byte Register – 0xE1B ..... 469
ROSD Title Vertical Postion for Full Mode High Byte Register – 0xE1C ..... 470
ROSD Title Vertical Top Postion for Half Mode Low Byte Register – 0xE1D ..... 470
ROSD Title Vertical Top Postion for Half Mode High Byte Register – 0xE1E ..... 470
ROSD Title Vertical Bottom Postion for Half Mode Low Byte Register – 0xE1F ..... 470
ROSD Title Vertical Bottom Postion for Half Mode High Byte Register – 0xE20 ..... 470
ROSD Status Horizontal Postion for Full Mode Low Byte Register – 0xE21 ..... 470
ROSD Status Horizontal Postion for Full Mode High Byte Register – 0xE22 ..... 470
ROSD Status Horizontal Left Postion for Half Mode Low Byte Register – 0xE23 ..... 471
ROSD Status Horizontal Left Postion for Half Mode High Byte Register – 0xE24 ..... 471
ROSD Status Horizontal Right Postion for Half Mode Low Byte Register – 0xE25 ..... 471
ROSD Status Horizontal Right Postion for Half Mode High Byte Register – 0xE26 ..... 471
ROSD Status Vertical Postion for Full Mode Low Byte Register – 0xE27 ..... 471
ROSD Status Vertical Postion for Full Mode High Byte Register – 0xE28 ..... 471
ROSD Status Vertical Top Postion for Half Mode Low Byte Register – 0xE29 ..... 471
ROSD Status Vertical Top Postion for Half Mode High Byte Register – 0xE2A ..... 472
ROSD Status Vertical Bottom Postion for Half Mode Low Byte Register – 0xE2B ..... 472
ROSD Status Vertical Bottom Postion for Half Mode High Byte Register – 0xE2C ..... 472
ROSD Font Y Color 1 Register – 0xE2D ..... 472
ROSD Font Cb Color 1 Register – 0xE2E ..... 472
ROSD Font Cr Color 1 Register – 0xE2F ..... 472
ROSD Font Y Color 2 Register – 0xE30 ..... 472
ROSD Font Cb Color 2 Register – 0xE31 ..... 473
ROSD Font Cr Color 2 Register – 0xE32 ..... 473
ROSD Font Y Color 3 Register – 0xE33 ..... 473
ROSD Font Cb Color 3 Register – 0xE34 ..... 473
ROSD Font Cr Color 3 Register – 0xE35 ..... 473
ROSD Background Y Color Register – 0xE36 ..... 473
ROSD Background Cb Color Register – 0xE37 ..... 473
ROSD Background Cr Color Register – 0xE38 ..... 474

ROSD Boundary Y Color Register – 0xE39 ..... 474
ROSD Boundary Cb Color Register – 0xE3A ..... 474
ROSD Boundary Cr Color Register – 0xE3B ..... 474
ROSD Font RAM Address Low Byte Register – 0xE3C ..... 474
ROSD Font RAM Address High Byte Register – 0xE3D ..... 474
ROSD Font RAM Data Low Byte Register – 0xE3E ..... 474
ROSD Font RAM Data High Byte Register – 0xE3F ..... 475
ROSD Display RAM Address Register – 0xE40 ..... 475
ROSD Display RAM Address Register – 0xE41 ..... 475
ROSD Display RAM Data Register – 0xE42 ..... 475
ROSD Color Bar Enable Register – 0xE43 ..... 475
ROSD Enable for Port 9 Register – 0xE44 ..... 476
ROSD Colbar Veritical Black Bar Position Register – 0xE45 ..... 476
ROSD Field Switching Control Register – 0xE46 ..... 476
Privacy Window Overflow Register 1 – 0xE47 ..... 476
Privacy Window Overflow Register 2 – 0xE48 ..... 476
Privacy Window Shadow Set Control Register – 0xE4F ..... 476
Privacy Window 1 Horizontal Start Register – 0xE50 – 0xE5F ..... 477
Privacy Window 1 Vertical Start Register – 0xE60 – 0xE6F ..... 477
Privacy Window 1 Control Register – 0xE70 – 0xE7F ..... 477
Privacy Window 2 Horizontal Start Register – 0xE80 – 0xE8F ..... 478
Privacy Window 2 Vertical Start Register – 0xE90 – 0xE9F ..... 478
Privacy Window 2 Control Register – 0xEA0 – 0xEAF ..... 479
Privacy Window 3 Horizontal Start Register – 0xEB0 – 0xEBF ..... 479
Privacy Window 3 Vertical Start Register – 0xEC0 – 0xECF ..... 479
Privacy Window 3 Control Register – 0xED0 – 0xEDF ..... 480
Privacy Window 4 Horizontal Start Register – 0xDD0 – 0xDDF ..... 480
Privacy Window 4 Vertical Start Register – 0xDE0 – 0xDEF ..... 480
Privacy Window 4 Control Register – 0xDF0 – 0xDFF ..... 481
SPOT Controller Unit ..... 482
Introduction ..... 482
Features ..... 482
SPOT Controller Architecture ..... 482
Memory access control ..... 482
Frame Data Synchonization ..... 482
SPOT Controller Block Diagram ..... 483
Memory Interface ..... 483
Rotation control ..... 484
FIFO Interface ..... 484
HV Timing Control ..... 484
Down Scalar ..... 485
LBUF Control ..... 485
OSD, Background and Border control ..... 485
TV Encoder ..... 486
Theory of Operation ..... 486
Registers Table ..... 488
Registers ..... 500
SPOT1 Control Register– 0xF00 ..... 500
SPOT1 Control Register– 0xF01 ..... 500
SPOT1 Control Register– 0xF02 ..... 500
SPOT1 Control Register– 0xF03 ..... 500
SPOT1 Control Register– 0xF04 ..... 500
SPOT1 Control Register– 0xF05 ..... 500
SPOT1 Control Register– 0xF06 ..... 500
SPOT1 Control Register– 0xF07 ..... 501
SPOT1 Control Register– 0xF08 ..... 501

SPOT1 Control Register– 0xF09 ..... 501
SPOT1 Control Register– 0xF0A ..... 501
SPOT1 Control Register– 0xF0B ..... 502
SPOT1 Control Register– 0xF0C ..... 502
SPOT1 Control Register – 0xF0D ..... 502
SPOT1 Control Register – 0xF0E ..... 502
SPOT1 Control Register – 0xF0F ..... 503
SPOT1 Control Register – 0xF10 ..... 503
SPOT1 Control Register – 0xF11 ..... 503
SPOT1 Control Register– 0xF12 ..... 503
SPOT1 Control Register – 0xF13 ..... 504
SPOT1 Control Register– 0xF14 ..... 504
SPOT1 Control Register – 0xF15 ..... 504
SPOT1 Control Register– 0xF16 ..... 504
SPOT1 Control Register– 0xF17 ..... 504
SPOT1 Control Register– 0xF18 ..... 505
SPOT1 Control Register– 0xF19 ..... 506
SPOT1 Control Register– 0xF1A ..... 506
SPOT1 Control Register– 0xF1B ..... 506
SPOT1 Control Register– 0xF1C ..... 506
SPOT1 Control Register– 0xF1D ..... 506
SPOT1 Control Register– 0xF1E ..... 507
SPOT1 Control Register– 0xF1F ..... 507
SPOT1 Control Register– 0xF20 ..... 507
SPOT Control Register– 0xF21 ..... 507
SPOT Control Register– 0xF22 ..... 508
SPOT Control Register– 0xF23 ..... 508
SPOT Control Register– 0xF24 ..... 508
SPOT Control Register– 0xF25 ..... 508
SPOT Control Register– 0xF26 ..... 508
SPOT Control Register– 0xF27 ..... 508
SPOT Control Register – 0xF28 ..... 509
SPOT Control Register – 0xF29 ..... 509
SPOT2 Control Register – 0xF2A ..... 509
SPOT2 Control Register – 0xF2B ..... 509
SPOT3 Control Register – 0xF2C ..... 509
SPOT3 Control Register – 0xF2D ..... 509
SPOT4 Control Register – 0xF2E ..... 509
SPOT4 Control Register – 0xF2F ..... 510
SPOT1 Control Register – 0xF30 ..... 510
SPOT1 Control Register – 0xF31 ..... 511
SPOT1 OSD Font Size Register – 0xF32 ..... 512
SPOT1 Osd channel Number Horizontal Position Low Byte Register – 0xF33 ..... 512
SPOT1 Osd channel Number Horizontal Position High Byte Register – 0xF34 ..... 512
SPOT1 Osd channel Number Vertical Position Low Byte Register – 0xF35 ..... 512
SPOT1 Osd channel Number Horizontal Position High Byte Register – 0xF36 ..... 513
SPOT1 Osd channel Picture Horizontal Position Low Byte Register – 0xF37 ..... 513
SPOT1 Osd channel Picture Horizontal Position High Byte Register – 0xF38 ..... 513
SPOT1 Osd channel Picture Vertical Position Low Byte Register – 0xF39 ..... 513
SPOT1 Osd channel Picture Horizontal Position High Byte Register – 0xF3A ..... 514
SPOT1 OSD Time and Date Horizontal Position Low Byte Register – 0xF3B ..... 514
SPOT1 OSD Time and Date Horizontal Position High Byte Register – 0xF3C ..... 514
SPOT1 OSD Time and Date Vertical Position Low Byte Register – 0xF3D ..... 514
SPOT1 OSD Time and Date Horizontal Position High Byte Register – 0xF3E ..... 515
SPOT1 OSD Title Horizontal Position Low Byte Register – 0xF3F ..... 515
SPOT1 OSD Title Horizontal Position High Byte Register – 0xF40 ..... 515
SPOT1 OSD Title Vertical Position Low Byte Register – 0xF41 ..... 515

SPOT1 OSD Title Horizontal Position High Byte Register – 0xF42 ..... 515
SPOT OSD Font Red Color 1 Register – 0xF43 ..... 515
SPOT OSD Font Green Color 1 Register – 0xF44 ..... 515
SPOT OSD Font Blue Color 1 Register – 0xF45 ..... 516
SPOT OSD Font Red Color 2 Register – 0xF46 ..... 516
SPOT OSD Font Green Color 2 Register – 0xF47 ..... 516
SPOT OSD Font Blue Color 2 Register – 0xF48 ..... 516
SPOT OSD Font Red Color 3 Register – 0xF49 ..... 516
SPOT OSD Font Green Color 3 Register – 0xF4A ..... 516
SPOT OSD Font Blue Color 3 Register – 0xF4B ..... 516
SPOT OSD Picture Red Color 0 Register – 0xF4C ..... 517
SPOT OSD Picture Green Color 0 Register – 0xF4D ..... 517
SPOT OSD Picture Blue Color 0 Register – 0xF4E ..... 517
SPOT OSD Picture Red Color 1 Register – 0xF4F ..... 517
SPOT OSD Picture Green Color 1 Register – 0xF50 ..... 517
SPOT OSD Picture Blue Color 1 Register – 0xF51 ..... 517
SPOT OSD Picture Red Color 2 Register – 0xF52 ..... 517
SPOT OSD Picture Green Color 2 Register – 0xF53 ..... 518
SPOT OSD Picture Blue Color 2 Register – 0xF54 ..... 518
SPOT OSD Picture Red Color 3 Register – 0xF55 ..... 518
SPOT OSD Picture Green Color 3 Register – 0xF56 ..... 518
SPOT OSD Picture Blue Color 3 Register – 0xF57 ..... 518
SPOT1 OSD Font and Picture RAM Address Low Byte Register – 0xF58 ..... 518
SPOT1 OSD Font and Picture RAM Address High Byte Register – 0xF59 ..... 518
SPOT1 OSD Font and Picture RAM Data Low Byte Register – 0xF5A ..... 519
SPOT1 OSD Font and Picture RAM Data High Byte Register – 0xF5B ..... 519
SPOT1 OSD Display RAM Address Register – 0xF5C ..... 519
SPOT1 OSD Display RAM Data Register – 0xF5D ..... 519
SPOT DISP_PITCH Register – 0xF5E ..... 519
SPOT ADJUSTMENT ENABLE Register – 0xF5F ..... 519
SPOT2 Control Register – 0xF60 ..... 520
SPOT2 Control Register – 0xF61 ..... 520
SPOT1 Control Register – 0xF62 ..... 520
SPOT2 Control Register – 0xF63 ..... 520
SPOT2 Control Register – 0xF64 ..... 520
SPOT2 Control Register – 0xF65 ..... 520
SPOT2 Control Register – 0xF66 ..... 521
SPOT2 Control Register – 0xF67 ..... 521
SPOT2 Control Register – 0xF68 ..... 521
SPOT2 Control Register – 0xF69 ..... 521
SPOT2 Control Register – 0xF6A ..... 522
SPOT2 Control Register – 0xF6B ..... 522
SPOT2 Control Register – 0xF6C ..... 522
SPOT2 Control Register – 0xF6D ..... 522
SPOT2 Control Register – 0xF6E ..... 522
SPOT2 Control Register – 0xF6F ..... 523
SPOT2 Control Register – 0xF70 ..... 523
SPOT2 Control Register – 0xF71 ..... 523
SPOT2 Control Register – 0xF72 ..... 523
SPOT2 Control Register – 0xF73 ..... 524
SPOT2 Control Register – 0xF74 ..... 524
SPOT2 Control Register – 0xF75 ..... 524
SPOT2 Control Register – 0xF76 ..... 524
SPOT2 Control Register – 0xF77 ..... 524
SPOT2 Control Register – 0xF78 ..... 525
SPOT2 Control Register – 0xF79 ..... 526
SPOT2 Control Register – 0xF7A ..... 526

SPOT2 Control Register – 0xF7B ..... 526
SPOT2 Control Register – 0xF7C ..... 526
SPOT2 Control Register – 0xF7D ..... 527
SPOT2 Control Register – 0xF7E ..... 527
SPOT2 Control Register – 0xF7F ..... 527
SPOT2 Control Register – 0xF80 ..... 527
SPOT2 Control Register – 0xF81 ..... 528
SPOT2 Control Register – 0xF82 ..... 528
SPOT2 OSD Font Size Register – 0xF83 ..... 529
SPOT2 OSD Time and Date Horizontal Position Low Byte Register – 0xF84 ..... 529
SPOT2 OSD Time and Date Horizontal Position High Byte Register – 0xF85 ..... 530
SPOT2 OSD Time and Date Vertical Position Low Byte Register – 0xF86 ..... 530
SPOT2 OSD Time and Date Vertical Position High Byte Register – 0xF87 ..... 530
SPOT2 OSD Title Horizontal Position Low Byte Register – 0xF88 ..... 530
SPOT2 OSD Title Horizontal Position High Byte Register – 0xF89 ..... 530
SPOT2 OSD Title Vertical Position Low Byte Register – 0xF8A ..... 530
SPOT2 OSD Title Horizontal Position High Byte Register – 0xF8B ..... 530
SPOT2 OSD Font and Picture RAM Address Low Byte Register – 0xF8C ..... 531
SPOT2 OSD Font and Picture RAM Address High Byte Register – 0xF8D ..... 531
SPOT2 OSD Font and Picture RAM Data Low Byte Register – 0xF8E ..... 531
SPOT2 OSD Font and Picture RAM Data High Byte Register – 0xF8F ..... 531
SPOT3 Control Register – 0xF90 ..... 531
SPOT3 Control Register – 0xF91 ..... 531
SPOT3 Control Register – 0xF92 ..... 532
SPOT3 Control Register – 0xF93 ..... 532
SPOT3 Control Register – 0xF94 ..... 532
SPOT3 Control Register – 0xF95 ..... 532
SPOT3 Control Register – 0xF96 ..... 532
SPOT3 Control Register – 0xF97 ..... 533
SPOT3 Control Register – 0xF98 ..... 533
SPOT3 Control Register – 0xF99 ..... 533
SPOT3 Control Register – 0xF9A ..... 533
SPOT3 Control Register – 0xF9B ..... 533
SPOT3 Control Register – 0xF9C ..... 534
SPOT3 Control Register – 0xF9D ..... 534
SPOT3 Control Register – 0xF9E ..... 534
SPOT3 Control Register – 0XF9F ..... 534
SPOT3 Control Register – 0xFA0 ..... 535
SPOT3 Control Register – 0xFA1 ..... 535
SPOT3 Control Register – 0xFA2 ..... 535
SPOT3 Control Register – 0xFA3 ..... 535
SPOT3 Control Register – 0xFA4 ..... 536
SPOT3 Control Register – 0xFA5 ..... 536
SPOT3 Control Register – 0xFA6 ..... 536
SPOT3 Control Register – 0xFA7 ..... 536
SPOT3 Control Register – 0xFA8 ..... 536
SPOT3 Control Register – 0xFA9 ..... 537
SPOT3 Control Register – 0xFAA ..... 538
SPOT3 Control Register – 0xFAB ..... 538
SPOT3 Control Register – 0xFAC ..... 538
SPOT3 Control Register – 0xFAD ..... 538
SPOT3 Control Register – 0xFAE ..... 538
SPOT3 Control Register – 0xFAF ..... 539
SPOT3 Control Register – 0xFB0 ..... 539
SPOT3 Control Register – 0xFB1 ..... 539
SPOT3 Control Register – 0xFB2 ..... 540
SPOT3 OSD Font Size Register – 0xFB3 ..... 541

SPOT3 OSD Time and Date Horizontal Position Low Byte Register – 0xFB4 ..... 541
SPOT3 OSD Time and Date Horizontal Position High Byte Register – 0xFB5 ..... 541
SPOT3 OSD Time and Date Vertical Position Low Byte Register – 0xFB6 ..... 541
SPOT3 OSD Time and Date Vertical Position High Byte Register – 0xFB7 ..... 542
SPOT3 OSD Title Horizontal Position Low Byte Register – 0xFB8 ..... 542
SPOT3 OSD Title Horizontal Position High Byte Register – 0xFB9 ..... 542
SPOT3 OSD Title Vertical Position Low Byte Register – 0xFBA ..... 542
SPOT3 OSD Title Horizontal Position High Byte Register – 0xFBB ..... 542
SPOT3 OSD Font and Picture RAM Address Low Byte Register – 0xFBC ..... 542
SPOT3 OSD Font and Picture RAM Address High Byte Register – 0xFBD ..... 542
SPOT3 OSD Font and Picture RAM Data Low Byte Register – 0xFBE ..... 543
SPOT3 OSD Font and Picture RAM Data High Byte Register – 0xFBF ..... 543
SPOT4 Control Register – 0xFC0 ..... 543
SPOT4 Control Register – 0xFC1 ..... 543
SPOT4 Control Register – 0xFC2 ..... 543
SPOT4 Control Register – 0xFC3 ..... 543
SPOT4 Control Register – 0xFC4 ..... 543
SPOT4 Control Register – 0xFC5 ..... 544
SPOT4 Control Register – 0xFC6 ..... 544
SPOT4 Control Register – 0xFC7 ..... 544
SPOT4 Control Register – 0xFC8 ..... 544
SPOT4 Control Register – 0xFC9 ..... 545
SPOT4 Control Register – 0xFCA ..... 545
SPOT4 Control Register – 0xFCB ..... 545
SPOT4 Control Register – 0xFCC ..... 545
SPOT4 Control Register – 0xFCD ..... 546
SPOT4 Control Register – 0xFCE ..... 546
SPOT4 Control Register – 0XFCF ..... 546
SPOT4 Control Register – 0xFD0 ..... 546
SPOT4 Control Register – 0xFD1 ..... 546
SPOT4 Control Register – 0xFD2 ..... 547
SPOT4 Control Register – 0xFD3 ..... 547
SPOT4 Control Register – 0xFD4 ..... 547
SPOT4 Control Register – 0xFD5 ..... 547
SPOT4 Control Register – 0xFD6 ..... 547
SPOT4 Control Register – 0xFD7 ..... 548
SPOT4 Control Register – 0xFD8 ..... 548
SPOT4 Control Register – 0xFD9 ..... 549
SPOT4 Control Register – 0xFDA ..... 549
SPOT4 Control Register – 0xFDB ..... 549
SPOT4 Control Register – 0xFDC ..... 550
SPOT4 Control Register – 0xFDD ..... 550
SPOT4 Control Register – 0xFDE ..... 550
SPOT4 Control Register – 0xFDF ..... 550
SPOT4 Control Register – 0xFE0 ..... 551
SPOT4 Control Register – 0xFE1 ..... 551
SPOT4 Control Register – 0xFE2 ..... 552
SPOT4 OSD Font Size Register – 0xFE3 ..... 553
SPOT4 OSD Time and Date Horizontal Position Low Byte Register – 0xFE4 ..... 553
SPOT4 OSD Time and Date Horizontal Position High Byte Register – 0xFE5 ..... 553
SPOT4 OSD Time and Date Vertical Position Low Byte Register – 0xFE6 ..... 553
SPOT4 OSD Time and Date Vertical Position High Byte Register – 0xFE7 ..... 554
SPOT4 OSD Title Horizontal Position Low Byte Register – 0xFE8 ..... 554
SPOT4 OSD Title Horizontal Position High Byte Register – 0xFE9 ..... 554
SPOT4 OSD Title Vertical Position Low Byte Register – 0xFEA ..... 554
SPOT4 OSD Title Horizontal Position High Byte Register – 0xFEB ..... 554
SPOT4 OSD Font and Picture RAM Address Low Byte Register – 0xFEC ..... 554

SPOT4 OSD Font and Picture RAM Address High Byte Register – 0xFED ..... 554
SPOT4 OSD Font and Picture RAM Data Low Byte Register – 0xFEE ..... 555
SPOT4 OSD Font and Picture RAM Data High Byte Register – 0xFEF ..... 555
SPOT Channel0 position mapping Register – 0XFF0 ..... 555
SPOT Channel1 position mapping Register – 0XFF1 ..... 556
SPOT Channel2 position mapping Register – 0XFF2 ..... 556
SPOT Channel3 position mapping Register – 0XFF3 ..... 557
SPOT Channel4 position mapping Register – 0XFF4 ..... 558
SPOT Channel5 position mapping Register – 0XFF5 ..... 558
SPOT Channel6 position mapping Register – 0XFF6 ..... 559
SPOT Channel7 position mapping Register – 0XFF7 ..... 560
SPOT Channel8 position mapping Register – 0XFF8 ..... 560
SPOT Channel9 position mapping Register – 0XFF9 ..... 561
SPOT Channel10 position mapping Register – 0XFFA ..... 562
SPOT Channel11 position mapping Register – 0XFFB ..... 562
SPOT Channel12 position mapping Register – 0XFFC ..... 563
SPOT Channel13 position mapping Register – 0XFFD ..... 564
SPOT Channel14 position mapping Register – 0XFFE ..... 564
SPOT Channel15 position mapping Register – 0XFFF ..... 565
Bitmap OSD for SPOT ..... 566
Introduction ..... 566
Features ..... 566
Theory of Operation ..... 566
Register Table ..... 568
Registers Definition ..... 571
Bitmap source vertical postion Register 1 – 0x080 ..... 571
Bitmap source vertical postion Register 2 – 0x081 ..... 571
Bitmap source horizontal position Register 1 – 0x082 ..... 571
Bitmap source horizontal position Register 2 – 0x083 ..... 571
Bitmap source vertical Size Register 1 – 0x088 ..... 571
Bitmap source vertical Size Register 2 – 0x089 ..... 571
Bitmap source horizontal Size Register – 0x08a ..... 571
SPOT1 Bitmap Destination vertical postion Register 1 – 0x084 ..... 571
SPOT1 Bitmap Destination vertical postion Register 2 – 0x085 ..... 572
SPOT1 Bitmap Destination horizontal position Register 3 – 0x086 ..... 572
SPOT1 Bitmap Destination horizontal position Register 4 – 0x087 ..... 572
SPOT1 Bitmap Destination vertical Size Register 1 – 0x08b ..... 572
SPOT1 Bitmap Destination vertical Size Register 2 – 0x08c ..... 572
SPOT1 Bitmap Destination horizontal Size Register 1 – 0x08d ..... 572
SPOT1 Bitmap Start Line Register 1 – 0x08e ..... 572
SPOT1 Bitmap Start Line Register 2 – 0x08f ..... 572
SPOT1 Bitmap Start Pixel Register 1 – 0x090 ..... 572
SPOT1 Bitmap Start Pixel Register 2 – 0x091 ..... 572
SPOT1 Bitmap Control Register 1 – 0x092 ..... 573
SPOT1 Bitmap Control Register 2 – 0x093 ..... 573
SPOT1 Bitmap Control Register 3 – 0x094 ..... 573
Bitmap DMA enable Control Register 3 – 0x095 ..... 573
SPOT1 Bitmap Control Register 3 – 0x096 ..... 574
SPOT1 Color look-up table1.Y – 0x097 ..... 574
SPOT1 Color look-up table1.Cb – 0x098 ..... 574
SPOT1 Color look-up table1.Cr – 0x099 ..... 574
SPOT1 Color look-up table2.Y – 0x09a ..... 574
SPOT1 Color look-up table2.Cb – 0x09b ..... 574
SPOT1 Color look-up table2.Cr – 0x09c ..... 574
SPOT1 Color look-up table3.Y – 0x09d ..... 574
SPOT1 Color look-up table3.Cb – 0x09e ..... 574

SPOT1 Color look-up table3.Cr – 0x09f ..... 574
SPOT2 Bitmap Destination vertical postion Register 1 – 0x0a4 ..... 575
SPOT2 Bitmap Destination vertical postion Register 2 – 0x0a5 ..... 575
SPOT2 Bitmap Destination horizontal position Register 1 – 0x0a6 ..... 575
SPOT2 Bitmap Destination horizontal position Register 2 – 0x0a7 ..... 575
SPOT2 Bitmap Destination vertical Size Register 1 – 0x0ab ..... 575
SPOT2 Bitmap Destination vertical Size Register 2 – 0x0ac ..... 575
SPOT2 Bitmap Destination horizontal Size Register – 0x0ad ..... 575
SPOT2 Bitmap Start Line Register 1 – 0x0ae ..... 575
SPOT2 Bitmap Start Line Register 2 – 0x0af ..... 576
SPOT2 Bitmap Start Pixel Register 1 – 0x0b0 ..... 576
SPOT2 Bitmap Start Pixel Register 2 – 0x0b1 ..... 576
SPOT2 Bitmap Control Register 1 – 0x0b2 ..... 576
SPOT2 Bitmap Control Register 2 – 0x0b3 ..... 576
SPOT2 Bitmap Control Register 3 – 0x0b4 ..... 577
SPOT2 Bitmap Control Register 3 – 0x0b6 ..... 577
SPOT2 Color look-up table1.Y – 0x0b7 ..... 577
SPOT2 Color look-up table1.Cb – 0x0b8 ..... 577
SPOT2 Color look-up table1.Cr – 0x0b9 ..... 577
SPOT2 Color look-up table2.Y – 0x0ba ..... 577
SPOT2 Color look-up table2.Cb – 0x0bb ..... 577
SPOT2 Color look-up table2.Cr – 0x0bc ..... 577
SPOT2 Color look-up table3.Y – 0x0bd ..... 577
SPOT2 Color look-up table3.Cb – 0x0be ..... 577
SPOT2 Color look-up table3.Cr – 0x0bf ..... 578
SPOT3 Bitmap Destination vertical postion Register 1 – 0x0c4 ..... 578
SPOT3 Bitmap Destination vertical postion Register 2 – 0x0c5 ..... 578
SPOT3 Bitmap Destination horizontal position Register 1 – 0x0c6 ..... 578
SPOT3 Bitmap Destination horizontal position Register 2 – 0x0c7 ..... 578
SPOT3 Bitmap Destination vertical Size Register 1 – 0x0cb ..... 578
SPOT3 Bitmap Destination vertical Size Register 2 – 0x0cc ..... 578
SPOT3 Bitmap Destination horizontal Size Register – 0x0cd ..... 578
SPOT3 Bitmap Start Line Register 1 – 0x0ce ..... 578
SPOT3 Bitmap Start Line Register 2 – 0x0cf ..... 579
SPOT3 Bitmap Start Pixel Register 1 – 0x0d0 ..... 579
SPOT3 Bitmap Start Pixel Register 2 – 0x0d1 ..... 579
SPOT3 Bitmap Control Register 1 – 0x0d2 ..... 579
SPOT3 Bitmap Control Register 2 – 0x0d3 ..... 579
SPOT3 Bitmap Control Register 3 – 0x0d4 ..... 580
SPOT3 Bitmap Control Register 3 – 0x0d6 ..... 580
SPOT3 Color look-up table1.Y – 0x0d7 ..... 580
SPOT3 Color look-up table1.Cb – 0x0d8 ..... 580
SPOT3 Color look-up table1.Cr – 0x0d9 ..... 580
SPOT3 Color look-up table2.Y – 0x0da ..... 580
SPOT3 Color look-up table2.Cb – 0x0db ..... 580
SPOT3 Color look-up table2.Cr – 0x0dc ..... 580
SPOT3 Color look-up table3.Y – 0x0dd ..... 580
SPOT3 Color look-up table3.Cb – 0x0de ..... 580
SPOT3 Color look-up table3.Cr – 0x0df ..... 581
SPOT4 Bitmap Destination vertical postion Register 1 – 0x0e4 ..... 581
SPOT4 Bitmap Destination vertical postion Register 2 – 0x0e5 ..... 581
SPOT4 Bitmap Destination horizontal position Register 3 – 0x0e6 ..... 581
SPOT4 Bitmap Destination horizontal position Register 4 – 0x0e7 ..... 581
SPOT4 Bitmap Destination vertical Size Register 1 – 0x0eb ..... 581
SPOT4 Bitmap Destination vertical Size Register 2 – 0x0ec ..... 581
SPOT4 Bitmap Destination horizontal Size Register – 0x0ed ..... 581
SPOT4 Bitmap Start Line Register 1 – 0x0ee ..... 581

SPOT4 Bitmap Start Line Register 2 – 0x0ef....582
SPOT4 Bitmap Start Pixel Register 1 – 0x0f0....582
SPOT4 Bitmap Start Pixel Register 2 – 0x0f1....582
SPOT4 Bitmap Control Register 1 – 0x0f2....582
SPOT4 Bitmap Control Register 2 – 0x0f3....582
SPOT4 Bitmap Control Register 3 – 0x0f4....583
SPOT4 Bitmap Control Register 3 – 0x0f6....583
SPOT4 Color look-up table1.Y – 0x0f7....583
SPOT4 Color look-up table1.Cb – 0x0f8....583
SPOT4 Color look-up table1.Cr – 0x0f9....583
SPOT4 Color look-up table2.Y – 0x0fa....583
SPOT4 Color look-up table2.Cb – 0x0fb....583
SPOT4 Color look-up table2.Cr – 0x0fc....583
SPOT4 Color look-up table3.Y – 0x0fd....583
SPOT4 Color look-up table3.Cb – 0x0fe....583
SPOT4 Color look-up table3.Cr – 0x0ff....584
Bitmap Write Data – 0x23e....584
PB Input Clock Delay Register – 0x23f....584
Bitmap Display Pitch Register 1 – 0x25d....584
Bitmap Initial Address Register 1 – 0x265....584
Bitmap Initial Address Register 2 – 0x266....584
Bitmap Initial Address Register 3 – 0x267....584
Bitmap Wait Enable Register 1 – 0x249....584
Bitmap Wait Enable Register 2 – 0x24a....585
OSG Bitmap Total Size Register 1 – 0x24b....585
OSG Bitmap Total Size Register 2 – 0x24c....585
OSG Bitmap Total Size Register 3 – 0x24d....585
OSG Bitmap Total Size Register 4 – 0x24e....585
OSG Bitmap Control Register – 0x24f....585
Host Interface....586
Introduction....586
Pin Assignments for Serial and Parallel Interface....586
Serial Interface....587
Parallel Interface....588
Single Transfer....588
Burst Transfer....590
External Host Interface Connection....591
External Host to Display Memory access Interface....591
DMA Interface....591
Interrupt Interface....592
Register Table....594
Registers Description....595
No video IRQ Enable Register 1 – 0xCB0....595
No video IRQ Enable Register 2 – 0xCB1....595
Motion Detection IRQ Enable Register 1 – 0xCB2....595
Motion Detection IRQ Enable Register 2 – 0xCB3....595
Blind Detection IRQ Enable Register 1 – 0xCB4....595
Blind Detection IRQ Enable Register 2 – 0xCB5....595
Night Detection IRQ Enable Register 1 – 0xCB6....596
Night Detection IRQ Enable Register 2 – 0xCB7....596
No video IRQ Status Register 1 – 0xCB8....596
No video IRQ Status Register 2 – 0xCB9....596
Motion Detection IRQ Status Register 1 – 0xCBA....596
Motion Detection IRQ Status Register 2 – 0xCBB....596
Blind Detection IRQ Status Register 1 – 0xCBC....597
Blind Detection IRQ Status Register 2 – 0xCBD....597

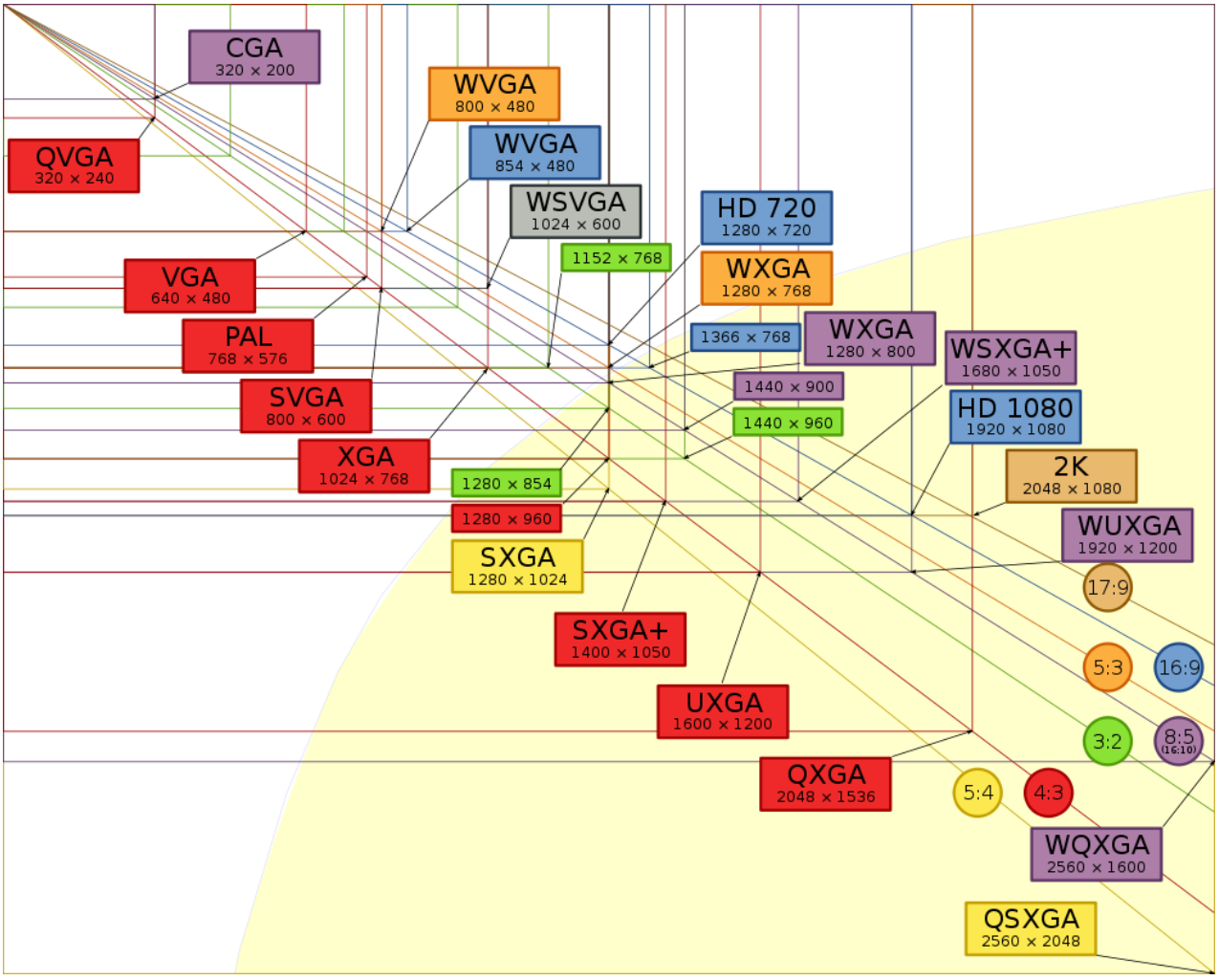
Night Detection IRQ Status Register 1 – 0xCBE ..... 597
Night Detection IRQ Status Register 2 – 0xCBF ..... 597
IRQ Main Control Register —Offset 0xCC0 ..... 597
IRQ Spacing Register – 0xCC1 ..... 597
IRQ Report Counter Register 1 – 0xCC2 ..... 598
IRQ Report Counter Register 2 – 0xCC3 ..... 598
IRQ Report Counter Register 3 – 0xCC4 ..... 598
AUX. IRQ Status Register —Offset 0xCC6 ..... 598
MPOUT Selection Register —Offset 0xCC7 ..... 599
Display DRAM Address Register 0 – 0x000 ..... 599
Display DRAM Address Register 1– 0x001 ..... 599
Display DRAM Address Register 2– 0x002 ..... 599
Display DRAM access Control Register —Offset 0x003 ..... 600
Display DRAM Data Port Register– 0x004 ..... 600
Clock Synthesizer and Pin MUX ..... 601
Introduction ..... 601
Clock Synthesizer Circuit ..... 601
Output Clock Summary ..... 601
Initial Conditions and Changing PLL values ..... 602
RESET Logic ..... 602
Playback Pin Mux ..... 602
Power On Sequence ..... 603
Live Video Pin Mux ..... 603
Port 80 Display ..... 604
HDMI Transmitter Signal Adjustment ..... 604
Register Table ..... 606
Registers Description ..... 608
Audio Control Register —Offset 0x200 ..... 608
Misc. Control Register —Offset 0x201 ..... 608
MPLL M Duty Cycle Control Register (MPLLMR) —Offset 0x202 ..... 608
MPLL N Multipler Factor Register (MPLLNR) —Offset 0x203 ..... 608
VPLL M Duty Cycle Control Register (VPLLMR) —Offset 0x204 ..... 609
VPLL N Multipler Factor Register (VPLLNR) —Offset 0x205 ..... 609
SPLL M Duty Cycle Control Register (SPLLMR) —Offset 0x206 ..... 609
SPLL N Multiply Factor Register (SPLLNR) —Offset 0x207 ..... 609
MCLK PLL Divider Control Register 1 —Offset 0x208 (Upper Byte at 0x220) ..... 609
VCLK PLL Divider Control Register 1—Offset 0x209 (Upper byte at x221 .. x223) ..... 610
IN Rate Select Register —Offset 0x20A ..... 610
Digital Block Reset Control Register 1 —Offset 0x20B ..... 610
Digital Block Reset Control Register 2 —Offset 0x20C ..... 611
Digital Block Reset Control Register 3 —Offset 0x20D ..... 611
Digital Block Reset Control Register 4 —Offset 0x20E ..... 612
Digital Block Reset Control Register 5 —Offset 0x20F ..... 612
LCD Display Pitch Register — Offset 0x210 ..... 613
Display DRAM Controller Configuration Register — 0x211 ..... 613
Recording DRAM Controller Configuration Register — 0x212 ..... 613
Port 80 Control Register — 0x213 ..... 614
Port 80 Content Register 1 — 0x214 ..... 614
Port 80 Content Register 2 — 0x215 ..... 614
SPLL Phase Control Register I —Offset 0x216 ..... 615
PLL Soft Reset Control Register —Offset 0x217 ..... 615
REC output Clock phase Control Register 1 —Offset 0x218 ..... 615
REC output Clock phase Control Register 2 —Offset 0x219 ..... 616
TV decoder output Clock select Register —Offset 0x21A ..... 616
HDMI physical interface Control Register —Offset 0x21B ..... 617
Live input clock phase select Register —Offset 0x21C ..... 617

DRAM Space access select Register —Offset 0x21D....618
Network port data source select Register —Offset 0x21E....618
SCLK Phase Control Register 2 —Offset 0x21F....619
MCLK Phase Control Register 2 —Offset 0x220....619
VCLK PLL Divider Control Register 2 —Offset 0x221....620
VCLK PLL Divider Control Register 3 —Offset 0x222....620
VCLK PLL Divider Control Register 4 —Offset 0x223....621
LP_CON1 x Control Register —Offset 0x224....621
LP_CON2 x Control Register —Offset 0x225....622
LP_CON3 x Control Register —Offset 0x226....623
LP_CON4 x Control Register —Offset 0x227....623
Audio Control 1 Register – 0x228....624
Audio Control 2 Register – 0x229....624
Recording Pin Port Data Select Register — 0x22A....625
CPU Fast Access Control Register — 0x22C....625
CPU Freeze Display Pitch Register — 0x22D....626
SCLK Phase Control Register 3 —Offset 0x22E....626
DRAM Address Mapping Register — 0x26D....626
Chip ID Register — 0x270....626
Display DRAM Priority Arbitration Register 1 — 0x280....627
Display DRAM Priority Arbitration Register 2 — 0x281....627
Display DRAM Priority Arbitration Register 3 — 0x282....627
Record DRAM Priority Arbitration Register 1 — 0x284....627
Record DRAM Priority Arbitration Register 2 — 0x285....628
Record DRAM Priority Arbitration Register 3 — 0x286....628
I²C MASTER CONTROL....629
Introduction....629
Features....629
Operation....629
Register Table....631
Registers....631
I²C Bus Frequency Control Register (Lower) —Offset 0x770....631
I²C Bus Frequency Control Register (Upper) —Offset 0x771....631
I²C Command / Data Register —Offset 0x772....631
I²C Data Read Register —Offset 0x773....631
I²C Command Register —Offset 0x774....631
I²C Command Status Register —Offset 0x775....632
Revision History....633

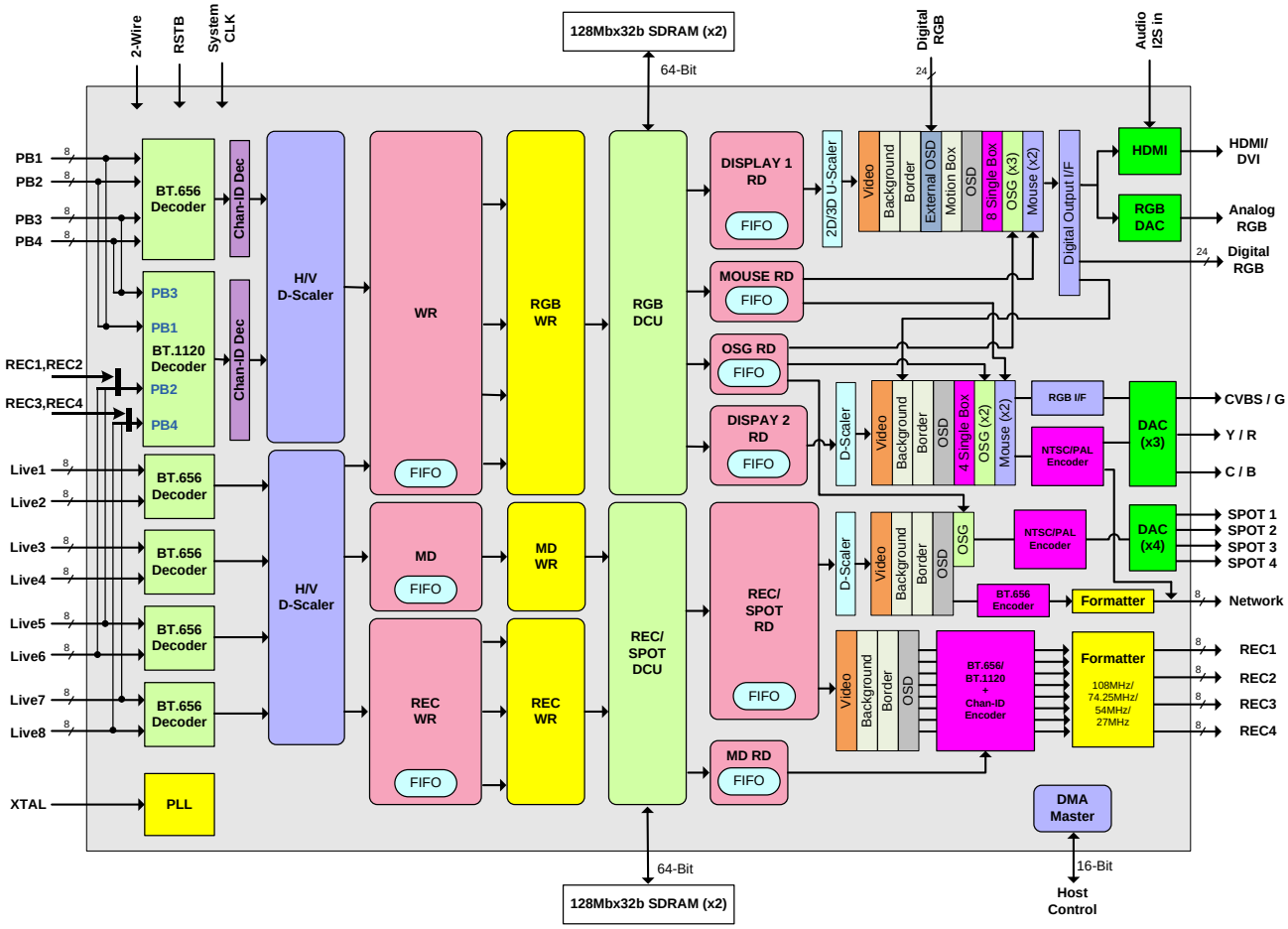
## Display Format

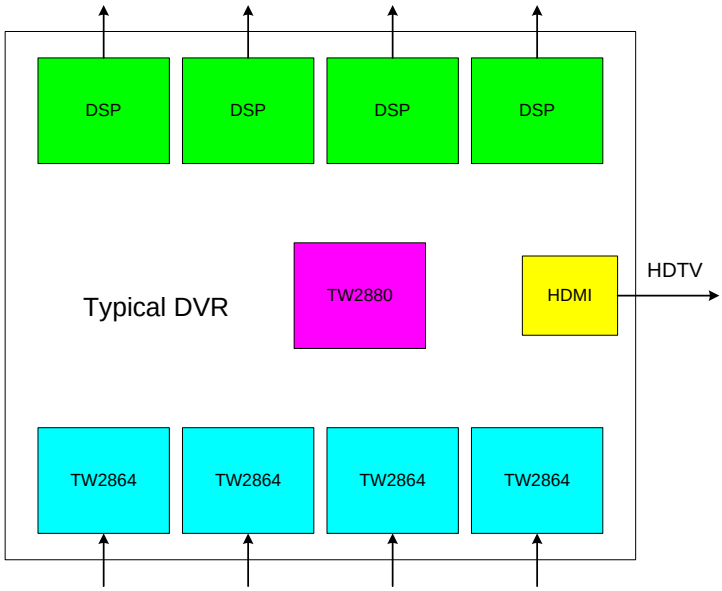
**The TW2880 supports many VESA standard display formats, as listed below. The following is a summary. Please remember that these are only a small fraction of the supported VESA modes. Refresh frequency is flexible and is adjustable based on the dot video clock, horizontal total and vertical total.**

| STANDARD | ACTIVE PIXEL | TOTAL PIXEL | $F_V$ (HZ) | $F_H$ (kHz) | CLOCK (MHz) |
|---|---|---|---|---|---|
| XGA | 1024x768 | 1312x797 | 50 | 42.50 | 51.75 |
| XGA | 1024x768 | 1344x806 | 70 | 56.47 | 75.00 |
| XGA | 1024x768 | 1344x806 | 75 | 60.02 | 78.75 |
| SXGA | 1280x1024 | 1696x1054 | 50 | 38.12 | 89.38 |
| SXGA | 1280x1024 | 1712x1080 | 60.020 | 49.31 | 108.10 |
| WXGA(II) | 1440x900 | 1600x950 | 59.90 | 55.47 | 100.00 |
| WSXGA+ | 1680x1050 | 1840x1080 | 59.98 | 64.67 | 122.00 |
| HD1080 | 1920x1080 | 2200x1125 | 59.98 | 67.37 | 148.50 |



# System Diagram



# Ball Description

**(436 signal pins + 40 Analog VCC/GND + 200 VCC/GND)**

TABLE 1. BALL DESCRIPTION OF VIDEO INPUT INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| A2 | P_ckvin15 | I | *Clock input for multiplexed Video1 and Video5* |
| H6 | P_ckvin26 | I | *Clock input for multiplexed Video2 and Video6* |
| J3 | P_ckvin37 | I | *Clock input for multiplexed Video3 and Video7* |
| M4 | P_ckvin48 | I | *Clock input for multiplexed Video4 and Video8* |
| C3 | P_din1[0] | I | *Port1 Video input bit 0* |
| D4 | P_din1[1] | I | *Port1 Video input bit 1* |
| E5 | P_din1[2] | I | *Port1 Video input bit 2* |
| G7 | P_din1[3] | I | *Port1 Video input bit 3* |
| B1 | P_din1[4] | I | *Port1 Video input bit 4* |
| C2 | P_din1[5] | I | *Port1 Video input bit 5* |
| D3 | P_din1[6] | I | *Port1 Video input bit 6* |
| E4 | P_din1[7] | I | *Port1 Video input bit 7* |
| F5 | P_din2[0] | I | *Port2 Video input bit 0* |
| G6 | P_din2[1] | I | *Port2 Video input bit 1* |
| C1 | P_din2[2] | I | *Port2 Video input bit 2* |
| D2 | P_din2[3] | I | *Port2 Video input bit 3* |
| H7 | P_din2[4] | I | *Port2 Video input bit 4* |
| E3 | P_din2[5] | I | *Port2 Video input bit 5* |
| F4 | P_din2[6] | I | *Port2 Video input bit 6* |
| G5 | P_din2[7] | I | *Port2 Video input bit 7* |
| D1 | P_din3[0] | I | *Port3 Video input bit 0* |
| E2 | P_din3[1] | I | *Port3 Video input bit 1* |
| F3 | P_din3[2] | I | *Port3 Video input bit 2* |
| G4 | P_din3[3] | I | *Port3 Video input bit 3* |
| H5 | P_din3[4] | I | *Port3 Video input bit 4* |
| E1 | P_din3[5] | I | *Port3 Video input bit 5* |
| G3 | P_din3[6] | I | *Port3 Video input bit 6* |
| H4 | P_din3[7] | I | *Port3 Video input bit 7* |
| J5 | P_din4[0] | I | *Port4 Video input bit 0* |
| G2 | P_din4[1] | I | *Port4 Video input bit 1* |
| H3 | P_din4[2] | I | *Port4 Video input bit 2* |
| G1 | P_din4[3] | I | *Port4 Video input bit 3* |
| H2 | P_din4[4] | I | *Port4 Video input bit 4* |
| J4 | P_din4[5] | I | *Port4 Video input bit 5* |
| H1 | P_din4[6] | I | *Port4 Video input bit 6* |
| J7 | P_din4[7] | I | *Port4 Video input bit 7* |
| K7 | P_din5[0] | B | *Port5 Video input bit 0 / Multi-purpose pin / R0* |
| K6 | P_din5[1] | B | *Port5 Video input bit 1 / Multi-purpose pin / R1* |
| L7 | P_din5[2] | B | *Port5 Video input bit 2 / Multi-purpose pin / R2* |
| L6 | P_din5[3] | B | *Port5 Video input bit 3 / Multi-purpose pin / R3* |
| J2 | P_din5[4] | B | *Port5 Video input bit 4 / Multi-purpose pin / R4* |
| J1 | P_din5[5] | B | *Port5 Video input bit 5 / Multi-purpose pin / R5* |
| K5 | P_din5[6] | B | *Port5 Video input bit 6 / Multi-purpose pin / R6* |
| K4 | P_din5[7] | B | *Port5 Video input bit 7 / Multi-purpose pin / R7* |
| K3 | P_din6[0] | B | *Port6 Video input bit 0 / Multi-purpose pin / G0* |
| K2 | P_din6[1] | B | *Port6 Video input bit 1 / Multi-purpose pin / G1* |
| K1 | P_din6[2] | B | *Port6 Video input bit 2 / Multi-purpose pin / G2* |
| L5 | P_din6[3] | B | *Port6 Video input bit 3 / Multi-purpose pin / G3* |
| L4 | P_din6[4] | B | *Port6 Video input bit 4 / Multi-purpose pin / G4* |
| L3 | P_din6[5] | B | *Port6 Video input bit 5 / Multi-purpose pin / G5* |

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| M5 | P_din6[6] | B | *Port6 Video input bit 6 / Multi-purpose pin / G6* |
| M7 | P_din6[7] | B | *Port6 Video input bit 7 / Multi-purpose pin / G7* |
| M3 | P_din7[0] | B | *Port7 Video input bit 0 / Multi-purpose pin / B0* |
| M2 | P_din7[1] | B | *Port7 Video input bit 1 / Multi-purpose pin / B1* |
| M1 | P_din7[2] | B | *Port7 Video input bit 2 / Multi-purpose pin / B2* |
| N6 | P_din7[3] | B | *Port7 Video input bit 3 / Multi-purpose pin / B3* |
| N7 | P_din7[4] | B | *Port7 Video input bit 4 / Multi-purpose pin / B4* |
| N5 | P_din7[5] | B | *Port7 Video input bit 5 / Multi-purpose pin / B5* |
| N4 | P_din7[6] | B | *Port7 Video input bit 6 / Multi-purpose pin / B6* |
| N3 | P_din7[7] | B | *Port7 Video input bit 7 / Multi-purpose pin / B7* |
| N2 | P_din8[0] | B | *Port8 Video input bit 0 / external OSD enable / LCD_DE (output)* |
| N1 | P_din8[1] | B | *Port8 Video input bit 1 / VCLK output* |
| P1 | P_din8[2] | I | *Port8 Video input bit 2* |
| P2 | P_din8[3] | I | *Port8 Video input bit 3* |
| P3 | P_din8[4] | I | *Port8 Video input bit 4* |
| P6 | P_din8[5] | I | *Port8 Video input bit 5* |
| P4 | P_din8[6] | I | *Port8 Video input bit 6* |
| P5 | P_din8[7] | I | *Port8 Video input bit 7* |

TABLE 2. BALL DESCRIPTION OF PLAYBACK INPUT INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| P7 | P_ck27pb1 | I | *Clock input for playback port1* |
| R1 | P_din_pb1[0] | I | *Playback Port1 input bit 0* |
| R2 | P_din_pb1[1] | I | *Playback Port1 input bit 1* |
| R3 | P_din_pb1[2] | I | *Playback Port1 input bit 2* |
| R4 | P_din_pb1[3] | I | *Playback Port1 input bit 3* |
| R7 | P_din_pb1[4] | I | *Playback Port1 input bit 4* |
| R5 | P_din_pb1[5] | I | *Playback Port1 input bit 5* |
| T7 | P_din_pb1[6] | I | *Playback Port1 input bit 6* |
| T6 | P_din_pb1[7] | I | *Playback Port1 input bit 7* |
| U6 | P_ck27pb2 | I | *Clock input for playback port2* |
| V1 | P_din_pb2[0] | I | *Playback Port2 input bit 0* |
| V2 | P_din_pb2[1] | I | *Playback Port2 input bit 1* |
| V3 | P_din_pb2[2] | I | *Playback Port2 input bit 2* |
| V4 | P_din_pb2[3] | I | *Playback Port2 input bit 3* |
| W1 | P_din_pb2[4] | I | *Playback Port2 input bit 4* |
| V5 | P_din_pb2[5] | I | *Playback Port2 input bit 5* |
| U7 | P_din_pb2[6] | I | *Playback Port2 input bit 6* |
| W2 | P_din_pb2[7] | I | *Playback Port2 input bit 7* |
| V7 | P_bt601_H1 | O | *DACK signal in DMA mode* |
| Y1 | P_bt601_V1 | I | *HDONE signal in DMA mode, tie to ground if not used* |
| W3 | P_ck27pb3 | I | *Clock input for playback port3* |
| Y2 | P_din_pb3[0] | I | *Playback Port3 input bit 0* |
| Y3 | P_din_pb3[1] | I | *Playback Port3 input bit 1* |
| W4 | P_din_pb3[2] | I | *Playback Port3 input bit 2* |
| AB1 | P_din_pb3[3] | I | *Playback Port3 input bit 3* |
| AC1 | P_din_pb3[4] | I | *Playback Port3 input bit 4* |
| AB2 | P_din_pb3[5] | I | *Playback Port3 input bit 5* |
| AA3 | P_din_pb3[6] | I | *Playback Port3 input bit 6* |
| Y4 | P_din_pb3[7] | I | *Playback Port3 input bit 7* |
| W5 | P_ck27pb4 | I | *Clock input for playback port4* |
| AD1 | P_din_pb4[0] | I | *Playback Port4 input bit 0* |
| AC2 | P_din_pb4[1] | I | *Playback Port4 input bit 1* |

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| AB3 | P_din_pb4[2] | I | *Playback Port4 input bit 2* |
| AA4 | P_din_pb4[3] | I | *Playback Port4 input bit 3* |
| Y5 | P_din_pb4[4] | I | *Playback Port4 input bit 4* |
| AE1 | P_din_pb4[5] | I | *Playback Port4 input bit 5* |
| AD2 | P_din_pb4[6] | I | *Playback Port4 input bit 6* |
| AC3 | P_din_pb4[7] | I | *Playback Port4 input bit 7* |
| AD3 | P_bt601_H2 | I | *Reserved* |
| W6 | P_bt601_V2 | I | *Reserved* |

TABLE 3. BALL DESCRIPTION OF PLL INPUT INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| T1 | P_xtal_in1 | I | *Crystal Input* |
| T2 | P_xtal_in2 | O | *Crystal Output* |

TABLE 4. BALL DESCRIPTION OF RRCORD PORT INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| AB4 | P_rec1_clkn | O | *Negative Clock output for record port1* |
| Y6 | P_rec1_clkp | O | *Positive Clock output for record port1* |
| AC4 | P_rec1_enc_data[0] | B | *Record Port1 output bit 0* |
| W7 | P_rec1_enc_data[1] | B | *Record Port1 output bit 1* |
| AA5 | P_rec1_enc_data[2] | B | *Record Port1 output bit 2* |
| AB5 | P_rec1_enc_data[3] | B | *Record Port1 output bit 3* |
| AD4 | P_rec1_enc_data[4] | B | *Record Port1 output bit 4* |
| AE3 | P_rec1_enc_data[5] | B | *Record Port1 output bit 5* |
| AF2 | P_rec1_enc_data[6] | B | *Record Port1 output bit 6* |
| Y7 | P_rec1_enc_data[7] | B | *Record Port1 output bit 7* |
| AA7 | P_rec1_bt601_H1 | B | *Record Port1 Horizontal sync* |
| AB6 | P_rec1_bt601_V1 | B | *Record Port1 Vertical sync* |
| AC5 | P_rec1_bt601_F1 | B | *Record Port1 Field / ext clock input* |
| AB7 | P_rec2_clkn | O | *Negative Clock output for record port2* |
| AC6 | P_rec2_clkp | O | *Positive Clock output for record port2* |
| AD5 | P_rec2_enc_data[0] | B | *Record Port2 output bit 0* |
| AE4 | P_rec2_enc_data[1] | B | *Record Port2 output bit 1* |
| AF3 | P_rec2_enc_data[2] | B | *Record Port2 output bit 2* |
| Y8 | P_rec2_enc_data[3] | B | *Record Port2 output bit 3* |
| AC7 | P_rec2_enc_data[4] | B | *Record Port2 output bit 4* |
| AA8 | P_rec2_enc_data[5] | B | *Record Port2 output bit 5* |
| AD6 | P_rec2_enc_data[6] | B | *Record Port2 output bit 6* |
| AE5 | P_rec2_enc_data[7] | B | *Record Port2 output bit 7* |
| AF4 | P_rec2_bt601_H2 | O | *Record Port2 Horizontal sync* |
| AF5 | P_rec2_bt601_V2 | O | *Record Port2 Vertical sync* |
| AB8 | P_rec2_bt601_F2 | O | *Record Port2 Field* |
| AC8 | P_rec3_clkn | O | *Negative Clock output for record port3* |
| AD7 | P_rec3_clkp | O | *Positive Clock output for record port3* |
| Y9 | P_rec3_enc_data[0] | O | *Record Port3 output bit 0* |
| AB9 | P_rec3_enc_data[1] | O | *Record Port3 output bit 1* |
| AC9 | P_rec3_enc_data[2] | O | *Record Port3 output bit 2* |
| AD8 | P_rec3_enc_data[3] | O | *Record Port3 output bit 3* |
| AA9 | P_rec3_enc_data[4] | O | *Record Port3 output bit 4* |

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| AE7 | P_rec3_enc_data[5] | O | *Record Port3 output bit 5* |
| AD9 | P_rec3_enc_data[6] | O | *Record Port3 output bit 6* |
| AE8 | P_rec3_enc_data[7] | O | *Record Port3 output bit 7* |
| Y10 | P_rec3_bt601_H3 | O | *Record Port3 Horizontal sync* |
| AF7 | P_rec3_bt601_V3 | O | *Record Port3 Vertical sync* |
| AB10 | P_rec3_bt601_F3 | O | *Record Port3 Field* |
| AC10 | P_rec4_clkn | O | *Negative Clock output for record port4* |
| AD10 | P_rec4_clkp | O | *Positive Clock output for record port4* |
| AA10 | P_rec4_enc_data[0] | O | *Record Port4 output bit 0* |
| AE9 | P_rec4_enc_data[1] | O | *Record Port4 output bit 1* |
| AF8 | P_rec4_enc_data[2] | O | *Record Port4 output bit 2* |
| AB11 | P_rec4_enc_data[3] | O | *Record Port4 output bit 3* |
| AC11 | P_rec4_enc_data[4] | O | *Record Port4 output bit 4* |
| AD11 | P_rec4_enc_data[5] | O | *Record Port4 output bit 5* |
| Y11 | P_rec4_enc_data[6] | O | *Record Port4 output bit 6* |
| AE10 | P_rec4_enc_data[7] | O | *Record Port4 output bit 7* |
| AA11 | P_rec4_bt601_H4 | O | *Record Port4 Horizontal sync* |
| AF9 | P_rec4_bt601_V4 | O | *Record Port4 Vertical sync* |
| AF10 | P_rec4_bt601_F4 | O | *Record Port4 Field* |

TABLE 5. BALL DESCRIPTION OF RECORD PORT DRAM INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| AA13 | P_rec_ma[0] | O | *Record port memory address 0* |
| AC13 | P_rec_ma[1] | O | *Record port memory address 1* |
| AD13 | P_rec_ma[2] | O | *Record port memory address 2* |
| AE13 | P_rec_ma[3] | O | *Record port memory address 3* |
| AF13 | P_rec_ma[4] | O | *Record port memory address 4* |
| AF14 | P_rec_ma[5] | O | *Record port memory address 5* |
| AE14 | P_rec_ma[6] | O | *Record port memory address 6* |
| AA14 | P_rec_ma[7] | O | *Record port memory address 7* |
| AD14 | P_rec_ma[8] | O | *Record port memory address 8* |
| Y14 | P_rec_ma[9] | O | *Record port memory address 9* |
| AC14 | P_rec_ma[10] | O | *Record port memory address 10* |
| AB14 | P_rec_ma[11] | O | *Record port memory address 11* |
| AF15 | P_rec_dq[0] | B | *Record port memory data 0* |
| AE15 | P_rec_dq[1] | B | *Record port memory data 1* |
| AD15 | P_rec_dq[2] | B | *Record port memory data 2* |
| Y15 | P_rec_dq[3] | B | *Record port memory data 3* |
| AC15 | P_rec_dq[4] | B | *Record port memory data 4* |
| AB15 | P_rec_dq[5] | B | *Record port memory data 5* |
| AF17 | P_rec_dq[6] | B | *Record port memory data 6* |
| AD16 | P_rec_dq[7] | B | *Record port memory data 7* |
| AE17 | P_rec_dq[8] | B | *Record port memory data 8* |

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| AF18 | P_rec_dq[9] | B | *Record port memory data 9* |
| AC16 | P_rec_dq[10] | B | *Record port memory data 10* |
| AA16 | P_rec_dq[11] | B | *Record port memory data 11* |
| AD17 | P_rec_dq[12] | B | *Record port memory data 12* |
| AE18 | P_rec_dq[13] | B | *Record port memory data 13* |
| AF19 | P_rec_dq[14] | B | *Record port memory data 14* |
| AB16 | P_rec_dq[15] | B | *Record port memory data 15* |
| Y16 | P_rec_dq[16] | B | *Record port memory data 16* |
| AC17 | P_rec_dq[17] | B | *Record port memory data 17* |
| W16 | P_rec_dq[18] | B | *Record port memory data 18* |
| AD18 | P_rec_dq[19] | B | *Record port memory data 19* |
| AE19 | P_rec_dq[20] | B | *Record port memory data 20* |
| AF20 | P_rec_dq[21] | B | *Record port memory data 21* |
| AA17 | P_rec_dq[22] | B | *Record port memory data 22* |
| AB17 | P_rec_dq[23] | B | *Record port memory data 23* |
| Y17 | P_rec_dq[24] | B | *Record port memory data 24* |
| AC18 | P_rec_dq[25] | B | *Record port memory data 25* |
| AD19 | P_rec_dq[26] | B | *Record port memory data 26* |
| AE20 | P_rec_dq[27] | B | *Record port memory data 27* |
| AA18 | P_rec_dq[28] | B | *Record port memory data 28* |
| AB18 | P_rec_dq[29] | B | *Record port memory data 29* |
| Y18 | P_rec_dq[30] | B | *Record port memory data 30* |
| AC19 | P_rec_dq[31] | B | *Record port memory data 31* |
| AF23 | P_rec_dq[32] | B | *Record port memory data 32* |
| AB20 | P_rec_dq[33] | B | *Record port memory data 33* |
| AC21 | P_rec_dq[34] | B | *Record port memory data 34* |
| AD22 | P_rec_dq[35] | B | *Record port memory data 35* |
| AE23 | P_rec_dq[36] | B | *Record port memory data 36* |
| AF24 | P_rec_dq[37] | B | *Record port memory data 37* |
| AB21 | P_rec_dq[38] | B | *Record port memory data 38* |
| AA19 | P_rec_dq[39] | B | *Record port memory data 39* |
| AC22 | P_rec_dq[40] | B | *Record port memory data 40* |
| AD23 | P_rec_dq[41] | B | *Record port memory data 41* |
| AE24 | P_rec_dq[42] | B | *Record port memory data 42* |
| AA20 | P_rec_dq[43] | B | *Record port memory data 43* |
| AF25 | P_rec_dq[44] | B | *Record port memory data 44* |
| Y19 | P_rec_dq[45] | B | *Record port memory data 45* |

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| AD24 | P_rec_dq[46] | B | *Record port memory data 46* |
| Y20 | P_rec_dq[47] | B | *Record port memory data 47* |
| AC23 | P_rec_dq[48] | B | *Record port memory data 48* |
| AB22 | P_rec_dq[49] | B | *Record port memory data 49* |
| AA22 | P_rec_dq[50] | B | *Record port memory data 50* |
| AB23 | P_rec_dq[51] | B | *Record port memory data 51* |
| W20 | P_rec_dq[52] | B | *Record port memory data 52* |
| AC24 | P_rec_dq[53] | B | *Record port memory data 53* |
| Y21 | P_rec_dq[54] | B | *Record port memory data 54* |
| AD25 | P_rec_dq[55] | B | *Record port memory data 55* |
| AE26 | P_rec_dq[56] | B | *Record port memory data 56* |
| Y22 | P_rec_dq[57] | B | *Record port memory data 57* |
| W21 | P_rec_dq[58] | B | *Record port memory data 58* |
| AA23 | P_rec_dq[59] | B | *Record port memory data 59* |
| AB24 | P_rec_dq[60] | B | *Record port memory data 60* |
| AC25 | P_rec_dq[61] | B | *Record port memory data 61* |
| AD26 | P_rec_dq[62] | B | *Record port memory data 62* |
| W22 | P_rec_dq[63] | B | *Record port memory data 63* |
| AD20 | P_rec_rasb | O | *Record port DRAM control RASB* |
| AF22 | P_rec_casb | O | *Record port DRAM control CASB* |
| AB19 | P_rec_web | O | *Record port DRAM control WEB* |
| AC20 | P_rec_dram_clk | O | *Record port DRAM control CLK* |
| AD21 | P_rec_ba[0] | O | *Record port DRAM control BANK0* |
| AE22 | P_rec_ba[1] | O | *Record port DRAM control BANK1* |

TABLE 6. BALL DESCRIPTION OF SPOT PORT INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| P26 | P_ana_spot1_VREF | I | *SPOT1 DAC voltage reference ( connect to 1.1 VDC)* |
| N23 | P_ana_spot1_VRO | O | *SPOT1 DAC Full-scale Current Adjust*<br>*Connect 2.7 KΩ (typ) between this pin and Analog GND* |
| N24 | P_ana_spot1_CVBS | O | *SPOT1, Analog composite TV signal* |
| N25 | P_ana_spot2_VREF | I | *SPOT2 DAC voltage reference ( connect to 1.1 VDC)* |
| N26 | P_ana_spot2_VRO | O | *SPOT2 DAC Full-scale Current Adjust*<br>*Connect 2.7 KΩ (typ) between this pin and Analog GND* |
| N22 | P_ana_spot2_CVBS | O | *SPOT2, Analog composite TV signal* |
| AB12 | P_ana_spot3_VREF | I | *SPOT3 DAC voltage reference ( connect to 1.1 VDC)* |
| AC12 | P_ana_spot3_VRO | O | *SPOT3 DAC Full-scale Current Adjust*<br>*Connect 2.7 KΩ (typ) between this pin and Analog GND* |
| AD12 | P_ana_spot3_CVBS | O | *SPOT3, Analog composite TV signal* |
| AE12 | P_ana_spot4_VREF | I | *SPOT4 DAC voltage reference ( connect to 1.1 VDC)* |
| AF12 | P_ana_spot4_VRO | O | *SPOT4 DAC Full-scale Current Adjust*<br>*Connect 2.7 KΩ (typ) between this pin and Analog GND* |
| AB13 | P_ana_spot4_CVBS | O | *SPOT4, Analog composite TV signal* |

TABLE 7. BALL DESCRIPTION OF HOST PORT INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| Y23 | P_rstb | I | *Chip Reset when it is LOW* |
| AC26 | P_hspb | I | *I²C Interface when this bit is 1. Parallel Interface if HSBP is 0* |
| V20 | P_hcsb | I | *In I²C mode, this bit is SCK, In Parallel Mode, it is Chip Select* |
| V22 | P_hwrb | I | *N/A In I²C mode. In Parallel Mode, it is WRITE enable when 0* |
| W23 | P_hrdb | I | *N/A In I²C mode. In Parallel Mode, it is READ enable when 0* |
| AA24 | P_wait_st | O | *host waiting indication from OSG* |
| AB25 | P_h16b_en | I | *8 / 16 bit mode select, high = Enable 16 bit host mode* |
| Y24 | P_haddr[0] | I | *Host address bit 0* |
| AB26 | P_haddr[1] | I | *Host address bit 1* |
| U20 | P_haddr[2] | I | *Host address bit 2* |
| U22 | P_haddr[3] | I | *Host address bit 3* |
| U21 | P_haddr[4] | I | *Host address bit 4* |
| V23 | P_haddr[5] | I | *Host address bit 5* |
| W24 | P_haddr[6] | I | *Host address bit 6* |
| Y25 | P_haddr[7] | I | *Host address bit 7* |
| T20 | P_haddr[8] | I | *Host address bit 8* |
| T22 | P_haddr[9] | I | *Host address bit 9* |
| T21 | P_haddr[10] | I | *Host address bit 10* |
| U23 | P_haddr[11] | I | *Host address bit 11* |
| V24 | P_hdat[0] | B | *Host data bit 0* |
| W25 | P_hdat[1] | B | *Host data bit 1* |
| Y26 | P_hdat[2] | B | *Host data bit 2* |
| T23 | P_hdat[3] | B | *Host data bit 3* |
| U24 | P_hdat[4] | B | *Host data bit 4* |
| V25 | P_hdat[5] | B | *Host data bit 5* |
| W26 | P_hdat[6] | B | *Host data bit 6* |
| T24 | P_hdat[7] | B | *Host data bit 7 / SDA for I²C mode* |
| U25 | P_hdat[8] | B | *Host data bit 8* |
| V26 | P_hdat[9] | B | *Host data bit 9* |
| U26 | P_hdat[10] | B | *Host data bit 10* |
| R20 | P_hdat[11] | B | *Host data bit 11* |
| R22 | P_hdat[12] | B | *Host data bit 12* |
| R23 | P_hdat[13] | B | *Host data bit 13* |
| R24 | P_hdat[14] | B | *Host data bit 14* |
| R25 | P_hdat[15] | B | *Host data bit 15* |
| R26 | P_hdat_iob | O | *Host read data valid signal / HDREQ* |

TABLE 8. BALL DESCRIPTION OF LCD DRAM INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| P22 | P_lcd_dram_clk | O | *LCD display dram clock out* |
| P23 | P_lcd_ma[0] | O | *LCD display DRAM control memory address bit 0* |
| P24 | P_lcd_ma[1] | O | *LCD display DRAM control memory address bit 1* |
| P25 | P_lcd_ma[2] | O | *LCD display DRAM control memory address bit 2* |
| M26 | P_lcd_ma[3] | O | *LCD display DRAM control memory address bit 3* |
| M20 | P_lcd_ma[4] | O | *LCD display DRAM control memory address bit 4* |
| M25 | P_lcd_ma[5] | O | *LCD display DRAM control memory address bit 5* |
| M24 | P_lcd_ma[6] | O | *LCD display DRAM control memory address bit 6* |
| M23 | P_lcd_ma[7] | O | *LCD display DRAM control memory address bit 7* |
| M22 | P_lcd_ma[8] | O | *LCD display DRAM control memory address bit 8* |
| L21 | P_lcd_ma[9] | O | *LCD display DRAM control memory address bit 9* |
| L24 | P_lcd_ma[10] | O | *LCD display DRAM control memory address bit 10* |
| L20 | P_lcd_ma[11] | O | *LCD display DRAM control memory address bit 11* |
| L23 | P_lcd_ba[0] | O | *LCD display DRAM control memory bank 0* |
| L22 | P_lcd_ba[1] | O | *LCD display DRAM control memory bank 1* |
| K26 | P_lcd_dq[0] | B | *LCD display DRAM control memory data bit 0* |
| J26 | P_lcd_dq[1] | B | *LCD display DRAM control memory data bit 1* |

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| K25 | P_lcd_dq[2] | B | *LCD display DRAM control memory data bit 2* |
| H26 | P_lcd_dq[3] | B | *LCD display DRAM control memory data bit 3* |
| J25 | P_lcd_dq[4] | B | *LCD display DRAM control memory data bit 4* |
| K24 | P_lcd_dq[5] | B | *LCD display DRAM control memory data bit 5* |
| G26 | P_lcd_dq[6] | B | *LCD display DRAM control memory data bit 6* |
| H25 | P_lcd_dq[7] | B | *LCD display DRAM control memory data bit 7* |
| J24 | P_lcd_dq[8] | B | *LCD display DRAM control memory data bit 8* |
| K21 | P_lcd_dq[9] | B | *LCD display DRAM control memory data bit 9* |
| K23 | P_lcd_dq[10] | B | *LCD display DRAM control memory data bit 10* |
| K20 | P_lcd_dq[11] | B | *LCD display DRAM control memory data bit 11* |
| G25 | P_lcd_dq[12] | B | *LCD display DRAM control memory data bit 12* |
| H24 | P_lcd_dq[13] | B | *LCD display DRAM control memory data bit 13* |
| J23 | P_lcd_dq[14] | B | *LCD display DRAM control memory data bit 14* |
| K22 | P_lcd_dq[15] | B | *LCD display DRAM control memory data bit 15* |
| E26 | P_lcd_dq[16] | B | *LCD display DRAM control memory data bit 16* |
| G24 | P_lcd_dq[17] | B | *LCD display DRAM control memory data bit 17* |
| H23 | P_lcd_dq[18] | B | *LCD display DRAM control memory data bit 18* |
| J22 | P_lcd_dq[19] | B | *LCD display DRAM control memory data bit 19* |
| D26 | P_lcd_dq[20] | B | *LCD display DRAM control memory data bit 20* |
| E25 | P_lcd_dq[21] | B | *LCD display DRAM control memory data bit 21* |
| F24 | P_lcd_dq[22] | B | *LCD display DRAM control memory data bit 22* |
| G23 | P_lcd_dq[23] | B | *LCD display DRAM control memory data bit 23* |
| J20 | P_lcd_dq[24] | B | *LCD display DRAM control memory data bit 24* |
| H22 | P_lcd_dq[25] | B | *LCD display DRAM control memory data bit 25* |
| C26 | P_lcd_dq[26] | B | *LCD display DRAM control memory data bit 26* |
| D25 | P_lcd_dq[27] | B | *LCD display DRAM control memory data bit 27* |
| E24 | P_lcd_dq[28] | B | *LCD display DRAM control memory data bit 28* |
| F23 | P_lcd_dq[29] | B | *LCD display DRAM control memory data bit 29* |
| G22 | P_lcd_dq[30] | B | *LCD display DRAM control memory data bit 30* |
| B26 | P_lcd_dq[31] | B | *LCD display DRAM control memory data bit 31* |
| C25 | P_lcd_dq[32] | B | *LCD display DRAM control memory data bit 32* |
| D24 | P_lcd_dq[33] | B | *LCD display DRAM control memory data bit 33* |
| E23 | P_lcd_dq[34] | B | *LCD display DRAM control memory data bit 34* |
| F22 | P_lcd_dq[35] | B | *LCD display DRAM control memory data bit 35* |
| C24 | P_lcd_dq[36] | B | *LCD display DRAM control memory data bit 36* |
| D23 | P_lcd_dq[37] | B | *LCD display DRAM control memory data bit 37* |
| E22 | P_lcd_dq[38] | B | *LCD display DRAM control memory data bit 38* |
| H21 | P_lcd_dq[39] | B | *LCD display DRAM control memory data bit 39* |
| A25 | P_lcd_dq[40] | B | *LCD display DRAM control memory data bit 40* |
| B24 | P_lcd_dq[41] | B | *LCD display DRAM control memory data bit 41* |
| C23 | P_lcd_dq[42] | B | *LCD display DRAM control memory data bit 42* |
| G21 | P_lcd_dq[43] | B | *LCD display DRAM control memory data bit 43* |
| D22 | P_lcd_dq[44] | B | *LCD display DRAM control memory data bit 44* |
| G20 | P_lcd_dq[45] | B | *LCD display DRAM control memory data bit 45* |
| A24 | P_lcd_dq[46] | B | *LCD display DRAM control memory data bit 46* |
| H20 | P_lcd_dq[47] | B | *LCD display DRAM control memory data bit 47* |
| B23 | P_lcd_dq[48] | B | *LCD display DRAM control memory data bit 48* |
| E21 | P_lcd_dq[49] | B | *LCD display DRAM control memory data bit 49* |
| E20 | P_lcd_dq[50] | B | *LCD display DRAM control memory data bit 50* |
| G19 | P_lcd_dq[51] | B | *LCD display DRAM control memory data bit 51* |
| D21 | P_lcd_dq[52] | B | *LCD display DRAM control memory data bit 52* |
| F20 | P_lcd_dq[53] | B | *LCD display DRAM control memory data bit 53* |
| C22 | P_lcd_dq[54] | B | *LCD display DRAM control memory data bit 54* |
| E19 | P_lcd_dq[55] | B | *LCD display DRAM control memory data bit 55* |
| D20 | P_lcd_dq[56] | B | *LCD display DRAM control memory data bit 56* |
| C21 | P_lcd_dq[57] | B | *LCD display DRAM control memory data bit 57* |
| F19 | P_lcd_dq[58] | B | *LCD display DRAM control memory data bit 58* |

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| B22 | P_lcd_dq[59] | B | *LCD display DRAM control memory data bit 59* |
| G18 | P_lcd_dq[60] | B | *LCD display DRAM control memory data bit 60* |
| A23 | P_lcd_dq[61] | B | *LCD display DRAM control memory data bit 61* |
| E18 | P_lcd_dq[62] | B | *LCD display DRAM control memory data bit 62* |
| D19 | P_lcd_dq[63] | B | *LCD display DRAM control memory data bit 63* |
| C20 | P_lcd_rasb | O | *LCD display DRAM control RASB* |
| A22 | P_lcd_casb | O | *LCD display DRAM control CASB* |
| G17 | P_lcd_web | O | *LCD display DRAM control WEB* |

TABLE 9. BALL DESCRIPTION OF DUAL VIEW PORT INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| E10 | P_ana_dm_R | O | *DualView R / S-Video Y* |
| D9 | P_ana_dm_G | O | *DualView G / CVBS* |
| C8 | P_ana_dm_B | O | *DualView B / S-Video C* |
| A8 | P_ana_dm_VREF | I | *DualView DAC voltage reference ( connect to 1.1 VDC)* |
| B8 | P_ana_dm_VRO | O | *DualView DAC Full-scale Current Adjust*<br>*Connect 2.7 KΩ (typ) between this pin and Analog GND* |
| B18 | P_dm_hsync | O | *Dual View Horizontal sync* |
| A19 | P_dm_vsync | O | *Dual View Vertical sync* |

TABLE 10. PIN DESCRIPTION OF JTAG PORT INTERFACE

| Pin(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| A7 | P_tck | I | *JTAG test pin connect to ground* |
| B7 | P_tms | I | *JTAG test pin connect to ground* |
| A5 | P_trst | I | *JTAG test pin connect to ground* |
| C7 | P_tdi | I | *JTAG test pin connect to ground* |
| A4 | P_tdo | O | *JTAG test pin out* |

TABLE 11. BALL DESCRIPTION OF MAIN DISPLAY PORT INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| B19 | P_lcd_hsync | O | *LCD Horizontal sync* |
| A20 | P_lcd_vsync | O | *LCD Vertical sync* |
| E17 | P_ana_lcd_R | O | *LCD R* |
| C17 | P_ana_lcd_G | O | *LCD G* |
| C16 | P_ana_lcd_B | O | *LCD B* |
| A15 | P_ana_lcd_VREF | I | *LCD DAC voltage reference ( connect to 1.1 VDC)* |
| C15 | P_ana_lcd_VRO | O | *LCD DAC Full-scale Current Adjust*<br>*Connect 2.7 KΩ (typ) between this pin and Analog GND* |
| D17 | P_ana_lcd_R_COM | I | *Reserved, no connection* |
| G16 | P_ana_lcd_G_COM | I | *Reserved, no connection* |
| B15 | P_ana_lcd_B_COM | I | *Reserved, no connection* |

TABLE 12. BALL DESCRIPTION OF HDMI PORT INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| B14 | EXCN | O | *HDMI differential clock output* |
| A14 | EXCP | O | *HDMI differential clock output* |
| A13 | EXN0 | O | *HDMI differential data signal output* |
| B13 | EXP0 | O | *HDMI differential data signal output* |
| B12 | EXN1 | O | *HDMI differential data signal output* |
| A12 | EXP1 | O | *HDMI differential data signal output* |
| A11 | EXN2 | O | *HDMI differential data signal output* |
| B11 | EXP2 | O | *HDMI differential data signal output* |
| A9 | P_HTMODE | I | *HDMI test mode select, tie to ground* |
| B9 | P_HTXRST | I | *HDMI reset, tie to 3.3V* |

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| C5 | P_i2s_sd[0] | I | *HDMI I²S serial data bit 0, tie to ground if not used* |
| E7 | P_i2s_sd[1] | I | *HDMI I²S serial data bit 1, tie to ground if not used* |
| G8 | P_i2s_sd[2] | I | *HDMI I²S serial data bit 2, tie to ground if not used* |
| E6 | P_i2s_sd[3] | I | *HDMI I²S serial data bit 3, tie to ground if not used* |
| D6 | P_i2s_sck | I | *HDMI I²S serial clock, tie to ground if not used* |
| F7 | P_i2s_ws | I | *HDMI I²S word select, tie to ground if not used* |
| A17 | P_MCLK_I | B | *HDMI Audio master clock, tie to ground if not used* |
| F11 | P_ASCLK_I | I | *HDMI SPDIF capture clock, tie to ground if not used* |
| F14 | P_SPDIF_I | I | *HDMI S/PDIF audio input, tie to ground if not used* |
| E14 | P_HPD_I | I | *HDMI hot plug detect, tie to ground via 10K resistor* |
| A18 | P_DSCL | B | *HDMI DDC clock pin* |
| B17 | P_DSDA | B | *HDMI DDC data pin* |
| D14 | P_VPD | I | *Test power control input, tie to ground* |

TABLE 13. BALL DESCRIPTION OF MANUFACTURER TEST INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| D18 | P_test_ena1 | I | *Test pin1, manufacturer test, tie to ground.* |
| F17 | P_test_ena2 | I | *Test pin2, manufacturer test, tie to ground.* |

TABLE 14. BALL DESCRIPTION OF PS2 MOUSE PORT INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| C4 | P_ps2_c | B | *PS2 mouse port clock pin* |
| B3 | P_ps2_d | B | *PS2 mouse port data pin* |

TABLE 15. BALL DESCRIPTION OF MISCELLANEOUS PORT INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| C19 | P_trigger_in | I | *Ext. Motion detect strobe input, tie to ground if not used* |
| B20 | P_irq_out | O | *Interrupt to the host* |
| C18 | P_mp_out | O | *Debugging output, (Multiplexed out of vsync, hsync, or field)* |
| D5 | P_ext_clko | O | *Output clock to TW2864 decoder to synchronize data* |

TABLE 16. BALL DESCRIPTION OF NETWORK PORT INTERFACE

| Ball(s) No. | Symbol | Attribute | Description |
|---|---|---|---|
| B5 | P_network_clk | O | *Network port clock output* |
| C6 | P_network_data[0] | O | *Network port data output bit 0* |
| F9 | P_network_data[1] | O | *Network port data output bit 1* |
| D7 | P_network_data[2] | O | *Network port data output bit 2* |
| G9 | P_network_data[3] | O | *Network port data output bit 3* |
| E8 | P_network_data[4] | O | *Network port data output bit 4* |
| A3 | P_network_data[5] | O | *Network port data output bit 5* |
| B4 | P_network_data[6] | O | *Network port data output bit 6* |
| F8 | P_network_data[7] | O | *Network port data output bit 7* |

TABLE 17. BALL DESCRIPTION OF POWER AND GROUND

| Symbol | Ball(s) No. | Voltage | Description |
|---|---|---|---|
| VDD | H9, H10, H13, H14, H17, H18, J8, J11, J12, J15, J16, J19, K8, K10, K13, K14, K17, K19, L9, L18, M9, M18, N8, N10, N17, N19, P8, P10, P17, P19, R9, R18, T9, T18, U8, U10, U13, U14, U17, U19, V8, V11, V12, V15, V16, V19, W9, W10, W14, W17, W18 | 1.2 Volts | *Power of the core logic* |
| VDE | B2, B6, B10, F2, F6, J6, L2, M6, R6, U2, V6, AA2, AA6, AA12, AE2, AE6, AE11, R21, T25, V21, AA15, AA21, AA25, AE16, AE21, AE25, G11, B16, B21, B25, F18, F21, F25, G15, J21, L25, M21 | 3.3 Volts | *Power of the external I/O* |
| VSS | A1, A6, A10, A16, A21, A26, F1, F26, G10, G14, H8, H11, H12, H15, H16, H19, J9, J10, J13, J14, J17, J18, K9, K11, K12, K15, K16, K18, L1, L8, L10, L11, L12, L13, L14, L15, L16, L17, L19, L26, M8, M10, M11, M12, M13, M14, M15, M16, M17, M19, N9, N11, N12, N13, N14, N15, N16, N18, P9, P11, P12, P13, P14, P15, P16, P18, R8, R10, R11, R12, R13, R14, R15, R16, R17, R19, T8, T10, T11, T12, T13, T14, T15, T16, T17, T19, T26, U1, U9, U11, U12, U15, U16, U18, V9, V10, V13, V14, V17, V18, W8, W11, W15, W19, AA1, AA26, AF1, AF6, AF11, AF16, AF21, AF26 | - | *Ground of the core logic* |
| | | | |
| DAC_DM_R_AVD | D10 | 3.3 volts | *Analog power of DAC for DM R* |
| DAC_DM_G_AVD | C9 | 3.3 volts | *Analog power of DAC for DM G* |
| DAC_DM_B_AVD | E9 | 3.3 volts | *Analog power of DAC for DM B* |
| DAC_LCD_R_AVD | E16 | 3.3 volts | *Analog power of DAC for LCD R* |
| DAC_LCD_G_AVD | F15 | 3.3 volts | *Analog power of DAC for LCD G* |
| DAC_LCD_B_AVD | D15 | 3.3 volts | *Analog power of DAC for LCD B* |
| DAC_SPOT1_AVD | P21 | 3.3 volts | *Analog power of DAC for SOPT1* |
| DAC_SPOT2_AVD | N20 | 3.3 volts | *Analog power of DAC for SOPT2* |
| DAC_SPOT3_AVD | Y12 | 3.3 volts | *Analog power of DAC for SOPT3* |
| DAC_SPOT4_AVD | W13 | 3.3 volts | *Analog power of DAC for SOPT4* |
| DAC_DM_R_AVS | C10 | - | *Analog ground of DAC for DM R* |
| DAC_DM_G_AVS | F10 | - | *Analog ground of DAC for DM G* |
| DAC_DM_B_AVS | D8 | - | *Analog ground of DAC for DM B* |

| Symbol | Ball(s) No. | Voltage | Description |
|---|---|---|---|
| DAC_LCD_R_AVS | F16 | - | *Analog ground of DAC for LCD R* |
| DAC_LCD_G_AVS | D16 | - | *Analog ground of DAC for LCD G* |
| DAC_LCD_B_AVS | E15 | - | *Analog ground of DAC for LCD B* |
| DAC_SPOT1_AVS | P20 | - | *Analog ground of DAC for SOPT1* |
| DAC_SPOT2_AVS | N21 | - | *Analog ground of DAC for SOPT2* |
| DAC_SPOT3_AVS | W12 | - | *Analog ground of DAC for SOPT3* |
| DAC_SPOT4_AVS | Y13 | - | *Analog ground of DAC for SOPT4* |
| | | | |
| MCK_PLL_AVD | T3 | 1.2 Volts | *Analog power of the MCK PLL* |
| VCK_PLL_AVD | T4 | 1.2 Volts | *Analog power of the VCK PLL* |
| SCK_PLL_AVD | T5 | 1.2 Volts | *Analog power of the SCK PLL* |
| MCK_PLL_AVS | U3 | - | *Analog ground of the MCK PLL* |
| VCK_PLL_AVS | U4 | - | *Analog ground of the VCK PLL* |
| SCK_PLL_AVS | U5 | - | *Analog ground of the SCK PLL* |
| | | | |
| HDMI_VDN | C11, C13, E11, F12, G13 | 1.2 volts | *Analog power of the HDMI core* |
| HDMI_VDP1 | D13 | 3.3 volts | *Analog power of the HDMI I/O* |
| HDMI_VDP2 | D12 | 3.3 volts | *Analog power of the HDMI I/O* |
| HDMI_VDU | E13 | 1.2 volts | *Analog power of HDMI PLL* |
| HDMI_VSN | C12, C14, D11, E12, F13, G12, | - | *Analog ground of the HDMI* |

# Ball Diagrams

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | VSS | P_ckvin12 | P_network_data[5] | P_tdo | P_trst | VSS | P_tck | P_ana_dm_VREF | P_HTMODE | VSS | EXN2 | EXP1 | EXN0 | EXCP | P_ana_lcd_VREF | VSS | P_MCLK_I | P_DSCL | P_dm_vsync | P_lcd_vsync | VSS | P_lcd_casb | P_lcd_dq[61] | P_lcd_dq[46] | P_lcd_dq[40] | VSS | A |
| B | P_din1[4] | VDE | P_ps2_d | P_network_data[6] | P_network_clk | VDE | P_tms | P_ana_dm_VRO | P_HTXRST | VDE | EXP2 | EXN1 | EXP0 | EXCN | P_ana_lcd_B_COM | VDE | P_DSDA | P_dm_hsync | P_lcd_hsync | P_irq_out | VDE | P_lcd_dq[59] | P_lcd_dq[48] | P_lcd_dq[41] | VDE | P_lcd_dq[31] | B |
| C | P_din2[2] | P_din1[5] | P_din1[0] | P_ps2_c | P_i2s_sd[0] | P_network_data[0] | P_tdi | P_ana_dm_B | DAC_DM_G_AVD | DAC_DM_R_AVS | HDMI_VDN | HDMI_VSN | HDMI_VDN | HDMI_VSN | P_ana_lcd_VRO | P_ana_lcd_B | P_ana_lcd_G | P_mp_out | P_trigger_in | P_lcd_rasb | P_lcd_dq[57] | P_lcd_dq[54] | P_lcd_dq[42] | P_lcd_dq[36] | P_lcd_dq[32] | P_lcd_dq[26] | C |
| D | P_din3[0] | P_din2[3] | P_din1[6] | P_din1[1] | P_ext_clko | P_i2s_sck | P_network_data[2] | DAC_DM_B_AVS | P_ana_dm_G | DAC_DM_R_AVD | HDMI_VSN | HDMI_VDP2 | HDMI_VDP1 | P_VPD | DAC_LCD_B_AVD | DAC_LCD_G_AVS | P_ana_lcd_R_COM | P_test_ena1 | P_lcd_dq[63] | P_lcd_dq[56] | P_lcd_dq[52] | P_lcd_dq[44] | P_lcd_dq[37] | P_lcd_dq[33] | P_lcd_dq[27] | P_lcd_dq[20] | D |
| E | P_din3[5] | P_din3[1] | P_din2[5] | P_din1[7] | P_din1[2] | P_i2s_sd[3] | P_i2s_sd[1] | P_network_data[4] | DAC_DM_B_AVD | P_ana_dm_R | HDMI_VDN | HDMI_VSN | HDMI_VDU | P_HPD_I | DAC_LCD_B_AVS | DAC_LCD_R_AVD | P_ana_lcd_R | P_lcd_dq[62] | P_lcd_dq[55] | P_lcd_dq[50] | P_lcd_dq[49] | P_lcd_dq[38] | P_lcd_dq[34] | P_lcd_dq[28] | P_lcd_dq[21] | P_lcd_dq[16] | E |
| F | VSS | VDE | P_din3[2] | P_din2[6] | P_din2[0] | VDE | P_i2s_ws | P_network_data[7] | P_network_data[1] | DAC_DM_G_AVS | P_ASCLK_I | HDMI_VDN | HDMI_VSN | P_SPDIF_I | DAC_LCD_G_AVD | DAC_LCD_R_AVS | P_test_ena2 | VDE | P_lcd_dq[58] | P_lcd_dq[53] | VDE | P_lcd_dq[35] | P_lcd_dq[29] | P_lcd_dq[22] | VDE | VSS | F |
| G | P_din4[3] | P_din4[1] | P_din3[6] | P_din3[3] | P_din2[7] | P_din2[1] | P_din1[3] | P_i2s_sd[2] | P_network_data[3] | VSS | VDE | HDMI_VSN | HDMI_VDN | VSS | VDE | P_ana_lcd_G_COM | P_lcd_wb | P_lcd_dq[60] | P_lcd_dq[51] | P_lcd_dq[45] | P_lcd_dq[43] | P_lcd_dq[30] | P_lcd_dq[23] | P_lcd_dq[17] | P_lcd_dq[12] | P_lcd_dq[6] | G |
| H | P_din4[6] | P_din4[4] | P_din4[2] | P_din3[7] | P_din3[4] | P_ckvin34 | P_din2[4] | VSS | VDD | VDD | VSS | VSS | VDD | VDD | VSS | VSS | VDD | VDD | VSS | P_lcd_dq[47] | P_lcd_dq[39] | P_lcd_dq[25] | P_lcd_dq[18] | P_lcd_dq[13] | P_lcd_dq[7] | P_lcd_dq[3] | H |
| J | P_din5[5] | P_din5[4] | P_ckvin56 | P_din4[5] | P_din4[0] | VDE | P_din4[7] | VDD | VSS | VSS | VDD | VDD | VSS | VSS | VDD | VDD | VSS | VSS | VDD | P_lcd_dq[24] | VDE | P_lcd_dq[19] | P_lcd_dq[14] | P_lcd_dq[8] | P_lcd_dq[4] | P_lcd_dq[1] | J |
| K | P_din6[2] | P_din6[1] | P_din6[0] | P_din5[7] | P_din5[6] | P_din5[1] | P_din5[0] | VDD | VSS | VDD | VSS | VSS | VDD | VDD | VSS | VSS | VDD | VSS | VDD | P_lcd_dq[11] | P_lcd_dq[9] | P_lcd_dq[15] | P_lcd_dq[10] | P_lcd_dq[5] | P_lcd_dq[2] | P_lcd_dq[0] | K |
| L | VSS | VDE | P_din6[5] | P_din6[4] | P_din6[3] | P_din5[3] | P_din5[2] | VSS | VDD | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VDD | VSS | P_lcd_ma[11] | P_lcd_ma[9] | P_lcd_ba[1] | P_lcd_ba[0] | P_lcd_ma[10] | VDE | VSS | L |
| M | P_din7[2] | P_din7[1] | P_din7[0] | P_ckvin78 | P_din6[6] | VDE | P_din6[7] | VSS | VDD | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VDD | VSS | P_lcd_ma[4] | VDE | P_lcd_ma[8] | P_lcd_ma[7] | P_lcd_ma[6] | P_lcd_ma[5] | P_lcd_ma[3] | M |
| N | P_din8[1] | P_din8[0] | P_din7[7] | P_din7[6] | P_din7[5] | P_din7[3] | P_din7[4] | VDD | VSS | VDD | VSS | VSS | VSS | VSS | VSS | VSS | VDD | VSS | VDD | DAC_SPOT2_AVD | DAC_SPOT2_AVS | P_ana_spot2_CVBS | P_ana_spot1_VRO | P_ana_spot1_CVBS | P_ana_spot2_VREF | P_ana_spot2_VRO | N |
| P | P_din8[2] | P_din8[3] | P_din8[4] | P_din8[6] | P_din8[7] | P_din8[5] | P_ck27pb1 | VDD | VSS | VDD | VSS | VSS | VSS | VSS | VSS | VSS | VDD | VSS | VDD | DAC_SPOT1_AVS | DAC_SPOT1_AVD | P_lcd_dram_clk | P_lcd_ma[0] | P_lcd_ma[1] | P_lcd_ma[2] | P_ana_spot1_VREF | P |
| R | P_din_pb1[0] | P_din_pb1[1] | P_din_pb1[2] | P_din_pb1[3] | P_din_pb1[5] | VDE | P_din_pb1[4] | VSS | VDD | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VDD | VSS | P_hdat[11] | VDE | P_hdat[12] | P_hdat[13] | P_hdat[14] | P_hdat[15] | P_hdat_iob | R |
| T | P_xtal_in1 | P_xtal_in2 | MCK_PLL_AVD | VCK_PLL_AVD | SCK_PLL_AVD | P_din_pb1[7] | P_din_pb1[6] | VSS | VDD | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VDD | VSS | P_haddr[8] | P_haddr[10] | P_haddr[9] | P_hdat[3] | P_hdat[7] | VDE | VSS | T |
| U | VSS | VDE | MCK_PLL_AVS | VCK_PLL_AVS | SCK_PLL_AVS | P_ck27pb2 | P_din_pb2[6] | VDD | VSS | VDD | VSS | VSS | VDD | VDD | VSS | VSS | VDD | VSS | VDD | P_haddr[2] | P_haddr[4] | P_haddr[3] | P_haddr[11] | P_hdat[4] | P_hdat[8] | P_hdat[10] | U |
| V | P_din_pb2[0] | P_din_pb2[1] | P_din_pb2[2] | P_din_pb2[3] | P_din_pb2[5] | VDE | P_bt601_H1 | VDD | VSS | VSS | VDD | VDD | VSS | VSS | VDD | VDD | VSS | VSS | VDD | P_hcsb | VDE | P_hwrb | P_haddr[5] | P_hdat[0] | P_hdat[5] | P_hdat[9] | V |
| W | P_din_pb2[4] | P_din_pb2[7] | P_ck27pb3 | P_din_pb3[2] | P_ck27pb4 | P_bt601_V2 | P_rec1_enc_data[1] | VSS | VDD | VDD | VSS | DAC_SPOT3_AVS | DAC_SPOT4_AVD | VDD | VSS | P_rec_dq[18] | VDD | VDD | VSS | P_rec_dq[52] | P_rec_dq[58] | P_rec_dq[63] | P_hrdb | P_haddr[6] | P_hdat[1] | P_hdat[6] | W |
| Y | P_bt601_V1 | P_din_pb3[0] | P_din_pb3[1] | P_din_pb3[7] | P_din_pb4[4] | P_rec1_clkp | P_rec1_enc_data[7] | P_rec2_enc_data[3] | P_rec3_enc_data[0] | P_rec3_bt601_H3 | P_rec4_enc_data[6] | DAC_SPOT3_AVD | DAC_SPOT4_AVS | P_rec_ma[9] | P_rec_dq[3] | P_rec_dq[16] | P_rec_dq[24] | P_rec_dq[30] | P_rec_dq[45] | P_rec_dq[47] | P_rec_dq[54] | P_rec_dq[57] | P_rstb | P_haddr[0] | P_haddr[7] | P_hdat[2] | Y |
| AA | VSS | VDE | P_din_pb3[6] | P_din_pb4[3] | P_rec1_enc_data[2] | VDE | P_rec1_bt601_H1 | P_rec2_enc_data[5] | P_rec3_enc_data[4] | P_rec4_enc_data[0] | P_rec4_bt601_H4 | VDE | P_rec_ma[0] | P_rec_ma[7] | VDE | P_rec_dq[11] | P_rec_dq[22] | P_rec_dq[28] | P_rec_dq[39] | P_rec_dq[43] | VDE | P_rec_dq[50] | P_rec_dq[59] | P_wait_st | VDE | VSS | AA |
| AB | P_din_pb3[3] | P_din_pb3[5] | P_din_pb4[2] | P_rec1_clkn | P_rec1_enc_data[3] | P_rec1_bt601_V1 | P_rec2_clkn | P_rec2_bt601_F2 | P_rec3_enc_data[1] | P_rec3_bt601_F3 | P_rec4_enc_data[3] | P_ana_spot3_VREF | P_ana_spot4_CVBS | P_rec_ma[11] | P_rec_dq[5] | P_rec_dq[15] | P_rec_dq[23] | P_rec_dq[29] | P_rec_web | P_rec_dq[33] | P_rec_dq[38] | P_rec_dq[49] | P_rec_dq[51] | P_rec_dq[60] | P_h16b_en | P_haddr[1] | AB |
| AC | P_din_pb3[4] | P_din_pb4[1] | P_din_pb4[7] | P_rec1_enc_data[0] | P_rec1_bt601_F1 | P_rec2_clkp | P_rec2_enc_data[4] | P_rec3_clkn | P_rec3_enc_data[2] | P_rec4_clkn | P_rec4_enc_data[4] | P_ana_spot3_VRO | P_rec_ma[1] | P_rec_ma[10] | P_rec_dq[4] | P_rec_dq[10] | P_rec_dq[17] | P_rec_dq[25] | P_rec_dq[31] | P_rec_dram_clk | P_rec_dq[34] | P_rec_dq[40] | P_rec_dq[48] | P_rec_dq[53] | P_rec_dq[61] | P_hspb | AC |
| AD | P_din_pb4[0] | P_din_pb4[6] | P_bt601_H2 | P_rec1_enc_data[4] | P_rec2_enc_data[0] | P_rec2_enc_data[6] | P_rec3_clkp | P_rec3_enc_data[3] | P_rec3_enc_data[6] | P_rec4_clkp | P_rec4_enc_data[5] | P_ana_spot3_CVBS | P_rec_ma[2] | P_rec_ma[8] | P_rec_dq[2] | P_rec_dq[7] | P_rec_dq[12] | P_rec_dq[19] | P_rec_dq[26] | P_rec_rasb | P_rec_ba[0] | P_rec_dq[35] | P_rec_dq[41] | P_rec_dq[46] | P_rec_dq[55] | P_rec_dq[62] | AD |
| AE | P_din_pb4[5] | VDE | P_rec1_enc_data[5] | P_rec2_enc_data[1] | P_rec2_enc_data[7] | VDE | P_rec3_enc_data[5] | P_rec3_enc_data[7] | P_rec4_enc_data[1] | P_rec4_enc_data[7] | VDE | P_ana_spot4_VREF | P_rec_ma[3] | P_rec_ma[6] | P_rec_dq[1] | VDE | P_rec_dq[8] | P_rec_dq[13] | P_rec_dq[20] | P_rec_dq[27] | VDE | P_rec_ba[1] | P_rec_dq[36] | P_rec_dq[42] | VDE | P_rec_dq[56] | AE |
| AF | VSS | P_rec1_enc_data[6] | P_rec2_enc_data[2] | P_rec2_bt601_H2 | P_rec2_bt601_V2 | VSS | P_rec3_bt601_V3 | P_rec4_enc_data[2] | P_rec4_bt601_V4 | P_rec4_bt601_F4 | VSS | P_ana_spot4_VRO | P_rec_ma[4] | P_rec_ma[5] | P_rec_dq[0] | VSS | P_rec_dq[6] | P_rec_dq[9] | P_rec_dq[14] | P_rec_dq[21] | VSS | P_rec_casb | P_rec_dq[32] | P_rec_dq[37] | P_rec_dq[44] | VSS | AF |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | |

**FIGURE 1. MAP OF ALL THE TW2880 SIGNAL POSITIONS**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | VSS | P_ckvin12 | P_network_data[5] | P_tdo | P_trst | VSS | P_tck | P_ana_dm_VREF | P_HTMODE | VSS | EXN2 | EXP1 | EXN0 |
| B | P_din1[4] | VDE | P_ps2_d | P_network_data[6] | P_network_clk | VDE | P_tms | P_ana_dm_VRO | P_HTXRST | VDE | EXP2 | EXN1 | EXP0 |
| C | P_din2[2] | P_din1[5] | P_din1[0] | P_ps2_c | P_i2s_sd[0] | P_network_data[0] | P_tdi | P_ana_dm_B | DAC_DM_G_AVD | DAC_DM_R_AVS | HDMI_VDN | HDMI_VSN | HDMI_VDN |
| D | P_din3[0] | P_din2[3] | P_din1[6] | P_din1[1] | P_ext_clko | P_i2s_sck | P_network_data[2] | DAC_DM_B_AVS | P_ana_dm_G | DAC_DM_R_AVD | HDMI_VSN | HDMI_VDP2 | HDMI_VDP1 |
| E | P_din3[5] | P_din3[1] | P_din2[5] | P_din1[7] | P_din1[2] | P_i2s_sd[3] | P_i2s_sd[1] | P_network_data[4] | DAC_DM_B_AVD | P_ana_dm_R | HDMI_VDN | HDMI_VSN | HDMI_VDU |
| F | VSS | VDE | P_din3[2] | P_din2[6] | P_din2[0] | VDE | P_i2s_ws | P_network_data[7] | P_network_data[1] | DAC_DM_G_AVS | P_ASCLK_I | HDMI_VDN | HDMI_VSN |
| G | P_din4[3] | P_din4[1] | P_din3[6] | P_din3[3] | P_din2[7] | P_din2[1] | P_din1[3] | P_i2s_sd[2] | P_network_data[3] | VSS | VDE | HDMI_VSN | HDMI_VDN |
| H | P_din4[6] | P_din4[4] | P_din4[2] | P_din3[7] | P_din3[4] | P_ckvin34 | P_din2[4] | VSS | VDD | VDD | VSS | VSS | VDD |
| J | P_din5[5] | P_din5[4] | P_ckvin56 | P_din4[5] | P_din4[0] | VDE | P_din4[7] | VDD | VSS | VSS | VDD | VDD | VSS |
| K | P_din6[2] | P_din6[1] | P_din6[0] | P_din5[7] | P_din5[6] | P_din5[1] | P_din5[0] | VDD | VSS | VDD | VSS | VSS | VDD |
| L | VSS | VDE | P_din6[5] | P_din6[4] | P_din6[3] | P_din5[3] | P_din5[2] | VSS | VDD | VSS | VSS | VSS | VSS |
| M | P_din7[2] | P_din7[1] | P_din7[0] | P_ckvin78 | P_din6[6] | VDE | P_din6[7] | VSS | VDD | VSS | VSS | VSS | VSS |
| N | P_din8[1] | P_din8[0] | P_din7[7] | P_din7[6] | P_din7[5] | P_din7[3] | P_din7[4] | VDD | VSS | VDD | VSS | VSS | VSS |

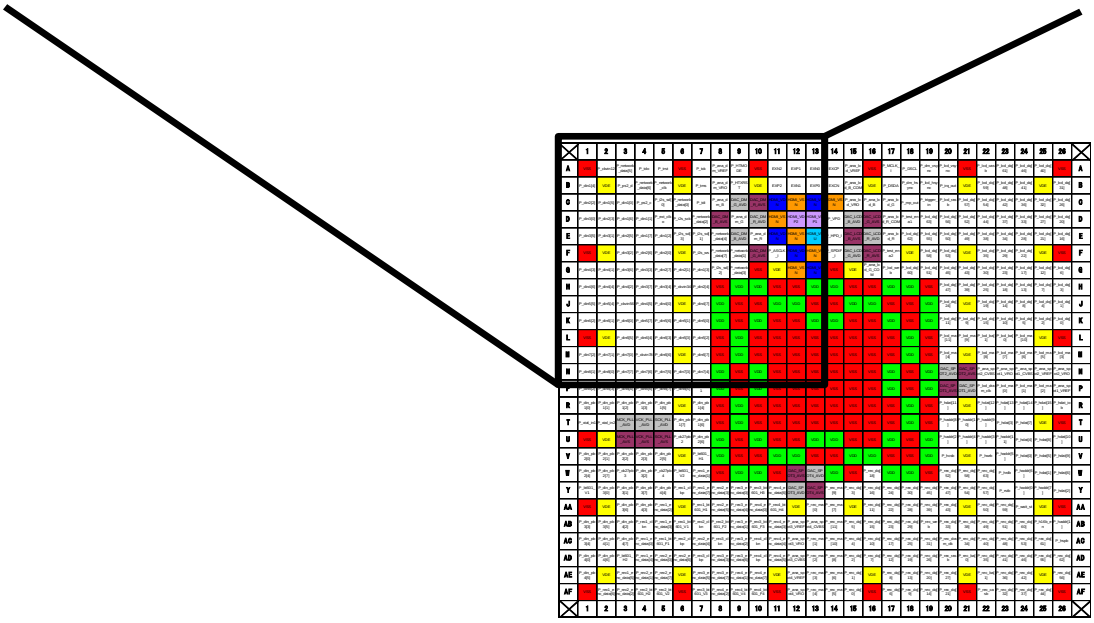


**FIGURE 2. MAP OF THE UPPER-LEFT QUADRANT**

| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| EXCP | P_ana_lcd_VREF | VSS | P_MCLK_I | P_DSCL | P_dm_vsync | P_lcd_vsync | VSS | P_lcd_casb | P_lcd_dq[61] | P_lcd_dq[46] | P_lcd_dq[40] | VSS | A |
| EXCN | P_ana_lcd_B_COM | VDE | P_DSDA | P_dm_hsync | P_lcd_hsync | P_irq_out | VDE | P_lcd_dq[59] | P_lcd_dq[48] | P_lcd_dq[41] | VDE | P_lcd_dq[31] | B |
| HDMI_VSN | P_ana_lcd_VRO | P_ana_lcd_B | P_ana_lcd_G | P_mp_out | P_trigger_in | P_lcd_rasb | P_lcd_dq[57] | P_lcd_dq[54] | P_lcd_dq[42] | P_lcd_dq[36] | P_lcd_dq[32] | P_lcd_dq[26] | C |
| P_VPD | DAC_LCD_B_AVD | DAC_LCD_G_AVS | P_ana_lcd_R_COM | P_test_ena1 | P_lcd_dq[63] | P_lcd_dq[56] | P_lcd_dq[52] | P_lcd_dq[44] | P_lcd_dq[37] | P_lcd_dq[33] | P_lcd_dq[27] | P_lcd_dq[20] | D |
| P_HPD_I | DAC_LCD_B_AVS | DAC_LCD_R_AVD | P_ana_lcd_R | P_lcd_dq[62] | P_lcd_dq[55] | P_lcd_dq[50] | P_lcd_dq[49] | P_lcd_dq[38] | P_lcd_dq[34] | P_lcd_dq[28] | P_lcd_dq[21] | P_lcd_dq[16] | E |
| P_SPDIF_I | DAC_LCD_G_AVD | DAC_LCD_R_AVS | P_test_ena2 | VDE | P_lcd_dq[58] | P_lcd_dq[53] | VDE | P_lcd_dq[35] | P_lcd_dq[29] | P_lcd_dq[22] | VDE | VSS | F |
| VSS | VDE | P_ana_lcd_G_COM | P_lcd_web | P_lcd_dq[60] | P_lcd_dq[51] | P_lcd_dq[45] | P_lcd_dq[43] | P_lcd_dq[30] | P_lcd_dq[23] | P_lcd_dq[17] | P_lcd_dq[12] | P_lcd_dq[6] | G |
| VDD | VSS | VSS | VDD | VDD | VSS | P_lcd_dq[47] | P_lcd_dq[39] | P_lcd_dq[25] | P_lcd_dq[18] | P_lcd_dq[13] | P_lcd_dq[7] | P_lcd_dq[3] | H |
| VSS | VDD | VDD | VSS | VSS | VDD | P_lcd_dq[24] | VDE | P_lcd_dq[19] | P_lcd_dq[14] | P_lcd_dq[8] | P_lcd_dq[4] | P_lcd_dq[1] | J |
| VDD | VSS | VSS | VDD | VSS | VDD | P_lcd_dq[11] | P_lcd_dq[9] | P_lcd_dq[15] | P_lcd_dq[10] | P_lcd_dq[5] | P_lcd_dq[2] | P_lcd_dq[0] | K |
| VSS | VSS | VSS | VSS | VDD | VSS | P_lcd_ma[11] | P_lcd_ma[9] | P_lcd_ba[1] | P_lcd_ba[0] | P_lcd_ma[10] | VDE | VSS | L |
| VSS | VSS | VSS | VSS | VDD | VSS | P_lcd_ma[4] | VDE | P_lcd_ma[8] | P_lcd_ma[7] | P_lcd_ma[6] | P_lcd_ma[5] | P_lcd_ma[3] | M |
| VSS | VSS | VSS | VDD | VSS | VDD | DAC_SPOT2_AVD | DAC_SPOT2_AVS | P_ana_spot2_CVBS | P_ana_spot1_VRO | P_ana_spot1_CVBS | P_ana_spot2_VREF | P_ana_spot2_VRO | N |

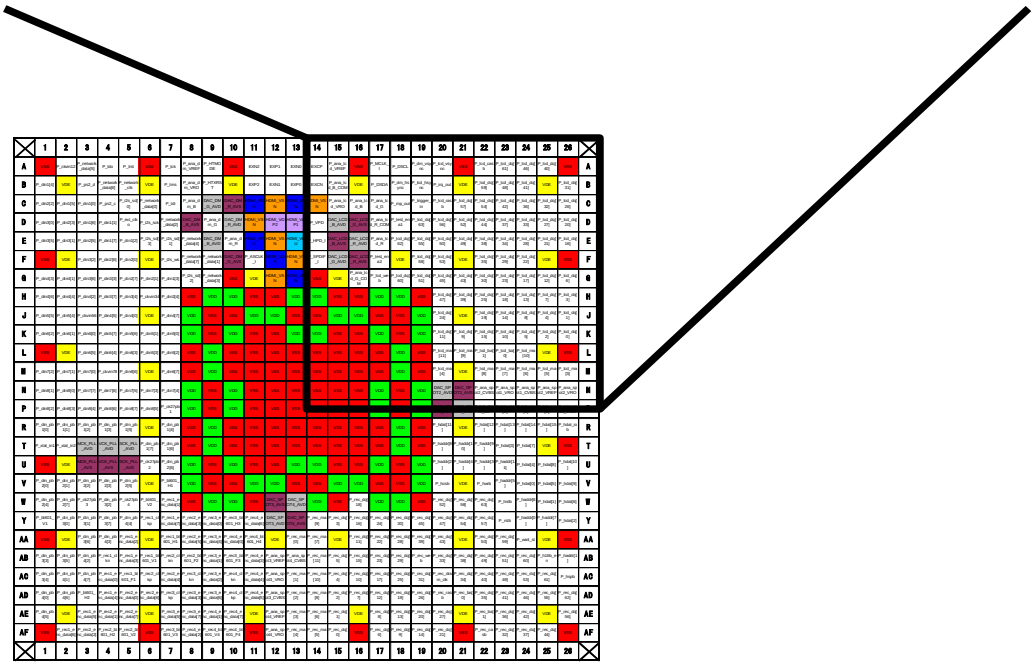
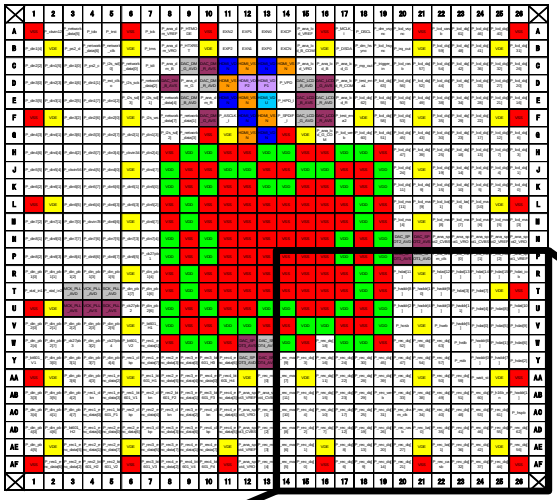
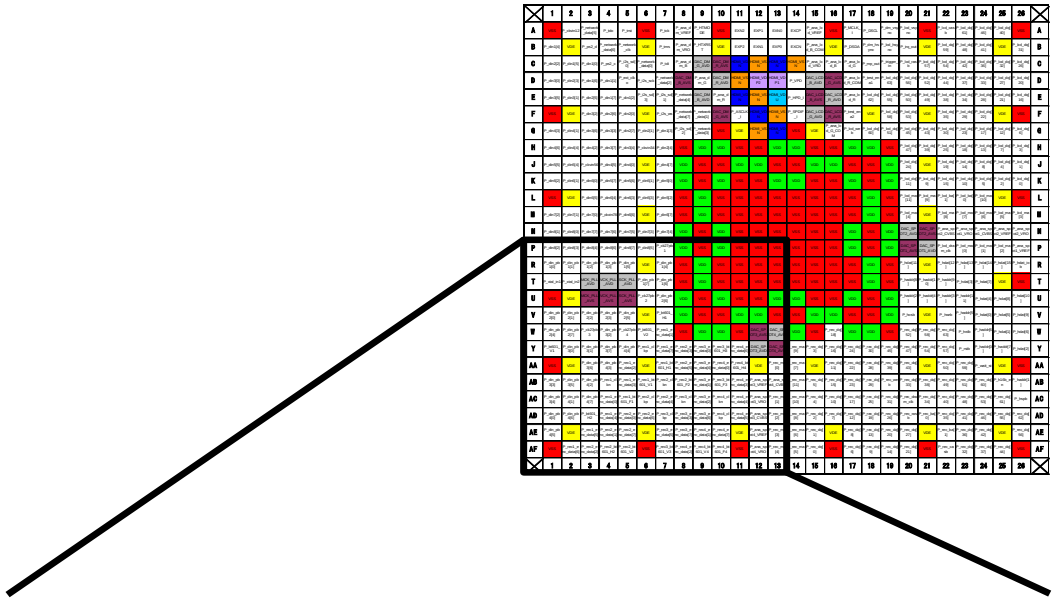


**FIGURE 3. MAP OF THE UPPER-RIGHT QUADRANT**



| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| VSS | VSS | VSS | VDD | VSS | VDD | DAC_SPOT1_AVS | DAC_SPOT1_AVD | P_lcd_dram_clk | P_lcd_ma[0] | P_lcd_ma[1] | P_lcd_ma[2] | P_ana_spot1_VREF | P |
| VSS | VSS | VSS | VSS | VDD | VSS | P_hdat[11] | VDE | P_hdat[12] | P_hdat[13] | P_hdat[14] | P_hdat[15] | P_hdat_iob | R |
| VSS | VSS | VSS | VSS | VDD | VSS | P_haddr[8] | P_haddr[10] | P_haddr[9] | P_hdat[3] | P_hdat[7] | VDE | VSS | T |
| VDD | VSS | VSS | VDD | VSS | VDD | P_haddr[2] | P_haddr[4] | P_haddr[3] | P_haddr[11] | P_hdat[4] | P_hdat[8] | P_hdat[10] | U |
| VSS | VDD | VDD | VSS | VSS | VDD | P_hcsb | VDE | P_hwrb | P_haddr[5] | P_hdat[0] | P_hdat[5] | P_hdat[9] | V |
| VDD | VSS | P_rec_dq[18] | VDD | VDD | VSS | P_rec_dq[52] | P_rec_dq[58] | P_rec_dq[63] | P_hrdb | P_haddr[6] | P_hdat[1] | P_hdat[6] | W |
| P_rec_ma[9] | P_rec_dq[3] | P_rec_dq[16] | P_rec_dq[24] | P_rec_dq[30] | P_rec_dq[45] | P_rec_dq[47] | P_rec_dq[54] | P_rec_dq[57] | P_rstb | P_haddr[0] | P_haddr[7] | P_hdat[2] | Y |
| P_rec_ma[7] | VDE | P_rec_dq[11] | P_rec_dq[22] | P_rec_dq[28] | P_rec_dq[39] | P_rec_dq[43] | VDE | P_rec_dq[50] | P_rec_dq[59] | P_wait_st | VDE | VSS | AA |
| P_rec_ma[11] | P_rec_dq[5] | P_rec_dq[15] | P_rec_dq[23] | P_rec_dq[29] | P_rec_web | P_rec_dq[33] | P_rec_dq[38] | P_rec_dq[49] | P_rec_dq[51] | P_rec_dq[60] | P_h16b_en | P_haddr[1] | AB |
| P_rec_ma[10] | P_rec_dq[4] | P_rec_dq[10] | P_rec_dq[17] | P_rec_dq[25] | P_rec_dq[31] | P_rec_dram_clk | P_rec_dq[34] | P_rec_dq[40] | P_rec_dq[48] | P_rec_dq[53] | P_rec_dq[61] | P_hspb | AC |
| P_rec_ma[8] | P_rec_dq[2] | P_rec_dq[7] | P_rec_dq[12] | P_rec_dq[19] | P_rec_dq[26] | P_rec_rasb | P_rec_ba[0] | P_rec_dq[35] | P_rec_dq[41] | P_rec_dq[46] | P_rec_dq[55] | P_rec_dq[62] | AD |
| P_rec_ma[6] | P_rec_dq[1] | VDE | P_rec_dq[8] | P_rec_dq[13] | P_rec_dq[20] | P_rec_dq[27] | VDE | P_rec_ba[1] | P_rec_dq[36] | P_rec_dq[42] | VDE | P_rec_dq[56] | AE |
| P_rec_ma[5] | P_rec_dq[0] | VSS | P_rec_dq[6] | P_rec_dq[9] | P_rec_dq[14] | P_rec_dq[21] | VSS | P_rec_casb | P_rec_dq[32] | P_rec_dq[37] | P_rec_dq[44] | VSS | AF |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | |

**FIGURE 4. MAP OF THE LOWER-RIGHT QUADRANT**



| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| P | P_din8[2] | P_din8[3] | P_din8[4] | P_din8[6] | P_din8[7] | P_din8[5] | P_ck27pb1 | VDD | VSS | VDD | VSS | VSS | VSS |
| R | P_din_pb1[0] | P_din_pb1[1] | P_din_pb1[2] | P_din_pb1[3] | P_din_pb1[5] | VDE | P_din_pb1[4] | VSS | VDD | VSS | VSS | VSS | VSS |
| T | P_xtal_in1 | P_xtal_in2 | MCK_PLL_AVD | VCK_PLL_AVD | SCK_PLL_AVD | P_din_pb1[7] | P_din_pb1[6] | VSS | VDD | VSS | VSS | VSS | VSS |
| U | VSS | VDE | MCK_PLL_AVS | VCK_PLL_AVS | SCK_PLL_AVS | P_ck27pb2 | P_din_pb2[6] | VDD | VSS | VDD | VSS | VSS | VDD |
| V | P_din_pb2[0] | P_din_pb2[1] | P_din_pb2[2] | P_din_pb2[3] | P_din_pb2[5] | VDE | P_bt601_H1 | VDD | VSS | VSS | VDD | VDD | VSS |
| W | P_din_pb2[4] | P_din_pb2[7] | P_ck27pb3 | P_din_pb3[2] | P_ck27pb4 | P_bt601_V2 | P_rec1_enc_data[1] | VSS | VDD | VDD | VSS | DAC_SPOT3_AVS | DAC_SPOT4_AVD |
| Y | P_bt601_V1 | P_din_pb3[0] | P_din_pb3[1] | P_din_pb3[7] | P_din_pb4[4] | P_rec1_clkp | P_rec1_enc_data[7] | P_rec2_enc_data[3] | P_rec3_enc_data[0] | P_rec3_bt601_H3 | P_rec4_enc_data[6] | DAC_SPOT3_AVD | DAC_SPOT4_AVS |
| AA | VSS | VDE | P_din_pb3[6] | P_din_pb4[3] | P_rec1_enc_data[2] | VDE | P_rec1_bt601_H1 | P_rec2_enc_data[5] | P_rec3_enc_data[4] | P_rec4_enc_data[0] | P_rec4_bt601_H4 | VDE | P_rec_ma[0] |
| AB | P_din_pb3[3] | P_din_pb3[5] | P_din_pb4[2] | P_rec1_clkn | P_rec1_enc_data[3] | P_rec1_bt601_V1 | P_rec2_clkn | P_rec2_bt601_F2 | P_rec3_enc_data[1] | P_rec3_bt601_F3 | P_rec4_enc_data[3] | P_ana_spot3_VREF | P_ana_spot4_CVBS |
| AC | P_din_pb3[4] | P_din_pb4[1] | P_din_pb4[7] | P_rec1_enc_data[0] | P_rec1_bt601_F1 | P_rec2_clkp | P_rec2_enc_data[4] | P_rec3_clkn | P_rec3_enc_data[2] | P_rec4_clkn | P_rec4_enc_data[4] | P_ana_spot3_VRO | P_rec_ma[1] |
| AD | P_din_pb4[0] | P_din_pb4[6] | P_bt601_H2 | P_rec1_enc_data[4] | P_rec2_enc_data[0] | P_rec2_enc_data[6] | P_rec3_clkp | P_rec3_enc_data[3] | P_rec3_enc_data[6] | P_rec4_clkp | P_rec4_enc_data[5] | P_ana_spot3_CVBS | P_rec_ma[2] |
| AE | P_din_pb4[5] | VDE | P_rec1_enc_data[5] | P_rec2_enc_data[1] | P_rec2_enc_data[7] | VDE | P_rec3_enc_data[5] | P_rec3_enc_data[7] | P_rec4_enc_data[1] | P_rec4_enc_data[7] | VDE | P_ana_spot4_VREF | P_rec_ma[3] |
| AF | VSS | P_rec1_enc_data[6] | P_rec2_enc_data[2] | P_rec2_bt601_H2 | P_rec2_bt601_V2 | VSS | P_rec3_bt601_V3 | P_rec4_enc_data[2] | P_rec4_bt601_V4 | P_rec4_bt601_F4 | VSS | P_ana_spot4_VRO | P_rec_ma[4] |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |

**FIGURE 5. MAP OF THE LOWER-LEFT QUDRANT**

# AC/DC Electrical Parameters

### ABSOLUTE MAXIMUM RATINGS

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|
| DAC AVD | DAC_VDD | -0.5 | 3.3 | 3.6 | V |
| Vref | VREF_VDD | -0.5 | 1.1 | 1.3 | V |
| PLL AVD | PLL_VDD | -0.5 | 1.2 | 1.8 | V |
| HDMI AVD | HDMI_VDD | -0.5 | 3.3 | 3.6 | V |
| VDD 1.2V | VDD | -0.5 | 1.2 | 1.8 | V |
| VDD 3.3V | VDE | -0.5 | 3.3 | 4.6 | V |
| Voltage on Any Digital Data Pin (See Caution statement) | - | -0.5 | - | 3.8 | V |
| Voltage on OSC Related Analog Pin | - | -0.5 | - | 3.8 | V |
| Storage Temperature | $T_S$ | -55 | - | 125 | °C |
| Junction Temperature | $T_J$ | -40 | - | 125 | °C |
| Reflow Soldering ( ≤ 10 Seconds) | $T_{PEAK}$ | - | - | 240 - 250 | °C |

*CAUTION: Do not operate at or near the maximum ratings listed for extended periods of time. Exposure to such conditions may adversely impact product reliability and result in failures not covered by warranty.*

### RECOMMENDED OPERATING CONDITIONS

| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|
| DAC AVD (measured to VSSDAC) | | 2.7 | 3.3 | 3.6 | V |
| Vref | | 1.05 | 1.1 | 1.15 | V |
| PLL AVD (measured to VSSPLL) | | 1.15 | 1.2 | 1.3 | V |
| HDMI VDP | | 3.0 | 3.3 | 3.6 | V |
| HDMI VDN,VDU | | 1.15 | 1.2 | 1.3 | V |
| VDD (measured to VSS) | | 1.15 | 1.2 | 1.3 | V |
| VDDO (measured to VSSO) | | 3.0 | 3.3 | 3.6 | V |
| Ambient Operating Temperature | $T_A$ | 0 | 25 | 70 | °C |

### SUPPLY CURRENT AND POWER DISSIPATION

| PARAMETER | SYMBOL | MIN (NOTE 1) | TYP | MAX (NOTE 1) | UNITS |
|---|---|---|---|---|---|
| PLL Supply Current (1.2V nom) | $I_{DDP}$ | 14 | 15.6 | 17 | mA |
| DAC Supply Current (3.3V nom) | $I_{DDD}$ | 280 | 280 | 285 | mA |
| HDMI Analog Supply Current (1.2V nom) | $I_{DDH}$ | 26.8 | 28.4 | 30.3 | mA |
| HDMI Digital Supply Current (1.2V nom) | $I_{DDH}$ | 11.3 | 12.36 | 13.8 | mA |

| PARAMETER | SYMBOL | MIN (NOTE 1) | TYP | MAX (NOTE 1) | UNITS |
|---|---|---|---|---|---|
| Digital Internal Supply Current (1.2V nom) | $I_{DDI}$ | 490 | 540 | 610 | mA |
| Digital I/O Supply Current (3.3V nom) | $I_{DDO}$ | 110 | 140 | 150 | mA |
| Total Power Dissipation | Pd | 1.81 | 2.16 | 2.5 | W |

**PLL AC PARAMETERS**

| PARAMETER | SYMBOL | MIN (NOTE 1) | TYP | MAX (NOTE 1) | UNITS |
|---|---|---|---|---|---|
| Period Jitter | $T_{jit}$ (per) | — | ±0.01 / Fout (Hz) | ±0.025 / Fout (Hz) | (P-P) s |
| Phase Jitter | $T_{jit}$ (φ) | — | — | ±150 | (P-P) ps |
| Phase Skew | $T_{pskew}$ | — | — | ±400 | (P-P) ps |
| Look up Time | $T_L$ | — | — | 250 | µs |
| Duty Cycle | Odc | 40 | 50 | 60 | % |

**DC CHARACTERISTICS**

| PARAMETER | SYMBOL | MIN (NOTE 1) | TYP | MAX (NOTE 1) | UNITS |
|---|---|---|---|---|---|
| Digital Inputs | | | | | |
| Input High Voltage (TTL) | $V_{IH}$ | 2.0 | — | $V_{DDE}$ + 0.3 | V |
| Input Low Voltage (TTL) | $V_{IL}$ | –0.3 | — | 0.8 | V |
| Input Leakage Current (@$V_I$=3.3V or 0V) | $I_L$ | — | — | ± 4 | µA |
| Input Capacitance | $C_{IN}$ | — | — | 16 | pF |
| Digital Outputs | | | | | |
| Output High Voltage | $V_{OH}$ | $V_{DDE}$ –0.2 | — | $V_{DDE}$ | V |
| Output Low Voltage | $V_{OL}$ | 0 | — | 0.2 | V |
| High Level Output Current (@$V_{OH}$=2.8V) | $I_{OH}$ | *1 (Note 2) | | | mA |
| Low Level Output Current (@$V_{OL}$=0.2V) | $I_{OL}$ | | | | mA |
| Tri-state Output Leakage Current (@$V_O$=2.5V or 0V) | $I_{OZ}$ | — | — | ± 4 | µA |
| Output Capacitance | $C_O$ | — | — | 16 | pF |
| Analog Outputs | | | | | |
| DAC DC Output | Voc | -0.1 | — | 1.2 | V |

| PARAMETER | SYMBOL | MIN (NOTE 1) | TYP | MAX (NOTE 1) | UNITS |
|---|---|---|---|---|---|
| HDMI DC Output | --- | 0.4 | 0.5 | 0.6 | V |

**ANALOG PERFORMANCE PARAMETER**

| PARAMETER | SYMBOL | MIN (NOTE 1) | TYP | MAX (NOTE 1) | UNITS |
|---|---|---|---|---|---|
| DAC Characteristic | | | | | |
| Differential Non-linearity | $D_{NL}$ | -1 | — | +1 | LSB |
| Output Impedance | $Z_o$ | — | 50 | — | KΩs |
| DAC-to-DAC Matching | | — | 3 | — | % |
| Signal-to-Noise and Distortion Ratio | SNDR | — | 40 | — | dB |

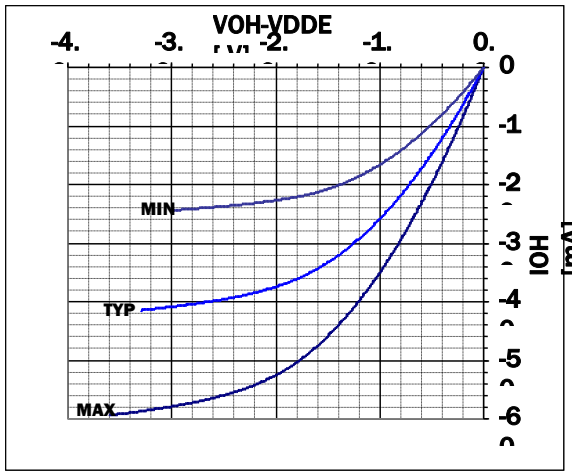
NOTES:
1. Compliance to datasheet limits is assured by one or more methods: production test, characterization and/or design.
2. *1 Please refer to the CS101 I-V graph.

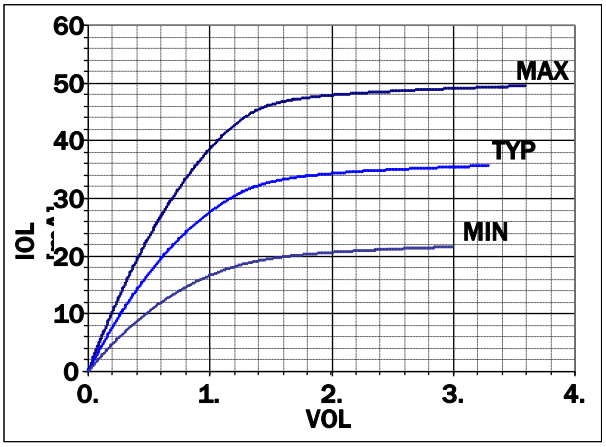
**3.3V CMOS OUTPUT I/O, M-TYPE V-I CHARACTERISTICS**

**Conditions:** MIN : Process = Slow, $T_j$ = 125°C, $V_{DDE}$ = 3.00V

TYP : Process = Typical, $T_j$ = 25°C, $V_{DDE}$ = 3.30V

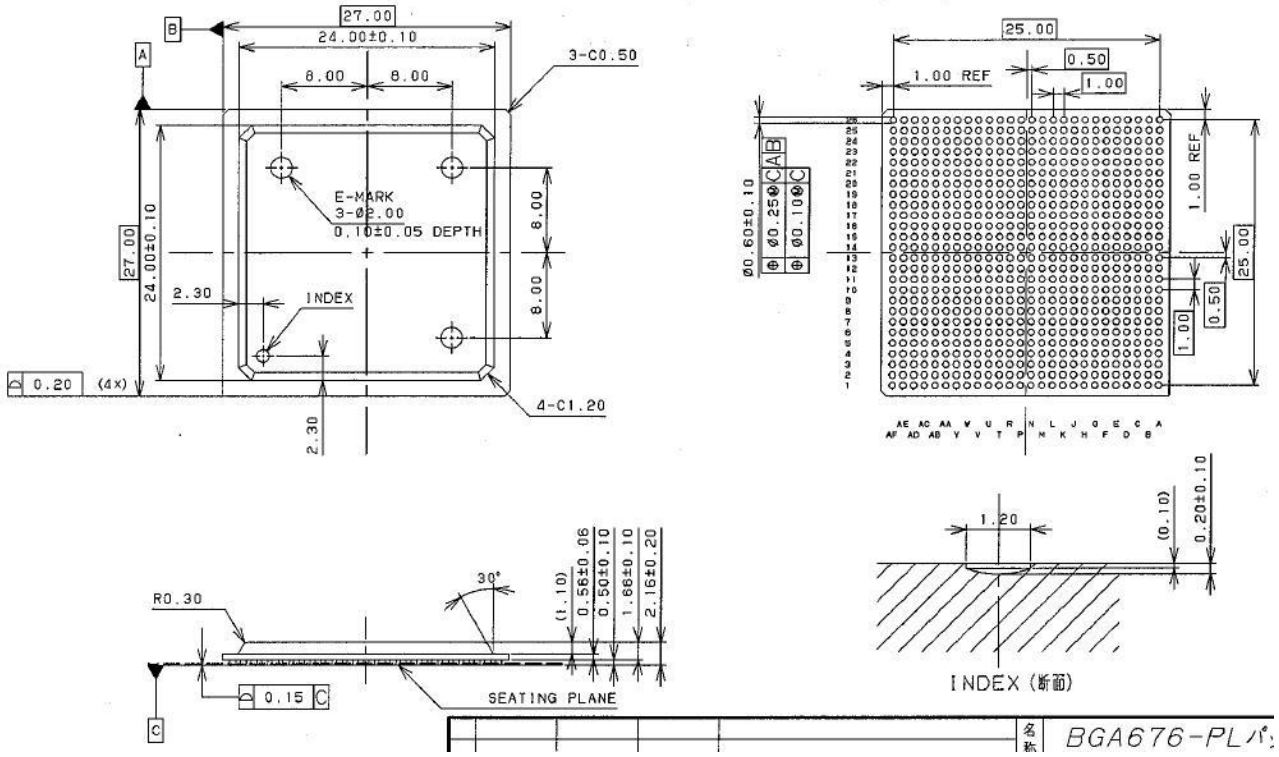
MAX : Process = Fast, $T_j$ = -40°C, $V_{DDE}$ = 3.60V



## Chip Package

**676-PIN Plastic BGA (27x27)**




**NOTE:** **All units in mm**

## Different Versions of TW2880

**There are five possible versions of TW2880, each is tailored to a specific market. Please see the following table to understand the details. Currently, only two versions are available to order. One is called TW2880-P and the other is called TW2880-N. The difference between the two is TW2880-N's ball M4 is tied to ground, therefore, it only supports up to 12 channels of live inputs.**

| NAME | TYPE | BALL #84 (PAD0) | BALL #96 (PAD1) | OTHER OPTIONS |
|---|---|---|---|---|
| TW2880P | Premium | H | H | Full option |
| TW2880N | 9 CH | H | H | Ball M4 (pad#70, P_ckvin_78) bond to low. |
| TW2880S | China, 3 SPOT | H | H | Ball AB13(pad#235, P_ana_SPOT4_CVBS) disconnect |
| TW2880M | Mid end | H | L | Display 64 bit, record 32 bit |
| TW2880B | Basic | L | L | Display 32 bit, record 32 bit |

# Video Interface and Down Scaler

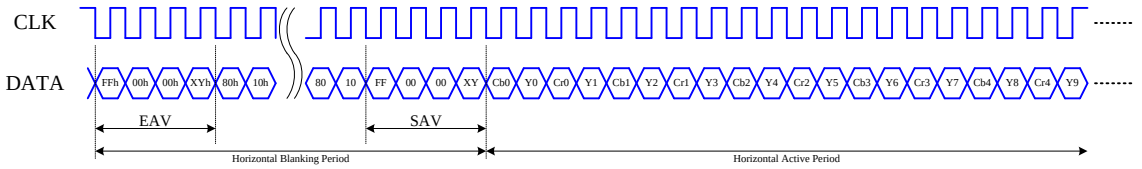
## Digital Video Input

The TW2880 has 8 digital video input ports which can receive video stream in normal ITU-R BT656 format running at 27 MHz, byte-interleaved BT.656 standard running at 54 or 108 MHz. In the latter configurations, TW2880 is expecting outputs from chips like TW2866 or TW2864 where high speed options are available. If TW2864 running at 108 MHz is used, only 4 input ports are needed. The unused ports can be used for other purposes. One thing needs to pay attention is the decoder has to use the clock provided by TW2880. This is for data synchronization purpose.

There is a small 16x16 FIFO in the beginning of the BT.656 decoder. The purpose is to accommodate possible drifting between input clock and the system clock.

In addition to the live inputs, there are four BT.656 ports running 27 (54 or 108) MHz or two BT.1120 HD ports available for playback monitoring. HD decoder will run at 74.25 MHz and in this mode, the display clock is set to twice the HD clock in 1080p mode.

After the digital data is decoded by the built-in BT 656 decoder, the image will be fed into a down scalar to adjust the final image to the desired sizes. The down scaler also generates corresponding vertical and horizontal timing signals and sends those to the write buffer.



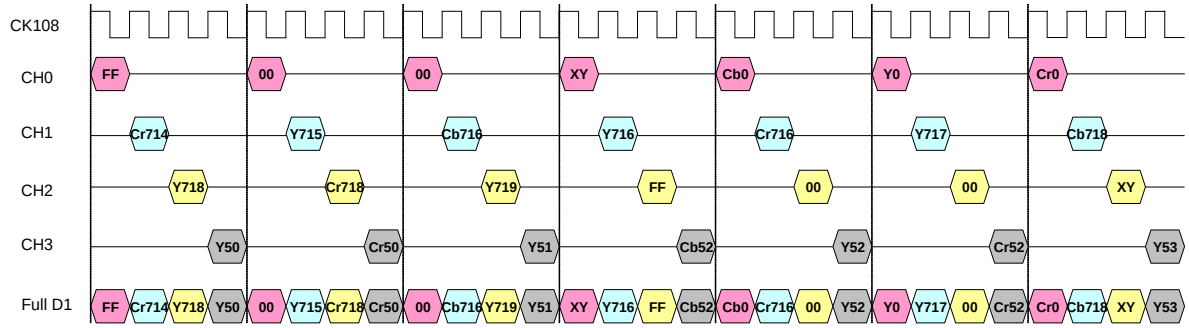

TIMING DIAGRAM OF BT.656 FORMAT FOR DIGITAL VIDEO INPUT

| CONDITION | | | 656 FVH VALUE | | | SAV/EAV CODE SEQUENCE | | | |
|---|---|---|---|---|---|---|---|---|---|
| FIELD | VERTICAL | HORIZONTAL | F | V | H | FIRST | SECOND | THIRD | FOURTH |
| EVEN | Blank | EAV | 1 | 1 | 1 | 0xFF | 0x00 | 0x00 | 0xF1 |
| | | SAV | | | 0 | | | | 0xEC |
| EVEN | Active | EAV | 1 | 0 | 1 | | | | 0xDA |
| | | SAV | | | 0 | | | | 0xC7 |
| ODD | Blank | EAV | 0 | 1 | 1 | | | | 0xB6 |
| | | SAV | | | 0 | | | | 0xAB |
| ODD | Active | EAV | 0 | 0 | 1 | | | | 0x9D |
| | | SAV | | | 0 | | | | 0x80 |

BT.656 SAV AND EAV CODE SEQUENCE

The playback port will also decode two kinds of channel IDs sent by recorder. The content is stored in registers and for CPU to use.

## Byte-interleaved Format




**FOUR CHANNEL INPUT WITH 1 PORT FOR FULL D1**

**The above diagram depicts how a high speed byte-interleaved video sequence looks like. In this case we are transmitting 4 D1 channel running at 108 MHz. This input sequence will go to a de-multiplexing module and split into four normal BT.656 sequence for later consumption.**

## Down Scaler for Live Inputs

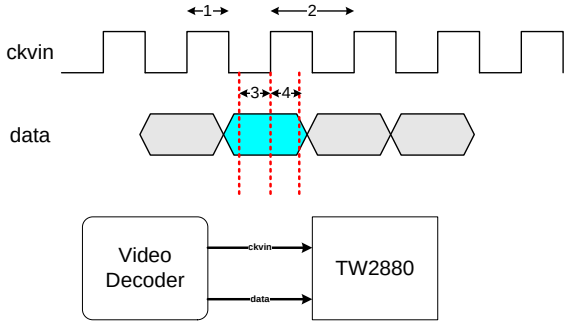
**For each live channel, two sixteen-bit registers control the final video stream size. One is for horizontal ratio and the other one is for vertical ratio. Take window 1 for example, 0x301 and 0x300 is the horizontal down scale ratio register and 0x321 and 0x320 are the vertical down scale register. The formula is:**

**Ratio = 65535 * target size / source size.**

**From this formula we can see if the When down scaler is set to 65535 (0xFFFF), the down scaler is disabled.**

**There are four down scaler for play back ports. A detailed description is provided in a separate section.**

## AC Timing for Live and PB Input




| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|
| Input Clock Half Period | 1 | 4.39 | 4.62(108) 9.26 (54), 18.52 (27) | 19.45 | ns |
| Input Clock Period | 2 | 8.78 | 9.25(108) 18.51 (54), 37.03 (27) | 38.80 | ns |
| Input Data Setup Time | 3 | 3 | | | ns |
| Input Data Hold Time | 4 | 1.5 | | | ns |

# Register Table

**Page 3 is for live channel and play back channel**

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x300 | R/W | 0xFF | HSCL_X1[7:0] |
| 0x301 | R/W | 0x7F | HSCL_X1[15:8] |
| 0x302 | R/W | 0xFF | HSCL_X2[7:0] |
| 0x303 | R/W | 0x7F | HSCL_X2[15:8] |
| 0x304 | R/W | 0xFF | HSCL_X3[7:0] |
| 0x305 | R/W | 0x7F | HSCL_X3[15:8] |
| 0x306 | R/W | 0xFF | HSCL_X4[7:0] |
| 0x307 | R/W | 0x7F | HSCL_X4[15:8] |
| 0x308 | R/W | 0xFF | HSCL_X5[7:0] |
| 0x309 | R/W | 0x7F | HSCL_X5[15:8] |
| 0x30A | R/W | 0xFF | HSCL_X6[7:0] |
| 0x30B | R/W | 0x7F | HSCL_X6[15:8] |
| 0x30C | R/W | 0xFF | HSCL_X7[7:0] |
| 0x30D | R/W | 0x7F | HSCL_X7[15:8] |
| 0x30E | R/W | 0xFF | HSCL_X8[7:0] |
| 0x30F | R/W | 0x7F | HSCL_X8[15:8] |
| 0x310 | R/W | 0xFF | HSCL_X9[7:0] |
| 0x311 | R/W | 0x7F | HSCL_X9[15:8] |
| 0x312 | R/W | 0xFF | HSCL_X10[7:0] |
| 0x313 | R/W | 0x7F | HSCL_X10[15:8] |
| 0x314 | R/W | 0xFF | HSCL_X11[7:0] |
| 0x315 | R/W | 0x7F | HSCL_X11[15:8] |
| 0x316 | R/W | 0xFF | HSCL_X12[7:0] |
| 0x317 | R/W | 0x7F | HSCL_X12[15:8] |
| 0x318 | R/W | 0xFF | HSCL_X13[7:0] |
| 0x319 | R/W | 0x7F | HSCL_X13[15:8] |
| 0x31A | R/W | 0xFF | HSCL_X14[7:0] |
| 0x31B | R/W | 0x7F | HSCL_X14[15:8] |
| 0x31C | R/W | 0xFF | HSCL_X15[7:0] |
| 0x31D | R/W | 0x7F | HSCL_X15[15:8] |
| 0x31E | R/W | 0xFF | HSCL_X16[7:0] |
| 0x31F | R/W | 0x7F | HSCL_X16[15:8] |
| 0x320 | R/W | 0xFF | VSCL_X1[7:0] |
| 0x321 | R/W | 0x7F | VSCL_X1[15:8] |
| 0x322 | R/W | 0xFF | VSCL_X2[7:0] |
| 0x323 | R/W | 0x7F | VSCL_X2[15:8] |
| 0x324 | R/W | 0xFF | VSCL_X3[7:0] |
| 0x325 | R/W | 0x7F | VSCL_X3[15:8] |
| 0x326 | R/W | 0xFF | VSCL_X4[7:0] |
| 0x327 | R/W | 0x7F | VSCL_X4[15:8] |
| 0x328 | R/W | 0xFF | VSCL_X5[7:0] |
| 0x329 | R/W | 0x7F | VSCL_X5[15:8] |
| 0x32A | R/W | 0xFF | VSCL_X6[7:0] |
| 0x32B | R/W | 0x7F | VSCL_X6[15:8] |
| 0x32C | R/W | 0xFF | VSCL_X7[7:0] |
| 0x32D | R/W | 0x7F | VSCL_X7[15:8] |
| 0x32E | R/W | 0xFF | VSCL_X8[7:0] |
| 0x32F | R/W | 0x7F | VSCL_X8[15:8] |
| 0x330 | R/W | 0xFF | VSCL_X9[7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x331 | R/W | 0x7F | VSCL_X9[15:8] |
| 0x332 | R/W | 0xFF | VSCL_X10[7:0] |
| 0x333 | R/W | 0x7F | VSCL_X10[15:8] |
| 0x334 | R/W | 0xFF | VSCL_X11[7:0] |
| 0x335 | R/W | 0x7F | VSCL_X11[15:8] |
| 0x336 | R/W | 0xFF | VSCL_X12[7:0] |
| 0x337 | R/W | 0x7F | VSCL_X12[15:8] |
| 0x338 | R/W | 0xFF | VSCL_X13[7:0] |
| 0x339 | R/W | 0x7F | VSCL_X13[15:8] |
| 0x33A | R/W | 0xFF | VSCL_X14[7:0] |
| 0x33B | R/W | 0x7F | VSCL_X14[15:8] |
| 0x33C | R/W | 0xFF | VSCL_X15[7:0] |
| 0x33D | R/W | 0x7F | VSCL_X15[15:8] |
| 0x33E | R/W | 0xFF | VSCL_X16[7:0] |
| 0x33F | R/W | 0x7F | VSCL_X16[15:8] |
| 0x340 | R/W | 0x00 | [7:6]: VSCL_MD_X4<br>[5:4]: VSCL_MD_X3<br>[3:2]: VSCL_MD_X2<br>[1:0]: VSCL_MD_X1 |
| 0x341 | R/W | 0x00 | [7:6]: VSCL_MD_X8<br>[5:4]: VSCL_MD_X7<br>[3:2]: VSCL_MD_X6<br>[1:0]: VSCL_MD_X5 |
| 0x342 | R/W | 0x00 | [7:6]: VSCL_MD_X12<br>[5:4]: VSCL_MD_X11<br>[3:2]: VSCL_MD_X10<br>[1:0]: VSCL_MD_X9 |
| 0x343 | R/W | 0x00 | [7:6]: VSCL_MD_X16<br>[5:4]: VSCL_MD_X15<br>[3:2]: VSCL_MD_X14<br>[1:0]: VSCL_MD_X13 |
| 0x344 | R/W | 0x00 | [7:6]: VSCL_MD_Y4<br>[5:4]: VSCL_MD_Y3<br>[3:2]: VSCL_MD_Y2<br>[1:0]: VSCL_MD_Y1 |
| 0x345 | R/W | 0x00 | [7:6]: VSCL_MD_Y8<br>[5:4]: VSCL_MD_Y7<br>[3:2]: VSCL_MD_Y6<br>[1:0]: VSCL_MD_Y5 |
| 0x346 | R/W | 0x00 | [7:6]: VSCL_MD_Y12<br>[5:4]: VSCL_MD_Y11<br>[3:2]: VSCL_MD_Y10<br>[1:0]: VSCL_MD_Y9 |
| 0x347 | R/W | 0x00 | [7:6]: VSCL_MD_Y16<br>[5:4]: VSCL_MD_Y15<br>[3:2]: VSCL_MD_Y14<br>[1:0]: VSCL_MD_Y13 |
| 0x348 | R/W | 0x00 | [7:6]: HLPF_MD_X4<br>[5:4]: HLPF_MD_X3<br>[3:2]: HLPF_MD_X2<br>[1:0]: HLPF_MD_X1 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x349 | R/W | 0x00 | [7:6]: HLPF_MD_X8<br>[5:4]: HLPF_MD_X7<br>[3:2]: HLPF_MD_X6<br>[1:0]: HLPF_MD_X5 |
| 0x34A | R/W | 0x00 | [7:6]: HLPF_MD_X12<br>[5:4]: HLPF_MD_X11<br>[3:2]: HLPF_MD_X10<br>[1:0]: HLPF_MD_X9 |
| 0x34B | R/W | 0x00 | [7:6]: HLPF_MD_X16<br>[5:4]: HLPF_MD_X15<br>[3:2]: HLPF_MD_X14<br>[1:0]: HLPF_MD_X13 |
| 0x34C | R/W | 0x00 | [7:6]: HLPF_MD_Y4<br>[5:4]: HLPF_MD_Y3<br>[3:2]: HLPF_MD_Y2<br>[1:0]: HLPF_MD_Y1 |
| 0x34D | R/W | 0x00 | [7:6]: HLPF_MD_Y8<br>[5:4]: HLPF_MD_Y7<br>[3:2]: HLPF_MD_Y6<br>[1:0]: HLPF_MD_Y5 |
| 0x34E | R/W | 0x00 | [7:6]: HLPF_MD_Y12<br>[5:4]: HLPF_MD_Y11<br>[3:2]: HLPF_MD_Y10<br>[1:0]: HLPF_MD_Y9 |
| 0x34F | R/W | 0x00 | [7:6]: HLPF_MD_Y16<br>[5:4]: HLPF_MD_Y15<br>[3:2]: HLPF_MD_Y14<br>[1:0]: HLPF_MD_Y13 |
| 0x350 | R/W | 0x00 | [0]: PAL_DLY_X |
| 0x351 | R/W | 0x00 | [0]: PAL_DLY_Y |
| 0x352 | R/W | 0x00 | [3:0]: ODD_SKEW |
| 0x353 | R/W | 0x00 | [3:0]: EVEN_SKEW |
| 0x354 | R/W | 0x00 | [7]: MAN_VSCL_LPFY8<br>[6]: MAN_VSCL_LPFY7<br>[5]: MAN_VSCL_LPFY6<br>[4]: MAN_VSCL_LPFY5<br>[3]: MAN_VSCL_LPFY4<br>[2]: MAN_VSCL_LPFY3<br>[1]: MAN_VSCL_LPFY2<br>[0]: MAN_VSCL_LPFY1 |
| 0x355 | R/W | 0x00 | [7]: MAN_VSCL_LPFY16<br>[6]: MAN_VSCL_LPFY15<br>[5]: MAN_VSCL_LPFY14<br>[4]: MAN_VSCL_LPFY13<br>[3]: MAN_VSCL_LPFY12<br>[2]: MAN_VSCL_LPFY11<br>[1]: MAN_VSCL_LPFY10<br>[0]: MAN_VSCL_LPFY9 |
| 0x356 | R/W | 0x00 | [0]: TST_VSCL_VAV |
| 0x35A | R/W | 0xf0 | VTAR_PB2[7:0] |
| 0x35B | R/W | 0xf0 | VTAR_PB3[7:0] |
| 0x35C | R/W | 0x00 | [6:4]: VTAR_PB4[10:8]<br>[2:0]: VTAR_PB3[10:8] |
| 0x35D | R/W | 0xf0 | VTAR_PB4[7:0] |
| 0x35E | R/W | 0xf0 | VSOR_PB1[7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x35F | R/W | 0x00 | [6:4]: VSOR_PB2[10:8]<br>[2:0]: VSOR_PB1[10:8] |
| 0x360 | R/W | 0xf0 | VSOR_PB2[7:0] |
| 0x361 | R/W | 0xf0 | VSOR_PB3[7:0] |
| 0x362 | R/W | 0x00 | [6:4]: VSOR_PB4[10:8]<br>[2:0]: VSOR_PB3[10:8] |
| 0x363 | R/W | 0xf0 | VSOR_PB4[7:0] |
| 0x364 | R/W | 0xd0 | HTAR_PB1[7:0] |
| 0x365 | R/W | 0x22 | [6:4]: HTAR_PB2[10:8]<br>[2:0]: HTAR_PB1[10:8] |
| 0x366 | R/W | 0xd0 | HTAR_PB2[7:0] |
| 0x367 | R/W | 0xd0 | HTAR_PB3[7:0] |
| 0x368 | R/W | 0x22 | [6:4]: HTAR_PB4[10:8]<br>[2:0]: HTAR_PB3[10:8] |
| 0x369 | R/W | 0xd0 | HTAR_PB4[7:0] |
| 0x36A | R/W | 0xd0 | HSOR_PB1[7:0] |
| 0x36B | R/W | 0x22 | [6:4]: HSOR_PB2[10:8]<br>[2:0]: HSOR_PB1[10:8] |
| 0x36C | R/W | 0xd0 | HSOR_PB2[7:0] |
| 0x36D | R/W | 0xd0 | HSOR_PB3[7:0] |
| 0x36E | R/W | 0x22 | [6:4]: HSOR_PB4[10:8]<br>[2:0]: HSOR_PB3[10:8] |
| 0x36F | R/W | 0xd0 | HSOR_PB4[7:0] |
| 0x370 | R/W | 0x00 | [7]: QUAD_AUTO_SCL_PB4<br>[6]: QUAD_AUTO_SCL_PB3<br>[5]: QUAD_AUTO_SCL_PB2<br>[4]: QUAD_AUTO_SCL_PB1<br>[3]: BYPASS_PB4<br>[2]: BYPASS_PB3<br>[1]: BYPASS_PB2<br>[0]: BYPASS_PB1 |
| 0x371 | R/W | 0x00 | [7]: H_601_INV<br>[6]: SYNC_SEL<br>[5]: YC_SWITCH<br>[4:2]: reserved<br>[1]: IN16_MODE_PB34<br>[0]: IN16_MODE_PB12 |
| 0x372 | RO | 0 | CHID_RD_BUS_PB1[39:32] |
| 0x373 | RO | 0 | CHID_RD_BUS_PB1[31:24] |
| 0x374 | RO | 0 | CHID_RD_BUS_PB1[23:16] |
| 0x375 | R/W | 0x00 | HDELAY_X1 |
| 0x376 | R/W | 0x00 | HDELAY_X2 |
| 0x377 | R/W | 0x00 | HDELAY_X3 |
| 0x378 | R/W | 0x00 | HDELAY_X4 |
| 0x379 | R/W | 0x00 | HDELAY_X5 |
| 0x37A | R/W | 0x00 | HDELAY_X6 |
| 0x37B | R/W | 0x00 | HDELAY_X7 |
| 0x37C | R/W | 0x00 | HDELAY_X8 |
| 0x37D | R/W | 0x00 | HDELAY_X9 |
| 0x37E | R/W | 0x00 | HDELAY_X10 |
| 0x37F | R/W | 0x00 | HDELAY_X11 |
| 0x380 | R/W | 0x00 | HDELAY_X12 |
| 0x381 | R/W | 0x00 | HDELAY_X13 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x382 | R/W | 0x00 | HDELAY_X14 |
| 0x383 | R/W | 0x00 | HDELAY_X15 |
| 0x384 | R/W | 0x00 | HDELAY_X16 |
| 0x385 | R/W | 0x00 | [3:0]: HDELAY_Y1 |
| 0x386 | R/W | 0x00 | [3:0]: HDELAY_Y2 |
| 0x387 | R/W | 0x00 | [3:0]: HDELAY_Y3 |
| 0x388 | R/W | 0x00 | [3:0]: HDELAY_Y4 |
| 0x389 | R/W | 0x00 | [3:0]: HDELAY_Y5 |
| 0x38A | R/W | 0x00 | [3:0]: HDELAY_Y6 |
| 0x38B | R/W | 0x00 | [3:0]: HDELAY_Y7 |
| 0x38C | R/W | 0x00 | [3:0]: HDELAY_Y8 |
| 0x38D | R/W | 0x00 | [3:0]: HDELAY_Y9 |
| 0x38E | R/W | 0x00 | [3:0]: HDELAY_Y10 |
| 0x38F | R/W | 0x00 | [3:0]: HDELAY_Y11 |
| 0x390 | R/W | 0x00 | [3:0]: HDELAY_Y12 |
| 0x391 | R/W | 0x00 | [3:0]: HDELAY_Y13 |
| 0x392 | R/W | 0x00 | [3:0]: HDELAY_Y14 |
| 0x393 | R/W | 0x00 | [3:0]: HDELAY_Y15 |
| 0x394 | R/W | 0x00 | [3:0]: HDELAY_Y16 |
| 0x395 | R/W | 0x00 | VDELAY_X1 |
| 0x396 | R/W | 0x00 | VDELAY_X2 |
| 0x397 | R/W | 0x00 | VDELAY_X3 |
| 0x398 | R/W | 0x00 | VDELAY_X4 |
| 0x399 | R/W | 0x00 | VDELAY_X5 |
| 0x39A | R/W | 0x00 | VDELAY_X6 |
| 0x39B | R/W | 0x00 | VDELAY_X7 |
| 0x39C | R/W | 0x00 | VDELAY_X8 |
| 0x39D | R/W | 0x00 | VDELAY_X9 |
| 0x39E | R/W | 0x00 | VDELAY_X10 |
| 0x39F | R/W | 0x00 | VDELAY_X11 |
| 0x3A0 | R/W | 0x00 | VDELAY_X12 |
| 0x3A1 | R/W | 0x00 | VDELAY_X13 |
| 0x3A2 | R/W | 0x00 | VDELAY_X14 |
| 0x3A3 | R/W | 0x00 | VDELAY_X15 |
| 0x3A4 | R/W | 0x00 | VDELAY_X16 |
| 0x3A5 | R/W | 0x00 | [3:0]: VDELAY_Y1 |
| 0x3A6 | R/W | 0x00 | [3:0]: VDELAY_Y2 |
| 0x3A7 | R/W | 0x00 | [3:0]: VDELAY_Y3 |
| 0x3A8 | R/W | 0x00 | [3:0]: VDELAY_Y4 |
| 0x3A9 | R/W | 0x00 | [3:0]: VDELAY_Y5 |
| 0x3AA | R/W | 0x00 | [3:0]: VDELAY_Y6 |
| 0x3AB | R/W | 0x00 | [3:0]: VDELAY_Y7 |
| 0x3AC | R/W | 0x00 | [3:0]: VDELAY_Y8 |
| 0x3AD | R/W | 0x00 | [3:0]: VDELAY_Y9 |
| 0x3AE | R/W | 0x00 | [3:0]: VDELAY_Y10 |
| 0x3AF | R/W | 0x00 | [3:0]: VDELAY_Y11 |
| 0x3B0 | R/W | 0x00 | [3:0]: VDELAY_Y12 |
| 0x3B1 | R/W | 0x00 | [3:0]: VDELAY_Y13 |
| 0x3B2 | R/W | 0x00 | [3:0]: VDELAY_Y14 |
| 0x3B3 | R/W | 0x00 | [3:0]: VDELAY_Y15 |
| 0x3B4 | R/W | 0x00 | [3:0]: VDELAY_Y16 |
| 0x3B5 | R/W | 0x00 | [7:0]: HDELAY_PB1 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x3B6 | R/W | 0x00 | [7:0]: VDELAY_PB1 |
| 0x3B7 | R/W | 0x00 | [7:0]: HDELAY_PB2 |
| 0x3B8 | R/W | 0x00 | [7:0]: VDELAY_PB2 |
| 0x3B9 | R/W | 0x00 | [7:0]: HDELAY_PB3 |
| 0x3BA | R/W | 0x00 | [7:0]: VDELAY_PB3 |
| 0x3BB | R/W | 0x00 | [7:0]: HDELAY_PB4 |
| 0x3BC | R/W | 0x00 | [7:0]: VDELAY_PB4 |
| 0x3BD | R/W | 0 | [7:6]: MAN_NOVID4<br>[5:4]: MAN_NOVID3<br>[3:2]: MAN_NOVID2<br>[1:0]: MAN_NOVID1 |
| 0x3BE | R/W | 0 | [7:6]:MAN_NOVID8<br>[5:4]: MAN_NOVID7<br>[3:2]: MAN_NOVID6<br>[1:0]: MAN_NOVID5 |
| 0x3BF | R/W | 0 | [7:6]:MAN_NOVID12<br>[5:4]: MAN_NOVID11<br>[3:2]: MAN_NOVID10<br>[1:0]: MAN_NOVID9 |
| 0x3C0 | R/W | 0 | [7:6]:MAN_NOVID16<br>[5:4]: MAN_NOVID15<br>[3:2]: MAN_NOVID14<br>[1:0]: MAN_NOVID13 |
| 0x3C1 | R/W | 0 | [7:6]:MAN_NON-STANDRED4<br>[5:4]: MAN_NON-STANDRED3<br>[3:2]: MAN_NON-STANDRED2<br>[1:0]: MAN_NON-STANDRED1 |
| 0x3C2 | R/W | 0 | [7:6]:MAN_NON-STANDRED8<br>[5:4]: MAN_NON-STANDRED7<br>[3:2]: MAN_NON-STANDRED6<br>[1:0]: MAN_NON-STANDRED5 |
| 0x3C3 | R/W | 0 | [7:6]:MAN_NON-STANDRED12<br>[5:4]: MAN_NON-STANDRED11<br>[3:2]: MAN_NON-STANDRED10<br>[1:0]: MAN_NON-STANDRED9 |
| 0x3C4 | R/W | 0 | [7:6]:MAN_NON-STANDRED16<br>[5:4]: MAN_NON-STANDRED15<br>[3:2]: MAN_NON-STANDRED14<br>[1:0]: MAN_NON-STANDRED13 |
| 0x3C5 | R/W | 0 | [6]: STILL_PAT<br>[5]: PALNT<br>[4]: IN_TEST<br>[3]: Reserved<br>[2]: CHID_BIT<br>[1]: NO_CHID<br>[0]: SYNC_CHID |
| 0x3C6 | R/W | 0 | [0]: SYS_60 |
| 0x3C7 | RO | 0 | CHID_RD_BUS_PB1[15:8] |
| 0x3C8 | RO | 0 | CHID_RD_BUS_PB1[7:0] |
| 0x3C9 | RO | 0 | CHID_RD_BUS_PB2[39:32] |
| 0x3CA | RO | 0 | CHID_RD_BUS_PB2[31:24] |
| 0x3CC | RO | 0 | CHID_RD_BUS_PB2[23:16] |
| 0x3CD | RO | 0 | CHID_RD_BUS_PB2[15:8] |
| 0x3CE | RO | 0 | CHID_RD_BUS_PB2[7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x3CF | RO | 0 | CHID_RD_BUS_PB3[39:32] |
| 0x3D0 | RO | 0 | CHID_RD_BUS_PB3[31:24] |
| 0x3D1 | RO | 0 | CHID_RD_BUS_PB3[23:16] |
| 0x3D2 | R/W | 0x10 | [7:4]: VIN_SEL_X2 / VIN_SEL_Y2<br>[3:0]: VIN_SEL_X1 / VIN_SEL_Y1 |
| 0x3D3 | R/W | 0x32 | [7:4]: VIN_SEL_X4 / VIN_SEL_Y4<br>[3:0]: VIN_SEL_X3 / VIN_SEL_Y3 |
| 0x3D4 | R/W | 0x54 | [7:4]: VIN_SEL_X6 / VIN_SEL_Y6<br>[3:0]: VIN_SEL_X5 / VIN_SEL_Y5 |
| 0x3D5 | R/W | 0x76 | [7:4]: VIN_SEL_X8 / VIN_SEL_Y8<br>[3:0]: VIN_SEL_X7 / VIN_SEL_Y7 |
| 0x3D6 | R/W | 0x98 | [7:4]: VIN_SEL_X10 / VIN_SEL_Y10<br>[3:0]: VIN_SEL_X9 / VIN_SEL_Y9 |
| 0x3D7 | R/W | 0xBA | [7:4]: VIN_SEL_X12 / VIN_SEL_Y12<br>[3:0]: VIN_SEL_X11 / VIN_SEL_Y11 |
| 0x3D8 | R/W | 0xDC | [7:4]: VIN_SEL_X14 / VIN_SEL_Y14<br>[3:0]: VIN_SEL_X13 / VIN_SEL_Y13 |
| 0x3D9 | R/W | 0xFE | [7:4]: VIN_SEL_X16 / VIN_SEL_Y16<br>[3:0]: VIN_SEL_X15 / VIN_SEL_Y15 |
| 0x3DA | R/W | 0x01 | [7]: VIN_SEL_XY<br>[0]: V_OFST_X |
| 0x3DB | R/W | 0 | [7:6]: PB_PATH_SEL4<br>[5:4]: PB_PATH_SEL3<br>[3:2]: PB_PATH_SEL2<br>[1:0]: PB_PATH_SEL1 |
| 0x3DC | R/W | 0x12 | PB_TEST_PATTEN<br>[4]: DIR2<br>[3]: DIR1<br>[2:1]: FRAME_RATE<br>[0]: IN_PB_TEST |
| 0x3DD | RW | 0x40 | PIXEL_SHIFT |
| 0x3DE | R/W | 0x0F | [7:4]: CHID_RD_SEL_PB1<br>[3]: VBI_RIC_ON_PB1<br>[2]: VBI_AUTO_DET_PB1<br>[1]: VBI_D_EN_PB1<br>[0]: VBI_A_EN_PB1 |
| 0x3DF | R/W | 0x00 | [4:0]: VBI_VOS_PB1 |
| 0x3E0 | R/W | 0x80 | [7:5]: VBI_PIXEL_WIDTH_PB1<br>[4:0]: VBI_FOS_PB1 |
| 0x3E1 | R/W | 0 | VBI_HOS_PB1 |
| 0x3E2 | R/W | 0x1F | VBI_MID_VAL_PB1 |
| 0x3E3 | R/W | 0x0F | [7:4]: CHID_RD_SEL_PB2<br>[3]: VBI_RIC_ON_PB2<br>[2]: VBI_AUTO_DET_PB2<br>[1]: VBI_D_EN_PB2<br>[0]: VBI_A_EN_PB2 |
| 0x3E4 | R/W | 0x00 | [4:0]: VBI_VOS_PB2 |
| 0x3E5 | R/W | 0x80 | [7:5]: VBI_PIXEL_WIDTH_PB2<br>[4:0]: VBI_FOS_PB2 |
| 0x3E6 | R/W | 0x00 | VBI_HOS_PB2 |
| 0x3E7 | R/W | 0x1F | VBI_MID_VAL_PB2 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x3E8 | R/W | 0x0F | [7:4]CHID_RD_SEL_PB3<br>[3]: VBI_RIC_ON_PB3<br>[2]: VBI_AUTO_DET_PB3<br>[1]: VBI_D_EN_PB3<br>[0]: VBI_A_EN_PB3 |
| 0x3E9 | R/W | 0x00 | [4:0]: VBI_VOS_PB3 |
| 0x3EA | R/W | 0x80 | [7:5]: VBI_PIXEL_WIDTH_PB3<br>[4:0]: VBI_FOS_PB3 |
| 0x3EB | R/W | 0x00 | VBI_HOS_PB3 |
| 0x3EC | R/W | 0x1F | VBI_MID_VAL_PB3 |
| 0x3ED | R/W | 0x0F | [7:4]: CHID_RD_SEL_PB4<br>[3]: VBI_RIC_ON_PB4<br>[2]: VBI_AUTO_DET_PB4<br>[1]: VBI_D_EN_PB4<br>[0]: VBI_A_EN_PB4 |
| 0x3EE | R/W | 0x00 | [4:0] VBI_VOS_PB4 |
| 0x3EF | R/W | 0x80 | [7:5]: VBI_PIXEL_WIDTH_PB4<br>[4:0]: VBI_FOS_PB4 |
| 0x3F0 | R/W | 0x00 | VBI_HOS_PB4 |
| 0x3F1 | R/W | 0x00 | VBI_MID_VAL_PB2 |
| 0x3F2 | R/W | 0x00 | [7]: FRAME_IL_PB4_EN<br>[6]: FRAME_IL_PB3_EN<br>[5]: FRAME_IL_PB2_EN<br>[4]: FRAME_IL_PB1_EN<br>[3]: FRAME_IL_PB4<br>[2]: FRAME_IL_PB3<br>[1]: FRAME_IL_PB2<br>[0]: FRAME_IL_PB1 |
| 0x3F3 | R/W | 0x00 | [7]: FLD_MODE_PB4_EN<br>[6]: FLD_MODE_PB3_EN<br>[5]: FLD_MODE_PB2_EN<br>[4]: FLD_MODE_PB1_EN<br>[3]: FLD_MODE_PB4<br>[2]: FLD_MODE_PB3<br>[1]: FLD_MODE_PB2<br>[0]: FLD_MODE_PB1 |
| 0x3F4 | R/W | 0xFF | [7]: VBI_E_EN_PB4<br>[6]: VBI_E_EN_PB3<br>[5]: VBI_E_EN_PB2<br>[4]: VBI_E_EN_PB1<br>[3]: VBI_O_EN_PB4<br>[2]: VBI_O_EN_PB3<br>[1]: VBI_O_EN_PB2<br>[0]: VBI_O_EN_PB1 |
| 0x3FC | R/W | 0 | [7]: VBI_DLY4<br>[6]: VBI_DLY3<br>[5]: VBI_DLY2<br>[4]: VBI_DLY1<br>[3]: FLD_SEL_EN_PB4<br>[2]: FLD_SEL_EN_PB3<br>[1]: FLD_SEL_EN_PB2<br>[0]: FLD_SEL_EN_PB1 |
| 0x3FD | R/W | EE | [7:4]: VBI_SIZE_PB2<br>[3:0]: VBI_SIZE_PB1 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x3FE | R/W | EE | [7:4]: VBI_SIZE_PB4<br>[3:0]: VBI_SIZE_PB3 |
| 0x3FF | R/W | 0 | [7]: selm<br>[6]: PBX4_SEL<br>[5]: PBX2_SEL<br>[4]: MG_DIG_CODE<br>[3]: VS_DIS_PB4<br>[2]: VS_DIS_PB3<br>[1]: VS_DIS_PB2<br>[0]: VS_DIS_PB1 |

## Registers

**Following are register descriptions for the live channel down scaler. Registers related to the PB down scaler will be described in a separate section.**

### LIVE VIDEO CHANNEL 1 HORIZONTAL SCALER LOW BYTE REGISTER – 0X300

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xFF | HSCL_X1[7:0]<br><br>Horizontal scaler factor. Maximum is 0xFFFF. The number is smaller, the down scaler factor is bigger. |

### LIVE VIDEO CHANNEL 1 HORIZONTAL SCALER HIGH BYTE REGISTER – 0X301

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x7F | HSCL_X1[15:8]<br><br>Horizontal scaler factor. Maximum is 0xFFFF. The number is smaller, the down scaler factor is bigger. |

**Register 0x302 to 0x31F are for channels 2 to 16, which are similar to 0x300 and 0x301.**

### LIVE VIDEO CHANNEL 1 VERTICAL SCALER LOW BYTE REGISTER – 0X320

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xFF | VSCL_X1[7:0]<br><br>Vertical scaler factor. Maximum is 0xFFFF. The number is smaller, the down scaler factor is bigger. |

### LIVE VIDEO CHANNEL 1 VERTICAL SCALER HIGH BYTE REGISTER – 0X321

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xFF | VSCL_X1[15:8]<br><br>Vertical scaler factor. Maximum is 0xFFFF. The number is smaller, the down scaler factor is bigger. |

**Register 0x322 to 0x33F are for channels 2 to 16, which are similar to 0x320 and 0x321.**

### LIVE VIDEO VERTICAL SCALE MODE REGISTER – 0X340

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | VSCL_MD_X<br><br>[7:6]: VSCL_MD_X4<br>[5:4]: VSCL_MD_X3<br>[3:2]: VSCL_MD_X2<br>[1:0]: VSCL_MD_X1<br><br>Vertical scale mode |

Register 0x341 to 0x343 are for other channels in display down scaler. Register 0x344 to 0x347 are for record down scaler.

### LIVE VIDEO HORIZONTAL SCALE MODE REGISTER – 0X348

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | HLPF_MD_X<br><br>[7:6]: HLPF_MD_X4<br>[5:4]: HLPF_MD_X3<br>[3:2]: HLPF_MD_X2<br>[1:0]: HLPF_MD_X1 |

Register 0x349 to 0x34B are for other channels in display down scaler. Register 0x34C to 0x34F are for record down scaler.

### LIVE VIDEO PAL DELAY MODE REGISTER – 0X350

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | R/W | 0x00 | PAL_DLY_X |

Register 0x351 is for record path.

### LIVE VIDEO ODD FIELD SKEW REGISTER – 0X352

| Bit | R/W | Default | Description |
|---|---|---|---|
| 3:0 | R/W | 0x00 | ODD_SKEW |

Register 0x353 is for even field.

### LIVE VIDEO VERTICAL SCALER LPF REGISTER – 0X354

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | MAN_VSCL_LPFY<br><br>[7]: MAN_VSCL_LPFY8<br>[6]: MAN_VSCL_LPFY7<br>[5]: MAN_VSCL_LPFY6<br>[4]: MAN_VSCL_LPFY5<br>[3]: MAN_VSCL_LPFY4<br>[2]: MAN_VSCL_LPFY3<br>[1]: MAN_VSCL_LPFY2<br>[0]: MAN_VSCL_LPFY1 |

**Register 0x355 is for channels 9 to 16**

### LIVE VIDEO VERTICAL SCALE VAV TEST REGISTER – 0X356

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | R/W | 0x00 | TST_VSCL_VAV |

### LIVE VIDEO CHANNEL 1 HORIZONTAL PIXEL ADJUSTMENT REGISTER – 0X375

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | HDELAY_X1[7:0]<br><br>Number of pixels times two. This is for display video. |

**Register 0x376 to 0x384 are for channels 2 to 16.**

### LIVE VIDEO CHANNEL 1 HORIZONTAL PIXEL ADJUSTMENT REGISTER – 0X385

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0 | Reserved |
| 3:0 | RW | 0 | HDELAY_Y1[3:0]<br><br>Number of pixels times four. This is for record video. |

**Register 0x386 to 0x394 are for channels 2 to 16.**

### LIVE VIDEO CHANNEL 1 VERTICAL PIXEL ADJUSTMENT REGISTER – 0X395

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | VDELAY_X1[7:0]<br><br>Number of lines times two. This is for display video. |

### LIVE VIDEO CHANNEL 1 VERTICAL PIXEL ADJUSTMENT REGISTER – 0X3A5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0 | Reserved |
| 3:0 | RW | 0 | VDELAY_Y1[3:0]<br><br>Number of lines times four. This is for record video. |

**Register 0x3A6 to 0x3B4 are for channel 2 to 16.**

### LIVE VIDEO CHANNEL 1~4 NO VIDEO REGISTER – 0X3BD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | MAN_NOVID4: definition same as MAN_NOVID1 |
| 5:4 | RW | 0 | MAN_NOVID3: definition same as MAN_NOVID1 |
| 3:2 | RW | 0 | MAN_NOVID2: definition same as MAN_NOVID1 |
| 1:0 | RW | 0 | MAN_NOVID1: no video status override enable<br><br>If bit 1 is set to 0, no video status is automatically detected by hardware (default). If bit 1 is set to 1, no video status is manually set by bit 0. To fully utilize this feature, user needs to enable TW2864/TW2866 MPP_MODE[5]<br><br>bit 1: 1= enable no video status override<br>bit 0: 1: no video, 0: has video |

**Register 0x3BE to 0x3C0 are for channels 5 to 16.**

### LIVE VIDEO CHANNEL 1~4 NON STANDARD REGISTER – 0X3C1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | MAN_NON-STANDRED4: definition same as MAN_NON-STANDRED1 |
| 5:4 | RW | 0 | MAN_NON-STANDRED3: definition same as MAN_NON-STANDRED1 |
| 3:2 | RW | 0 | MAN_NON-STANDRED2: definition same as MAN_NON-STANDRED1 |
| 1:0 | RW | 0 | MAN_NON-STANDRED1: non-standard video status override<br><br>If bit 1 is set to 0, non-standard status is automatically detected by hardware (default). If bit 1 is set to one, non-standard status is manually set by bit 0.<br><br>Bit 1: 1= enable non-standard video status override<br>Bit 0: 1: non-standard video 0: standard video |

**Register 0x3C2 to 0x3C4 are for channels 5 to 16.**

### LIVE VIDEO CHANNEL CONTROL REGISTER – 0X3C5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0 | Reserved |
| 6 | RW | 0 | STILL_PAT<br><br>0: moving pattern<br>1: still pattern |
| 5 | RW | 0 | PALNT: Only used for test pattern<br><br>0: NTSC pattern<br>1: PAL pattern |
| 4 | RW | 0 | IN_TEST<br><br>1 = Test pattern enable |
| 3 | RW | 0 | Reserved |
| 2 | RW | 0 | CHID_BIT<br><br>0: channel ID is 1 bit<br>1: channel ID is 2 bit |
| 1 | RW | 0 | NO_CHID<br><br>0: has channel ID in video data<br>1: no channel ID in video data |
| 0 | RW | 0 | SYNC_CHID: Channel ID location<br><br>0: in horizontal blank data last 2 bit<br>1: In timing reference data protection bit |

### LIVE VIDEO CHANNEL SYSTEM SELECT REGISTER – 0X3C6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | RECX4_SEL: This bit selects PB4 input in 16 bit mode<br><br>0: Input used is same as PBX4_SEL<br>1: Input comes from REC3 and REC4. |
| 6 | RW | | RECX2_SEL: This bit selects PB2 input in 16 bit mode<br><br>0: Input used is same as PBX2_SEL<br>1: Input comes from REC1 and REC2. |
| 5:1 | R | 0 | Reserved |
| 0 | RW | 0 | SYS_60<br><br>0: 50Hz, PAL<br>1: 60Hz, NTSC |

### LIVE/RECORD CHANNEL SELECT REGISTER – 0X3D2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 1 | VIN_SEL2<br><br>Input channel select for channel 2 for display or record.<br>When VIN_SEL_XY is “0”, this register is for display, when VIN_SEL_XY is “1”, this register is for record. |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 3:0 | RW | 0 | VIN_SEL1<br><br>Input channel select for channel 1. |

**Register 0x3D3 to 0x3D9 are for channels 3 to 16.**

### LIVE/RECORD SET SELECT REGISTER – 0X3DA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | VIN_SEL_XY<br><br>This bit is used for select VIN_SEL X or Y. VIN_SEL share address for display and record. When VIN_SEL_XY is 0, VIN_SEL can be read/write for display. When VIN_SEL_XY is 1, VIN_SEL can be read/write for record. |
| 6:1 | R | 0 | Reserved |
| 0 | RW | 1 | V_OFST_X<br><br>Vertical scale offset select |

# Play Back Interface

## Introduction

The TW2880 supports 4 play back port. It can be 4x8-bit input or 4x16-bit input (with external FPGA). Playback port can be BT.656 format or BT.1120 format or a similar format. If channel ID is deployed, each playback port can support up to 16 channels (auto mode). The maximum horizontal resolution is 1920 for each port. The data rate can be up to 148.5MHz. The playback port can support both interlace or progressive video. Four high quality down scalers reside in each port.

## Features

- Four channel HD input support ITU-R BT.1120 up to 148.5 MHz (with external FPGA)
- Four channel SD input support ITU-R BT.656 up to 108 MHz
- Channel ID detection in forms of analog and digital
- Cost efficient programmable down scalar with good quality
- Using same clock for input formatter and down scalar makes the system flexible for different applications

## Description

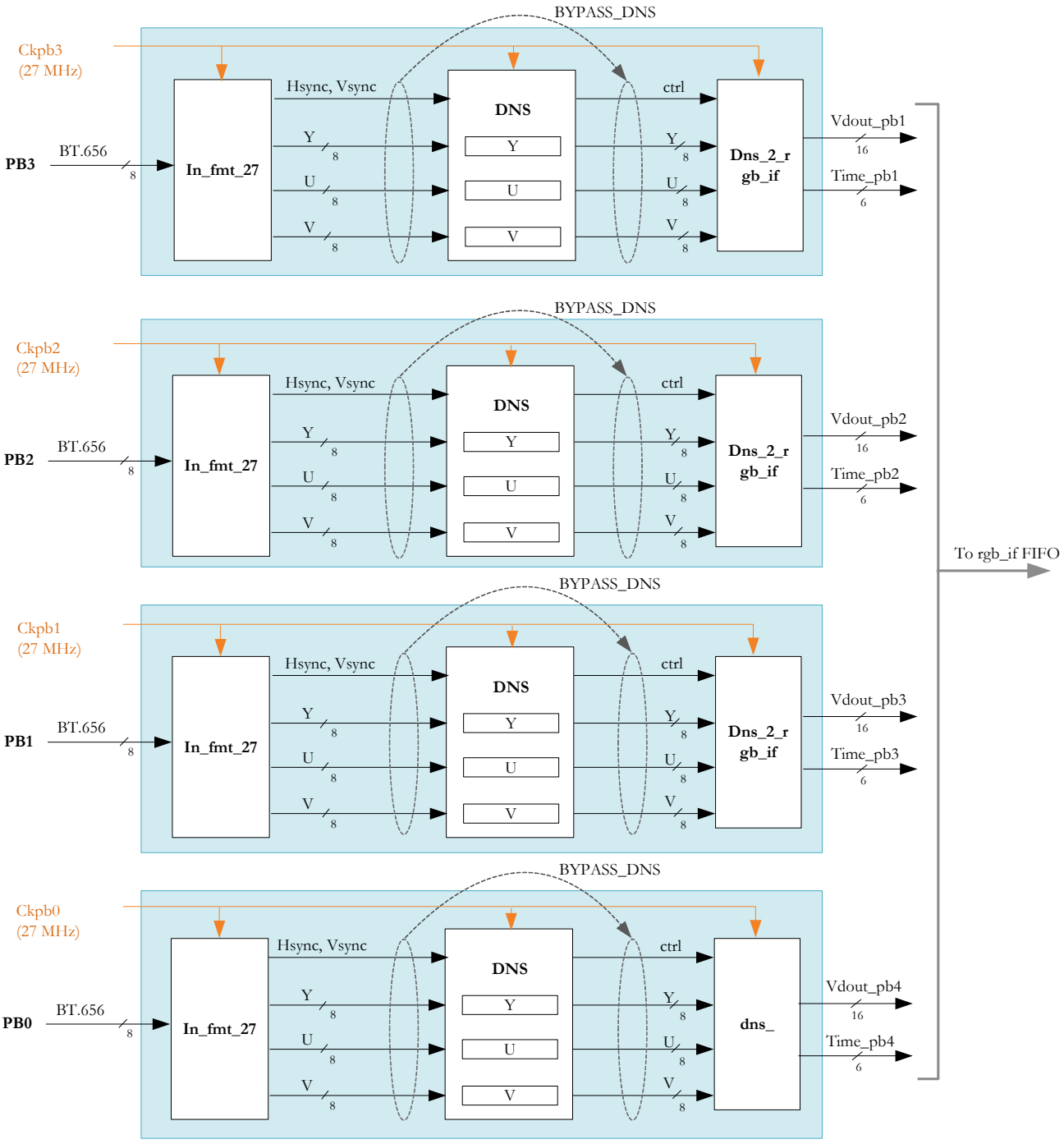
### PLAY BACK ARCHITECTURE

#### Play Back Architecture in BT.656 SD Mode

In BT.656 SD input mode, four channels are supported with the data rate at 27 MHz. Incoming 8-bit data format is Cb1 Y1 Cr1 Y2 Cb3 Y3 Cr3 Y4 ... Four input formatters in_fmt_27 and four down scalars DNS are occupied. In each channel, both in_fmt_27 and DNS use same clock ckpb and each channel has it own individual clock. In_fmt_27 outputs Y, U and V data in 4:4:4 to down scalar DNS. DNS outputs Y, U and V data in 4:4:4 as well. Data will be converted to 4:2:2 chroma format in dns_2_rgb_if, which sends data to rgb_interface.

Down scalar DNS is acquired from the design TW2835 aimed to save circuit gate count while maintaining down scalar quality. Down scalar's output Y, Cb and Cr are converted to 16 bit data word to dns_2_rgb_if in the form of (Cb1 Y1) (Cr1Y2) (Cr3Y3) (Cb3Y4)... along with control signals of hsync, vsync and field. Decoded H blanking, V blanking and Field and data bus input to channel ID decoder CHID_dec.v. Channel ID decoder will decode Auto Channel ID, Detection Channel ID and User Channel ID. The results are readable by Host Interface.

BT.656 SD Input Mode

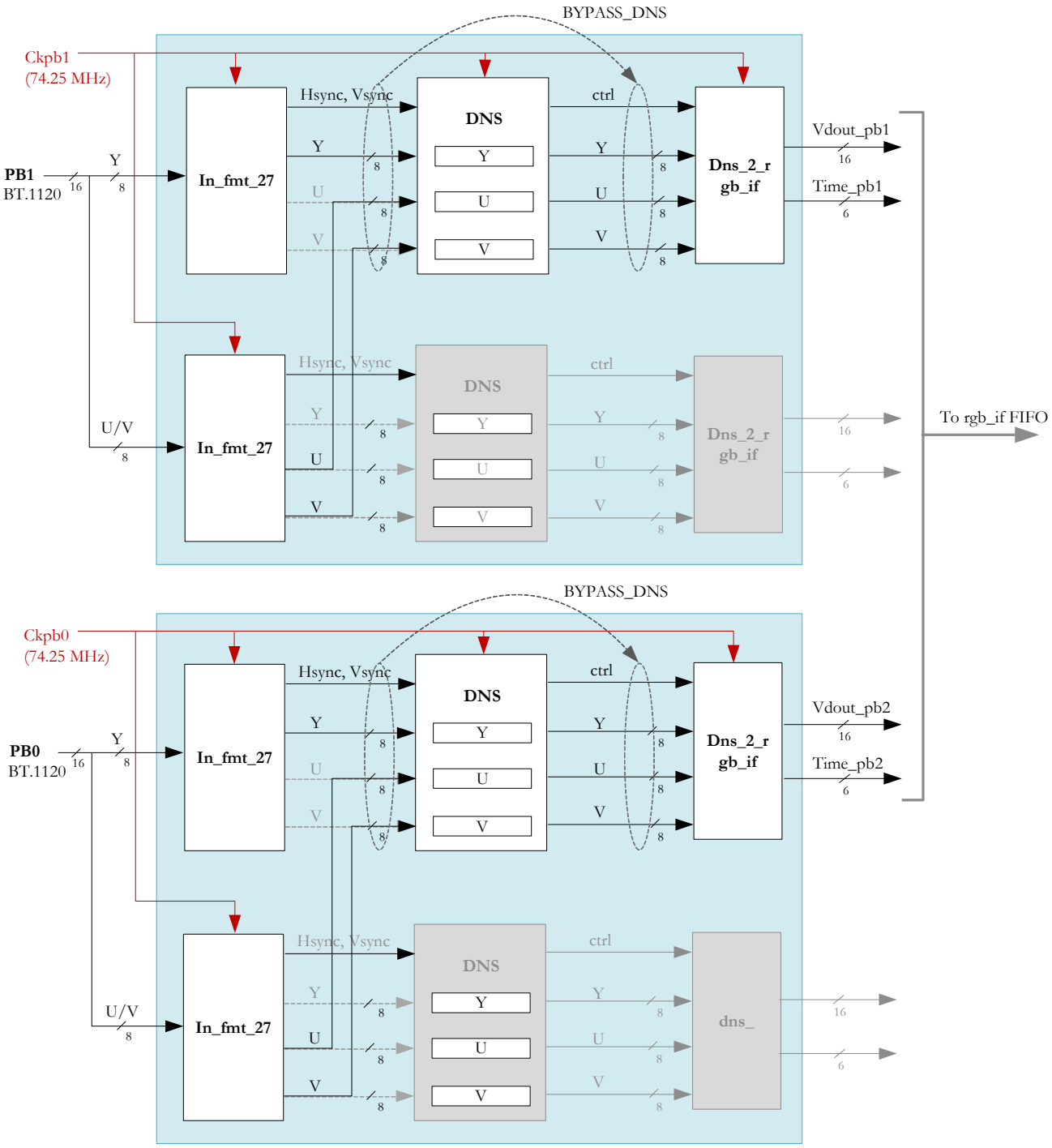



**Play Back Architecture in 16-Bit Mode**

**In BT.1120 HD input mode, two channels are supported with data rate at 74.25 MHz. Each channel carries 16-bit data in the sequence of (Cb1, Y1) (Cr1, Y2) (Cb3, Y3) (Cr3, Y4). MSB 8 bits are Cb/Cr, LSB 8 bits are Y. One channel BT.1120 HD input data will use two input formatters in_fmt_27. Y bus goes to one in_fmt_27, Cb/Cr bus goes to another in_fmt_27. Two in_fmt_27's outputs Y, U and V data as input data of one down scalar DNS. With two input**

**channels, only two DNSs will be used. In the following block diagram, the shaded block is denoted as not using module in BT.1120 mode. The interface format for dns_2_rgb_if is same as BT.656 mode.**

BT.1120 HD Input Mode



### Interface with rgb_Interface

The outputs Y, Cb and Cr of down scalars either in BT.656 mode or BT.1120 mode convert to 16-bit data bus vdout_pb1 ~ vdout_pb4 in the sequence of (Cb, Y) (Cr, Y). Meanwhile, 6-bit control signal time_pb1 ~ tim_pb4 (carry hsync, vsync and field) will be provided in a format of {fld_d, new_ln, end_ln, new_fld, end_fld, valid}. This video data is write to rgb_interface channel buffer and further transferred to SDRAM for display.

The system has flexibility for play back clock rate since it uses same clock pbck1 ~ pbck4 for each channel from incoming video data through in_fmt_27, DNS data to channel buffer.

## FUNCTIONS

### INPUT FORMATTER

Input formatter in_fmt_27 supports SD BT.656 in NTSC, PAL or non-standard video format. In HD BT.1120, it supports 1920x1080i. It generates Y, Cb and Cr data bus to down scalar. HSYNC, VSYNC and VALID signals generate accordingly. In BT.1120 mode, in_fmt_27 can be set either take Y data or take Cb/Cr data and output Y or Cb/Cr will be generated respectively. Two in_fmt_27's outputs go to one down scalar.

HDELSY and VDELAY can be configured to get portion of one frame for display.

### Down Scalar

There are four down scalar for four play back ports. Down scale ratio register is different from live channels. Source width and height must be set correctly. And target width and height will be set to desired size.

### Channel ID Decoder

TW2880 Channel ID Decoder is using same protocol of TW2880 Channel ID encoder. For the detailed definition and description of Channel ID, please refer to Channel ID Encoder Unit.

There are twelve words (lines) of channel ID: one word of Auto Channel ID, two words of Detection Channel ID, three words of User Channel ID and 6 words (192 bits) Motion Detection Channel ID. They can be represented in terms of digital and analog formats. One of them or both can be existed in BT.656 video bit stream. Usually, they appear in horizontal active area of vertical blanking period. Channel ID encoder may also possible program them to vertical active area. But in this design, it only allows it in the first two top lines of active video area to ensure proper display video image. Any Channel ID information program outside of top two lines of active video area in both fields will be ignored in both Channel ID encoder and decoder. Analog Channel ID follows after digital channel ID if digital channel ID exists.

There are two modes to decoding channel ID: Auto detection and Non-auto detection. Digital channel ID can be detected fully automatic. However, decoding analog channel ID even in Auto mode requires extra host configuration.

Decoded twelve words of channel ID data and corresponding valid signals are readable by host.

### Play Back Data Path Selection

Each channel has 2-bit selection control PB_PATH_SEL1 ~ PB_PATH_SEL4 to select one of four input data path. The selection can be made in the input of in_fmt_27. Please note each channel has its own clock and data should be corresponding to its own clock. This feature may useful in test and debugging.

### Bypass Down Scalar

Each channel has its configuration bit BYPASS_PB to bypass down scalar. Input formatter's output directly goes to rgb_interface for display. If down scalar is set or in its default values HSCL = 16'h100 and VSCL = 16'h100, the system will perform bypass to maintain best display quality.

### Display Part of Video Frame

Incoming video frame can be display partially by configuring HDELAY and VDELAY. Down scalar still can be applied.

## Registers Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x357 | R/W | 0x00 | [7]: TOGGLE_FLD4<br>[6]: TOGGLE_FLD3<br>[5]: TOGGLE_FLD2<br>[4]: TOGGLE_FLD1<br>[3]: TST_FLDLY_Y<br>[2]: TST_FLDLY_X<br>[1]: TST_VSCL_Y<br>[0]: TST_VSCL_X |
| 0x358 | R/W | 0xf0 | VTAR_PB1[7:0] |
| 0x359 | R/W | 0x00 | [6:4]: VTAR_PB2[10:8]<br>[2:0]: VTAR_PB1[10:8] |
| 0x35A | R/W | 0xf0 | VTAR_PB2[7:0] |
| 0x35B | R/W | 0xf0 | VTAR_PB3[7:0] |
| 0x35C | R/W | 0x00 | [6:4]: VTAR_PB4[10:8]<br>[2:0]: VTAR_PB3[10:8] |
| 0x35D | R/W | 0xf0 | VTAR_PB4[7:0] |
| 0x35E | R/W | 0xf0 | VSOR_PB1[7:0] |
| 0x35F | R/W | 0x00 | [6:4]: VSOR_PB2[10:8]<br>[2:0]: VSOR_PB1[10:8] |
| 0x360 | R/W | 0xf0 | VSOR_PB2[7:0] |
| 0x361 | R/W | 0xf0 | VSOR_PB3[7:0] |
| 0x362 | R/W | 0x00 | [6:4]: VSOR_PB4[10:8]<br>[2:0]: VSOR_PB3[10:8] |
| 0x363 | R/W | 0xf0 | VSOR_PB4[7:0] |
| 0x364 | R/W | 0xd0 | HTAR_PB1[7:0] |
| 0x365 | R/W | 0x22 | [6:4]: HTAR_PB2[10:8]<br>[2:0]: HTAR_PB1[10:8] |
| 0x366 | R/W | 0xd0 | HTAR_PB2[7:0] |
| 0x367 | R/W | 0xd0 | HTAR_PB3[7:0] |
| 0x368 | R/W | 0x22 | [6:4]: HTAR_PB4[10:8]<br>[2:0]: HTAR_PB3[10:8] |
| 0x369 | R/W | 0xd0 | HTAR_PB4[7:0] |
| 0x36A | R/W | 0xd0 | HSOR_PB1[7:0] |
| 0x36B | R/W | 0x22 | [6:4]: HSOR_PB2[10:8]<br>[2:0]: HSOR_PB1[10:8] |
| 0x36C | R/W | 0xd0 | HSOR_PB2[7:0] |
| 0x36D | R/W | 0xd0 | HSOR_PB3[7:0] |
| 0x36E | R/W | 0x22 | [6:4]: HSOR_PB4[10:8]<br>[2:0]: HSOR_PB3[10:8] |
| 0x36F | R/W | 0xd0 | HSOR_PB4[7:0] |
| 0x370 | R/W | 0x00 | [7]: QUAD_AUTO_SCL_PB4<br>[6]: QUAD_AUTO_SCL_PB3<br>[5]: QUAD_AUTO_SCL_PB2<br>[4]: QUAD_AUTO_SCL_PB1<br>[3]: BYPASS_PB4<br>[2]: BYPASS_PB3<br>[1]: BYPASS_PB2<br>[0]: BYPASS_PB1 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x371 | R/W | 0x00 | [7]: H_601_INV<br>[6]: SYNC_SEL<br>[5]: YC_SWITCH<br>[4:2]: reserved<br>[1]: IN16_MODE_PB34<br>[0]: IN16_MODE_PB12 |
| 0x372 | RO | 0 | CHID_RD_BUS_PB1[39:32] |
| 0x373 | RO | 0 | CHID_RD_BUS_PB1[31:24] |
| 0x374 | RO | 0 | CHID_RD_BUS_PB1[23:16] |
| 0x3B5 | R/W | 0x00 | [7:0]: HDELAY_PB1 |
| 0x3B6 | R/W | 0x00 | [7:0]: VDELAY_PB1 |
| 0x3B7 | R/W | 0x00 | [7:0]: HDELAY_PB2 |
| 0x3B8 | R/W | 0x00 | [7:0]: VDELAY_PB2 |
| 0x3B9 | R/W | 0x00 | [7:0]: HDELAY_PB3 |
| 0x3BA | R/W | 0x00 | [7:0]: VDELAY_PB3 |
| 0x3BB | R/W | 0x00 | [7:0]: HDELAY_PB4 |
| 0x3BC | R/W | 0x00 | [7:0]: VDELAY_PB4 |
| 0x3C7 | RO | 0 | CHID_RD_BUS_PB1[15:8] |
| 0x3C8 | RO | 0 | CHID_RD_BUS_PB1[7:0] |
| 0x3C9 | RO | 0 | CHID_RD_BUS_PB2[39:32] |
| 0x3CA | RO | 0 | CHID_RD_BUS_PB2[31:24] |
| 0x3CC | RO | 0 | CHID_RD_BUS_PB2[23:16] |
| 0x3CD | RO | 0 | CHID_RD_BUS_PB2[15:8] |
| 0x3CE | RO | 0 | CHID_RD_BUS_PB2[7:0] |
| 0x3CF | RO | 0 | CHID_RD_BUS_PB3[39:32] |
| 0x3D0 | RO | 0 | CHID_RD_BUS_PB3[31:24] |
| 0x3D1 | RO | 0 | CHID_RD_BUS_PB3[23:16] |
| 0x3DB | R/W | 0 | [7:6]: PB_PATH_SEL4<br>[5:4]: PB_PATH_SEL3<br>[3:2]: PB_PATH_SEL2<br>[1:0]: PB_PATH_SEL1 |
| 0x3DC | R/W | 0x12 | PB_TEST_PATTEN<br>[4]: DIR2<br>[3]: DIR1<br>[2:1]: FRAME_RATE<br>[0]: IN_PB_TEST |
| 0x3DD | RW | 0x40 | PIXEL_SHIFT |
| 0x3DE | R/W | 0x0F | [7:4]: CHID_RD_SEL_PB1<br>[3]: VBI_RIC_ON_PB1<br>[2]: VBI_AUTO_DET_PB1<br>[1]: VBI_D_EN_PB1<br>[0]: VBI_A_EN_PB1 |
| 0x3DF | R/W | 0x00 | [4:0]: VBI_VOS_PB1 |
| 0x3E0 | R/W | 0x80 | [7:5]: VBI_PIXEL_WIDTH_PB1<br>[4:0]: VBI_FOS_PB1 |
| 0x3E1 | R/W | 0 | VBI_HOS_PB1 |
| 0x3E2 | R/W | 0x1F | VBI_MID_VAL_PB1 |
| 0x3E3 | R/W | 0x0F | [7:4]: CHID_RD_SEL_PB2<br>[3]: VBI_RIC_ON_PB2<br>[2]: VBI_AUTO_DET_PB2<br>[1]: VBI_D_EN_PB2<br>[0]: VBI_A_EN_PB2 |
| 0x3E4 | R/W | 0x00 | [4:0]: VBI_VOS_PB2 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x3E5 | R/W | 0x80 | [7:5]: VBI_PIXEL_WIDTH_PB2<br>[4:0]: VBI_FOS_PB2 |
| 0x3E6 | R/W | 0x00 | VBI_HOS_PB2 |
| 0x3E7 | R/W | 0x1F | VBI_MID_VAL_PB2 |
| 0x3E8 | R/W | 0x0F | [7:4]CHID_RD_SEL_PB3<br>[3]: VBI_RIC_ON_PB3<br>[2]: VBI_AUTO_DET_PB3<br>[1]: VBI_D_EN_PB3<br>[0]: VBI_A_EN_PB3 |
| 0x3E9 | R/W | 0x00 | [4:0]: VBI_VOS_PB3 |
| 0x3EA | R/W | 0x80 | [7:5]: VBI_PIXEL_WIDTH_PB3<br>[4:0]: VBI_FOS_PB3 |
| 0x3EB | R/W | 0x00 | VBI_HOS_PB3 |
| 0x3EC | R/W | 0x1F | VBI_MID_VAL_PB3 |
| 0x3ED | R/W | 0x0F | [7:4]: CHID_RD_SEL_PB4<br>[3]: VBI_RIC_ON_PB4<br>[2]: VBI_AUTO_DET_PB4<br>[1]: VBI_D_EN_PB4<br>[0]: VBI_A_EN_PB4 |
| 0x3EE | R/W | 0x00 | [4:0] VBI_VOS_PB4 |
| 0x3EF | R/W | 0x80 | [7:5]: VBI_PIXEL_WIDTH_PB4<br>[4:0]: VBI_FOS_PB4 |
| 0x3F0 | R/W | 0x00 | VBI_HOS_PB4 |
| 0x3F1 | R/W | 0x00 | VBI_MID_VAL_PB2 |
| 0x3F2 | R/W | 0x00 | [7]: FRAME_IL_PB4_EN<br>[6]: FRAME_IL_PB3_EN<br>[5]: FRAME_IL_PB2_EN<br>[4]: FRAME_IL_PB1_EN<br>[3]: FRAME_IL_PB4<br>[2]: FRAME_IL_PB3<br>[1]: FRAME_IL_PB2<br>[0]: FRAME_IL_PB1 |
| 0x3F3 | R/W | 0x00 | [7]: FLD_MODE_PB4_EN<br>[6]: FLD_MODE_PB3_EN<br>[5]: FLD_MODE_PB2_EN<br>[4]: FLD_MODE_PB1_EN<br>[3]: FLD_MODE_PB4<br>[2]: FLD_MODE_PB3<br>[1]: FLD_MODE_PB2<br>[0]: FLD_MODE_PB1 |
| 0x3F4 | R/W | 0xFF | [7]: VBI_E_EN_PB4<br>[6]: VBI_E_EN_PB3<br>[5]: VBI_E_EN_PB2<br>[4]: VBI_E_EN_PB1<br>[3]: VBI_O_EN_PB4<br>[2]: VBI_O_EN_PB3<br>[1]: VBI_O_EN_PB2<br>[0]: VBI_O_EN_PB1 |
| 0x3F8 | R/W | 0x84 | Non-standard Video Upper Limit - NTSC Register, 0x3ff[7] = 0 these registers will read out PB4 channel ID |
| 0x3F9 | R/W | 0x6C | Non-standard Video Upper Limit - NTSC Register |
| 0x3FA | R/W | 0x9E | Non-standard Video Upper Limit – PAL Register |
| 0x3FB | R/W | 0x82 | Non-standard Video Upper Limit - PAL Register |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x3FC | R/W | 0 | [7]: VBI_DLY4<br>[6]: VBI_DLY3<br>[5]: VBI_DLY2<br>[4]: VBI_DLY1<br>[3]: FLD_SEL_EN_PB4<br>[2]: FLD_SEL_EN_PB3<br>[1]: FLD_SEL_EN_PB2<br>[0]: FLD_SEL_EN_PB1 |
| 0x3FD | R/W | EE | [7:4]: VBI_SIZE_PB2<br>[3:0]: VBI_SIZE_PB1 |
| 0x3FE | R/W | EE | [7:4]: VBI_SIZE_PB4<br>[3:0]: VBI_SIZE_PB3 |
| 0x3FF | R/W | 0 | [7]: selm<br>[6]: PBX4_SEL<br>[5]: PBX2_SEL<br>[4]: MG_DIG_CODE<br>[3]: VS_DIS_PB4<br>[2]: VS_DIS_PB3<br>[1]: VS_DIS_PB2<br>[0]: VS_DIS_PB1 |

# Registers Description

### TOGGLE FIELD SIGNAL ENABLE REGISTER – 0X357

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0 | **TOGGLE_FLD4 - TOGGLE_FLD1**<br><br>These bits are used for progressive play back capture mode. In progressive mode, field signal is always 0. If this bit is set to high, hardware will generate toggled field signal for channel ID decoder. It is toggled after 3 line of the beginning of vertical blank.<br><br>1: toggle field signal for channel ID decoder<br>0: use original field signal |
| 3 | RW | 0 | **TST_FLDLY_Y** |
| 2 | RW | 0 | **TST_FLDLY_X** |
| 1 | RW | 0 | **TST_VSCL_Y** |
| 0 | RW | 0 | **TST_VSCL_X** |

### PB1 VERTICAL TARGET SIZE LSB REGISTER – 0X358

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xf0 | **VTAR_PB1[7:0]**<br><br>PB1 down scalar vertical target size. **Unit is one line**. In interlaced mode, this register should set to half of the vertical lines of one frame. |

### PB1 AND PB2 VERTICAL TARGET SIZE MSB REGISTER – 0X359

| Bit | R/W | Default | Description |
|---|---|---|---|
| 6:4 | RW | 0x0 | **VTAR_PB2[10:8]**<br><br>PB2 down scalar vertical target size. **Unit is one line**. In interlaced mode, this register is set to half of the vertical lines of one frame. |
| 2:0 | RW | 0x0 | **VTAR_PB1[10:8]**<br><br>PB1 down scalar vertical target size. **Unit is one line**. In interlaced mode, this register is set to half of the vertical lines of one frame. |

### PB2 VERTICAL TARGET SIZE LSB REGISTER – 0X35A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xf0 | **VTAR_PB2[7:0]**<br><br>PB2 down scalar vertical target size. **Unit is one line**. In interlaced mode, this register is set to half of the vertical lines of one frame. |

**Registers 0x35B to 0x35D are for PB channels 3 and 4.**

### PB1 VERTICAL SOURCE SIZE LSB REGISTER – 0X35E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xf0 | VSOR_PB1[7:0]<br><br>PB1 down scalar vertical source size. **Unit is one line**. In interlaced mode, this register is set to half of the vertical lines of one frame. |

### PB1 AND PB2 VERTICAL SOURCE SIZE MSB REGISTER – 0X35F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 6:4 | RW | 0x00 | VSOR_PB2[10:8]<br><br>PB2 down scalar vertical source size. **Unit is one line**. In interlaced mode, this register is set to half of the vertical lines of one frame. |
| 2:0 | RW | 0x0 | VSOR_PB1[10:8]<br><br>PB1 down scalar vertical source size. **Unit is one line**. In interlaced mode, this register is set to half of the vertical lines of one frame. |

### PB2 VERTICAL SOURCE SIZE LSB REGISTER – 0X360

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xf0 | VSOR_PB2[7:0]<br><br>PB2 down scalar vertical source size. **Unit is one line**. In interlaced mode, this register is set to half of the vertical lines of one frame. |

**Registers 0x361 to 0x363 are for PB channels 3 and 4.**

### PB1 HORIZONTAL SCALER TARGET LSB REGISTER – 0X364

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xd0 | HTAR_PB1[7:0]<br><br>PB1 down scalar horizontal target size. **Unit is one pixel**. |

### PB1 AND PB2 HORIZONTAL TARGET SIZE MSB REGISTER – 0X365

| Bit | R/W | Default | Description |
|---|---|---|---|
| 6:4 | RW | 0x2 | HTAR_PB2[10:8]<br><br>PB2 down scalar horizontal target size. **Unit is one pixel**. |
| 2:0 | RW | 0x2 | HTAR_PB1[10:8]<br><br>PB1 down scalar horizontal target size. **Unit is one pixel**. |

### PB2 HORIZONTAL SCALER TARGET LSB REGISTER – 0X366

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xd0 | HTAR_PB2[7:0]<br><br>PB2 down scalar horizontal target size. **Unit is one pixel.** |

**Registers 0x367 to 0x369 are for channels 3 and 4.**

### PB1 HORIZONTAL SCALER SOURCE LSB REGISTER – 0X36A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xd0 | HSOR_PB1[7:0]<br><br>PB1 down scalar horizontal source size. **Unit is one pixel.** |

### PB1 AND PB2 HORIZONTAL SOURCE SIZE MSB REGISTER – 0X36B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 6:4 | RW | 0x2 | HSOR_PB2[10:8]<br><br>PB2 down scalar horizontal source size. **Unit is one pixel.** |
| 2:0 | RW | 0x2 | HSOR_PB1[10:8]<br><br>PB1 down scalar horizontal source size. **Unit is one pixel.** |

### PB2 HORIZONTAL SCALER SOURCE LSB REGISTER – 0X36C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xd0 | HSOR_PB2[7:0]<br><br>PB2 down scalar horizontal source size. **Unit is one pixel.** |

**Registers 0x36D to 0x36F are for channels 3 and 4.**

### PB AUTO DOWNSCALE AND BYPASS REGISTER – 0X370

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0 | **QUAD_AUTO_SCL_PB4 - QUAD_AUTO_SCL_PB1**<br><br>Down scale ratio can be changed in multi-mode for each play back port<br><br>1: enable, D1 and Quad is 2:1 in down scale ratio<br>0: disable, down scale ratio keep same |
| 3:0 | RW | 0x1 | **BYPASS_PB4 - BYPASS_PB1**<br><br>Bypass down scalar. Video data from in_fmt_27 directly go to rgb_interface for display.<br><br>1: enable<br>0: disable |

### PB MISC CONTROL REGISTER – 0X371

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0x0 | **Reserved** |
| 6 | RW | 0x0 | **SYNC_SEL:** When this bit is set to high, use external H/V/F sync.<br><br>1: use external sync<br>0: use embedded sync |
| 5 | RW | 0x1 | **YC_SWITCH:** YC switch in 16bit mode<br><br>1: Y in port 2, C in port 1<br>0: Y in port 1, C in port 2 |
| 4 | RW | 0 | **SYNC656_SEL**<br><br>1 = Play back port sync from state machine<br>0 = Play back port sync from byte compare |
| 3 | RW | 0 | **IN_16BIT_RGB_MODE34**<br><br>1 = 16 bit Play back port input format = 565RGB<br>0 = default |
| 2 | RW | 0 | **IN_16BIT_RGB_MODE12**<br><br>1 = 16 bit Play back port input format = 565RGB<br>0 = default |
| 1 | RW | 0 | **IN16_MODE_PB34:** Play back port 3 and 4 data is 16bit or 8bit<br><br>1: 16bit<br>0: 8bit |
| 0 | RW | 0 | **IN16_MODE_PB12:** Play back port 1 and 2 data is 16bit or 8bit<br><br>1: 16bit<br>0: 8bit |

### HORIZONTAL DELAY FOR PLAYBACK PORT 1 REGISTER – 0X3B5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x0 | HDELAY_PB1<br><br>Horizontal Delay (Cut off left side) of horizontal active picture in unit of two pixels. |

### VERTICAL DELAY FOR PLAYBACK PORT 1 REGISTER – 0X3B6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x0 | VDELAY_PB1<br><br>Horizontal Delay (Cut off top portion) of vertical active picture in unit of one line. |

**Registers 0x3B7 to 0x3BC are for PB ports 2 to 4**

### PB PATH SELECTION REGISTER – 0X3DB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 3 | **PB_PATH_PB4:** PB4 down scalar input select. If PBX4_SEL is low, the input is the same as PB3. If PBX4_SEL is one, input is coming from live input7, 8 or record 3, 4 |
| 5:4 | RW | 2 | **PB_PATH_PB3:** PB3 down scalar input select. Remember in 16 bit mode user needs to select input 4 as one set. |
| 3:2 | RW | 1 | **PB_PATH_PB2:** PB2 down scalar input select. If PBX2_SEL is low, the input is the same as PB1. If PBX2_SEL is one, input is coming from live input5, 6 or record 1, 2 |
| 1:0 | RW | 0 | **PB_PATH_PB1:** PB1 down scalar input select. Remember in 16 bit mode user needs to select input 2 as one set.<br><br>3: select PB4 input pins<br>2: select PB3 input pins<br>1: select PB2 input pins<br>0: select PB1 input pins |

### PB IN_PB_TEST REGISTER – 0X3DC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | RW | 0x0 | Reserved |
| 4 | RW | 0x0 | **DIR2:** PB test pattern generator BT1120_gen2 color bar shift direction.<br><br>1: shift to left<br>0: shift to right |
| 3 | RW | 0x1 | **DIR1:** PB test pattern generator BT1120_gen1 color bar shift direction.<br><br>1: shift to left<br>0: shift to right |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:1 | RW | 0x0 | **Frame Rate**<br><br>PB test pattern generator color bar shift frame rate for both BT1120_gen1 and BT1120_gen2<br>0: change every 16 frame<br>1: change every 32 frame<br>Others: change every 64 frame |
| 0 | RW | 0x0 | **IN_PB_TEST:** PB test pattern generator color bar for PB in BT.1120<br><br>1: enable PB HD color bar generator<br>0: disable. PB use data from port. |

### PB IN_PB_TEST PATTERN GENERATOR COLOR BAR SHIFT VOLUM – 0X3DD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x0 | **PIXEL_SHIFT**<br><br>HD Color bar shift volume in unit of pixel |

### PB CHANNEL ID DEC PB1 CTRL – 0X3DE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0x0 | **CHID_RD_SEL_PB1**<br><br>Host selects one of 11 CHID words to read.<br>b: User CHID3<br>a: User CHID2<br>9: User CHID1<br>8: Motion detection CHID6<br>7: Motion detection CHID5<br>6: Motion detection CHID4<br>5: Motion detection CHID3<br>4: Motion detection CHID2<br>3: Motion detection CHID1<br>2: Detection CHID2<br>1: Detection CHID1<br>0: Auto CHID |
| 3 | RW | 0x1 | **VBI_RIC_ON_PB1**<br><br>Enable run in clock detection. It should be same as CHID encoder configured.<br>1: enable<br>0: disable |
| 2 | RW | 0x1 | **VBI_AUTO_DET_PB1**<br><br>Automatic detect channel ID for digital channel ID vertical offset and analog channel ID run in clock frequency.<br>1: enable<br>0: disable |
| 1 | RW | 0x1 | **VBI_D_EN_PB1:** Enable digital CHID. It should be same as CHID encoder<br><br>1: enable<br>0: disable |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | RW | 0x1 | VBI_A_EN_PB1: Enable analog CHID. It should be same as CHID encoder, if both enable then use analog CHID.<br><br>1: enable<br>0: disable |

### PB CHANNEL ID DEC PB1 VOS – 0X3DF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | RW | 0x0 | VBI_VOS_PB1<br><br>Line number of vertical offset for channel id in First field in unit of one. |

### PB CHANNEL ID DEC PB1 FOS – 0X3E0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | RW | 0x4 | VBI_PIXEL_WIDTH_PB1<br><br>Define half width of analog CHID sample rate. |
| 4:0 | RW | 0x1 | VBI_FOS_PB1<br><br>Line number of vertical offset for channel id in second field in unit of one |

### PB CHANNEL ID DEC PB1 HOS – 0X3E1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x1 | VBI_HOS_PB1: Pixel horizontal offset for channel. |

### PB CHANNEL ID DEC PB1 MID_VAL – 0X3E2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x1 | VBI_MID_VAL_PB1<br><br>Middle value of VIS_HIGH_VAL and VIS_LOW_VAL in CHID encoder. |

### PB FRAME/FIELD CONTROL REGISTER – 0X3F2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0 | FRAME_IL_PB4_EN – FRAME_IL_PB1_EN:<br><br>Force the receiving mode of PB4 – PB1 to follow the value of bit [3:0].<br><br>1: enable<br>0: disable |
| 3:0 | RW | 0 | FRAME_IL_PB4 - FRAME_IL_PB1:<br><br>Host define the stream is frame interleaved or field interleaved.<br><br>1: field interleaved mode (auto), interlaced mode (normal)<br>0: frame interleaved mode (auto), progressive mode (normal) |

**PB FIELD MODE REGISTER – 0X3F3**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0 | **FLD_MODE_PB4_EN - FLD_MODE_PB1_EN:**<br><br>Manual set field mode<br><br>1: enable, get field mode from **FLD_MODE_PB4-1**<br>0: disable, get field mode from channel ID |
| 3:0 | RW | 0 | **FLD_MODE_PB4 - FLD_MODE_PB1:**<br><br>Field mode in manual mode for port 4-1<br><br>1: even field when fld is 0<br>0: even field when fld is 1 |

**PB DIGITAL CHANNEL ID ENABLE REGISTER – 0X3F4**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 1 | **VBI_E_EN_PB4 - VBI_E_EN_PB1:** Digital CHID even field enable<br><br>1: enable<br>0: disable |
| 3:0 | RW | 1 | **VBI_O_EN_PB4 - VBI_O_EN_PB1:** Digital CHID odd field enable<br><br>1: enable<br>0: disable |

**NON-STANDARD VIDEO UPPER LIMIT - NTSC REGISTER – 0X3F8**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x84 | **NT_UP_LIM:** NTSC line counter upper limit, unit is double line |

**NON-STANDARD VIDEO LOWER LIMIT - NTSC REGISTER – 0X3F9**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x6c | **NT_DOWN_LIM:** NTSC line counter lower limit, unit is double line |

**NON-STANDARD VIDEO UPPER LIMIT - PAL REGISTER – 0X3FA**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x9e | **NT_UP_LIM:** PAL line counter upper limit, unit is double line |

**NON-STANDARD VIDEO LOWER LIMIT - PAL REGISTER – 0X3FB**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x82 | **NT_DOWN_LIM:** PAL line counter lower limit, unit is double line |

**When 3FF [7] = 1, the read data of 0x3F8 - 0x3FB is CHID of PB4.**

### PB CHID DECODER CONTROL REGISTER – 0X3FC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0 | **VBI_DLY4 - VBI_DLY1:**<br><br>1 = New field signal delay one line. It is used for digital channel ID in first active line. |
| 3:0 | RW | 0 | **FLD_SEL_EN_PB4 - FLD_SEL_EN_PB1:**<br><br>Field signal source selection<br><br>1: from embedded field signature.<br>0: from channel ID |

**When 3C6 [1] = 1, the read data of 0x3FC and 0x3FD is non-standard video signal of live channels.**

### ANALOG CHANNEL ID LINE NUMBER REGISTER 1 – 0X3FD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0xE | **VBI_SIZE_PB2**<br><br>Analog channel ID line number for PB port 2 |
| 3:0 | RW | 0xE | **VBI_SIZE_PB1**<br><br>Analog channel ID line number for PB port 1 |

### ANALOG CHANNEL ID LINE NUMBER REGISTER 2 – 0X3FE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0xE | **VBI_SIZE_PB4:** Analog channel ID line number for PB port 4 |
| 3:0 | RW | 0xE | **VBI_SIZE_PB3:** Analog channel ID line number for PB port 3 |

### MISC. CONTROL REGISTER – 0X3FF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **selm:** This bit selects the output of 0x3f8 – 0x3fb<br><br>1: Output of PB port channel ID<br>0: Output of 0x3f8 – ox3fb. |
| 6 | RW | 0 | **PBX4_SEL:** This bit selects the PB4 input in 16 bit mode<br><br>1: Input coming from Live7 and Live8 or REC3 and REC4 determined by RECX4_SEL<br>0: Input used is same as PB3 |
| 5 | RW | 0 | **PBX2_SEL:** This bit selects the PB2 input in 16 bit mode<br><br>1: Input use Live5 and Live6 or REC1 and REC2 determined by RECX2_SEL<br>0: Input used is same as PB1 |
| 4 | RW | 0 | **DIG_CODE:** This bit selects digital channel ID start code<br><br>1: 0x80_9F_80_90<br>0: 0x00_9F_00_90 |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 3:0 | RW | 0 | VS_DIS4 – VS_DIS1:<br>New field signal disabled in even field for progressive mode<br><br>1: disable even field new field signal<br>0: don't disable even field new field signal |

**PB CHANNEL ID DEC READ BUS PB1 BYTE 1– 0X372**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x7 | CHID_RD_BUS_PB1[39:32]<br><br>Read CHID_rd_bus_pb1[39:32]. Depend on CHID_RD_SEL_PB1(REG OX3DE), it read one of six CHID word:<br>0: auto CHID<br>[7] digital_chid_valid_pb1, 1: D CHID valid, 0: A CHID valid<br>[6] auto_valid_pb1, 1: auto_CHID valid, 0 not valid<br>[5:0] auto_chid_pb1[37:32]<br><br>1: detection CHID1<br>[7] digital_chid_valid_pb1, 1: D CHID valid, 0: A CHID valid<br>[4] det1_valid_pb1, 1: det1_CHID valid, 0 not valid<br>[3:0] det1_chid_pb1[35:32]<br><br>2: detection CHID2<br>[7] digital_chid_valid_pb1, 1: D CHID valid, 0: A CHID valid<br>[4] det2_valid_pb1, 1: det2_CHID valid, 0 not valid<br>[3:0] det2_chid_pb1[35:32]<br><br>3: Motion detection CHID1<br>[7] digital_chid_valid_pb1, 1: D CHID valid, 0: A CHID valid<br>[4] det3_valid_pb1, 1: det2_CHID valid, 0 not valid<br>[3:0] de3_chid_pb1[35:32]<br><br>4: Motion detection CHID1<br>[7] digital_chid_valid_pb1, 1: D CHID valid, 0: A CHID valid<br>[4] det4_valid_pb1, 1: det4_CHID valid, 0 not valid<br>[3:0] det4_chid_pb1[35:32]<br><br>5: Motion detection CHID1<br>[7] digital_chid_valid_pb1, 1: D. CHID valid, 0: A. CHID valid<br>[4] det5_valid_pb1, 1: det5_CHID valid, 0 not valid<br>[3:0] det5_chid_pb1[35:32]<br><br>6: Motion detection CHID1<br>[7] digital_chid_valid_pb1, 1: D CHID valid, 0: A CHID valid<br>[4] det6_valid_pb1, 1: det6_CHID valid, 0 not valid<br>[3:0] det6_chid_pb1[35:32]<br><br>7: Motion detection CHID1<br>[7] digital_chid_valid_pb1, 1: D CHID valid, 0: A CHID valid<br>[4] det7_valid_pb1, 1: det7_CHID valid, 0 not valid<br>[3:0] det7_chid_pb1[35:32]<br><br>8: Motion detection CHID1<br>[7] digital_chid_valid_pb1, 1: D CHID valid, 0: A CHID valid |

| Bit | R/W | Default | Description |
|---|---|---|---|
| | | | [4] det8_valid_pb1, 1: det8_CHID valid, 0 not valid<br>[3:0] det8_chid_pb1[35:32]<br><br>9: user CHID1<br>[7] digital_chid_valid_pb1, 1: D CHID valid, 0: A CHID valid<br>[4] usr1_valid_pb1, 1: usr1_CHID valid, 0 not valid<br>[3:0] usr1_chid_pb1[35:32]<br><br>a: user CHID2<br>[7] digital_chid_valid_pb1, 1: D CHID valid, 0: A CHID valid<br>[4] usr2_valid_pb1, 1: usr2_CHID valid, 0 not valid<br>[3:0] usr2_chid_pb1[35:32]<br><br>b: user CHID3<br>[7] digital_chid_valid_pb1, 1: D CHID valid, 0: A CHID valid<br>[4] usr3_valid_pb1, 1: usr3_CHID valid, 0 not valid<br>[3:0] usr3_chid_pb1[35:32] |

**To get most up-to-date channel ID on four PB ports, write any value to 0x372 to trigger a self update process.**

### PB CHANNEL ID DEC READ BUS PB1 BYTE 2 – 0X373

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | CHID_RD_BUS_PB1[31:24]<br><br>Read CHID_rd_bus_pb1[31:24]. Depend on CHID_RD_SEL_PB1(REG OX3DE), it read one of six CHID word:<br><br>0: auto CHID [7:0] auto_chid_pb1[31:24]<br>1: detection CHID1<br>[7:0] det1_chid_pb1[31:24]<br>2: detection CHID2<br>[7:0] det2_chid_pb1[31:24]<br>3: Motion detection CHID1<br>[7:0] det3_chid_pb1[31:24]<br>4: Motion detection CHID1<br>[7:0] det4_chid_pb1[31:24]<br>5: Motion detection CHID1<br>[7:0] det5_chid_pb1[31:24]<br>6: Motion detection CHID1<br>[7:0] det6_chid_pb1[31:24]<br>7: Motion detection CHID1<br>[7:0] det7_chid_pb1[31:24]<br>8: Motion detection CHID1<br>[7:0] det8_chid_pb1[31:24]<br>9: user CHID1<br>[7:0] usr1_chid_pb1[31:24]<br>a: user CHID2<br>[7:0] usr2_chid_pb1[31:24]<br>b: user CHID3<br>[7:0] usr3_chid_pb1[31:24] |

### PB CHANNEL ID DEC READ BUS PB1 BYTE 3 – 0X374

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x7 | **CHID_RD_BUS_PB1[23:16]**<br><br>Read CHID_rd_bus_pb1[23:16]. Depend on CHID_RD_SEL_PB1(REG OX3DE), it read one of six CHID word:<br>0: auto CHID<br>[7:0] auto_chid_pb1[23:16]<br>1: detection CHID1<br>[7:0] det1_chid_pb1[23:16]<br>2: detection CHID2<br>[7:0] det2_chid_pb1[23:16]<br>3: Motion detection CHID1<br>[7:0] det3_chid_pb1[23:16]<br>4: Motion detection CHID1<br>[7:0] det4_chid_pb1[23:16]<br>5: Motion detection CHID1<br>[7:0] det5_chid_pb1[23:16]<br>6: Motion detection CHID1<br>[7:0] det6_chid_pb1[23:16]<br>7: Motion detection CHID1<br>[7:0] det7_chid_pb1[23:16]<br>8: Motion detection CHID1<br>[7:0] det8_chid_pb1[23:16]<br>9: user CHID1<br>[7:0] usr1_chid_pb1[23:16]<br>a: user CHID2<br>[7:0] usr2_chid_pb1[23:16]<br>b: user CHID3<br>[7:0] usr3_chid_pb1[23:16] |

### PB CHANNEL ID DEC READ BUS PB1 BYTE 4 – 0X3C7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x7 | **CHID_RD_BUS_PB1[15:8]**<br><br>Read CHID_rd_bus_pb1[15:8]. Depend on CHID_RD_SEL_PB1(REG OX3DE), it read one of six CHID word:<br>0: auto CHID<br>[7:0] auto_chid_pb1[15:8]<br>1: detection CHID1<br>[7:0] det1_chid_pb1[15:8]<br>2: detection CHID2<br>[7:0] det2_chid_pb1[15:8]<br>3: Motion detection CHID1<br>[7:0] det3_chid_pb1[15:8]<br>4: Motion detection CHID1<br>[7:0] det4_chid_pb1[15:8]<br>5: Motion detection CHID1<br>[7:0] det5_chid_pb1[15:8]<br>6: Motion detection CHID1<br>[7:0] det6_chid_pb1[15:8]<br>7: Motion detection CHID1<br>[7:0] det7_chid_pb1[15:8]<br>8: Motion detection CHID1<br>[7:0] det8_chid_pb1[15:8] |

| Bit | R/W | Default | Description |
|---|---|---|---|
| | | | 9: user CHID1<br>[7:0] usr1_chid_pb1[15:8]<br>a: user CHID2<br>[7:0] usr2_chid_pb1[15:8]<br>b: user CHID3<br>[7:0] usr3_chid_pb1[15:8] |

### PB CHANNEL ID DEC READ BUS PB1 BYTE 5 – 0X3C8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x7 | CHID_RD_BUS_PB1[7:0]<br><br>Read CHID_rd_bus_pb1[7:0]. Depend on CHID_RD_SEL_PB1(REG OX3DE), it read one of six CHID word:<br>0: auto CHID<br>[7:0] auto_chid_pb1[7:0]<br>1: detection CHID1<br>[7:0] det1_chid_pb1[7:0]<br>2: detection CHID2<br>[7:0] det2_chid_pb1[7:0]<br>3: Motion detection CHID1<br>[7:0] det3_chid_pb1[7:0]<br>4: Motion detection CHID1<br>[7:0] det4_chid_pb1[7:0]<br>5: Motion detection CHID1<br>[7:0] det5_chid_pb1[7:0]<br>6: Motion detection CHID1<br>[7:0] det6_chid_pb1[7:0]<br>7: Motion detection CHID1<br>[7:0] det7_chid_pb1[7:0]<br>8: Motion detection CHID1<br>[7:0] det8_chid_pb1[7:0]<br>9: user CHID1<br>[7:0] usr1_chid_pb1[7:0]<br>a: user CHID2<br>[7:0] usr2_chid_pb1[7:0]<br>b: user CHID3<br>[7:0] usr3_chid_pb1[7:0] |

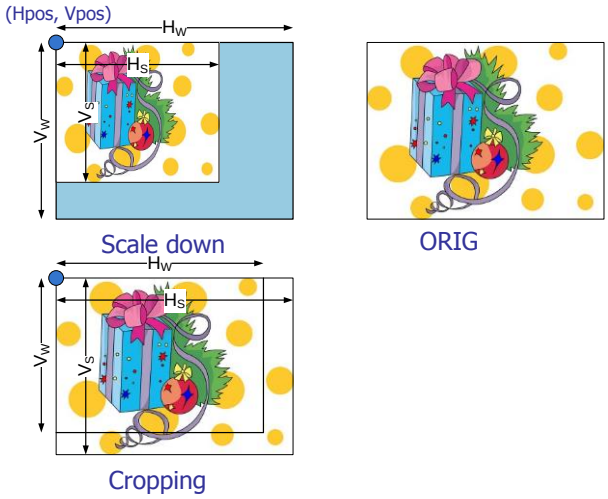
# Video Write Buffer (RGBW)

## Introduction

TW2880 has incorporated 32 write buffers to support 16 live video channels and 16 playback channels. Each channel has its own registers for users to adjust window width, height, starting positions and orientations. Each live channel can support up to D1 size while for playback channel the resolution can go up to HD (1920x1080). One thing to remember is in this HD configuration, only two windows are available. All write requests from the buffers will be summed up and prioritized in the multiplex unit and later sent to SDRAM controller. Each channel need to be programmed to write to different locations in SDRAM.

## Window Control

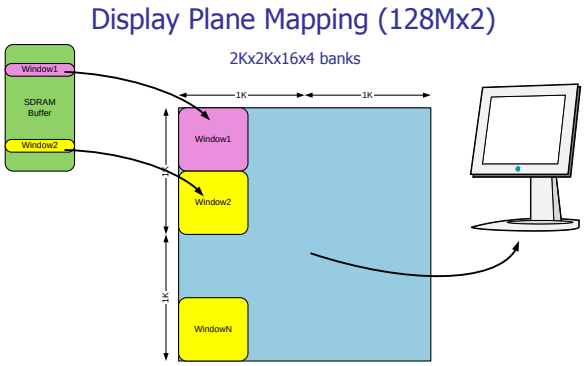
Each window in the display path has its own set of control. It can be enabled and disable. The height, width and the starting position of the window is adjustable. User can mirror the image horizontally or vertically to compensate the position of the camera. Cropping is also supported if the display window size is smaller than the incoming signal. One thing to be remembered is windows cannot be overlapped otherwise artifact will appear in the display area.




## Display Map Control

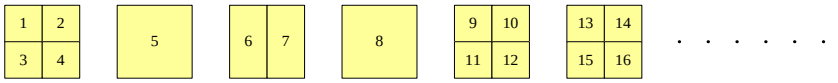
When updating video information, all channels are writing into the same bank of display memory. When display controller read data, it will read out the data sequentially. The bank control of the reading and writing controller is handled by Frame Rate Control Module (FRSC) and is discussed in detailed in later sections.

The off-screen memory is used by OSG and OSD module. TW2880 has a different display map control when 3D de-interlacing and 2D de-interlacing is deployed.



## Play Back Channels

**Play back channels can support several modes, such as interlaced or progressive. And also for one play back port, different formats can be support in different frames, such as D1, half D1, CIF and Quad. And also different channels being in one port is possible. Data rate in play back channel can be 27MHz, 54MHz or 108MHz. In BT1120 case, clock can be 74.25MHz. Here is an example for one play back port.**




**Playback write control unit can write different channels to different location in SDRAM. User can set size and location register for different channels. There are total 16 groups of register for play back channels. These registers share address with live channels. Channel number and format information is given by play back channel ID decoder unit.**

**If play back port has only one channel and data rate is 27MHz, it can be set to normal write buffer mode. In normal mode, channel number is not used.**

**In summary, playback write control unit can support two mode: auto mode and normal mode. In auto mode, channel data will be written to different location according to channel number. In normal mode, channel data will be written to one location according to port number.**

## Register Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x600 | R/W | 0 | rgb_wr_ctrl0<br>[3]: freeze0<br>[2]: wr_en0<br>[1]: hinv0<br>[0]: vinv0 |
| 0x601 | R/W | 0 | rgb_wr_ctrl1<br>[3]: freeze1<br>[2]: wr_en1<br>[1]: hinv1<br>[0]: vinv1 |
| 0x602 | R/W | 0 | rgb_wr_ctrl2<br>[3]: freeze2<br>[2]: wr_en2<br>[1]: hinv2<br>[0]: vinv2 |
| 0x603 | R/W | 0 | rgb_wr_ctrl3<br>[3]: freeze3<br>[2]: wr_en3<br>[1]: hinv3<br>[0]: vinv3 |
| 0x604 | R/W | 0 | rgb_wr_ctrl4<br>[3]: freeze4<br>[2]: wr_en4<br>[1]: hinv4<br>[0]: vinv4 |
| 0x605 | R/W | 0 | rgb_wr_ctrl5<br>[3]: freeze5<br>[2]: wr_en5<br>[1]: hinv5<br>[0]: vinv5 |
| 0x606 | R/W | 0 | rgb_wr_ctrl6<br>[3]: freeze6<br>[2]: wr_en6<br>[1]: hinv6<br>[0]: vinv6 |
| 0x607 | R/W | 0 | rgb_wr_ctrl7<br>[3]: freeze7<br>[2]: wr_en7<br>[1]: hinv7<br>[0]: vinv7 |
| 0x608 | R/W | 0 | rgb_wr_ctrl8<br>[3]: freeze8<br>[2]: wr_en8<br>[1]: hinv8<br>[0]: vinv8 |
| 0x609 | R/W | 0 | rgb_wr_ctrl9<br>[3]: freeze9<br>[2]: wr_en9<br>[1]: hinv9<br>[0]: vinv9 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x60A | R/W | 0 | rgb_wr_ctrl10<br>[3]: freeze10<br>[2]: wr_en10<br>[1]: hinv10<br>[0]: vinv10 |
| 0x60B | R/W | 0 | rgb_wr_ctrl11<br>[3]: freeze11<br>[2]: wr_en11<br>[1]: hinv11<br>[0]: vinv11 |
| 0x60C | R/W | 0 | rgb_wr_ctrl12<br>[3]: freeze12<br>[2]: wr_en12<br>[1]: hinv12<br>[0]: vinv12 |
| 0x60D | R/W | 0 | rgb_wr_ctrl13<br>[3]: freeze13<br>[2]: wr_en13<br>[1]: hinv13<br>[0]: vinv13 |
| 0x60E | R/W | 0 | rgb_wr_ctrl14<br>[3]: freeze14<br>[2]: wr_en14<br>[1]: hinv14<br>[0]: vinv14 |
| 0x60F | R/W | 0 | rgb_wr_ctrl15<br>[3]: freeze15<br>[2]: wr_en15<br>[1]: hinv15<br>[0]: vinv15 |
| 0x610 | R/W | 0 | rgb_wr_ctrl16<br>[3]: freeze16<br>[2]: wr_en16<br>[1]: hinv16<br>[0]: vinv16 |
| 0x611 | R/W | 0 | rgb_wr_ctrl17<br>[3]: freeze17<br>[2]: wr_en17<br>[1]: hinv17<br>[0]: vinv17 |
| 0x612 | R/W | 0 | rgb_wr_ctrl18<br>[3]: freeze18<br>[2]: wr_en18<br>[1]: hinv18<br>[0]: vinv18 |
| 0x613 | R/W | 0 | rgb_wr_ctrl19<br>[3]: freeze19<br>[2]: wr_en19<br>[1]: hinv19<br>[0]: vinv19 |
| 0x614 | R/W | 0 | rgb_vpos0_live[7:0] / rgb_vpos0_pb[7:0] |
| 0x615 | R/W | 0 | rgb_vpos0_live[11:8] / rgb_vpos0_pb[11:8] |
| 0x616 | R/W | 0 | rgb_vpos1_live[7:0] / rgb_vpos1_pb[7:0] |
| 0x617 | R/W | 0 | rgb_vpos1_live[11:8] / rgb_vpos1_pb[11:8] |
| 0x618 | R/W | 0 | rgb_vpos2_live[7:0] / rgb_vpos2_pb[7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x619 | R/W | 0 | rgb_vpos2_live[11:8] / rgb_vpos2_pb[11:8] |
| 0x61A | R/W | 0 | rgb_vpos3_live[7:0] / rgb_vpos3_pb[7:0] |
| 0x61B | R/W | 0 | rgb_vpos3_live[11:8] / rgb_vpos3_pb[11:8] |
| 0x61C | R/W | 0 | rgb_vpos4_live[7:0] / rgb_vpos4_pb[7:0] |
| 0x61D | R/W | 0 | rgb_vpos4_live[11:8] / rgb_vpos4_pb[11:8] |
| 0x61E | R/W | 0 | rgb_vpos5_live[7:0] / rgb_vpos5_pb[7:0] |
| 0x61F | R/W | 0 | rgb_vpos5_live[11:8] / rgb_vpos5_pb[11:8] |
| 0x620 | R/W | 0 | rgb_vpos6_live[7:0] / rgb_vpos6_pb[7:0] |
| 0x621 | R/W | 0 | rgb_vpos6_live[11:8] / rgb_vpos6_pb[11:8] |
| 0x622 | R/W | 0 | rgb_vpos7_live[7:0] / rgb_vpos7_pb[7:0] |
| 0x623 | R/W | 0 | rgb_vpos7_live[11:8] / rgb_vpos7_pb[11:8] |
| 0x624 | R/W | 0 | rgb_vpos8_live[7:0] / rgb_vpos8_pb[7:0] |
| 0x625 | R/W | 0 | rgb_vpos8_live[11:8] / rgb_vpos8_pb[11:8] |
| 0x626 | R/W | 0 | rgb_vpos9_live[7:0] / rgb_vpos9_pb[7:0] |
| 0x627 | R/W | 0 | rgb_vpos9_live[11:8] / rgb_vpos9_pb[11:8] |
| 0x628 | R/W | 0 | rgb_vpos10_live[7:0] / rgb_vpos10_pb[7:0] |
| 0x629 | R/W | 0 | rgb_vpos10_live[11:8] / rgb_vpos10_pb[11:8] |
| 0x62A | R/W | 0 | rgb_vpos11_live[7:0] / rgb_vpos11_pb[7:0] |
| 0x62B | R/W | 0 | rgb_vpos11_live[11:8] / rgb_vpos11_pb[11:8] |
| 0x62C | R/W | 0 | rgb_vpos12_live[7:0] / rgb_vpos12_pb[7:0] |
| 0x62D | R/W | 0 | rgb_vpos12_live[11:8] / rgb_vpos12_pb[11:8] |
| 0x62E | R/W | 0 | rgb_vpos13_live[7:0] / rgb_vpos13_pb[7:0] |
| 0x62F | R/W | 0 | rgb_vpos13_live[11:8] / rgb_vpos13_pb[11:8] |
| 0x630 | R/W | 0 | rgb_vpos14_live[7:0] / rgb_vpos14_pb[7:0] |
| 0x631 | R/W | 0 | rgb_vpos14_live[11:8] / rgb_vpos14_pb[11:8] |
| 0x632 | R/W | 0 | rgb_vpos15_live[7:0] / rgb_vpos15_pb[7:0] |
| 0x633 | R/W | 0 | rgb_vpos15_live[11:8] / rgb_vpos15_pb[11:8] |
| 0x634 | R/W | 0 | rgb_vpos16_pb[7:0] |
| 0x635 | R/W | 0 | rgb_vpos16_pb[11:8] |
| 0x636 | R/W | 0 | rgb_vpos17_pb[7:0] |
| 0x637 | R/W | 0 | rgb_vpos17_pb[11:8] |
| 0x638 | R/W | 0 | rgb_vpos18_pb[7:0] |
| 0x639 | R/W | 0 | rgb_vpos18_pb[11:8] |
| 0x63A | R/W | 0 | rgb_vpos19_pb[7:0] |
| 0x63B | R/W | 0 | rgb_vpos19_pb[11:8] |
| 0x63C | R/W | 0 | rgb_vsize0_live[7:0] / rgb_vsize0_pb[7:0] |
| 0x63D | R/W | 0 | rgb_vsize0_live[10:8] / rgb_vsize0_pb[10:8] |
| 0x63E | R/W | 0 | rgb_vsize1_live[7:0] / rgb_vsize1_pb[7:0] |
| 0x63F | R/W | 0 | rgb_vsize1_live[10:8] / rgb_vsize1_pb[10:8] |
| 0x640 | R/W | 0 | rgb_vsize2_live[7:0] / rgb_vsize2_pb[7:0] |
| 0x641 | R/W | 0 | rgb_vsize2_live[10:8] / rgb_vsize2_pb[10:8] |
| 0x642 | R/W | 0 | rgb_vsize3_live[7:0] / rgb_vsize3_pb[7:0] |
| 0x643 | R/W | 0 | rgb_vsize3_live[10:8] / rgb_vsize3_pb[10:8] |
| 0x644 | R/W | 0 | rgb_vsize4_live[7:0] / rgb_vsize4_pb[7:0] |
| 0x645 | R/W | 0 | rgb_vsize4_live[10:8] / rgb_vsize4_pb[10:8] |
| 0x646 | R/W | 0 | rgb_vsize5_live[7:0] / rgb_vsize5_pb[7:0] |
| 0x647 | R/W | 0 | rgb_vsize5_live[10:8] / rgb_vsize5_pb[10:8] |
| 0x648 | R/W | 0 | rgb_vsize6_live[7:0] / rgb_vsize6_pb[7:0] |
| 0x649 | R/W | 0 | rgb_vsize6_live[10:8] / rgb_vsize6_pb[10:8] |
| 0x64A | R/W | 0 | rgb_vsize7_live[7:0] / rgb_vsize7_pb[7:0] |
| 0x64B | R/W | 0 | rgb_vsize7_live[10:8] / rgb_vsize7_pb[10:8] |
| 0x64C | R/W | 0 | rgb_vsize8_live[7:0] / rgb_vsize8_pb[7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x64D | R/W | 0 | rgb_vsize8_live[10:8] / rgb_vsize8_pb[10:8] |
| 0x64E | R/W | 0 | rgb_vsize9_live[7:0] / rgb_vsize9_pb[7:0] |
| 0x64F | R/W | 0 | rgb_vsize9_live[10:8] / rgb_vsize9_pb[10:8] |
| 0x650 | R/W | 0 | rgb_vsize10_live[7:0] / rgb_vsize10_pb[7:0] |
| 0x651 | R/W | 0 | rgb_vsize10_live[10:8] / rgb_vsize10_pb[10:8] |
| 0x652 | R/W | 0 | rgb_vsize11_live[7:0] / rgb_vsize11_pb[7:0] |
| 0x653 | R/W | 0 | rgb_vsize11_live[10:8] / rgb_vsize11_pb[10:8] |
| 0x654 | R/W | 0 | rgb_vsize12_live[7:0] / rgb_vsize12_pb[7:0] |
| 0x655 | R/W | 0 | rgb_vsize12_live[10:8] / rgb_vsize12_pb[10:8] |
| 0x656 | R/W | 0 | rgb_vsize13_live[7:0] / rgb_vsize13_pb[7:0] |
| 0x657 | R/W | 0 | rgb_vsize13_live[10:8] / rgb_vsize13_pb[10:8] |
| 0x658 | R/W | 0 | rgb_vsize14_live[7:0] / rgb_vsize14_pb[7:0] |
| 0x659 | R/W | 0 | rgb_vsize14_live[10:8] / rgb_vsize14_pb[10:8] |
| 0x65A | R/W | 0 | rgb_vsize15_live[7:0] / rgb_vsize15_pb[7:0] |
| 0x65B | R/W | 0 | rgb_vsize15_live[10:8] / rgb_vsize15_pb[10:8] |
| 0x65C | R/W | 0 | rgb_vsize16_pb[7:0] |
| 0x65D | R/W | 0 | rgb_vsize16_pb[10:8] |
| 0x65E | R/W | 0 | rgb_vsize17_pb[7:0] |
| 0x65F | R/W | 0 | rgb_vsize17_pb[10:8] |
| 0x660 | R/W | 0 | rgb_vsize18_pb[7:0] |
| 0x661 | R/W | 0 | rgb_vsize18_pb[10:8] |
| 0x662 | R/W | 0 | rgb_vsize19_pb[7:0] |
| 0x663 | R/W | 0 | rgb_vsize19_pb[10:8] |
| 0x664 | R/W | 0 | rgb_hpos0_live[7:0] / rgb_hpos0_pb[7:0] |
| 0x665 | R/W | 0 | rgb_hpos0_live[9:8] / rgb_hpos0_pb[9:8] |
| 0x666 | R/W | 0 | rgb_hpos1_live[7:0] / rgb_hpos1_pb[7:0] |
| 0x667 | R/W | 0 | rgb_hpos1_live[9:8] / rgb_hpos1_pb[9:8] |
| 0x668 | R/W | 0 | rgb_hpos2_live[7:0] / rgb_hpos2_pb[7:0] |
| 0x669 | R/W | 0 | rgb_hpos2_live[9:8] / rgb_hpos2_pb[9:8] |
| 0x66A | R/W | 0 | rgb_hpos3_live[7:0] / rgb_hpos3_pb[7:0] |
| 0x66B | R/W | 0 | rgb_hpos3_live[9:8] / rgb_hpos3_pb[9:8] |
| 0x66C | R/W | 0 | rgb_hpos4_live[7:0] / rgb_hpos4_pb[7:0] |
| 0x66D | R/W | 0 | rgb_hpos4_live[9:8] / rgb_hpos4_pb[9:8] |
| 0x66E | R/W | 0 | rgb_hpos5_live[7:0] / rgb_hpos5_pb[7:0] |
| 0x66F | R/W | 0 | rgb_hpos5_live[9:8] / rgb_hpos5_pb[9:8] |
| 0x670 | R/W | 0 | rgb_hpos6_live[7:0] / rgb_hpos6_pb[7:0] |
| 0x671 | R/W | 0 | rgb_hpos6_live[9:8] / rgb_hpos6_pb[9:8] |
| 0x672 | R/W | 0 | rgb_hpos7_live[7:0] / rgb_hpos7_pb[7:0] |
| 0x673 | R/W | 0 | rgb_hpos7_live[9:8] / rgb_hpos7_pb[9:8] |
| 0x674 | R/W | 0 | rgb_hpos8_live[7:0] / rgb_hpos8_pb[7:0] |
| 0x675 | R/W | 0 | rgb_hpos8_live[9:8] / rgb_hpos8_pb[9:8] |
| 0x676 | R/W | 0 | rgb_hpos9_live[7:0] / rgb_hpos9_pb[7:0] |
| 0x677 | R/W | 0 | rgb_hpos9_live[9:8] / rgb_hpos9_pb[9:8] |
| 0x678 | R/W | 0 | rgb_hpos10_live[7:0] / rgb_hpos10_pb[7:0] |
| 0x679 | R/W | 0 | rgb_hpos10_live[9:8] / rgb_hpos10_pb[9:8] |
| 0x67A | R/W | 0 | rgb_hpos11_live[7:0] / rgb_hpos11_pb[7:0] |
| 0x67B | R/W | 0 | rgb_hpos11_live[9:8] / rgb_hpos11_pb[9:8] |
| 0x67C | R/W | 0 | rgb_hpos12_live[7:0] / rgb_hpos12_pb[7:0] |
| 0x67D | R/W | 0 | rgb_hpos12_live[9:8] / rgb_hpos12_pb[9:8] |
| 0x67E | R/W | 0 | rgb_hpos13_live[7:0] / rgb_hpos13_pb[7:0] |
| 0x67F | R/W | 0 | rgb_hpos13_live[9:8] / rgb_hpos13_pb[9:8] |
| 0x680 | R/W | 0 | rgb_hpos14_live[7:0] / rgb_hpos14_pb[7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x681 | R/W | 0 | rgb_hpos14_live[9:8] / rgb_hpos14_pb[9:8] |
| 0x682 | R/W | 0 | rgb_hpos15_live[7:0] / rgb_hpos15_pb[7:0] |
| 0x683 | R/W | 0 | rgb_hpos15_live[9:8] / rgb_hpos15_pb[9:8] |
| 0x684 | R/W | 0 | rgb_hpos16_pb[7:0] |
| 0x685 | R/W | 0 | rgb_hpos16_pb[9:8] |
| 0x686 | R/W | 0 | rgb_hpos17_pb[7:0] |
| 0x687 | R/W | 0 | rgb_hpos17_pb[9:8] |
| 0x688 | R/W | 0 | rgb_hpos18_pb[7:0] |
| 0x689 | R/W | 0 | rgb_hpos18_pb[9:8] |
| 0x68A | R/W | 0 | rgb_hpos19_pb[7:0] |
| 0x68B | R/W | 0 | rgb_hpos19_pb[9:8] |
| 0x68C | R/W | 0 | rgb_hsize0_live[7:0] / rgb_hsize0_pb[7:0] |
| 0x68D | R/W | 0 | rgb_hsize0_live[8] / rgb_hsize0_pb[8] |
| 0x68E | R/W | 0 | rgb_hsize1_live[7:0] / rgb_hsize1_pb[7:0] |
| 0x68F | R/W | 0 | rgb_hsize1_live[8] / rgb_hsize1_pb[8] |
| 0x690 | R/W | 0 | rgb_hsize2_live[7:0] / rgb_hsize2_pb[7:0] |
| 0x691 | R/W | 0 | rgb_hsize2_live[8] / rgb_hsize2_pb[8] |
| 0x692 | R/W | 0 | rgb_hsize3_live[7:0] / rgb_hsize3_pb[7:0] |
| 0x693 | R/W | 0 | rgb_hsize3_live[8] / rgb_hsize3_pb[8] |
| 0x694 | R/W | 0 | rgb_hsize4_live[7:0] / rgb_hsize4_pb[7:0] |
| 0x695 | R/W | 0 | rgb_hsize4_live[8] / rgb_hsize4_pb[8] |
| 0x696 | R/W | 0 | rgb_hsize5_live[7:0] / rgb_hsize5_pb[7:0] |
| 0x697 | R/W | 0 | rgb_hsize5_live[8] / rgb_hsize5_pb[8] |
| 0x698 | R/W | 0 | rgb_hsize6_live[7:0] / rgb_hsize6_pb[7:0] |
| 0x699 | R/W | 0 | rgb_hsize6_live[8] / rgb_hsize6_pb[8] |
| 0x69A | R/W | 0 | rgb_hsize7_live[7:0] / rgb_hsize7_pb[7:0] |
| 0x69B | R/W | 0 | rgb_hsize7_live[8] / rgb_hsize7_pb[8] |
| 0x69C | R/W | 0 | rgb_hsize8_live[7:0] / rgb_hsize8_pb[7:0] |
| 0x69D | R/W | 0 | rgb_hsize8_live[8] / rgb_hsize8_pb[8] |
| 0x69E | R/W | 0 | rgb_hsize9_live[7:0] / rgb_hsize9_pb[7:0] |
| 0x69F | R/W | 0 | rgb_hsize9_live[8] / rgb_hsize9_pb[8] |
| 0x6A0 | R/W | 0 | rgb_hsize10_live[7:0] / rgb_hsize10_pb[7:0] |
| 0x6A1 | R/W | 0 | rgb_hsize10_live[8] / rgb_hsize10_pb[8] |
| 0x6A2 | R/W | 0 | rgb_hsize11_live[7:0] / rgb_hsize11_pb[7:0] |
| 0x6A3 | R/W | 0 | rgb_hsize11_live[8] / rgb_hsize11_pb[8] |
| 0x6A4 | R/W | 0 | rgb_hsize12_live[7:0] / rgb_hsize12_pb[7:0] |
| 0x6A5 | R/W | 0 | rgb_hsize12_live[8] / rgb_hsize12_pb[8] |
| 0x6A6 | R/W | 0 | rgb_hsize13_live[7:0] / rgb_hsize13_pb[7:0] |
| 0x6A7 | R/W | 0 | rgb_hsize13_live[8] / rgb_hsize13_pb[8] |
| 0x6A8 | R/W | 0 | rgb_hsize14_live[7:0] / rgb_hsize14_pb[7:0] |
| 0x6A9 | R/W | 0 | rgb_hsize14_live[8] / rgb_hsize14_pb[8] |
| 0x6AA | R/W | 0 | rgb_hsize15_live[7:0] / rgb_hsize15_pb[7:0] |
| 0x6AB | R/W | 0 | rgb_hsize15_live[8] / rgb_hsize15_pb[8] |
| 0x6AC | R/W | 0 | rgb_hsize16_pb[7:0] |
| 0x6AD | R/W | 0 | rgb_hsize16_pb[8] |
| 0x6AE | R/W | 0 | rgb_hsize17_pb[7:0] |
| 0x6AF | R/W | 0 | rgb_hsize17_pb[8] |
| 0x6B0 | R/W | 0 | rgb_hsize18_pb[7:0] |
| 0x6B1 | R/W | 0 | rgb_hsize18_pb[8] |
| 0x6B2 | R/W | 0 | rgb_hsize19_pb[7:0] |
| 0x6B3 | R/W | 0 | rgb_hsize19_pb[8] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x6B4 | R/W | 0 | [7:4]: newfld_sel<br>[3]: mode<br>[2]: TST_IN_PB<br>[1]: PALNT<br>[0]: TST_IN |
| 0x6B5 | R/W | 0 | Test_timing_replace[7:0] |
| 0x6B6 | R/W | 0 | [7]: line_bigger_en_pb<br>[6]: line_bigger_en<br>[5]: use_novid<br>[4:3]: skip_sel<br>[2]: prot_en<br>[1]: CHID_sel<br>[0]: live_pb |
| 0x6B7 | R/W | 0 | [7:4]: pb1_chnum<br>[3:0]: pb0_chnum |
| 0x6B8 | R/W | 0 | [7:4]: pb3_chnum<br>[3:0]: pb2_chnum |
| 0x6B9 | R/W | 0 | [7:6]: IGNON_NON-STANDRED[1:0]<br>[5]: NON-STANDRED_IRQ_MASK<br>[4]: read select<br>[3:0]: pb3_chnum_en – pb0_chnum_en |
| 0x6BA | R/W | 0 | Test_timing_replace[15:8] |
| 0x6BB | R/W | 0 | Pb_ch_frez[7:0] |
| 0x6BC | R/W | 0 | Pb_ch_frez[15:8] |
| 0x6BD | R/W | 0 | Prot_addr[7:0] |
| 0x6BE | R/W | 0 | Prot_addr[15:8] |
| 0x6BF | R/W | 0 | Prot_addr[21:16] |
| 0x6C0 | R/W | 0 | rgb_hstart0_live[7:0] / rgb_hstart0_pb[7:0] |
| 0x6C1 | R/W | 0 | rgb_hstart1_live[7:0] / rgb_hstart1_pb[7:0] |
| 0x6C2 | R/W | 0 | rgb_hstart2_live[7:0] / rgb_hstart2_pb[7:0] |
| 0x6C3 | R/W | 0 | rgb_hstart3_live[7:0] / rgb_hstart3_pb[7:0] |
| 0x6C4 | R/W | 0 | rgb_hstart4_live[7:0] / rgb_hstart4_pb[7:0] |
| 0x6C5 | R/W | 0 | rgb_hstart5_live[7:0] / rgb_hstart5_pb[7:0] |
| 0x6C6 | R/W | 0 | rgb_hstart6_live[7:0] / rgb_hstart6_pb[7:0] |
| 0x6C7 | R/W | 0 | rgb_hstart7_live[7:0] / rgb_hstart7_pb[7:0] |
| 0x6C8 | R/W | 0 | rgb_hstart8_live[7:0] / rgb_hstart8_pb[7:0] |
| 0x6C9 | R/W | 0 | rgb_hstart9_live[7:0] / rgb_hstart9_pb[7:0] |
| 0x6CA | R/W | 0 | rgb_hstart10_live[7:0] / rgb_hstart10_pb[7:0] |
| 0x6CB | R/W | 0 | rgb_hstart11_live[7:0] / rgb_hstart11_pb[7:0] |
| 0x6CC | R/W | 0 | rgb_hstart12_live[7:0] / rgb_hstart12_pb[7:0] |
| 06CD | R/W | 0 | rgb_hstart13_live[7:0] / rgb_hstart13_pb[7:0] |
| 0x6CE | R/W | 0 | rgb_hstart14_live[7:0] / rgb_hstart14_pb[7:0] |
| 0x6CF | R/W | 0 | rgb_hstart15_live[7:0] / rgb_hstart15_pb[7:0] |
| 0x6D0 | R/W | 0 | rgb_hstart16_pb[7:0] |
| 0x6D1 | R/W | 0 | rgb_hstart17_pb[7:0] |
| 0x6D2 | R/W | 0 | rgb_hstart18_pb[7:0] |
| 0x6D3 | R/W | 0 | rgb_hstart19_pb[7:0] |
| 0x6D4 | R/W | 0 | rgb_vstart0_live[7:0] / rgb_vstart0_pb[7:0] |
| 0x6D5 | R/W | 0 | rgb_vstart1_live[7:0] / rgb_vstart1_pb[7:0] |
| 0x6D6 | R/W | 0 | rgb_vstart2_live[7:0] / rgb_vstart2_pb[7:0] |
| 0x6D7 | R/W | 0 | rgb_vstart3_live[7:0] / rgb_vstart3_pb[7:0] |
| 0x6D8 | R/W | 0 | rgb_vstart4_live[7:0] / rgb_vstart4_pb[7:0] |
| 0x6D9 | R/W | 0 | rgb_vstart5_live[7:0] / rgb_vstart5_pb[7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x6DA | R/W | 0 | rgb_vstart6_live[7:0] / rgb_vstart6_pb[7:0] |
| 0x6DB | R/W | 0 | rgb_vstart7_live[7:0] / rgb_vstart7_pb[7:0] |
| 0x6DC | R/W | 0 | rgb_vstart8_live[7:0] / rgb_vstart8_pb[7:0] |
| 0x6DD | R/W | 0 | rgb_vstart9_live[7:0] / rgb_vstart9_pb[7:0] |
| 0x6DE | R/W | 0 | rgb_vstart10_live[7:0] / rgb_vstart10_pb[7:0] |
| 0x6DF | R/W | 0 | rgb_vstart11_live[7:0] / rgb_vstart11_pb[7:0] |
| 0x6E0 | R/W | 0 | rgb_vstart12_live[7:0] / rgb_vstart12_pb[7:0] |
| 06E1 | R/W | 0 | rgb_vstart13_live[7:0] / rgb_vstart13_pb[7:0] |
| 0x6E2 | R/W | 0 | rgb_vstart14_live[7:0] / rgb_vstart14_pb[7:0] |
| 0x6E3 | R/W | 0 | rgb_vstart15_live[7:0] / rgb_vstart15_pb[7:0] |
| 0x6E4 | R/W | 0 | rgb_vstart16_pb[7:0] |
| 0x6E5 | R/W | 0 | rgb_vstart17_pb[7:0] |
| 0x6E6 | R/W | 0 | rgb_vstart18_pb[7:0] |
| 0x6E7 | R/W | 0 | rgb_vstart19_pb[7:0] |
| 0x6E8 | R/W | 0 | Live port 1, 0 correction control |
| 0x6E9 | R/W | 0 | Live port 3, 2 correction control |
| 0x6EA | R/W | 0 | Live port 5, 4 correction control |
| 0x6EB | R/W | 0 | Live port 7, 6 correction control |
| 0x6EC | R/W | 0 | Live port 9, 8 correction control |
| 0x6ED | R/W | 0 | Live port 11, 10 correction control |
| 0x6EE | R/W | 0 | Live port 13, 12 correction control |
| 0x6EF | R/W | 0 | Live port 15, 14 correction control |
| 0x6F0 | RO | 0 | linenum_pb0[7:0] |
| 0x6F1 | RO | 0 | linenum_pb0[11:8] |
| 0x6F0 | RW | 0 | PB port 1, 0 correction control, shared |
| 0x6F1 | RW | 0 | PB port 3, 2 correction control, shared |
| 0x6F2 | RO | 0 | linenum_pb1[7:0] |
| 0x6F3 | RO | 0 | linenum_pb1[11:8] |
| 0x6F4 | RO | 0 | linenum_pb2[7:0] |
| 0x6F5 | RO | 0 | linenum_pb2[11:8] |
| 0x6F6 | RO | 0 | linenum_pb3[7:0] |
| 0x6F7 | RO | 0 | linenum_pb3[11:8] |
| 0x6F8 | R/W | 0 | [7]: write to clear<br>[3:0]: pb3_valid – pb0_valid |
| 0x6F9 | R/W | 0 | [7:0]: period2 external correction period |
| 0x6FA | R/W | 0 | [7:4]: Enable ignore function<br>[3:0]: pb_free_3 – pb_free_0 |
| 0x6FB | R/W | 0x1C | [7:4]: fifo_cnt_limit<br>[3]: fld_update<br>[2]: fld_sel_fix<br>[1]: fld_sel_en<br>[0]: vstart_unit_pb |
| 0x6FC | R/W | 0 | [7:6]: address select<br>[5]: Force live use old method<br>[4]: seld<br>[3:2]: reserved<br>[1]: hstart_unit_pb, 1= 8 pixels<br>[0]: vinv_sel |
| 0x6FD | R/W | 0 | pb_ch_en[7:0] |
| 0x6FE | R/W | 0 | pb_ch_en[15:8] |
| 0x6FF | R/W | 0 | period[7:0] internal correction period |

## Register

**DISPLAY WRITE BUFFER CONTROL REGISTER – 0X600**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R | 0 | Reserved |
| 3 | R/W | 0 | Freeze<br><br>1 = Enable<br>0 = Disable |
| 2 | R/W | 0 | Buffer Operation<br><br>1 = Enable<br>0 = Disable |
| 1 | R/W | 0 | Horizontal Inversion<br><br>1 = Enable<br>0 = Disable |
| 0 | R/W | 0 | Vertical Inversion<br><br>1 = Enable<br>0 = Disable |

Similar registers are assigned to control Display Write Buffer 2 to 20 using register 0x601 – 0x613 respectively.

**DISPLAY WRITE BUFFER VERTICAL POSITION REGISTER A – 0X614**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | Vertical Position[7:0] |

**DISPLAY WRITE BUFFER VERTICAL POSITION REGISTER B – 0X615**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R | 0 | Reserved |
| 3:0 | R/W | 0 | Vertical Position[11:8] |

Similar registers are assigned to control Display Write Buffer 2 to 20 using register 0x616 – 0x63B respectively.

Note: Vertical position must be even number.

In auto mode, when register live_pb (0x6B6) is set to 0, these registers are for live channels. When live_pb is set to 1, these registers are for PB channels. Register rgb_vpos0_pb to rgb_vpos15_pb are used in auto mode. They share address with rgb_vpos0_live to rgb_vpos15_live. Register rgb_vpos16_pb to rgb_vpos19_pb are used in normal mode.

**DISPLAY WRITE BUFFER VERTICAL SIZE REGISTER A – 0X63C**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | Vertical Size[7:0] |

**DISPLAY WRITE BUFFER VERTICAL SIZE REGISTER B – 0X63D**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0 | Reserved |
| 2:0 | R/W | 0 | Vertical Size[10:8] |

Similar registers are assigned to control Display Write Buffer 2 to 20 using register 0x63E – 0x663 respectively.

**Vertical size must be even number.**

**In auto mode, when register live_pb (0x6B6) is set to 0, these registers are for live channels. When live_pb is set to 1, these registers are for PB channels. Register rgb_vsize0_pb to rgb_vsize15_pb are used in auto mode. They share address with rgb_vsize0_live to rgb_vsize15_live. Register rgb_vsize16_pb to rgb_vsize19_pb are used in normal mode.**

### DISPLAY WRITE BUFFER HORIZONTAL POSITION REGISTER A – 0X664

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | Horizontal Position[7:0]l, unit is 4 pixels |

### DISPLAY WRITE BUFFER HORIZONTAL POSITION REGISTER B – 0X665

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | R | 0 | Reserved |
| 0 | R/W | 0 | Horizontal Position[8] |

**Similar registers are assigned to control Display Write Buffer 2 to 20 using register 0x666 – 0x68B respectively.**

**In auto mode, when register live_pb (0x6B6) is set to 0, these registers are for live channels. When live_pb is set to 1, these registers are for PB channels. Register rgb_hpos0_pb to rgb_hpos15_pb are used in auto mode. They share address with rgb_hpos0_live to rgb_hos15_live. Register rgb_hpos16_pb to rgb_hpos19_pb are used in normal mode.**

### DISPLAY WRITE BUFFER HORIZONTAL SIZE REGISTER A – 0X68C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | Horizontal Size[7:0]l, unit is 4 pixels |

### DISPLAY WRITE BUFFER HORIZONTAL SIZE REGISTER B – 0X68D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | R | 0 | Reserved |
| 0 | R/W | 0 | Horizontal Size[8] |

**Similar registers are assigned to control Display Write Buffer 2 to 20 using register 0x68E – 0x6B3 respectively.**

**In auto mode, when register live_pb (0x6B6) is set to 0, these registers are for live channels. When live_pb is set to 1, these registers are for PB channels. Register rgb_hsize0_pb to rgb_hsize15_pb are used in auto mode. They share address with rgb_hsize0_live to rgb_hos15_live. Register rgb_hsize16_pb to rgb_hsize19_pb are used in normal mode.**

### DISPLAY WRITE BUFFER CONTROL REGISTER 1 – 0X6B4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0 | **Newfld_sel:** New field signal for LCD frame rate adjustment. There are total 16 live channels. Only one channel is selected for LCD frame rate adjustment. |
| 3 | R/W | 0 | **MODE:** This bit set play back channel buffer mode.<br><br>1: auto mode, use channel ID information, can support 16 channels in one PB port.<br>0: normal mode, support only one channel in one PB port |
| 2 | R/W | 0 | **TST_IN_PB:** Test pattern enable for PB channels |
| 1 | R/W | 0 | **PALNT:** Test pattern PAL/NTSC selection.<br><br>0: NTSC<br>1: PAL |
| 0 | R/W | 0 | **TST_IN:** Test pattern enable for live channels |

### TEST PATTERN TIMING REPLACE REGISTER 1 – 0X6B5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **TST_PAT_CH[7:0]:** This register control live channel [7:0]. If the respective bit is set to "1", write process will use internal test pattern timing instead of camera timing. Set 0x6fc.4 = 1 to read. |

### MISC CONTROL REGISTER – 0X6B6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **Line_bigger_en_pb:** This bit cut vertical active signal when input video is crashed |
| 6 | RW | 0 | **Line_bigger_en:** This bit cut vertical active signal when input video is crashed |
| 5 | RW | 0 | **Use_novid:** When enable this bit, no video signal from TW2864 will determine write or not. |
| 4:3 | RW | 0 | **Skip_sel:** Frame drop selection for live videos<br><br>00: no frame drop<br>01: every five frames will drop one frame<br>10: every six frames will drop one frame<br>11: every seven frames will drop one frame |
| 2 | RW | 0 | **Prot_en**<br><br>When this bit turns on, rgb_interface will not write data to address bigger than prot_addr. |
| 1 | RW | 0 | **chid_sel**<br><br>When this bit turn on, channel ID for PB port will be set to 0x0123 for port 0, 0x4567 for port 1, 0x89AB for port 2 and 0xCDEF for port 3. When this bit is 0, channel ID will decode from channel ID area. |
| 0 | R/W | 0 | **Live_Pb Select**<br><br>1: PB channels<br>0: live channels<br>For register hpos0-15, vpos0-15, hsize0-15, vsize0-15, hstart0-15, vstart0-15, they share same address between live channels and PB channels. If this bit is set to low, live channels values can be read or write. If this bit is set to high, PB channels values can be read or write |

### PB FIRST CHANNEL NUMBER REGISTER 1 – 0X6B7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0 | **Pb1_chnum:** This is used in frame interleave mode. It should be set to first frame number in PB port 1 |
| 3:0 | RW | 0 | **Pb0_chnum:** This is used in frame interleave mode. It should be set to first frame number in PB port 0 |

### PB FIRST CHANNEL NUMBER REGISTER 2 – 0X6B8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0 | **Pb3_chnum:** This is used in frame interleave mode. It should be set to first frame number in PB port 3 |
| 3:0 | RW | 0 | **Pb2_chnum:** This is used in frame interleave mode. It should be set to first frame number in PB port 2 |

### PB FIRST CHANNEL NUMBER ENABLE REGISTER – 0X6B9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | **IGNOR_NON-STANDRED[1]**<br>0: if non-standard sequence appears, switch to test pattern timing<br>1: disable the switching, always use original camera timing<br><br>**IGNOR_NON-STANDRED[0]**<br>0: report non-standard video sequence status<br>1: disable the non-standard status report |
| 5 | RW | 0 | **IRQENA_NON-STANDRED:** Non-standard interrupt masking bit<br><br>0: off (masked)<br>1: on |
| 4 | R/W | 0 | **Read out select**<br><br>0: normal<br>1: Read address 0x6f0, 0x6f1 will become the PB correction register |
| 3 | RW | 0 | **Pb3_chnum_en:** Same as channel 0 |
| 2 | RW | 0 | **Pb2_chnum_en:** Same as channel 0 |
| 1 | RW | 0 | **Pb1_chnum_en:** Same as channel 0 |
| 0 | RW | 0 | **Pb0_chnum_en:** this is used in frame interleave mode. First channel number enable<br><br>0: update wr_page any time<br>1: update wr_page when chnum equals to pb0_chnum |

### TEST PATTERN TIMING REPLACE REGISTER 2 – 0X6BA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **TST_PAT_CH[15:8]:** This register control live channel [15:8]. If the respective bit is set to "1", write process will use internal test pattern timing instead of camera timing. Set 0x6fc.4 = 1 to read. |

**PB CHANNEL FREEZE REGISTER 1 – 0X6BB**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | Pb_ch_frez[7:0]<br><br>1: freeze<br>0: normal operation |

**PB CHANNEL FREEZE REGISTER 2 – 0X6BC**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | Pb_ch_frez[15:8] |

**PROTECTION ADDRESS REGISTER 1 – 0X6BD**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | Prot_addr[7:0]<br><br>Protection area. It is 8 bytes unit. RGB interface will not write data to address more than or equal to prot_addr. |

**PROTECTION ADDRESS REGISTER 2 – 0X6BE**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | Prot_addr[15:8] |

**PROTECTION ADDRESS REGISTER 3 – 0X6BF**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5:0 | R/W | 0 | Prot_addr[21:16] |

**Register 0x6bd, 0x6be, 0x6bf is controlled by 0x6fc[7:6]. It can be redefined as external OSG address1 and 2.**

**DISPLAY WRITE BUFFER HORIZONTAL START – 0X6C0**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | Horizontal Start[7:0]<br><br>Image horizontal start position used for cropping. Unit is 4 pixels |

**Similar registers are assigned to control Display Write Buffer 2 to 20 using register 0x6C1 – 0x6D3 respectively.**

**In auto mode, when register live_pb (0x6B6) bit 0 is set to 0, these registers are for live channels. When live_pb is set to 1, these registers are for PB channels. Register rgb_hstart0_pb to rgb_hstart15_pb are used in auto mode. They share address with rgb_hstart0_live to rgb_hos15_live. Register rgb_hstart16_pb to rgb_hstart19_pb are used in normal mode.**

**DISPLAY WRITE BUFFER VERTICAL START – 0X6D4**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | Vertical Start[7:0]<br><br>Image vertical start position used for cropping. |

**Similar registers are assigned to control Display Write Buffer 2 to 20 using register 0x6D5 – 0x6E7 respectively.**

**In auto mode, when register live_pb (0x6B6) is set to 0, these registers are for live channels. When live_pb is set to 1, these registers are for PB channels. Register rgb_vstart0_pb to rgb_vstart15_pb are used in auto mode. They share address with rgb_vstart0_live to rgb_hos15_live. Register rgb_vstart16_pb to rgb_vstart19_pb are used in normal mode. Unit is 2 lines for live channels. Unit can be changed for PB channels. When vstart_unit_pb is set to 1, the unit is 4 lines. When vstart_unit_pb is set to 0, the unit is 2 lines. In auto mode, vstart for PB has different meaning. Vstart0_pb to vstart3_pb are used for PB port 0. They are for 4 positions in Quad mode. Vstart4_pb to vstart7_pb are used for PB port 1. Vstart8_pb to Vstart11_pb are used for PB port 2. Vstart12_pb to vstart15_pb are used for PB port 3.**

### LIVE CHANNEL FRAME RATE CORRECTION CONTROL REGISTER 1 – 0X6E8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | F | **Correction Control[3:0]:** This is for channel 1 |
| 3:0 | RW | F | **Correction Control[3:0]:** This is for channel 0<br><br>Bit 3: 1 = Enable non-standard FRSC correction<br>Bit 2: 1 = Enable non-standard write page correction<br>Bit 1: 1 = Enable external write page correction<br>Bit 0: 1 = Enable internal write page correction |

### LIVE CHANNEL FRAME RATE CORRECTION CONTROL REGISTER 2 – 0X6E9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | F | **Correction Control[3:0]:** This is for channel 3 |
| 3:0 | RW | F | **Correction Control[3:0]:** This is for channel 2 |

### LIVE CHANNEL FRAME RATE CORRECTION CONTROL REGISTER 3 – 0X6EA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | F | **Correction Control[3:0]:** This is for channel 5 |
| 3:0 | RW | F | **Correction Control[3:0]:** This is for channel 4 |

### LIVE CHANNEL FRAME RATE CORRECTION CONTROL REGISTER 4 – 0X6EB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | F | **Correction Control[3:0]:** This is for channel 7 |
| 3:0 | RW | F | **Correction Control[3:0]:** This is for channel 6 |

### LIVE CHANNEL FRAME RATE CORRECTION CONTROL REGISTER 1 – 0X6EC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | F | **Correction Control[3:0]:** This is for channel 9 |
| 3:0 | RW | F | **Correction Control[3:0]:** This is for channel 8 |

### LIVE CHANNEL FRAME RATE CORRECTION CONTROL REGISTER 2 – 0X6ED

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | F | **Correction Control[3:0]:** This is for channel 11 |
| 3:0 | RW | F | **Correction Control[3:0]:** This is for channel 10 |

### LIVE CHANNEL FRAME RATE CORRECTION CONTROL REGISTER 7 – 0X6EE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | F | Correction Control[3:0]: This is for channel 13 |
| 3:0 | RW | F | Correction Control[3:0]: This is for channel 12 |

### LIVE CHANNEL FRAME RATE CORRECTION CONTROL REGISTER 8 – 0X6EF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | F | Correction Control[3:0]: This is for channel 15 |
| 3:0 | RW | F | Correction Control[3:0]: This is for channel 14 |

### PB0 LINE NUMBER STATUS REGISTER 1 – 0X6F0 (READ ONLY)

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RO | - | Linenum_pb0[7:0]: Total line number for PB port 0. |

### PB0 LINE NUMBER STATUS REGISTER 2 – 0X6F1 (READ ONLY)

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RO | 0 | Reserved |
| 3:0 | RO | - | Linenum_pb0[11:8]: Total line number for PB port 0. |

**Register 0x6F2 to 0x6F7 are port 1 to 3, When 0x6b9[4] = 1, 0x6f0 and 0x6f1 is redefine as PB Correction Control Register.**

### PB CHANNEL FRAME RATE CORRECTION CONTROL REGISTER 1 – 0X6F0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0x33 | Correction Control[3:0]: This is for PB channel 1 |
| 3:0 | RW | 0x33 | Correction Control[3:0]: This is for PB channel 0<br><br>Bit 3 = Double save mode control (1 = enable)<br>Bit 2 = Select Top or Bottom field (1 = Bottom)<br>Bit 1,0 (Auto mode)<br>1X = Set compatible mode<br>01 = Correction enable<br>00 = Free running<br><br>Bit1,0 (Normal mode)<br>0X = Set compatible mode<br>11 = Correction enable<br>10 = Free running |

### PB CHANNEL FRAME RATE CORRECTION CONTROL REGISTER 2 – 0X6F1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0x33 | Correction Control[3:0]: This is for PB channel 3 |
| 3:0 | RW | 0x33 | Correction Control[3:0]: This is for PB channel 2 |

### PB VALID STATUS REGISTER – 0X6F8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | **Write this bit to clear bit [3:0]** |
| 6:4 | RO | 0 | **Reserved** |
| 3 | RO | 0 | **Pb3_valid**: Valid signal bit for PB channel 3 |
| 2 | RO | 0 | **Pb2_valid**: Valid signal bit for PB channel 2 |
| 1 | RO | 0 | **Pb1_valid**: Valid signal bit for PB channel 1 |
| 0 | RO | 0 | **Pb0_valid:** This bit will be high if there is valid field signal coming. This bit will be clear when write to this register. |

### LIVE CHANNEL EXTERNAL CORRECTION PERIOD REGISTER – 0X6F9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x25 | **period2[7:0]:** Control the write page self checking period, the unit is the LCD Vsync. |

### PB CHANNEL IGNORE AND FREE RUNNING REGISTER – 0X6FA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **SEL_IG3:** Ignore control of channel 3 |
| 6 | RW | 0 | **SEL_IG2:** Ignore control of channel 2 |
| 5 | RW | 0 | **SEL_IG1:** Ignore control of channel 1 |
| 4 | RW | 0 | **SEL_IG0:** Ignore control of channel 0<br><br>Set this bit to "1" and PB write buffer will use ignore bits in digital channel ID to determine which channel or channels should be disabled. |
| 3 | RW | 0 | **Pb_free_3:** Free running control of channel 3 |
| 2 | RW | 0 | **Pb_free_2:** Free running control of channel 2 |
| 1 | RW | 0 | **Pb_free_1:** Free running control of channel 1 |
| 0 | RW | 0 | **Pb_free_0:** Free running control of channel 0<br><br>Set this bit to "1", PB video will be free running, write_page will automatically plus 1, frame rate control has no use. |

### FIFO COUNT LIMIT REGISTER – 0X6FB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 1 | **Fifo_cnt_limit:** RGB FIFO count limit. |
| 3 | RW | 1 | **Fld_update:** In interlaced PB mode, when this bit set to 1, frame buffer only update when last field is even field and current field is odd filed. |
| 2 | RW | 1 | **Fld_sel_fix:** In progressive PB mode, when fld_sel_en set to 0, fld signal is fixed to setting value: fld_sel_fix. |
| 1 | RW | 0 | **Fld_sel_en:** In progressive PB mode, when this bit set to 0, fld signal is fixed to setting valued: fld_sel_fix. When this bit set to 1, fld signal is the detected fld signal. |
| 0 | RW | 0 | **Vstart_unit_pb:** Unit for vstart for pb channels<br><br>0: unit is 2 line<br>1: unit is 4 line, used for BT1120 or 4D1 case |

### VERTICAL INVERTER METHOD SELECTION – 0X6FC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | **Address Select**<br><br>00 = Protection address register<br>01 = External OSG buffer address 1<br>10 = External OSG buffer address 2<br>11 = Reserved |
| 5 | RW | 0 | **1 = Force live channel to use old frame rate algorithm** |
| 4 | RW | 1 | **seld:** 1 = change the read content of 0x6b5 and 0x6b9 |
| 3 | R | 0 | **Reserved** |
| 2 | RW | 0 | **1 = PB and Live with the same priority**<br>**0 = Force display arbiter adopt PB has no round robin** |
| 1 | RW | 0 | **Hstart_unit_pb:** Unit for hstart for pb channels<br><br>0: unit is 4 pixels<br>1: unit is 8 pixels, used for BT1120 or 4D1 case |
| 0 | RW | 0 | **Vinv_sel:** Vertical invert method selection.<br><br>0: odd field will write to even field<br>1: odd field will write to odd field |

### AUTO MODE PB CHANNEL ENABLE REGISTER 1 – 0X6FD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | **PB_CH_EN[7:0]:** Control auto mode PB channel 7 - 0<br><br>1 = Playback channel enable |

### AUTO MODE PB CHANNEL ENABLE REGISTER 2 – 0X6FE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | **PB_CH_EN[15:8]:** Control auto mode PB channel 15 - 8<br><br>1 = Playback channel enable |

### LIVE CHANNEL INTERNAL CORRECTION PERIOD REGISTER – 0X6FF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x5 | **period[7:0]:** Control the write page self checking period, the unit is the incoming channel Vsync. |

# DRAM Control Unit (DCU)

## Introduction

**The TW2880's DRAM controller unit provides the necessary bridging for the display path and recording path functional units to get access to the off-chip SDRAM. The recording unit is covered in the recording section of this datasheet. Here, we mainly discuss the memory map and the clients in the display unit. However, the basic functionalities are the same for both controllers. The current display clients are: CPU, RGBW, LCD, Dual monitor and OSG. In these DRAM clients, only CPU and RGBW have the write requirement. The detailed DRAM data bus flow is shown in the chip block diagram.**

## Memory Organization

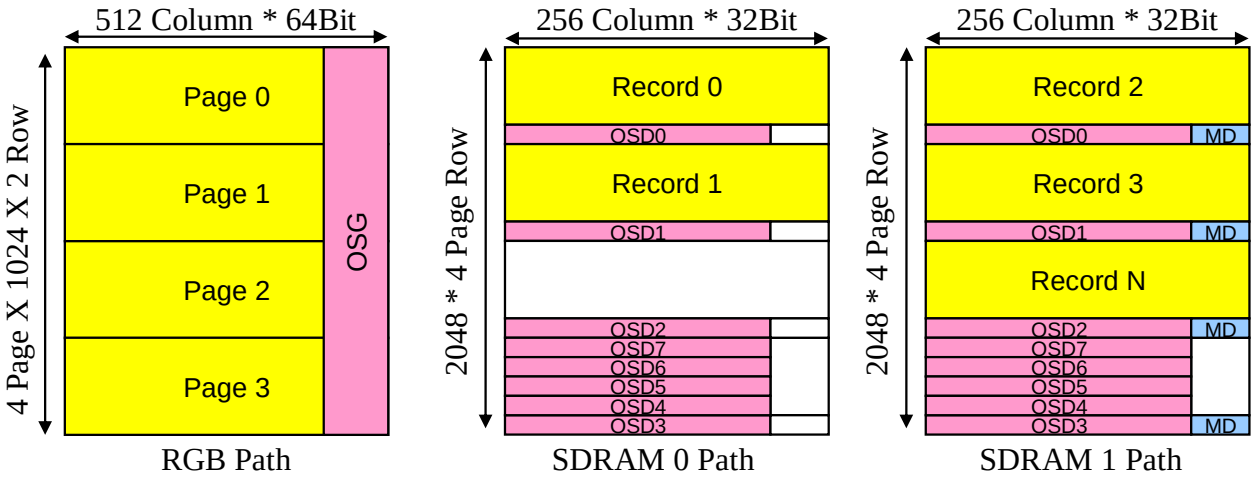
TW2880's DCU supports SDRAM subsystem size up to 128 Mbytes by using four 64, 128 or 256 Mbit generation DRAMs with 16 or 32 wide bus. The overall memory bus is 64 bit wide. Although using 32 bit bus is possible, the number of incoming channels to display has to be minimized for reduced memory bandwidth. The operating speed is ranged from 166 MHz to 182 MHz. The sustained bandwidth when running at 166 MHz is 166 * 8 * 0.8 = 1062 MB/sec.

The internal linear address (LA) is 24 bit and classified as:

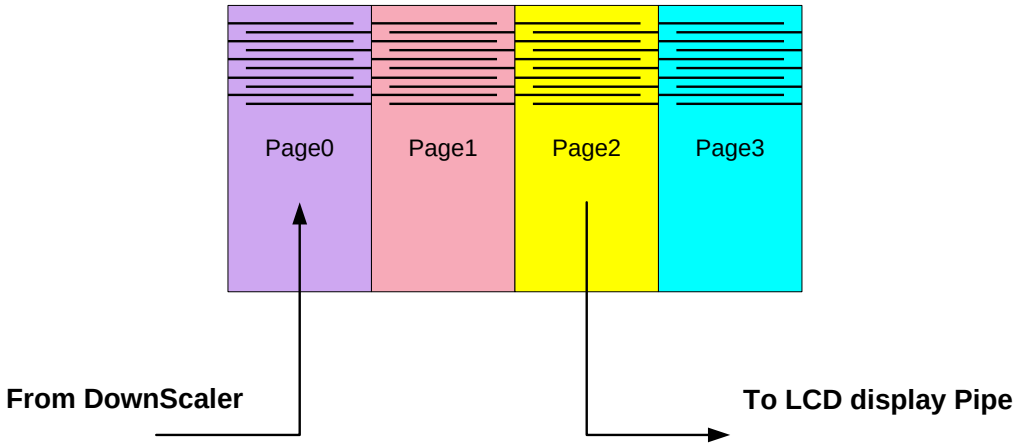
$2^{10}$ (Row) x $2^{9}$ (Column) x $2^{2}$ (Page) x $2^{3}$ (64 bit) = $2^{24}$

The next table shows the resolutions and memory required.

| Item | | Total: 16 MB installed |
|---|---|---|
| Video | 1280 x 1024 x 2 x 4 = 10.48 MB | |
| OSG | 1280 x 1024 x 2 = 2.62 MB | |
| MISC | 3MB | |
| | | Total: 32 MB installed |
| Video | 1680 x 1050 x 2 x 4 = 14.08 MB | |
| OSG | 1680 x 1050 x 2 = 3.52 MB | |
| MISC | | |



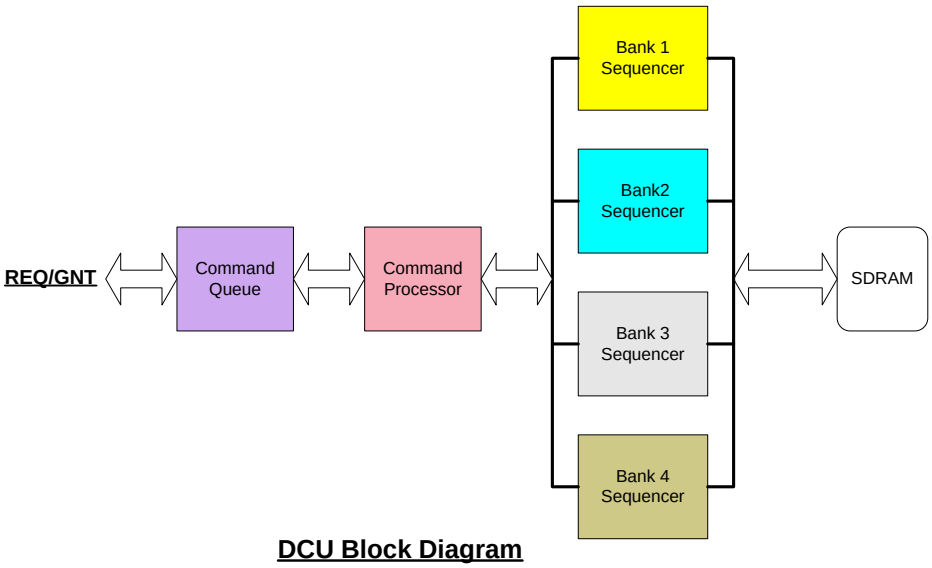
## Display Plane and DRAM Arrangement




## Channel Video Storage Diagram

To prevent video tearing and help simplifying the incoming video field management, TW2880's display processor is storing four incoming video frame (8 fields) into the DRAM. A frame rate synchronization circuit (FRSC) will guarantee there are two frames difference between active display frame and active storing field. Because the monitoring process of the channel can go active at any time, FRSC is used to direct active channel dump image at the right place in a video frame while display process is fetching pixels two frames away. If the frame rate of the display pipe is higher than the incoming video pipe, frame repeating process is employed to make sure the two frame boundary is maintained.

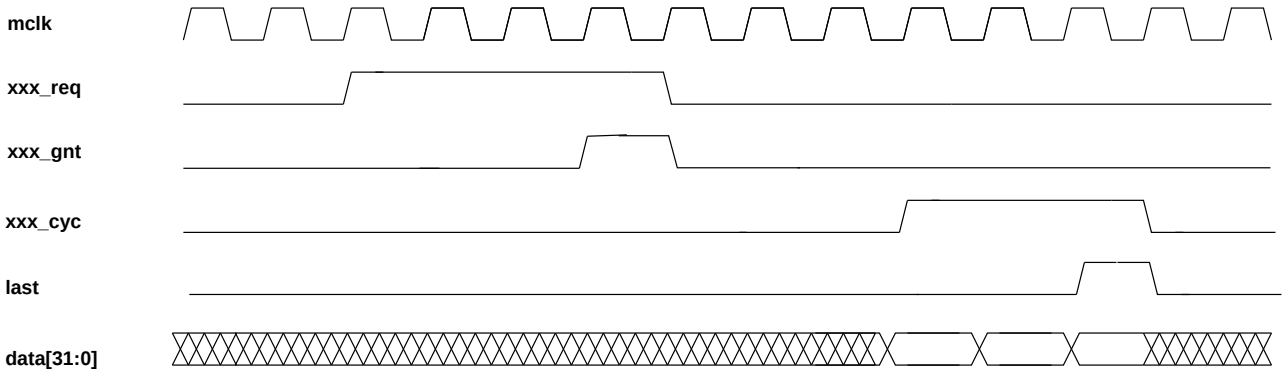
The four stored frames served also served another feature for 3D de-interlacing. If 3D de-interlacing is employed then there will be five pixel read stream from the neighboring frame to determine the merging process. More detailed information is in next chapter.




DCU Block Diagram

## DCU Handshake

DCU consist of four major blocks: command queue, command processor, IO register and data collect state machine, bank sequencer. Command queue is the place stores all the pending DRAM transactions. Its buffer has an 8 level deep FIFO. The transaction protocol used with each functional unit is the dbus. dbus is a synchronous bus based on simple request / grant relationship. The following diagram is a typical transaction (read or write).




For requesting a DRAM transaction service, functional unit first write burst length, starting address, read/write information into its SDRAM information bus and raise its request line to DCU. Upon receiving the request, DCU will have the data collect state machine analyze the data and put them into request FIFO. Multiple requests are supported as long as there is no pending request of others. After this action is finished, a grant signal will be sent to the functional unit to acknowledge the receipt the request. In normal circumstances, DCU should send out grant signal to functional units very quickly unless there is some long transaction going on. To finish the cycle, the functional unit should prepare its data access FIFO and wait for its xxx_cycle line to go high. Exchange of data (read or write) will continue as long as the cycle line is high and using the system clock as a counter. Read data bus is shared and write data bus is multiplexed using cycle arbiter. The last cycle flag is sent out in the last data phase.

If several DRAM requests are received simultaneously, an arbiter will arbitrate the incoming requests on a first-come-first-served basis and round-robin scheme is used for arbitration to ensure equal share of bandwidth to each unit. This means successive DRAM requests will get masked if other clients have requests. The masking will automatically disappear once other request is served.

It also supports several DRAM parameters including $t_{RAS}$, $t_{RCD}$ for different DRAM venders.

## TW2880 DRAM Configuration Options

In forming the TW2880's DRAM subsystem, users have many options as TW2880 supports SDRAM density from 64Mbit, 128 Mbit to 256 Mbit. Normally we recommend our customers using 64 bit memory bus to get the most features but TW2880 also support 32 bit memory bus as well to save cost. The speed grade of the SDRAM chip is also very important as it affects how high the memory clock can go and thus it affects the memory bandwidth. Because the TW2880's memory PLL can generate many values so in the following tables I am only using 133 MHz, 147 MHz and 166 MHz as examples. The user should use these tables as a guide and figure out the features and options for a particular kind of memory.

Because memory venders provide many selections in choosing memory like densities, data width and speed grade, customer can build their memory sub-system based on their needs.

**DISPLAY PATH, 64 BIT, 256 MBIT, 8MX32X2 (SUGGESTED)**

| | 133 MHz | 147 MHz | 166 MHz |
|---|---|---|---|
| 1080p | Supported | Supported | Supported |
| 2D / 3D | Supported | Supported | Supported |
| OSG | Not supported | Limited functions | Supported |
| Dual display | Not supported | Supported | Supported |

**DISPLAY PATH, 64 BIT, 128 MBIT, 4MX32X2 (SUGGESTED)**

| | 133 MHz | 147 MHz | 166 MHz |
|---|---|---|---|
| 1080p | Supported | Supported | Supported |
| 2D / 3D | Supported | Supported | Supported |
| OSG | Not supported | Limited functions | Supported |
| Dual display | Not supported | Limited functions | Supported |

**DISPLAY PATH, 64 BIT, 64 MBIT, 2MX32X2**

| | 133 MHz | 147 MHz | 166 MHz |
|---|---|---|---|
| 1080p | Not supported | Not supported | Not supported |
| 2D / 3D | 2D | Supported | Supported |
| OSG | Not supported | Limited functions | Limited functions |
| Dual display | Not supported | Limited functions | Limited functions |

**DISPLAY PATH, 32 BIT, 128 MBIT, 4MX32X1**

| | 133 MHz | 147 MHz | 166 MHz |
|---|---|---|---|
| 1080p | Limited functions | Supported | Supported |
| 2D / 3D | Not supported | Only 2D | Only 2D |
| OSG | Not supported | Limited functions | Limited functions |
| Dual display | Not supported | Not supported | Supported |

**DISPLAY PATH, 32 BIT, 64 MBIT, 2MX32X1 (NOT RECOMMENDED)**

| | 133 MHz | 147 MHz | 166 MHz |
|---|---|---|---|
| 1080p | Limited functions | Limited functions | Limited functions |
| 2D / 3D | Only 2D | Only 2D | Only 2D |
| OSG | Not supported | Not supported | Not supported |
| Dual display | Not supported | Not supported | Not supported |

**RECORDING PATH, 64 BIT, 256 MBIT, 8MX32X2 (SUGGESTED)**

| | 133 MHz | 147 MHz | 166 MHz |
|---|---|---|---|
| D1 FMI num# | 16 channel | 16 real time | 16 real time |
| SPOT | Not supported | Not supported | 1 |
| Network port | Not supported | Not supported | Supported |
| 4D1 / 6VGA | Not supported | Limited functions | Supported |

**RECORDING PATH, 64 BIT, 128 MBIT, 4MX32X2**

| | 133 MHz | 147 MHz | 166 MHz |
|---|---|---|---|
| D1 FMI num# | 16 channel (field) | 16 real time (field) | 16 real time (field) |
| SPOT | Not supported | Not supported | 1 |
| Network port | Not supported | Not supported | Supported |
| 4D1 / 6VGA | Not supported | Limited functions | Supported |

**RECORDING PATH, 64 BIT, 64 MBIT, 2MX32X2**

| | 133 MHz | 147 MHz | 166 MHz |
|---|---|---|---|
| D1 FMI num# | 8 channel (field) | 8 real time (field) | 8 real time (field) |
| SPOT | Not supported | Not supported | 1 |
| Network port | Not supported | Not supported | Supported |
| 4D1 / 6VGA | Not supported | Limited functions | Supported |

**RECORDING PATH, 32 BIT, 256 MBIT, 8MX32X1**

| | 133 MHz | 147 MHz | 166 MHz |
|---|---|---|---|
| D1 FMI num# | Not supported | Not supported | Not supported |
| SPOT | Not supported | Not supported | Not supported |
| Network port | Not supported | Not supported | Not supported |
| 4D1 / 6VGA | Not supported | Not supported | Not supported |

**RECORDING PATH, 32 BIT, 128 MBIT, 4MX32X1**

| | 133 MHz | 147 MHz | 166 MHz |
|---|---|---|---|
| D1 FMI num# | 8 channel (field) | 8 real time (field) | 8 real time (field) |
| SPOT | Not supported | 1 | 1 |
| Network port | Not supported | Not supported | Not supported |
| 4D1 / 6VGA | Not supported | Not supported | Limited functions |

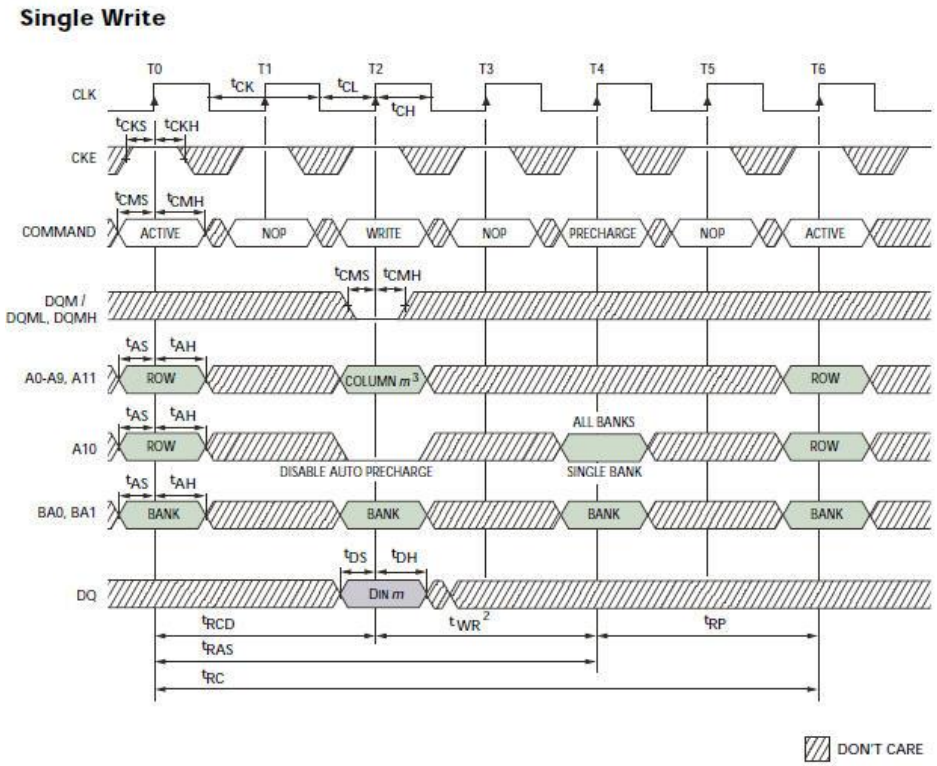
**RECORDING PATH, 32 BIT, 64 MBIT, 2MX32X1**

| | 133 MHz | 147 MHz | 166 MHz |
|---|---|---|---|
| D1 FMI num# | 4 channel (field) | 4 real time (field) | 4 real time (field) |
| SPOT | Not supported | Not supported | 1 |
| Network port | Not supported | Not supported | Not supported |
| 4D1 / 6VGA | Not supported | Not supported | Limited functions |

## TW2880 DRAM Parameter Adjustments

**The DRAM controller in TW2880 is designed to support different speed grades of DRAM from different manufacturers. Normally we suggest user use higher speed grade of DRAM because it will generate more bandwidth for the entire system. The SDRAM devices can be used in TW2880 based system are: -5 (200 MHz), -6 (166 MHz), -7 (142 MHz) and -7.5 (133 MHz). TW2880 always use CL-3 setting and do not use DQM or CS. So these signals can be static in the schematics. To determine whether a certain DRAM speed grade is suitable for a particular TW2880 system, we need to pay attention to $t_{RAS}$, $t_{RCD}$, $t_{RC}$, $t_{WR}$ , $t_{RP}$, $t_{WS}$ and $t_{WH}$ for different DRAM vendors.**

| PARAMETERS | EXPLANATION | |
|---|---|---|
| $t_{RAS}$ | RAS to pre-charge time | 6T, 7T, 8T |
| $t_{RCD}$ | RAS to CAS time | 3T, 4T |
| $t_{RC}$ | Access cycle time | 9T, 10T, 11T |
| $t_{WR}$ | Write turnaround time | 2T |
| $t_{RP}$ | Bank address pre-charge time | 3T, 4T |
| $t_{DS}$ | Data setup time, same as address | 1 ns |
| $t_{DH}$ | Data hold time, same as address | 1 ns |




### COMMAND AND ADDRESS TIMING

**The transaction illustrated above is a single write operation. The basic definitions of each parameter can be found in the SDRAM data sheet. I choose the values from a typical manufacturer. Other manufacturer's value can also be used in this calculation. TW2880 has a fixed timing for each operation. $t_{RAS}$ is 6T (clock period) minimum, $t_{RCD}$ is 3T minimum, $t_{RC}$ is 9T minimum, $t_{WR}$ is 2T and $t_{RP}$ is 3T. However if you set DRAM control register 0x211, 0x212**

**bit 3 to 0, you will get one more T for $t_{RAS}$, $t_{RCD}$ and $t_{RC}$. Setting DRAM control register 0x211, 0x212 bit 6 to 1, you will get another T for $t_{Rp}$ and $t_{RC}$. The table below also shows the time delta variations in each clock setting.**

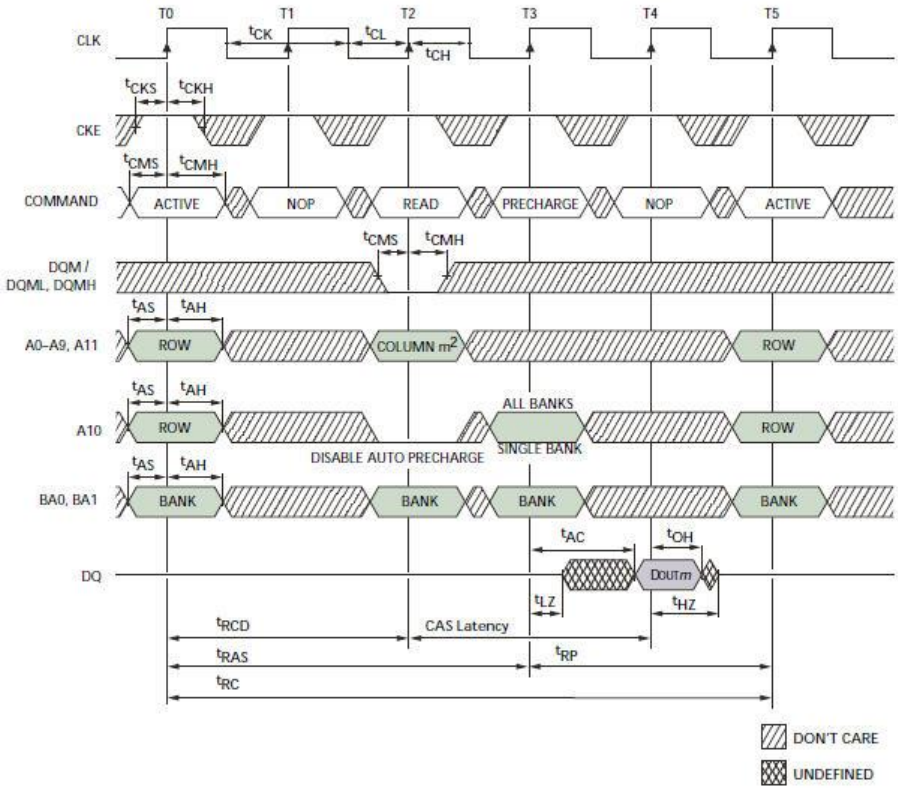
**Based on this we can calculate the five important constants in the DRAM operations:**

| Name | -7.5 | -7 | -6 | -5 | 133-n | 133-e | 142-n | 142-e | 162-n | 162-e |
|---|---|---|---|---|---|---|---|---|---|---|
| $t_{RAS}$ | 45 | 42 | 42 | 40 | 45 | 52.5 | 42 | 49 | 37.02 | 43.19 |
| $t_{RCD}$ | 20 | 18 | 18 | 15 | 22.5 | 30 | 21 | 28 | 18.51 | 24.68 |
| $t_{RC}$ | 65 | 60 | 60 | 55 | 67.5 | 75 | 63 | 70 | 55.53 | 61.7 |
| $t_{WR}$ | 2T | 2T | 2T | 2T | 2T | | 2T | | 2T | |
| $t_{RP}$ | 20 | 18 | 18 | 15 | 22.5 | | 21 | | 18.51 | |

**After this calculation user can make sure the device they choose are suitable running at certain frequencies. For example, a system run at 142 MHz can choose -7, -6 or -5 device.**

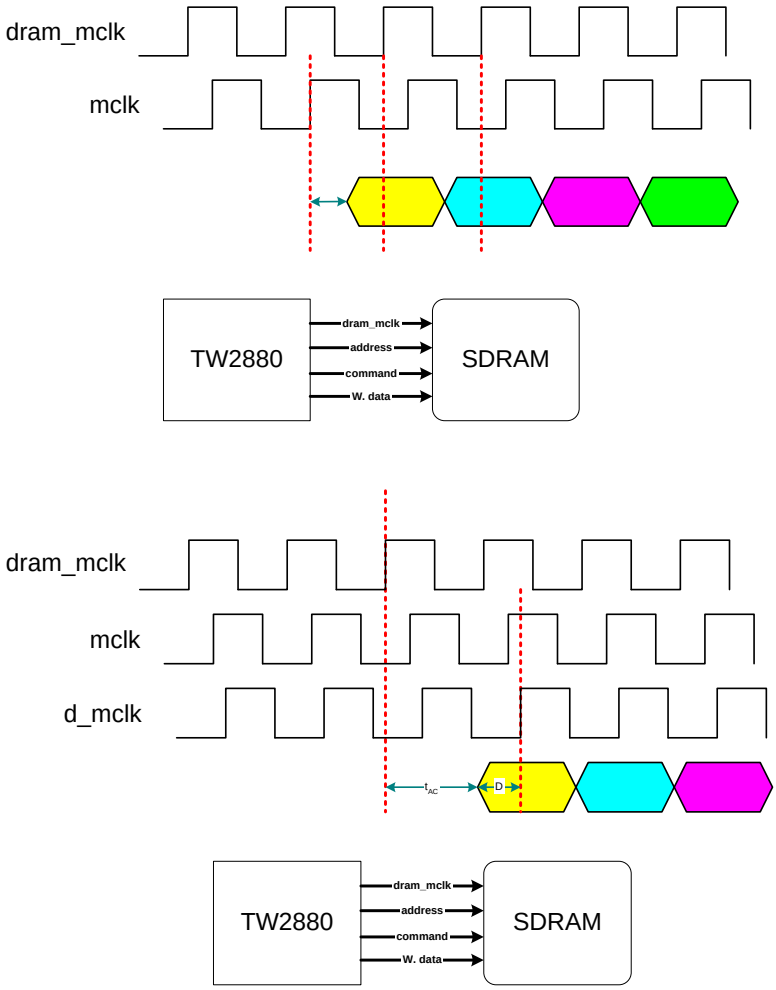
**SDRAM transaction always starts with sending command and later on address to the devices no matter it is a read or writes operation. For write operation it will followed by the data phase which has the same setup / hold requirements as the address. Page operation will make the timing parameters unimportant except $t_{RCD}$. Read operation is very similar to the write operation in command and address phase. However, the data phase is quite different and is discussed in the next section.**

**Single READ – Without Auto Precharge**



**DATA READ / WRITE TIMING**

**The above discussion is for the command and address phase operations. In TW2880, we actually squeeze the timing on this side to help speed up the read operation. The concept is illustrated in the below diagram. dram_mclk is actual clock used by the output DRAM device. d_mclk is a variation of the memory clock derived from inside. The timing relationship is controlled by register 0x220 bit 5:4 (display) and bit 7:6 (record). To get optimal read / write operations, we usually let dram_clock's edge a little ahead than the inside memory clock. Any command or address will be sent from the internal register clocked at mclk, travel through I/O pad and PCB traces and arrive at DRAM device but latched by slightly ahead dram_clk. As a result of this setting, the actual write propagation time is squeezed a little because we did not use the whole cycle. This is the case because we want to extend the read cycle timing as illustrated in the next drawing.**




**The following discussion is based on the diagram above. In this case, TW2880 is trying to do a read operation. The command phase and the address phase is the same as the write operation. In terms of read data travel path, data is clocked by dram_clock and after data access time $t_{AC}$ will appear in the device's pad and after some delay in PCB traces and TW2880's I/O pad, data will be to the DCU's read latch and get latched by a delayed version of memory clock to buy more time. By using this technique, we can extend the read clock to the maximum.**

**The delay of d_mclk is adjustable by setting register 0x220 bit 5:4 (display) and bit 7:6 (record). The counting logic is controlled by the VCO clock in PLL, so if selecting divided by four option when you make the clock you will have four**

selections, 8 selections will choosing divided by 8 option. So together with the dram_mclk option we will have at least 16 steps for adjusting $t_{DS}$ , $t_{DH}$ in the data side.

**REFRESH OPTIONS**

Refresh in TW2880 is controlled by register 0x211, 0x212 bit 5 and bit 4. Because read is the best form of refresh, user can set bit 5 to 1 to disable the refresh operation to save some band width. However, this is not possible if off screen memory is used. Bit 4 is used to choose 4K or 8K refresh option normally found in the DRAM manufacturers.

# Frame Rate Control Unit

## Introduction

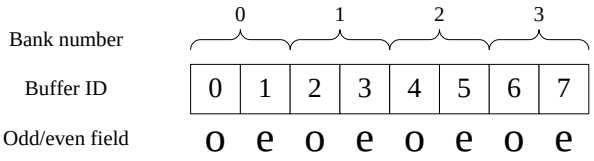
Frame rate control module controls the relationship between read bank pointer for the display processor and the write bank pointer for the 20 input channels. Since the display frequency and the channel buffer write frequency is quite different, we need to make some adjustments to the pointers such that read and write operations do not happen in the same DRAM bank.

Generally speaking, if the reading operation is faster than writing operation, TW2880 will repeat the read bank image from time to time to avoid video tearing. On the other hand, if writing operation is faster than reading operation, TW2880 will skip frame by forcing the two frames get written into same bank. That means one underplayed video frame will be overwritten and not display. This mechanism is used in Weave de-interlacing method.

When 3D de-interlacing method is chosen, the four banks of memory will be treated as 8 fields. Five fields are needed at the same time to generate one field. TW2880 will rotate the banks and keep track of incoming fields and not to overwrite will the existing fields at the same time. Because the extra bandwidth requirements for the 3D de-interlacing, user has make sure the resolution and DRAM clock combination in order to get a display free from artifact. TW2880 also has 2D de-interlacing engine available, it only need one extra read at the same time. Overall, Weave de-interlacing method will be the bandwidth conscious method to display a progressive image.

## SDRAM Allocation

In TW2880, fields are saved in banks. Field 0 and field 1 are saved in bank 0, field 2 and 3 are saved in bank 1, field 4 and 5 are saved in bank 2, and field 6 and field 7 are saved in bank 3. And then wrap around.




Bank number is from 0 to 3. Buffer ID number is from 0 to 7.

## Read Buffer ID Calculation

There are three read out method: 3D de-interlacing, 2D de-interlacing, and the Weave de-interlacing method. If 3D de-interlacing is enabled, 5 pixels of data from 5 adjacent fields are needed to generate one pixel. For example, if buffer 0 is current display field, buffer 4, 5, 6 and 7 are needed to be read out at the same time. At this time only buffer 1 and 2 and 3 can be updated by write units. If 2D de-interlacing is enabled, only one field is needed to be read out. In this case, read out buffer ID is same as 3D de-interlacing.

Read out buffer ID is updated at the end of display frame. To be able to advance to the next read bank all the next read buffer IDs should not collide with their current write bank IDs. If not the read ID will stay the same for the new frame so old frame data is sent out repeated to the outside again. A frame repeating signal is available to indicate this action.

If Weave de-interlacing method is used, two fields in one bank are needed to be read out. In this case, bank number is current bank minus 1.

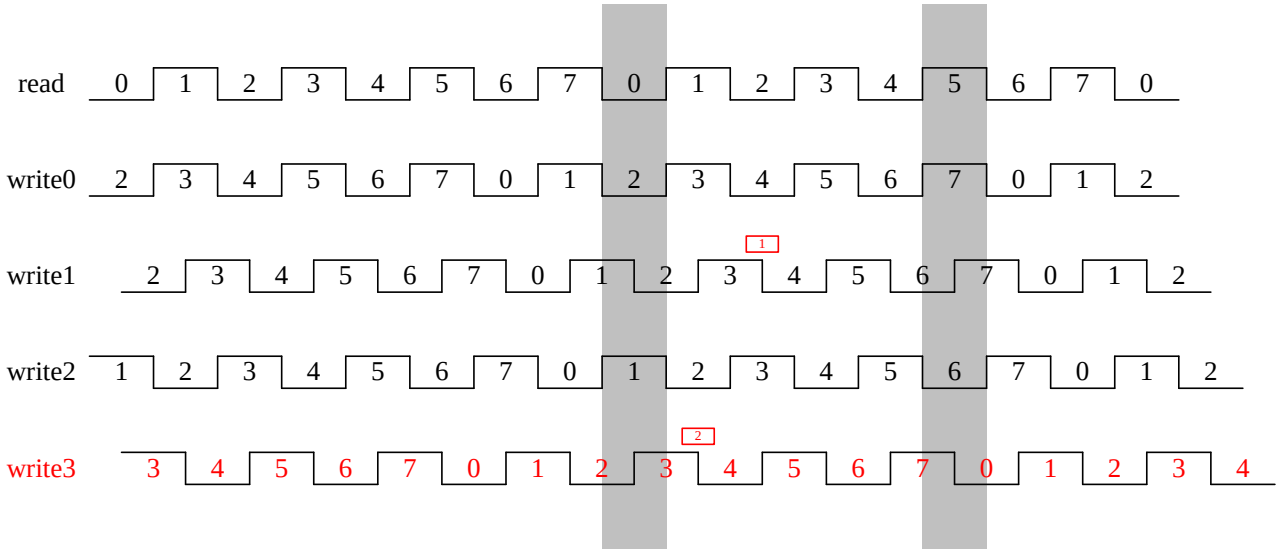
bank_sel = rd_buf_id / 2 -1;

If we use rd_buf_id / 2, read out field may be conflict with write field. For example, if rd_buffer_id is 0, then buffer 0 and 1 will be read out. But buffer 1 may be written at the same time.

## Write Buffer ID Calculation

For write, if frequency is fast, the previous frame will be overwritten. For example, if current write frame is in bank 0, that means odd field is written to buffer 0, even field is written to buffer 1. If next written bank is still bank 0, then next odd field is still written to buffer 0 and even field is still written to buffer 1. Write buffer is determined by write bank and field information. Write bank is always not less than previous one. But write buffer ID can be less than 1 then previous one. Write page number is output from frame rate control module.

There are totally 20 live video write units in TW2880. Because the incoming timing for each channel is totally asynchronous, we need to consider all the possible timing combinations. The following diagram shows four different write timing. In here, low level represents odd field and high level means even filed. The number in the sequence means buffer ID.




In this diagram, when read buffer ID equals to 0, write buffer ID could be 1 and 2 and 3. When read buffer ID is 5, write buffer could be 6 and 7 and 0. Write buffer will always get update in the beginning of odd field. So when we reading one buffer, there are altogether three buffers are being updated by write units.

From this timing diagram, we can find next write page is current read buffer plus 3 then divided by 2. Use write1 as first example, in position 1, write page need to be updated. Current read buffer ID is 2 and then next write buffer ID is 2+3=5. Buffer ID 5 is in bank 2, so next write page is 2. That channel will write buffer 4 and 5. Use write2 as second example, in position 2, write page need to be updated. Current read buffer ID is 1, then next write buffer ID is 1+3=4. Buffer ID 4 is in bank 2 so next write page is 2. That channel will write buffer 4 and 5.

In conclusion, we can use the following formulas:

wr_buf_id_next = rd_buf_id + 3

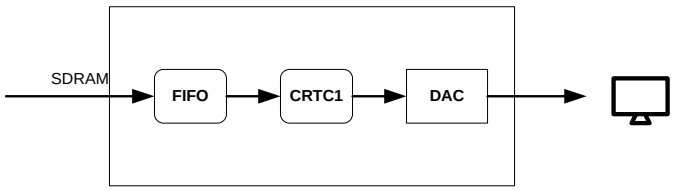
wr_page = wr_buf_id_next / 2.

If write operation is faster, that means read operation is slower so read buffer ID will keep same. From the last section we know the suggested write page number will be the same.

# VGA Controller and Video interface

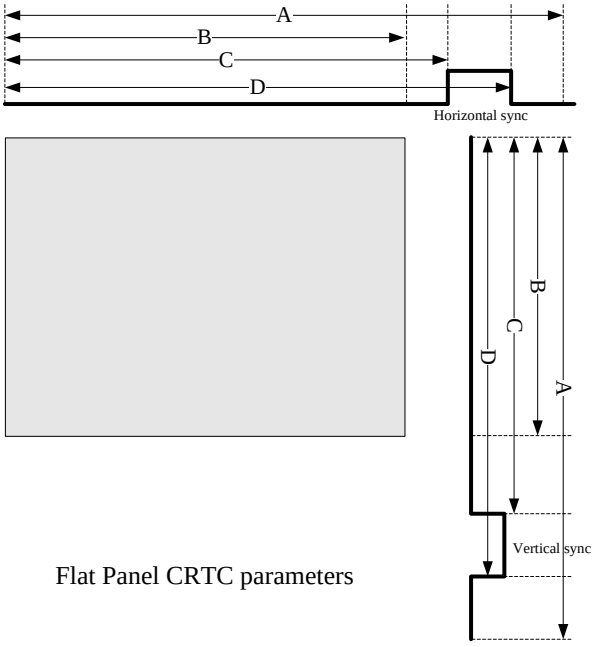
## Introduction

**TW2880 has glue-less video output interface capable of driving VGA-compatible display devices with screen size up to 1920x1080 at 60 Hz. As shown in the next diagram, display data is pulled from the SDRAM and sent to the VGA device through three 10 bit DACs according to the VGA timing. The display path used here is just an example, the detailed explanation of the display plane is in the next section.**




TW2880 Display Controller

**Display controller includes Frame buffer controller, CRT controller and DAC for the progressive data stream. Based on the display requirements, Data FIFO will raise request through DBUS and get display data in advance. When the display is active, the data will be sent to display device and a series of requests will go into DCU. The FIFOs are designed to fetch display data based on the display pattern.**




Flat Panel CRTC parameters

**The frame display size is determined by horizontal total register (unit in pixel clock) and vertical total register (unit in scan line). Display end register is used to determine whether they are in active display area and if not data stream request to the FIFO can be turned off. Synchronization pulses are used to generate frame pulse and line pulse to the panel. The sync pulse start time and end time can adjust to fit different panels.**

## Features

- Support LCD monitor and HDMI TV
- Support interlaced mode and progressive mode for HDMI TV
- Resolution up to 1080p
- Support 2D/3D de-interlace
- Support upscale for video
- Support edge enhancement
- Support 16 live channel and 16 play back channel windows
- Support 4 Privacy windows / per channel
- Support simple OSD for 32 channels
- Support 16 motion box
- Support 8 single box
- Support 3 layer OSG, each layer has 8 sub-window
- Support upscale for OSG
- Support 2 layer mouse cursors
- Support up to 16 mouse shapes in SDRAM
- Support digital RGB gain
- Support CVBS output using dual monitor's TV encoder

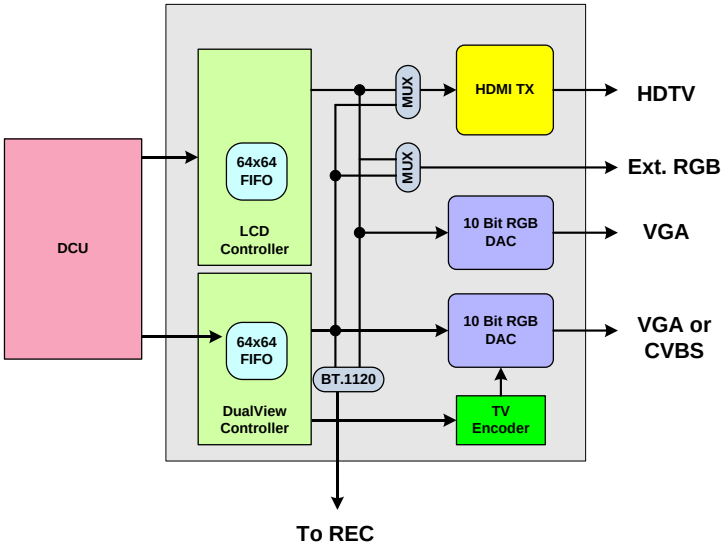
## Frame Buffer Controller

Frame buffer controller is the display data agent responsible for raising the request to the DCU and getting the display data and put into the FIFO. The frame buffer is a virtual buffer as it is actually a piece of DRAM address space and the fetch locations are controlled by the display buffer starting address. For progressive device the address counter will get reset when vertical end is reached. A reset in display buffer address occurs when a sequence of odd / even fields reach the vertical ends. At the end of the horizontal line CPU need to pad zero if the data is not aligned to the DWORD boundary. Line end is from horizontal display end.

## Output Interface

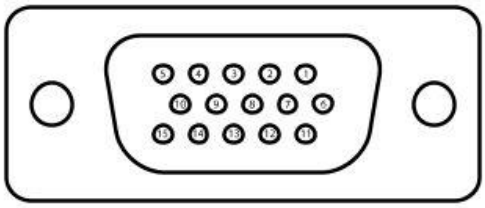
TW2880 supports three display output interface formats on three different ports. These ports can be turned on and off independently. The main output port supports standard VGA socket (HD15) or the HDMI format from XGA to Full HD resolutions. And HDMI can also support interlaced video format such as 1080i. As the DVI-D port is compatible with HDMI port, user can choose to layout a DVI head in the board to support digital RGB LCD monitors.

Another output is the 24 bit RGB output port and the pins are shared with live video inputs. To use this output, incoming video decoder has to run at 108 MHz mode and user needs to turn on live video pin function control. The DualView port will be covered in next section and is not mentioned here anymore.




## VGA Interface

**The diagram on the right shows a VGA pin out. TW2880 uses R, G, B, Hsync and Vsync to drive video. Pin 10, 12, and 15 is used to support VESA E-DDC.**

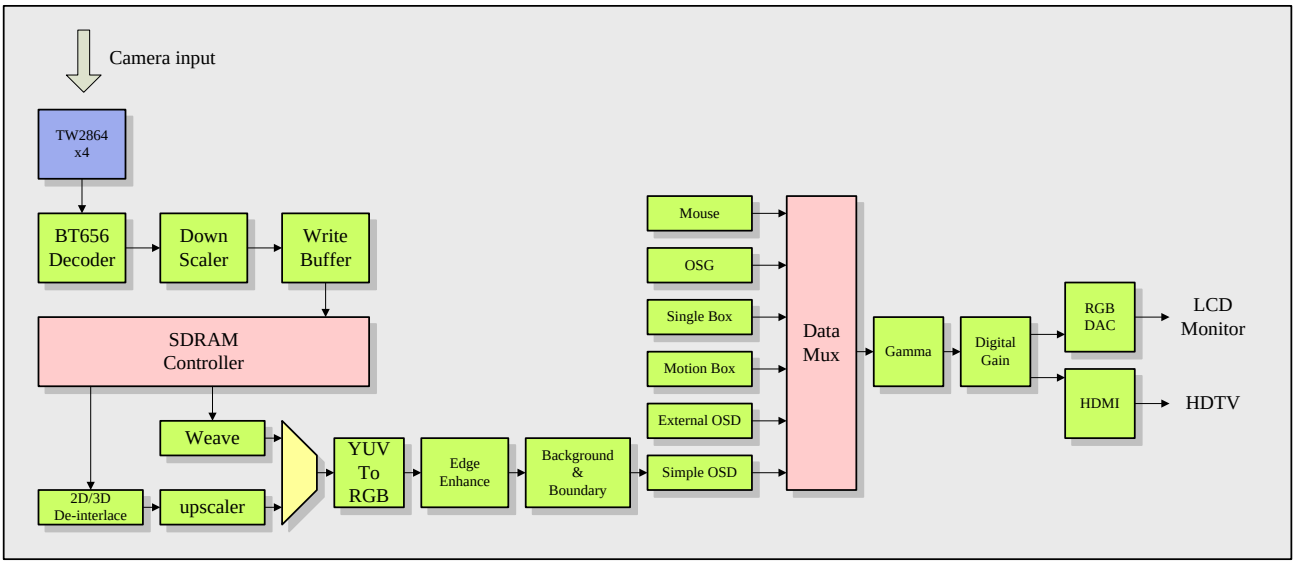


**A female DE15 socket (video card side).**

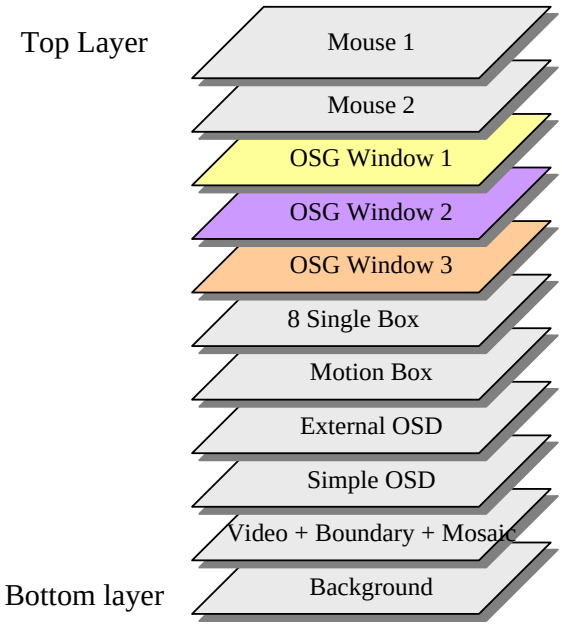
| | | |
|---|---|---|
| Pin 1 | RED | Red video |
| Pin 2 | GREEN | Green video |
| Pin 3 | BLUE | Blue video |
| Pin 4 | N/C | Not connected |
| Pin 5 | GND | Ground (Hsync) |
| Pin 6 | RED_RTN | Red return |
| Pin 7 | GREEN_RTN | Green return |
| Pin 8 | BLUE_RTN | Blue return |
| Pin 9 | SENSE | +5 V DC from gfx adapter |
| Pin 10 | GND | Ground (VSync, DDC) |
| Pin 11 | N/C | Monitor ID |
| Pin 12 | SDA | I²C data |
| Pin 13 | HSync | Horizontal sync |
| Pin 14 | VSync | Vertical sync |
| Pin 15 | SCL | I²C clock |

## Block Diagram

**There are ten display planes in the TW2880 display subsystem. There are, from the highest precedence to lowest: mouse, text cursor, OSG, single box, MD box, external OSD, simple OSD, live display, boundary and background. We have detailed description of each plane in the subsequent sections. The live display windows and background are compared and displayed on a pixel by pixel basis.**




## Display Layers



## Display Resolutions

| RESOLUTION | VCLK | HTT | HDE | HS | HSPW | HPOL | VTT | VDE | VS | VSPW | VSPOL |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1280x1024 | 108 | 1688 | 1280 | 48 | 112 | 1 | 1066 | 1024 | 1 | 3 | 1 |
| 1440x900 | 106.5 | 1904 | 1440 | 80 | 152 | 0 | 934 | 900 | 3 | 6 | 1 |
| 1680x1050r | 119 | 1840 | 1680 | 48 | 32 | 1 | 1080 | 1050 | 3 | 6 | 0 |
| 1920x1080p | 148.5 | 2200 | 1920 | 88 | 44 | 1 | 1125 | 1080 | 4 | 5 | 0 |
| 1920x1080r | 138.5 | 2080 | 1920 | 48 | 32 | 1 | 1111 | 1080 | 3 | 5 | 0 |
| 1920x1080 | 173 | 2576 | 1920 | 128 | 200 | 0 | 1120 | 1080 | 3 | 5 | 1 |
| 1920x1080i | 74.25 | 2200 | 1920 | 88 | 44 | 1 | 562 | 540 | 2 | 5 | 0 |
| 1280x720 | 74.25 | 1650 | 1280 | 110 | 40 | 1 | 750 | 720 | 5 | 5 | 1 |
| NTSC (480i) | 13.5 | 858 | 720 | 16 | 64 | 1 | 262 | 240 | 3 | 4 | 0 |
| PAL (576i) | 13.5 | 864 | 720 | 16 | 64 | 1 | 312 | 288 | 3 | 4 | 0 |

This table is based on VESA standard. Some registers need to minus 1 such as HTT, HDE, VTT and VDE. 1920x1080p and 1920x1080i are based on EIA/CEA-861-B standard. "r" means reduced blank.

## De-interlacing

TW2880 is equipped with advanced motion adaptive de-interlacing circuit. If multi-channel display is desired, set the de-interlacing mode to MD or by-pass to use original Weave method. If one or two particular channels zoom up view are needed, user can only turn on few channels and pipe the data through scale up circuit to get the blow up view. The biggest size of the incoming video stream is limited to 1920 pixels. There are three de-interlacing method: Weave, 2D or 3D. In weave mode, no upscale supported.

## Upscale

Up scaling function is only supported in 2D or 3D de-interlaced mode is selected. TW2880's upscaler does not support channel based upscaling activities. That means all channels must be upscaled at the same time with the same ratio. When upscaling function is enabled, original channel position, size and boundaries will be changed. User must set POS_UPS_EN and POS_HSCALE and POS_VSCALE register accordingly to prevent wrong image.

# Mouse Interface

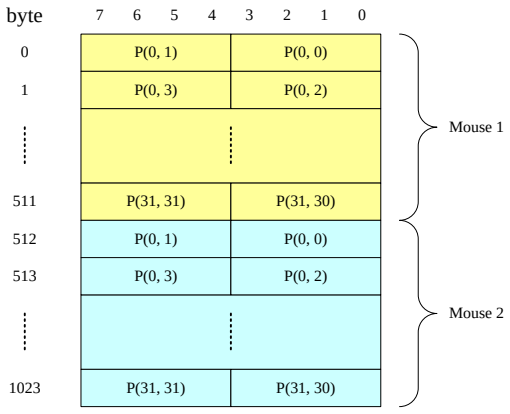
**TW2880 supports two identical 32x32 mouse pointer. One will be used as text cursor and the other will be used as mouse pointer. The location of the mouse0 is controlled by MOUSE0_HPOS and MOUSE0_VPOS register. User can write mouse data to local SRAM directly. Or user can write up to 16 mouse shape to SDRAM and then hardware load mouse shape to local SRAM automatically. Each mouse pixel has 4 bits.**

| Bit | Description |
|---|---|
| 3 | **Blinking Control**<br><br>1: Blink<br>0: Static |
| 2 | **Mixing Control**<br><br>1: Based on mouse mix control<br>0: mouse color |
| 1:0 | **Mouse Type**<br><br>00: Transparent<br>01: Filled with white color<br>10: Foreground color<br>11: Background color |

**To load mouse data to SRAM, there are total 1024 bytes in local SRAM. One mouse uses 512 bytes. Sample code is shown below:**

```
for (i=0;i<1024;i=i+4)
begin
  PWRITE(12'h54c, i/4);  //mouse data location
  PWRITE(12'h54d, mouse_data[i]);  //mouse data
  PWRITE(12'h54d, mouse_data[i+1]);  //mouse data
  PWRITE(12'h54d, mouse_data[i+2]);  //mouse data
  PWRITE(12'h54d, mouse_data[i+3]);  //mouse data
  PWRITE(12'h54f, 8'h00);  //mouse data write enable
End
```

**Mouse_data is 8 bit which is combined by two pixels. Bit[7:4] is p1, bit[3:0] is p0. Here is the SRAM allocation:**



**To load mouse shape to SDRM, sample code is shown below:**

```
   MOUSE_BASE_ADDR = 24'h3ffc00;
 PWRITE(12'h46c, MOUSE_BASE_ADDR[7:0]); //MOUSE_BASE_ADDR[7:0]
 PWRITE(12'h46d, MOUSE_BASE_ADDR[15:8]); //MOUSE_BASE_ADDR[15:8]
 PWRITE(12'h46e, MOUSE_BASE_ADDR[23:16]); //MOUSE_BASE_ADDR[23:16]

 host_dram_addr = MOUSE_BASE_ADDR;
 PWRITE(12'h003, 8'he0); //dram_rw_ctrl, [7]:0:r, 1:w, [6]: enable, [5:0] bl

 for(j=0;j<16;j=j+1)
 begin
   PWRITE(12'h000, host_dram_addr[7:0]); //dram_addr_l
   PWRITE(12'h001, host_dram_addr[15:8]); //dram_addr_m
   PWRITE(12'h002, host_dram_addr[23:16]); //dram_addr_h

   for (i=0;i<256;i=i+1) PWRITE(12'h004, mouse_data[i]); //dram_data
   PREAD(12'h044); //check status, if bit 0 is high, done

   host_dram_addr = host_dram_addr + 8'h20;


   PWRITE(12'h000, host_dram_addr[7:0]); //dram_addr_l
   PWRITE(12'h001, host_dram_addr[15:8]); //dram_addr_m
   PWRITE(12'h002, host_dram_addr[23:16]); //dram_addr_h

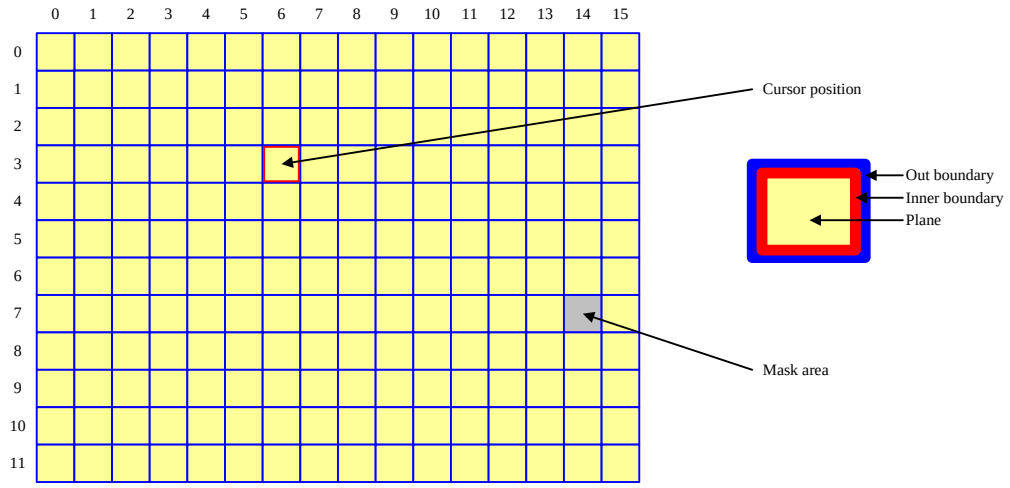
   for (i=256;i<512;i=i+1) PWRITE(12'h004, mouse_data[i]); //dram_data
   PREAD(12'h044); //check status, if bit 0 is high, done

   host_dram_addr = host_dram_addr + 8'h20;
 end

 PWRITE(12'h470, 8'h00); //MOUSE_BUF,MOUSE_INDEX
 PWRITE(12'h46f, 8'h01); //MOUSE_UPDATE_EN
 PREAD(12'h46f); //check bit 0, if 0 done
```

## Motion Box

**TW2880 supports 16 motion box for live channels. Motion box positions and size can be set by registers. User must set these register according to live channels down scale ratio.**




**The user can enable plane, out boundary and inner boundary. Cursor and motion are using inner boundary. When motion detection mode is enabled, user must set horizontal cell number to 15 and vertical cell number to 12.**

## Single Box

TW2880 provides 8 single boxes which can be used for highlighting portion of the display. The effects include a single box or box cursor, a masking box and a box blending with a plane color. Each box has programmable location and sizes and controlled by BOX_HL (0x513 - 0x51A), BOX_HW (0x51B - 0x522), BOX_VT (0x523 – 0x52A) and BOX_VW (0x52B – 0x532) registers. The BOX_HL is the horizontal location of box with 2 pixel unit and the BOX_HW is the horizontal size of box with 2 pixel unit. The BOX_VT is the vertical location of box with 1 line unit and the BOX_VW is the vertical size of box with 1 line unit.

The display option is controlled by registers (0x50C – 0x510). BOX_PLNEN bit in these register enables each plane color and its R, G, B components are defined by registers 0x536, 0x537, 0x538 for box 1-4, by registers 0x496, 0x497, 0x498 for box 5-8. The color of box boundary is enabled via the BOX_BNDEN bit in the control registers and its color is defined by registers 0x533, 0x534, 0x535 for box 1-4, 0x493, 0x494, 0x495 for box 5-8.

In case that several boxes have same region specified, there will be a conflict of what to display for that region. Generally the TW2880 defines that box 0 has priority over box 7. So if a conflict happens between more than 2 boxes, box 0 will be displayed first as top layer and box 1 to box 7 are hidden beneath that are not supported for pop-up attribute unlike channel display.

## Peak Function

TW2880B support peak function. This function make video looks sharp. It applies both horizontal and vertical directions.

## OSG Layer

TW2880B main display support 3 layers OSG window. Each window has 8 sub-windows. But 8 sub-window cannot overlap. Each sub-window has its own memory location and screen location and size. Detail descriptions are in OSG chapter.

## Simple OSD

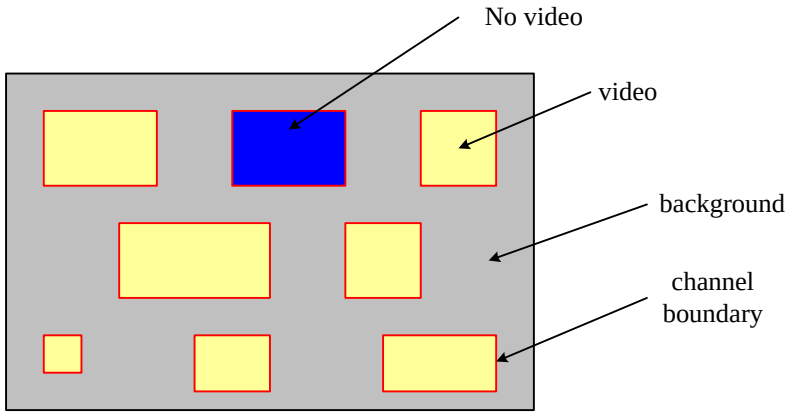
TW2880B support 32 channel simple OSD. Simple OSD is font based OSD. In each channel, it can support 8 font channel number information and 32x32 pictures. In whole screen, it support 32 title and 32 time information. Detail descriptions are in OSD chapter.

## Background and Channel Boundary

The area without video will show background. Background color is 24bit color which can be set by registers.

Each channel has its own boundary. TW2880B can support 32 channel boundary. Channel position and size information are got from rgb_interface related registers. If upscale is enable, user must turn on pos_ups_en and set correct pos_hscale and pos_vscale registers. In some cases, video is not turn on, user still wanted to show boundary for this channel. User must set the following registers:

```
[0x4F0] to [0x4F3]: BND_CH_EN
[0x4CE] bit 4: BND_CH_EN_SEL, this bit must be set to “1”
[0x4F8] to [0x4FA]: NOVID_R/G/B
```




Display Background and Boundary

## Digital Gain

**Digital gain is the last stage of the video pipe and is operated on the RGB color space. The formulary is:**

```
R = R * R_GAIN + R_OFFSET
G = G * G_GAIN + G_OFFSET
B = B * B_GAIN + B_OFFSET
```

**Gain and offset are set by register. 0x40 is default value which gain is "1". Offset is 2's complement value. "0" means no offset. 0x7f is maximum offset. 0x80 is -128 and 0xFF is -1.**

## Register Table

**Following are pages 4 and 5. Page 1 for OSG and simple OSD is described in a separate section.**

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x400 | R/W | 0x10 | [7:4]: rg_afm_ctl<br>[2]: rg_di_use_bob<br>[1]: rg_di_use_3field<br>[0]: rg_force_2ddi |
| 0x401 | R/W | 0x00 | rg_di_err_base[7:0] |
| 0x402 | R/W | 0x00 | rg_di_err_base[15:8] |
| 0x403 | R/W | 0x00 | rg_di_err_base[21:16] |
| 0x404 | R/W | 0xF0 | rg_man_wr_height[7:0] |
| 0x405 | R/W | 0x00 | rg_man_wr_height[10:8] |
| 0x406 | R/W | 0xD0 | rg_man_wr_width[7:0] |
| 0x407 | R/W | 0x02 | rg_man_wr_width[10:8] |
| 0x408 | R/W | 0x00 | rg_ditb0[6:0] |
| 0x409 | R/W | 0x04 | rg_ditb4[6:0] |
| 0x40A | R/W | 0x08 | rg_ditb8 |
| 0x40B | R/W | 0x0C | rg_ditb12 |
| 0x40C | R/W | 0x40 | rg_ditb16 |
| 0x40D | R/W | 0x40 | rg_ditb24 |
| 0x40E | R/W | 0x40 | rg_ditb32 |
| 0x40F | R/W | 0x40 | rg_ditb48 |
| 0x410 | R/W | 0x30 | rg_afm_yf02_thd |
| 0x411 | R/W | 0x68 | rg_afm_yf02_h_thd |
| 0x412 | R/W | 0x08 | rg_afm_yf01_thd |
| 0x413 | R/W | 0x80 | rg_afm_yf01_h_thd |
| 0x414 | R/W | 0x01 | rg_32_caption_thd |
| 0x415 | R/W | 0x00 | [7:4]: rg_ups1_vsharp_gain<br>[1]: rg_ups1_vlimit_on<br>[0]: rg_disable_man_disp |
| 0x416 | R/W | 0x00 | rg_BGcolor |
| 0x417 | R/W | 0x00 | rg_ups1_hscale[7:0] |
| 0x418 | R/W | 0x10 | rg_ups1_hscale[12:8] |
| 0x419 | R/W | 0x00 | rg_ups1_vscale[7:0] |
| 0x41A | R/W | 0x04 | rg_ups1_vscale[10:8] |
| 0x41B | R/W | 0x00 | rg_ups1_Xst[7:0] |
| 0x41C | R/W | 0x00 | rg_ups1_Xst[10:8] |
| 0x41D | R/W | 0x00 | rg_ups1_Yst[7:0] |
| 0x41E | R/W | 0x00 | rg_ups1_Yst[10:8] |
| 0x41F | R/W | 0xD0 | rg_ups1_Xmax[7:0] |
| 0x420 | R/W | 0x02 | rg_ups1_Xmax[10:8] |
| 0x421 | R/W | 0xE0 | rg_ups1_Ymax[7:0] |
| 0x422 | R/W | 0x01 | rg_ups1_Ymax[10:8] |
| 0x423 | R/W | 0x00 | rg_ups1_Xoff |
| 0x424 | R/W | 0x00 | rg_ups1_Yoff |
| 0x425 | R/W | 0x00 | [7]: rg_panorama<br>[6]: rg_ups1_hs_type<br>[5:4]: rg_ups1_hs_step<br>[3:0]: rg_ups1_hs_inc |
| 0x426 | R/W | 0x00 | rg_ups1_hs_center[7:0] |
| 0x427 | R/W | 0x00 | rg_ups1_hs_center[12:8] |
| 0x428 | R/W | 0x00 | rg_2ddi_ctl |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x429 | R/W | 0x28 | rg_2ddi_thd1 |
| 0x42A | R/W | 0x1E | rg_2ddi_thd2 |
| 0x42B | R/W | 0x28 | rg_2ddi_thd3 |
| 0x42C | R/W | 0x10 | rg_2ddi_thd4 |
| 0x42D | R/W | 0x24 | rg_2ddi_thd5 |
| 0x42E | R/W | 0x6F | rg_2ddi_thd6 |
| 0x42F | R/W | 0x00 | rg_3ddi_ctl |
| 0x430 | R/W | 0x0A | rg_3ddi_thd1 |
| 0x431 | R/W | 0x95 | rg_3ddi_thd2 |
| 0x432 | R/W | 0x00 | rg_ups1_dbg |
| | | | |
| 0x440 | R/W | 0x00 | SBOX4_CTRL[4:0] |
| 0x441 | R/W | 0x00 | SBOX5_CTRL[4:0] |
| 0x442 | R/W | 0x00 | SBOX6_CTRL[4:0] |
| 0x443 | R/W | 0x00 | SBOX7_CTRL[4:0] |
| 0x444 | R/W | 0x00 | [7:6]: SBOX5_V_LINE<br>[5:4]: SBOX5_H_LINE<br>[3:2]: SBOX4_V_LINE<br>[1:0]: SBOX4_H_LINE |
| 0x445 | R/W | 0x00 | [7:6]: SBOX5_V_LINE<br>[5:4]: SBOX5_H_LINE<br>[3:2]: SBOX4_V_LINE<br>[1:0]: SBOX4_H_LINE |
| 0x446 | R/W | 0x00 | SBOX4_HL[7:0] |
| 0x447 | R/W | 0x00 | SBOX4_HL[10:8] |
| 0x448 | R/W | 0x00 | SBOX5_HL[7:0] |
| 0x449 | R/W | 0x00 | SBOX5_HL[10:8] |
| 0x44A | R/W | 0x00 | SBOX6_HL[7:0] |
| 0x44B | R/W | 0x00 | SBOX6_HL[10:8] |
| 0x44C | R/W | 0x00 | SBOX7_HL[7:0] |
| 0x44D | R/W | 0x00 | SBOX7_HL[10:8] |
| 0x44E | R/W | 0x00 | SBOX4_HR[7:0] |
| 0x44F | R/W | 0x00 | SBOX4_HR[10:8] |
| 0x450 | R/W | 0x00 | SBOX5_HR[7:0] |
| 0x451 | R/W | 0x00 | SBOX5_HR[10:8] |
| 0x452 | R/W | 0x00 | SBOX6_HR[7:0] |
| 0x453 | R/W | 0x00 | SBOX6_HR[10:8] |
| 0x454 | R/W | 0x00 | SBOX7_HR[7:0] |
| 0x455 | R/W | 0x00 | SBOX7_HR[10:8] |
| 0x456 | R/W | 0x00 | SBOX4_VT[7:0] |
| 0x457 | R/W | 0x00 | SBOX4_VT[10:8] |
| 0x458 | R/W | 0x00 | SBOX5_VT[7:0] |
| 0x459 | R/W | 0x00 | SBOX5_VT[10:8] |
| 0x45A | R/W | 0x00 | SBOX6_VT[7:0] |
| 0x45B | R/W | 0x00 | SBOX6_VT[10:8] |
| 0x45C | R/W | 0x00 | SBOX7_VT[7:0] |
| 0x45D | R/W | 0x00 | SBOX7_VT[10:8] |
| 0x45E | R/W | 0x00 | SBOX4_VB[7:0] |
| 0x45F | R/W | 0x00 | SBOX4_VB[10:8] |
| 0x460 | R/W | 0x00 | SBOX5_VB[7:0] |
| 0x461 | R/W | 0x00 | SBOX5_VB[10:8] |
| 0x462 | R/W | 0x00 | SBOX6_VB[7:0] |
| 0x463 | R/W | 0x00 | SBOX6_VB[10:8] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x464 | R/W | 0x00 | SBOX7_VB[7:0] |
| 0x465 | R/W | 0x00 | SBOX7_VB[10:8] |
| 0x466 | R/W | 0x00 | BBR2: Box border R |
| 0x467 | R/W | 0x00 | BBG2: Box border G |
| 0x468 | R/W | 0x00 | BBB2: Box border B |
| 0x469 | R/W | 0x00 | BPR2: Box plane R |
| 0x46A | R/W | 0x00 | BPG2: Box plane G |
| 0x46B | R/W | 0x00 | BPB2: Box plane B |
| 0x46C | R/W | 0x00 | MOUSE_BASE_ADDR[7:0] |
| 0x46D | R/W | 0x00 | MOUSE_BASE_ADDR[15:8] |
| 0x46E | R/W | 0x00 | MOUSE_BASE_ADDR[23:16] |
| 0x46F | R/W | 0x00 | [0]: MOUSE_UPDATE_EN, MOUSE_UPDATE_BUSY |
| 0x470 | R/W | 0x00 | [4]: MOUSE_BUF<br>[3:0]: MOUSE_INDEX |
| 0x471 | | | |
| 0x472 | R/W | 0x00 | VPOS_32[7:0] |
| 0x473 | R/W | 0x00 | HPOS_32[7:0] |
| 0x474 | R/W | 0x00 | VSIZE_32[7:0] |
| 0x475 | R/W | 0x00 | HSIZE_32[7:0] |
| 0x476 | RW | 0x00 | [5:4]:HPOS_32[9:8]<br>[3:0]: VPOS_32[11:8] |
| 0x477 | R/W | 0x00 | [4]:HSIZE_32[8]<br>[2:0]: VSIZE_32[10:8] |
| 0x478 | R/W | 0x00 | DM_VPOS_32[7:0] |
| 0x479 | R/W | 0x00 | DM_HPOS_32[7:0] |
| 0x47A | R/W | 0x00 | DM_VSIZE_32[7:0] |
| 0x47B | R/W | 0x00 | DM_HSIZE_32[7:0] |
| 0x47C | RW | 0x00 | [5:4]: DM_HPOS_32[9:8]<br>[3:0]: DM_VPOS_32[11:8] |
| 0x47D | R/W | 0x00 | [4]: DM_HSIZE_32[8]<br>[2:0]: DM_VSIZE_32[10:8] |
| 0x47E | R/W | 0x00 | [1]: CH32_BND_EN<br>[0]: CH32_EN |
| 0x47F | R/W | 0x00 | [1]: DM_CH32_BND_EN<br>[0]: DM_CH32_EN |
| 0x480 | R/W | 0x80 | [7]: OUT_DIS<br>[6]: MDBOX_POS_SEL<br>[5]: Reserved<br>[4]: BND_EN<br>[3]: COLBAR_EN<br>[2]: SWITCH_EN<br>[1]: HSPOL<br>[0]: VSPOL |
| 0x481 | R/W | 0x2F | [7:4]: BND_WIDTH<br>[3:0]: VSDEL |
| 0x482 | R/W | 0xFF | BND_R: Channel Boundary R |
| 0x483 | R/W | 0xFF | BND_G: Channel Boundary G |
| 0x484 | R/W | 0xFF | BND_B: Channel Boundary B |
| 0x485 | R/W | 0x00 | POS_HSCALE[7:0] |
| 0x486 | R/W | 0x10 | POS_HSCALE[15:8] |
| 0x487 | R/W | 0x00 | POS_VSCALE[7:0] |
| 0x488 | R/W | 0x10 | POS_VSCALE[15:8] |
| 0x489 | R/W | 0x00 | [0]: POS_UPS_EN |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x48A | R/W | 0x00 | [1:0]: OSD_BLINK_TIME |
| 0x48B | R/W | 0x00 | [7:4]: CUR1_VPOS<br>[3:0]: CUR0_VPOS |
| 0x48C | R/W | 0x00 | [7:4]: CUR3_VPOS<br>[3:0]: CUR2_VPOS |
| 0x48D | R/W | 0x00 | [7:4]: CUR5_VPOS<br>[3:0]: CUR4_VPOS |
| 0x48E | R/W | 0x00 | [7:4]: CUR7_VPOS<br>[3:0]: CUR6_VPOS |
| 0x48F | R/W | 0x00 | [7:4]: CUR9_VPOS<br>[3:0]: CUR8_VPOS |
| 0x490 | R/W | 0x00 | [7:4]: CUR11_VPOS<br>[3:0]: CUR10_VPOS |
| 0x491 | R/W | 0x00 | [7:4]: CUR13_VPOS<br>[3:0]: CUR12_VPOS |
| 0x492 | R/W | 0x00 | [7:4]: CUR15_VPOS<br>[3:0]: CUR14_VPOS |
| 0x493 | R/W | 0x00 | MDBOX_OBR |
| 0x494 | R/W | 0x00 | MDBOX_OBG |
| 0x495 | R/W | 0x00 | MDBOX_OBB |
| 0x496 | R/W | 0x00 | MDBOX_IBR |
| 0x497 | R/W | 0x00 | MDBOX_IBG |
| 0x498 | R/W | 0x00 | MDBOX_IBB |
| 0x499 | R/W | 0x00 | MDBOX_MSKR |
| 0x49A | R/W | 0x00 | MDBOX_MSKG |
| 0x49B | R/W | 0x00 | MDBOX_MSKB |
| 0x49C | R/W | 0x00 | MDBOX_PR |
| 0x49D | R/W | 0x00 | MDBOX_PG |
| 0x49E | R/W | 0x00 | MDBOX_PB |
| 0x49F | | | |
| 0x4A0 | R/W | 0x00 | HSTART[7:0] |
| 0x4A1 | R/W | 0x00 | HSTART[9:8] |
| 0x4A2 | R/W | 0x00 | VSTART[7:0] |
| 0x4A3 | R/W | 0x00 | VSTART[11:8] |
| 0x4A4 | R/W | 0xFF | VTT_WIN |
| 0x4A5 | R/W | 0x00 | [2]: VTT_ADJUST<br>[1]: VTT_ADJUST_MODE<br>[0]: VTT_ADJUST_EN |
| 0x4A6 | RO | - | NEW_VTT_RGBW[7:0] |
| 0x4A7 | RO | - | NEW_VTT_RGBW[11:8] |
| 0x4A8 | RO | - | NEW_VTT[7:0] |
| 0x4A9 | RO | - | NEW_VTT[10:8] |
| 0x4AA | R/W | 0xFF | [7:4]: BND_EN_1<br>[3:0]: BND_EN_0 |
| 0x4AB | R/W | 0xFF | [7:4]: BND_EN_3<br>[3:0]: BND_EN_2 |
| 0x4AC | R/W | 0xFF | [7:4]: BND_EN_5<br>[3:0]: BND_EN_4 |
| 0x4AD | R/W | 0xFF | [7:4]: BND_EN_7<br>[3:0]: BND_EN_6 |
| 0x4AE | R/W | 0xFF | [7:4]: BND_EN_9<br>[3:0]: BND_EN_8 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x4AF | R/W | 0xFF | [7:4]: BND_EN_11<br>[3:0]: BND_EN_10 |
| 0x4B0 | R/W | 0xFF | [7:4]: BND_EN_13<br>[3:0]: BND_EN_12 |
| 0x4B1 | R/W | 0xFF | [7:4]: BND_EN_15<br>[3:0]: BND_EN_14 |
| 0x4B2 | R/W | 0xFF | [7:4]: BND_EN_17<br>[3:0]: BND_EN_16 |
| 0x4B3 | R/W | 0xFF | [7:4]: BND_EN_19<br>[3:0]: BND_EN_18 |
| 0x4B4 | R/W | 0xFF | PS2CON[7:0] |
| 0x4B5 | R/W | 0xFF | PS2CON[15:8] |
| 0x4B6 | RO | - | PS2DATA_OUT[7:0] |
| 0x4B7 | RO | - | PS2DATA_OUT[15:8] |
| 0x4B8 | RO | - | PS2DATA_OUT[23:16] |
| 0x4B9 | RO | - | PS2DATA_OUT[31:24] |
| 0x4BA | R/W | | [0]: ps2_int_reg |
| 0x4BB | R/W | 0x00 | [4]: EOSD_SEL<br>[3]: EOSD_EN<br>[2]: EOSD_MODE<br>[1]: EOSD_EN_DIS<br>[0]: EOSD_ALPHA_EN |
| 0x4BC | R/W | 0x00 | [3:0]: EOSD_ALPHA |
| 0x4BD | R/W | 0x00 | EOSD_R |
| 0x4BE | R/W | 0x00 | EOSD_G |
| 0x4BF | R/W | 0x00 | EOSD_B |
| 0x4C0 | R/W | 0xFF | [3:0]: INT_MASK<br>[3]: osg_vblank_int_mask<br>[2]: ps2_int_mask<br>[1]: osg_host_busy_int_mask<br>[0]: osg_host_wait_int_mask |
| 0x4C1 | R/W | 0x00 | [3]: osg_vblank_int / clear<br>[2]: ps2_int / clear<br>[1]: osg_host_busy_int / clear<br>[0]: osg_host_wait_int / clear |
| 0x4C2 | R/W | 0x00 | FRSC_DBG_CTRL<br>[7]: rd_buf_id_toggle<br>[6]: rd_buf_inc<br>[5]: wr_page_fix<br>[4]: rd_page_fix<br>[3:2]: wr_page_sel<br>[1:0]: rd_page_sel |
| 0x4C3 | R/W | 0x60 | [7:5]: WR_BUF_OFST<br>[4]: INTERLACE_DBG<br>[3]: VTT_INV<br>[2]:EVEN_REPEAT<br>[1]:DMON_VS_EN<br>[0]:interlace |
| 0x4C4 | R/W | 0xFF | [7:4]: BND_EN_21<br>[3:0]: BND_EN_20 |
| 0x4C5 | R/W | 0xFF | [7:4]: BND_EN_23<br>[3:0]: BND_EN_22 |
| 0x4C6 | R/W | 0xFF | [7:4]: BND_EN_25<br>[3:0]: BND_EN_24 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x4C7 | R/W | 0xFF | [7:4]: BND_EN_27<br>[3:0]: BND_EN_26 |
| 0x4C8 | R/W | 0xFF | [7:4]: BND_EN_29<br>[3:0]: BND_EN_28 |
| 0x4C9 | R/W | 0xFF | [7:4]: BND_EN_31<br>[3:0]: BND_EN_30 |
| 0x4CA | R/W | 0x00 | GAMA_ADDR[7:0] |
| 0x4CB | R/W | 0x00 | GAMA_ADDR[9:8] |
| 0x4CC | R/W | 0x00 | GAMA_DATA[7:0] |
| 0x4CD | R/W | 0x00 | GAMA_DATA[9:8] |
| 0x4CE | R/W | 0x00 | [6]: OSD_CH_EN_SEL<br>[5]: NOVID_SWITCH_EN<br>[4]: BND_CH_EN_SEL<br>[3]: PEAK_C_DUP<br>[2]: GAIN_EN<br>[1]: PEAK_EN<br>[0]: GAMA_EN |
| 0x4CF | R/W | 0x00 | FRSC_DBG_CTRL[15:8]<br>[5:4]: rd_buf_id_sel_dm<br>[3:2]: rd_buf_id_sel<br>[1]: fld_inv<br>[0]: all_available |
| 0x4D0 | R/W | 0x30 | [7:0]: PEAK_CTRL |
| 0x4D1 | R/W | 0x00 | PEAK_TB0 |
| 0x4D2 | R/W | 0x00 | PEAK_TB1 |
| 0x4D3 | R/W | 0x00 | PEAK_TB2 |
| 0x4D4 | R/W | 0x00 | PEAK_TB3 |
| 0x4D5 | R/W | 0x00 | PEAK_TB4 |
| 0x4D6 | R/W | 0x00 | PEAK_TB5 |
| 0x4D7 | R/W | 0x00 | PEAK_VTB |
| 0x4D8 | R/W | 0x00 | [7:4]: PEAK_FTHD<br>[3:0]: PEAK_CORING |
| 0x4D9 | R/W | 0x00 | [4:0]: PEAK_OVERSHOT |
| 0x4DA | R/W | 0x40 | R_GAIN |
| 0x4DB | R/W | 0x40 | G_GAIN |
| 0x4DC | R/W | 0x40 | B_GAIN |
| 0x4DD | R/W | 0 | R_OFST |
| 0x4DE | R/W | 0 | G_OFST |
| 0x4DF | R/W | 0 | B_OFST |
| 0x4F0 | R/W | 0 | BND_CH_EN[7:0] |
| 0x4F1 | R/W | 0 | BND_CH_EN[15:8] |
| 0x4F2 | R/W | 0 | BND_CH_EN[23:16] |
| 0x4F3 | R/W | 0 | BND_CH_EN[31:24] |
| 0x4F4 | R/W | 0xFF | FRSC_CH_EN[7:0] |
| 0x4F5 | R/W | 0xFF | FRSC_CH_EN[15:8] |
| 0x4F6 | R/W | 0xFF | FRSC_CH_EN[23:16] |
| 0x4F7 | R/W | 0xFF | FRSC_CH_EN[31:24] |
| 0x4F8 | R/W | 0 | NOVID_R |
| 0x4F9 | R/W | 0 | NOVID_G |
| 0x4FA | R/W | 0xFF | NOVID_B |
| | | | |
| 0x500 | R/W | 0x2F | HTT[7:0] |
| 0x501 | R/W | 0x07 | HTT[12:8] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x502 | R/W | 0x37 | VTT[7:0] |
| 0x503 | R/W | 0x04 | VTT[10:8] |
| 0x504 | R/W | 0x8F | HDE[7:0] |
| 0x505 | R/W | 0x06 | HDE[10:8] |
| 0x506 | R/W | 0x19 | VDE[7:0] |
| 0x507 | R/W | 0x04 | VDE[10:8] |
| 0x508 | R/W | 0x30 | PHSYNC[8:1] |
| 0x509 | R/W | 0x02 | PVSYNC[7:0] |
| 0x50A | R/W | 0x20 | HSPW[7:0] |
| 0x50B | R/W | 0x06 | VSPW[7:0] |
| 0x50C | R/W | 0x00 | SBOX0_CTRL[4:0] |
| 0x50D | R/W | 0x00 | SBOX1_CTRL[4:0] |
| 0x50E | R/W | 0x00 | SBOX2_CTRL[4:0] |
| 0x50F | R/W | 0x00 | SBOX3_CTRL[4:0] |
| | | | |
| 0x511 | R/W | 0x00 | [7:6]: SBOX3_V_LINE<br>[5:4]: SBOX3_H_LINE<br>[3:2]: SBOX2_V_LINE<br>[1:0]: SBOX2_H_LINE |
| 0x512 | R/W | 0x00 | [7:6]: SBOX1_V_LINE<br>[5:4]: SBOX1_H_LINE<br>[3:2]: SBOX0_V_LINE<br>[1:0]: SBOX0_H_LINE |
| 0x513 | R/W | 0x00 | SBOX0_HL[7:0] |
| 0x514 | R/W | 0x00 | SBOX0_HL[10:8] |
| 0x515 | R/W | 0x00 | SBOX1_HL[7:0] |
| 0x516 | R/W | 0x00 | SBOX1_HL[10:8] |
| 0x517 | R/W | 0x00 | SBOX2_HL[7:0] |
| 0x518 | R/W | 0x00 | SBOX2_HL[10:8] |
| 0x519 | R/W | 0x00 | SBOX3_HL[7:0] |
| 0x51A | R/W | 0x00 | SBOX3_HL[10:8] |
| 0x51B | R/W | 0x00 | SBOX0_HR[7:0] |
| 0x51C | R/W | 0x00 | SBOX0_HR[10:8] |
| 0x51D | R/W | 0x00 | SBOX1_HR[7:0] |
| 0x51E | R/W | 0x00 | SBOX1_HR[10:8] |
| 0x51F | R/W | 0x00 | SBOX2_HR[7:0] |
| 0x520 | R/W | 0x00 | SBOX2_HR[10:8] |
| 0x521 | R/W | 0x00 | SBOX3_HR[7:0] |
| 0x522 | R/W | 0x00 | SBOX3_HR[10:8] |
| 0x523 | R/W | 0x00 | SBOX0_VT[7:0] |
| 0x524 | R/W | 0x00 | SBOX0_VT[10:8] |
| 0x525 | R/W | 0x00 | SBOX1_VT[7:0] |
| 0x526 | R/W | 0x00 | SBOX1_VT[10:8] |
| 0x527 | R/W | 0x00 | SBOX2_VT[7:0] |
| 0x528 | R/W | 0x00 | SBOX2_VT[10:8] |
| 0x529 | R/W | 0x00 | SBOX3_VT[7:0] |
| 0x52A | R/W | 0x00 | SBOX3_VT[10:8] |
| 0x52B | R/W | 0x00 | SBOX0_VB[7:0] |
| 0x52C | R/W | 0x00 | SBOX0_VB[10:8] |
| 0x52D | R/W | 0x00 | SBOX1_VB[7:0] |
| 0x52E | R/W | 0x00 | SBOX1_VB[10:8] |
| 0x52F | R/W | 0x00 | SBOX2_VB[7:0] |
| 0x530 | R/W | 0x00 | SBOX2_VB[10:8] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x531 | R/W | 0x00 | SBOX3_VB[7:0] |
| 0x532 | R/W | 0x00 | SBOX3_VB[10:8] |
| 0x533 | R/W | 0x00 | BBR: Box border R |
| 0x534 | R/W | 0x00 | BBG: Box border G |
| 0x535 | R/W | 0x00 | BBB: Box border B |
| 0x536 | R/W | 0x00 | BPR: Box plane R |
| 0x537 | R/W | 0x00 | BPG: Box plane G |
| 0x538 | R/W | 0x00 | BPB: Box plane B |
| 0x539 | R/W | 0x00 | BAR: Background R |
| 0x53A | R/W | 0x00 | BAG: Background G |
| 0x53B | R/W | 0x00 | BAB: Background B |
| 0x53C | R/W | 0x00 | MOUSE0_HPOS[7:0] |
| 0x53D | R/W | 0x00 | MOUSE0_HPOS[10:8] |
| 0x53E | R/W | 0x00 | MOUSE0_VPOS[7:0] |
| 0x53F | R/W | 0x00 | MOUSE0_VPOS[10:8] |
| 0x540 | R/W | 0x00 | MOUSE1_HPOS[7:0] |
| 0x541 | R/W | 0x00 | MOUSE1_HPOS[10:8] |
| 0x542 | R/W | 0x00 | MOUSE1_VPOS[7:0] |
| 0x543 | R/W | 0x00 | MOUSE1_VPOS[10:8] |
| 0x544 | R/W | 0x00 | [6:4]: MOUSE1_CTRL<br>[2:0]: MOUSE0_CTRL |
| 0x545 | R/W | 0x00 | BR: mouse background R |
| 0x546 | R/W | 0x00 | BG: mouse background G |
| 0x547 | R/W | 0x00 | BB: mouse background B |
| 0x548 | R/W | 0x00 | FR: mouse foreground R |
| 0x549 | R/W | 0x00 | FG: mouse foreground G |
| 0x54A | R/W | 0x00 | FB: mouse foreground B |
| 0x54B | R/W | 0x00 | DMODE[1:0] |
| 0x54C | R/W | 0x00 | MOUSE_WR_LOC[7:0] |
| 0x54D | R/W | 0x00 | MOUSE_WR_DATA |
| 0x54E | R/W | 0x01 | MOUSE_REG_UPDATE |
| 0x54F | WO | 0x00 | MOUSE_WR_EN |
| 0x550 | R/W | 0x00 | MDBOX0_CTRL[7:0] |
| 0x551 | R/W | 0x00 | MDBOX1_CTRL[7:0] |
| 0x552 | R/W | 0x00 | MDBOX2_CTRL[7:0] |
| 0x553 | R/W | 0x00 | MDBOX3_CTRL[7:0] |
| 0x554 | R/W | 0x00 | MDBOX4_CTRL[7:0] |
| 0x555 | R/W | 0x00 | MDBOX5_CTRL[7:0] |
| 0x556 | R/W | 0x00 | MDBOX6_CTRL[7:0] |
| 0x557 | R/W | 0x00 | MDBOX7_CTRL[7:0] |
| 0x558 | R/W | 0x00 | MDBOX8_CTRL[7:0] |
| 0x559 | R/W | 0x00 | MDBOX9_CTRL[7:0] |
| 0x55A | R/W | 0x00 | MDBOX10_CTRL[7:0] |
| 0x55B | R/W | 0x00 | MDBOX11_CTRL[7:0] |
| 0x55C | R/W | 0x00 | MDBOX12_CTRL[7:0] |
| 0x55D | R/W | 0x00 | MDBOX13_CTRL[7:0] |
| 0x55E | R/W | 0x00 | MDBOX14_CTRL[7:0] |
| 0x55F | R/W | 0x00 | MDBOX15_CTRL[7:0] |
| 0x560 | R/W | 0x00 | [7:6]: MDBOX1_V_LINE<br>[5:4]: MDBOX1_H_LINE<br>[3:2]: MDBOX0_V_LINE<br>[1:0]: MDBOX0_H_LINE |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x561 | R/W | 0x00 | [7:6]: MDBOX3_V_LINE<br>[5:4]: MDBOX3_H_LINE<br>[3:2]: MDBOX2_V_LINE<br>[1:0]: MDBOX2_H_LINE |
| 0x562 | R/W | 0x00 | [7:6]: MDBOX5_V_LINE<br>[5:4]: MDBOX5_H_LINE<br>[3:2]: MDBOX4_V_LINE<br>[1:0]: MDBOX4_H_LINE |
| 0x563 | R/W | 0x00 | [7:6]: MDBOX7_V_LINE<br>[5:4]: MDBOX7_H_LINE<br>[3:2]: MDBOX6_V_LINE<br>[1:0]: MDBOX6_H_LINE |
| 0x564 | R/W | 0x00 | [7:6]: MDBOX9_V_LINE<br>[5:4]: MDBOX9_H_LINE<br>[3:2]: MDBOX8_V_LINE<br>[1:0]: MDBOX8_H_LINE |
| 0x565 | R/W | 0x00 | [7:6]: MDBOX11_V_LINE<br>[5:4]: MDBOX11_H_LINE<br>[3:2]: MDBOX10_V_LINE<br>[1:0]: MDBOX10_H_LINE |
| 0x566 | R/W | 0x00 | [7:6]: MDBOX13_V_LINE<br>[5:4]: MDBOX13_H_LINE<br>[3:2]: MDBOX12_V_LINE<br>[1:0]: MDBOX12_H_LINE |
| 0x567 | R/W | 0x00 | [7:6]: MDBOX15_V_LINE<br>[5:4]: MDBOX15_H_LINE<br>[3:2]: MDBOX14_V_LINE<br>[1:0]: MDBOX14_H_LINE |
| 0x568 | R/W | 0x00 | MDBOX0_HL[7:0] |
| 0x569 | R/W | 0x00 | MDBOX0_HL[10:8] |
| 0x56A | R/W | 0x00 | MDBOX1_HL[7:0] |
| 0x56B | R/W | 0x00 | MDBOX1_HL[10:8] |
| 0x56C | R/W | 0x00 | MDBOX2_HL[7:0] |
| 0x56D | R/W | 0x00 | MDBOX2_HL[10:8] |
| 0x56E | R/W | 0x00 | MDBOX3_HL[7:0] |
| 0x56F | R/W | 0x00 | MDBOX3_HL[10:8] |
| 0x570 | R/W | 0x00 | MDBOX4_HL[7:0] |
| 0x571 | R/W | 0x00 | MDBOX4_HL[10:8] |
| 0x572 | R/W | 0x00 | MDBOX5_HL[7:0] |
| 0x573 | R/W | 0x00 | MDBOX5_HL[10:8] |
| 0x574 | R/W | 0x00 | MDBOX6_HL[7:0] |
| 0x575 | R/W | 0x00 | MDBOX6_HL[10:8] |
| 0x576 | R/W | 0x00 | MDBOX7_HL[7:0] |
| 0x577 | R/W | 0x00 | MDBOX7_HL[10:8] |
| 0x578 | R/W | 0x00 | MDBOX8_HL[7:0] |
| 0x579 | R/W | 0x00 | MDBOX8_HL[10:8] |
| 0x57A | R/W | 0x00 | MDBOX9_HL[7:0] |
| 0x57B | R/W | 0x00 | MDBOX9_HL[10:8] |
| 0x57C | R/W | 0x00 | MDBOX10_HL[7:0] |
| 0x57D | R/W | 0x00 | MDBOX10_HL[10:8] |
| 0x57E | R/W | 0x00 | MDBOX11_HL[7:0] |
| 0x57F | R/W | 0x00 | MDBOX11_HL[10:8] |
| 0x580 | R/W | 0x00 | MDBOX12_HL[7:0] |
| 0x581 | R/W | 0x00 | MDBOX12_HL[10:8] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x582 | R/W | 0x00 | MDBOX13_HL[7:0] |
| 0x583 | R/W | 0x00 | MDBOX13_HL[10:8] |
| 0x584 | R/W | 0x00 | MDBOX14_HL[7:0] |
| 0x585 | R/W | 0x00 | MDBOX14_HL[10:8] |
| 0x586 | R/W | 0x00 | MDBOX15_HL[7:0] |
| 0x587 | R/W | 0x00 | MDBOX15_HL[10:8] |
| 0x588 | R/W | 0x00 | MDBOX0_VT[7:0] |
| 0x589 | R/W | 0x00 | MDBOX0_VT[10:8] |
| 0x58A | R/W | 0x00 | MDBOX1_VT[7:0] |
| 0x58B | R/W | 0x00 | MDBOX1_VT[10:8] |
| 0x58C | R/W | 0x00 | MDBOX2_VT[7:0] |
| 0x58D | R/W | 0x00 | MDBOX2_VT[10:8] |
| 0x58E | R/W | 0x00 | MDBOX3_VT[7:0] |
| 0x58F | R/W | 0x00 | MDBOX3_VT[10:8] |
| 0x590 | R/W | 0x00 | MDBOX4_VT[7:0] |
| 0x591 | R/W | 0x00 | MDBOX4_VT[10:8] |
| 0x592 | R/W | 0x00 | MDBOX5_VT[7:0] |
| 0x593 | R/W | 0x00 | MDBOX5_VT[10:8] |
| 0x594 | R/W | 0x00 | MDBOX6_VT[7:0] |
| 0x595 | R/W | 0x00 | MDBOX6_VT[10:8] |
| 0x596 | R/W | 0x00 | MDBOX7_VT[7:0] |
| 0x597 | R/W | 0x00 | MDBOX7_VT[10:8] |
| 0x598 | R/W | 0x00 | MDBOX8_VT[7:0] |
| 0x599 | R/W | 0x00 | MDBOX8_VT[10:8] |
| 0x59A | R/W | 0x00 | MDBOX9_VT[7:0] |
| 0x59B | R/W | 0x00 | MDBOX9_VT[10:8] |
| 0x59C | R/W | 0x00 | MDBOX10_VT[7:0] |
| 0x59D | R/W | 0x00 | MDBOX10_VT[10:8] |
| 0x59E | R/W | 0x00 | MDBOX11_VT[7:0] |
| 0x59F | R/W | 0x00 | MDBOX11_VT[10:8] |
| 0x5A0 | R/W | 0x00 | MDBOX12_VT[7:0] |
| 0x5A1 | R/W | 0x00 | MDBOX12_VT[10:8] |
| 0x5A2 | R/W | 0x00 | MDBOX13_VT[7:0] |
| 0x5A3 | R/W | 0x00 | MDBOX13_VT[10:8] |
| 0x5A4 | R/W | 0x00 | MDBOX14_VT[7:0] |
| 0x5A5 | R/W | 0x00 | MDBOX14_VT[10:8] |
| 0x5A6 | R/W | 0x00 | MDBOX15_VT[7:0] |
| 0x5A7 | R/W | 0x00 | MDBOX15_VT[10:8] |
| 0x5A8 | R/W | 0x00 | MDBOX0_HS[7:0] |
| 0x5A9 | R/W | 0x00 | MDBOX0_HS[10:8] |
| 0x5AA | R/W | 0x00 | MDBOX1_HS[7:0] |
| 0x5AB | R/W | 0x00 | MDBOX1_HS[10:8] |
| 0x5AC | R/W | 0x00 | MDBOX2_HS[7:0] |
| 0x5AD | R/W | 0x00 | MDBOX2_HS[10:8] |
| 0x5AE | R/W | 0x00 | MDBOX3_HS[7:0] |
| 0x5AF | R/W | 0x00 | MDBOX3_HS[10:8] |
| 0x5B0 | R/W | 0x00 | MDBOX4_HS[7:0] |
| 0x5B1 | R/W | 0x00 | MDBOX4_HS[10:8] |
| 0x5B2 | R/W | 0x00 | MDBOX5_HS[7:0] |
| 0x5B3 | R/W | 0x00 | MDBOX5_HS[10:8] |
| 0x5B4 | R/W | 0x00 | MDBOX6_HS[7:0] |
| 0x5B5 | R/W | 0x00 | MDBOX6_HS[10:8] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x5B6 | R/W | 0x00 | MDBOX7_HS[7:0] |
| 0x5B7 | R/W | 0x00 | MDBOX7_HS[10:8] |
| 0x5B8 | R/W | 0x00 | MDBOX8_HS[7:0] |
| 0x5B9 | R/W | 0x00 | MDBOX8_HS[10:8] |
| 0x5BA | R/W | 0x00 | MDBOX9_HS[7:0] |
| 0x5BB | R/W | 0x00 | MDBOX9_HS[10:8] |
| 0x5BC | R/W | 0x00 | MDBOX10_HS[7:0] |
| 0x5BD | R/W | 0x00 | MDBOX10_HS[10:8] |
| 0x5BE | R/W | 0x00 | MDBOX11_HS[7:0] |
| 0x5BF | R/W | 0x00 | MDBOX11_HS[10:8] |
| 0x5C0 | R/W | 0x00 | MDBOX12_HS[7:0] |
| 0x5C1 | R/W | 0x00 | MDBOX12_HS[10:8] |
| 0x5C2 | R/W | 0x00 | MDBOX13_HS[7:0] |
| 0x5C3 | R/W | 0x00 | MDBOX13_HS[10:8] |
| 0x5C4 | R/W | 0x00 | MDBOX14_HS[7:0] |
| 0x5C5 | R/W | 0x00 | MDBOX14_HS[10:8] |
| 0x5C6 | R/W | 0x00 | MDBOX15_HS[7:0] |
| 0x5C7 | R/W | 0x00 | MDBOX15_HS[10:8] |
| 0x5C8 | R/W | 0x00 | MDBOX0_VS[7:0] |
| 0x5C9 | R/W | 0x00 | MDBOX0_VS[10:8] |
| 0x5CA | R/W | 0x00 | MDBOX1_VS[7:0] |
| 0x5CB | R/W | 0x00 | MDBOX1_VS[10:8] |
| 0x5CC | R/W | 0x00 | MDBOX2_VS[7:0] |
| 0x5CD | R/W | 0x00 | MDBOX2_VS[10:8] |
| 0x5CE | R/W | 0x00 | MDBOX3_VS[7:0] |
| 0x5CF | R/W | 0x00 | MDBOX3_VS[10:8] |
| 0x5D0 | R/W | 0x00 | MDBOX4_VS[7:0] |
| 0x5D1 | R/W | 0x00 | MDBOX4_VS[10:8] |
| 0x5D2 | R/W | 0x00 | MDBOX5_VS[7:0] |
| 0x5D3 | R/W | 0x00 | MDBOX5_VS[10:8] |
| 0x5D4 | R/W | 0x00 | MDBOX6_VS[7:0] |
| 0x5D5 | R/W | 0x00 | MDBOX6_VS[10:8] |
| 0x5D6 | R/W | 0x00 | MDBOX7_VS[7:0] |
| 0x5D7 | R/W | 0x00 | MDBOX7_VS[10:8] |
| 0x5D8 | R/W | 0x00 | MDBOX8_VS[7:0] |
| 0x5D9 | R/W | 0x00 | MDBOX8_VS[10:8] |
| 0x5DA | R/W | 0x00 | MDBOX9_VS[7:0] |
| 0x5DB | R/W | 0x00 | MDBOX9_VS[10:8] |
| 0x5DC | R/W | 0x00 | MDBOX10_VS[7:0] |
| 0x5DD | R/W | 0x00 | MDBOX10_VS[10:8] |
| 0x5DE | R/W | 0x00 | MDBOX11_VS[7:0] |
| 0x5DF | R/W | 0x00 | MDBOX11_VS[10:8] |
| 0x5E0 | R/W | 0x00 | MDBOX12_VS[7:0] |
| 0x5E1 | R/W | 0x00 | MDBOX12_VS[10:8] |
| 0x5E2 | R/W | 0x00 | MDBOX13_VS[7:0] |
| 0x5E3 | R/W | 0x00 | MDBOX13_VS[10:8] |
| 0x5E4 | R/W | 0x00 | MDBOX14_VS[7:0] |
| 0x5E5 | R/W | 0x00 | MDBOX14_VS[10:8] |
| 0x5E6 | R/W | 0x00 | MDBOX15_VS[7:0] |
| 0x5E7 | R/W | 0x00 | MDBOX15_VS[10:8] |
| 0x5E8 | R/W | 0xFF | [7:4]: MDBOX1_HCELL<br>[3:0]: MDBOX0_HCELL |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x5E9 | R/W | 0xFF | [7:4]: MDBOX3_HCELL<br>[3:0]: MDBOX2_HCELL |
| 0x5EA | R/W | 0xFF | [7:4]: MDBOX5_HCELL<br>[3:0]: MDBOX4_HCELL |
| 0x5EB | R/W | 0xFF | [7:4]: MDBOX7_HCELL<br>[3:0]: MDBOX6_HCELL |
| 0x5EC | R/W | 0xFF | [7:4]: MDBOX9_HCELL<br>[3:0]: MDBOX8_HCELL |
| 0x5ED | R/W | 0xFF | [7:4]: MDBOX11_HCELL<br>[3:0]: MDBOX10_HCELL |
| 0x5EE | R/W | 0xFF | [7:4]: MDBOX13_HCELL<br>[3:0]: MDBOX12_HCELL |
| 0x5EF | R/W | 0xFF | [7:4]: MDBOX15_HCELL<br>[3:0]: MDBOX14_HCELL |
| 0x5F0 | R/W | 0xBB | [7:4]: MDBOX1_VCELL<br>[3:0]: MDBOX0_VCELL |
| 0x5F1 | R/W | 0xBB | [7:4]: MDBOX3_VCELL<br>[3:0]: MDBOX2_VCELL |
| 0x5F2 | R/W | 0xBB | [7:4]: MDBOX5_VCELL<br>[3:0]: MDBOX4_VCELL |
| 0x5F3 | R/W | 0xBB | [7:4]: MDBOX7_VCELL<br>[3:0]: MDBOX6_VCELL |
| 0x5F4 | R/W | 0xBB | [7:4]: MDBOX9_VCELL<br>[3:0]: MDBOX8_VCELL |
| 0x5F5 | R/W | 0xBB | [7:4]: MDBOX11_VCELL<br>[3:0]: MDBOX10_VCELL |
| 0x5F6 | R/W | 0xBB | [7:4]: MDBOX13_VCELL<br>[3:0]: MDBOX12_VCELL |
| 0x5F7 | R/W | 0xBB | [7:4]: MDBOX15_VCELL<br>[3:0]: MDBOX14_VCELL |
| 0x5F8 | R/W | 0x00 | [7:4]: CUR1_HPOS<br>[3:0]: CUR0_HPOS |
| 0x5F9 | R/W | 0x00 | [7:4]: CUR3_HPOS<br>[3:0]: CUR2_HPOS |
| 0x5FA | R/W | 0x00 | [7:4]: CUR5_HPOS<br>[3:0]: CUR4_HPOS |
| 0x5FB | R/W | 0x00 | [7:4]: CUR7_HPOS<br>[3:0]: CUR6_HPOS |
| 0x5FC | R/W | 0x00 | [7:4]: CUR9_HPOS<br>[3:0]: CUR8_HPOS |
| 0x5FD | R/W | 0x00 | [7:4]: CUR11_HPOS<br>[3:0]: CUR10_HPOS |
| 0x5FE | R/W | 0x00 | [7:4]: CUR13_HPOS<br>[3:0]: CUR12_HPOS |
| 0x5FF | R/W | 0x00 | [7:4]: CUR15_HPOS<br>[3:0]: CUR14_HPOS |

# Register

### DE-INTERLACE MODE SELECTION REGISTER 1 – 0X400

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x1 | **rg_afm_ctl**<br><br>[7]: debug_alpha, internal use only<br>[6]: disable_line_conv, internal use only<br>[5]: disable 32 pulldown<br>1: Turn off 32 pulldown detection.<br>0: Turn on 32 pulldown detection.<br>[4]: disable 22 pulldown<br>1: Turn off 22 Pulldown detection<br>0: Turn on 22 pulldown detection. |
| 3 | R/W | 0 | **WEAVE_UP_EN:** Weave mode up-scaler on/off control<br><br>1 = turn on up-scaler<br>0 = turn off up-scaler |
| 2 | R/W | 0 | **rg_dl_use_bob**<br><br>1: Use BOB for 2D de-interlace.<br>0: Use low angle 2D de-interlace. |
| 1 | R/W | 0 | **rg_dl_use_3field**<br><br>1 = use 3 fields for de-interlace<br>0 = use 5 fields for de-interlace |
| 0 | R/W | 0 | **rg_force_2ddi**<br><br>1 = turn off 3D de-interlace, use 2D de-interlace<br>0 = turn on 3D de-interlace |

### ERROR BUFFER BASE LOW BYTE REGISTER – 0X401

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **rg_dl_err_base [7:0]**<br><br>Start address of error field buffer allocated in the SDRAM. This number is in unit of 8 bytes. [23:22] is bank address. (2048 x height x 2) / 8 |

### ERROR BUFFER BASE MIDDLE BYTE REGISTER – 0X402

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **rg_dl_err_base [15:8]**<br><br>Start address of error field buffer allocated in the SDRAM. This number is in unit of 8 bytes. [23:22] is bank address. (2048 x height x 2) / 8 |

### ERROR BUFFER BASE HIGH BYTE REGISTER – 0X403

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | rg_di_err_base [23:16]<br><br>Start address of error field buffer allocated in the SDRAM. This number is in unit of 8 bytes. [23:22] is bank address. (2048 x height x 2) / 8 |

### HEIGHT OF ONE FIELD IN BUFFER REGISTER LOW BYTE – 0X404

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xF0 | rg_man_wr_height [7:0]<br><br>This number specifies the number of lines in the field. It is half of frame height. |

### HEIGHT OF ONE FIELD IN BUFFER REGISTER HIGH BYTE – 0X405

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | rg_man_wr_height [10:8]<br><br>This number specifies the number of lines in the field. It is half of frame height. |

### WIDTH OF ONE FIELD IN BUFFER REGISTER LOW BYTE – 0X406

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xD0 | rg_man_wr_width [7:0]<br><br>This number specifies the number of pixels per line in the buffer. |

### WIDTH OF ONE FIELD IN BUFFER REGISTER HIGH BYTE – 0X407

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x2 | rg_man_wr_width [10:8]<br><br>This number specifies the number of pixels per line in the buffer. |

### 3D DE-INTERLACE DITB0 REGISTER – 0X408

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | Reserved |
| 6:0 | R/W | 0x00 | rg_ditb0 |

### 3D DE-INTERLACE DITB4 REGISTER – 0X409

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | Reserved |
| 6:0 | R/W | 0x04 | rg_ditb4 |

### 3D DE-INTERLACE DITB8 REGISTER – 0X40A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | Reserved |
| 6:0 | R/W | 0x08 | rg_ditb8 |

### 3D DE-INTERLACE DITB12 REGISTER – 0X40B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | Reserved |
| 6:0 | R/W | 0x0C | rg_ditb12 |

### 3D DE-INTERLACE DITB16 REGISTER – 0X40C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | Reserved |
| 6:0 | R/W | 0x40 | rg_ditb16 |

### 3D DE-INTERLACE DITB24 REGISTER – 0X40D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | Reserved |
| 6:0 | R/W | 0x40 | rg_ditb24 |

### 3D DE-INTERLACE DITB32 REGISTER – 0X40E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | Reserved |
| 6:0 | R/W | 0x40 | rg_ditb32 |

### 3D DE-INTERLACE DITB48 REGISTER – 0X40F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | Reserved |
| 6:0 | R/W | 0x40 | rg_ditb48 |

### 3D DE-INTERLACE AFM_YF02_THD REGISTER – 0X410

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x30 | rg_afm_yf02_thd |

### 3D DE-INTERLACE AFM_YF02_H_THD REGISTER – 0X411

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x68 | rg_afm_yf02_h_thd |

### 3D DE-INTERLACE AFM_YF01_THD REGISTER – 0X412

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x08 | rg_afm_yf01_thd |

### 3D DE-INTERLACE AFM_YF01_H_THD REGISTER – 0X413

| Bit | R/W | Default | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 7:0 | R/W | 0x80 | rg_afm_yf01_h_thd |

### 3D DE-INTERLACE 32 CAPTION THRESHOLD REGISTER – 0X414

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x01 | rg_32_caption_thd |

### UP-SCALE CONTROL REGISTER – 0X415

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | rg_ups1_vsharp_gain<br><br>There is a sharpener before upscaler. This register controls the sharpening weighting. |
| 3:2 | R/W | 0x0 | Reserved |
| 1 | R/W | 0x0 | rg_ups1_vlimit_on<br><br>1 = remove overshoot<br>0 = turn off overshoot removal |
| 0 | R/W | 0x0 | rg_disable_man_disp<br><br>1 = black out the panel<br>0 = normal operation |

### UP-SCALE BACKGROUND COLOR – 0X416

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | rg_BGcolor<br><br>The background color used when there is no valid data in the main stream. This is also the color used in the border of the video window. The color is specified as 2 bit Y, 3 bit Cb, 2 bit Cr |

### UP-SCALE HORIZONTAL FACTOR LOW BYTE REGISTER – 0X417

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | rg_ups1_hscale [7:0]<br><br>When rg_ups1_hscale == 0x1000, the horizontal scaling factor is 1. When rg_ups1_hscale is any number smaller than 0x1000, the main path picture is up scaled horizontally.<br><br>When rg_ups1_hscale Is larger than 0x1000, the main video window is down scaled horizontally. The down scaling has to be less than 0x2000 |

### UP-SCALE HORIZONTAL FACTOR HIGH BYTE REGISTER – 0X418

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R | 0x0 | Reserved |
| 4:0 | R/W | 0x10 | rg_ups1_hscale [12:8]<br><br>When rg_ups1_hscale == 0x1000, the horizontal scaling factor is 1.<br>When rg_ups1_hscale is any number smaller than 0x1000, the main path picture is up scaled horizontally.<br>When rg_ups1_hscale Is greater than 0x1000, the main video window is down scaled horizontally. The down scaling has to be less than 0x2000 |

### UP-SCALE VERTICAL FACTOR LOW BYTE REGISTER – 0X419

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | rg_ups1_vscale [7:0]<br><br>When rg_ups1_vscale == 0x400, the vertical scaling factor is 1<br>When rg_ups1_vscale is any number smaller than 0x400, the main path picture is up scaled vertically.<br>When rg_ups1_vscale is any number greater than 0x400, the main path picture is down scaled vertically The downscaling has to be less than 0x500 |

### UP-SCALE VERTICAL FACTOR HIGH BYTE REGISTER – 0X41A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x4 | rg_ups1_vscale [10:8]<br><br>When rg_ups1_vscale == 0x400, the vertical scaling factor is 1<br>When rg_ups1_vscale is any number smaller than 0x400, the main path picture is up scaled vertically.<br>When rg_ups1_vscale is any number greater than 0x400, the main path picture is down scaled vertically The downscaling has to be less than 0x500 |

### UP-SCALE HORIZONTAL START LOW BYTE REGISTER – 0X41B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | rg_ups1_Xst [7:0]<br><br>Main Video Window Horizontal Starting location, unit is pixel |

### UP-SCALE HORIZONTAL START HIGH BYTE REGISTER – 0X41C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | rg_ups1_Xst [10:8]<br><br>Main Video Window Horizontal Starting location, unit is pixel |

### UP-SCALE VERTICAL START LOW BYTE REGISTER – 0X41D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | rg_ups1_Yst [7:0]<br><br>Main Video Window Vertical Starting location, unit is line |

### UP-SCALE VERTICAL START HIGH BYTE REGISTER – 0X41E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | rg_ups1_Yst [10:8]<br><br>Main Video Window Vertical Starting location, unit is line |

### UP-SCALE OUTPUT WIDTH LOW BYTE REGISTER – 0X41F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xD0 | rg_ups1_Xmax [7:0]<br><br>Main Video Window width after scaling, unit is pixel |

### UP-SCALE OUTPUT WIDTH HIGH BYTE REGISTER – 0X420

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x2 | rg_ups1_Xmax [10:8]<br><br>Main Video Window width after scaling, unit is pixel |

### UP-SCALE OUTPUT HEIGHT LOW BYTE REGISTER – 0X421

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xE0 | rg_ups1_Ymax [7:0]<br><br>Main Video Window height after scaling, unit is line |

### UP-SCALE OUTPUT HEIGHT HIGH BYTE REGISTER – 0X422

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x1 | rg_ups1_Ymax [10:8]<br><br>Main Video Window height after scaling, unit is line |

### UP-SCALE X BOUNDARY WIDTH REGISTER – 0X423

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **rg_ups1_Xoff**<br><br>Main Video Window horizontal colored boundary width.<br>When the panorama / Water glass mode is on, this offset specifies the width of the band at two side of the picture that will be scaled into panorama / water glass |

### UP-SCALE Y BOUNDARY WIDTH REGISTER – 0X424

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **rg_ups1_Yoff**<br><br>Main Video Window vertical colored boundary width. |

### UP-SCALE PANORAMA / WATER GLASS CONTROL REGISTER – 0X425

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **rg_ups1_panorama**<br><br>1 = Enable Panorama / Water glass scaling.<br>0 = Disable Panorama / Water glass scaling |
| 6 | R/W | 0x0 | **rg_ups1_hs_type**<br><br>0 = Concave (Panorama)<br>1 = Convex (Water Glass) |
| 5:4 | R/W | 0x0 | **rg_ups1_hs_step**<br><br>When Panorama / Water glass is on, the scaling factor increased/decreased every N pixels at the input side.<br>0 : N = 1, 1: N = 2, 2: N = 4, 3: N = 8. |
| 3:0 | R/W | 0x0 | **rg_ups1_hs_inc**<br><br>The scaling factor change between steps of input pixels as specified in bit 5:4. This change is added/subtracted from the scaling factor of the previous step closer to the center. |

### UP-SCALE HORIZONTAL CENTER SCALE FACTOR LOW BYTE REGISTER – 0X426

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **rg_ups1_hs_center [7:0]**<br><br>The scaling factor at the center portion. When panorama/water glass mode is on, the horizontal scaling factor rg_ups1_hscale is not used. Instead, rg_ups1_hs_center is used for the center portion. |

### UP-SCALE HORIZONTAL CENTER SCALE FACTOR HIGH BYTE REGISTER – 0X427

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R | 0x00 | Reserved |
| 4:0 | R/W | 0x00 | rg_ups1_hs_center [12:8]<br><br>The scaling factor at the center portion. When panorama/water glass mode is on, the horizontal scaling factor rg_ups1_hscale is not used. Instead, rg_ups1_hs_center is used for the center portion. |

### 2D DE-INTERLACE CONTROL REGISTER – 0X428

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | rg_2ddi_ctl |

### 2D DE-INTERLACE THRESHOLD 1 REGISTER – 0X429

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x28 | rg_2ddi_thd1 |

### 2D DE-INTERLACE THRESHOLD 2 REGISTER – 0X42A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x1E | rg_2ddi_thd2 |

### 2D DE-INTERLACE THRESHOLD 3 REGISTER – 0X42B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x28 | rg_2ddi_thd3 |

### 2D DE-INTERLACE THRESHOLD 4 REGISTER – 0X42C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x28 | rg_2ddi_thd4 |

### 2D DE-INTERLACE THRESHOLD 5 REGISTER – 0X42D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x24 | rg_2ddi_thd5 |

### 2D DE-INTERLACE THRESHOLD 6 REGISTER – 0X42E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x6F | rg_2ddi_thd6 |

### 3D DE-INTERLACE CONTROL REGISTER – 0X42F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | rg_3ddi_ctl |

### 3D DE-INTERLACE THRESHOLD 1 REGISTER – 0X430

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0A | rg_3ddi_thd1 |

### 3D DE-INTERLACE THRESHOLD 2 REGISTER – 0X431

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x95 | rg_3ddi_thd2 |

### UPSCALER OPTION REGISTER – 0X432

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | R | 0 | Reserved |
| 1 | R/W | 0 | Ups1_dbg[1]<br><br>Vertical upscale parameter select |
| 0 | R/W | 0 | Ups1_dbg[0]<br><br>Horizontal upscale parameter select |

### SINGLE BOX 4 TO 7 REGISTER – 0X440 TO 0X46B

Single box 4 to 7 are same as single box 0 to 3. Please refer to register 0x50C 0x538

### MOUSE BASE ADDRESS LOW BYTE REGISTER – 0X46C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | MOUSE_BASE_ADDR[7:0]<br><br>There are 16 mouse shape can be written to SDRAM. This is address for the first mouse. All mouse data is continuous. This address is 8 bytes unit. |

### MOUSE BASE ADDRESS MIDDLE BYTE REGISTER – 0X46D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | MOUSE_BASE_ADDR[15:8]<br><br>There are 16 mouse shape can be written to SDRAM. This is address for the first mouse. All mouse data is continuous. This address is 8 bytes unit. |

### MOUSE BASE ADDRESS HIGH BYTE REGISTER – 0X46E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | MOUSE_BASE_ADDR[23:16]<br><br>There are 16 mouse shape can be written to SDRAM. This is address for the first mouse. All mouse data is continuous. This address is 8 bytes unit. |

### MOUSE UPDATE ENABLE REGISTER – 0X46F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | R | 0 | Reserved |
| 0 | R/W | 0 | MOUSE_UPDATE_EN<br><br>Set this bit to high, hardware will automatically load mouse data from SDRAM to local SRAM. This bit will be clear when load is done. |

### MOUSE INDEX REGISTER – 0X470

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R | 0 | Reserved |
| 4 | R/W | 0 | MOUSE_BUF: This bit select which buffer will be updated.<br><br>0: first mouse buffer<br>1: second mouse buffer |
| 3:0 | R/W | 0 | MOUSE_INDEX[3:0]<br><br>This register set which mouse shape will be loaded from SDRAM. Up to 16 mouse shapes can be saved in SDRAM. If user wants more than 16 mouse shape, user can set different mouse base address. |

### WINDOW 32 VERTICAL POSITION REGISTER LOW BYTE – 0X472

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | VPOS_32[7:0]<br><br>Window 32 vertical position. This window will show memory content instead of video. It can be used for LOGO display. 1 line unit. |

### WINDOW 32 HORIZONTAL POSITION REGISTER LOW BYTE – 0X473

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | HPOS_32[7:0]<br><br>Window 32 horizontal position. This window will show memory content instead of video. It can be used for LOGO display. 4 pixels unit. |

### WINDOW 32 VERTICAL SIZE REGISTER LOW BYTE – 0X474

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | VSIZE_32[7:0]<br><br>Window 32 vertical size. This window will show memory content instead of video. It can be used for LOGO display. 1 line unit. |

### WINDOW 32 HORIZONTAL SIZE REGISTER LOW BYTE – 0X475

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | HSIZE_32[7:0]<br><br>Window 32 horizontal size. This window will show memory content instead of video. It can be used for LOGO display. 4 pixels unit. |

### WINDOW 32 VERTICAL AND HORIZONTAL POSITION REGISTER HIGH BYTE – 0X476

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R | 0 | Reserved |
| 5:4 | R/W | 0 | HPOS_32[9:8]<br><br>Window 32 horizontal position. This window will show memory content instead of video. It can be used for LOGO display. 4 pixels unit. |
| 3:0 | R/W | 0 | VPOS_32[11:8]<br><br>Window 32 vertical position. This window will show memory content instead of video. It can be used for LOGO display. 1 line unit. |

### WINDOW 32 VERTICAL AND HORIZONTAL SIZE REGISTER HIGH BYTE – 0X477

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R | 0 | Reserved |
| 4 | R/W | 0 | HSIZE_32[8]<br><br>Window 32 horizontal size. This window will show memory content instead of video. It can be used for LOGO display. 4 pixels unit. |
| 2:0 | R/W | 0 | VSIZE_32[10:8]<br><br>Window 32 vertical size. This window will show memory content instead of video. It can be used for LOGO display. 1 line unit. |

**Register 0x478 to 0x47D are for dual monitor window 32.**

### WINDOW 32 ENABLE REGISTER – 0X47E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | R | 0 | Reserved |
| 1 | R/W | 0 | CH32_BND_EN<br><br>Window 32 boundary enable |
| 0 | R/W | 0 | CH32_EN<br><br>Window 32 DRAM data display enable |

**Register 0x47F is for dual monitor window 32.**

### DISPLAY MISC. CONTROL REGISTER – 0X480

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 1 | **OUT_DIS**<br><br>Output disable. If set to high, VGA output will have all signals low. |
| 6 | R/W | 0 | **MDBOX_POS_SEL**<br><br>Motion box position selection. If set to high, motion box position is automatically set to channel position |
| 5 | R/W | 0 | **Reserved** |
| 4 | R/W | 0 | **BND_EN**<br><br>Channel boundary enable. Only enabled channel has boundary. |
| 3 | R/W | 0 | **COLBAR_EN**<br><br>Color bar enable. Output is color bar instead of live video. |
| 2 | R/W | 0 | **SWITCH_EN**<br><br>Switch to SDRAM contents if channel is disabled. If this bit is set to low, disabled channel will show background. If this bit is set to high, background is not shown. The contents in SDRAM will shows up. This is used for debug |
| 1 | R/W | 0 | **HS_POL:** Horizontal sync polarity<br><br>1: positive<br>0: negative |
| 0 | R/W | 0 | **VS_POL:** Vertical sync polarity<br><br>1: positive<br>0: negative |

### VERTICAL SYNC DELAY AND BOUNDARY WIDTH REGISTER – 0X481

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x02 | **BND_WIDTH**<br><br>Boundary width. Unit is pixel. |
| 3:0 | R/W | 0x0F | **VSDEL**<br><br>Vertical sync delay cycle after horizontal sync |

### BOUNDRARY RED COLOR INTENSITY REGISTER – 0X482

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xFF | **BND_R[7:0]**<br><br>Boundary red color. |

### BOUNDRARY GREEN COLOR INTENSITY REGISTER – 0X483

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xFF | BND_G[7:0]<br><br>Boundary green color. |

### BOUNDRARY BLUE COLOR INTENSITY REGISTER – 0X484

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xFF | BND_B[7:0]<br><br>Boundary blue color. |

### HORIZONTAL UPSCALE FACTOR LOW BYTE REGISTER – 0X485

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | POS_HSCALE[7:0]<br><br>Channel position information upscale factor. 0x1000 means no upscale. Value more than 0x1000 means upscale. If image is upscaled, this register must be set correctly. |

### HORIZONTAL UPSCALE FACTOR HIGH BYTE REGISTER – 0X486

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x10 | POS_HSCALE[15:8]<br><br>Channel position information upscale factor. 0x1000 means no upscale. Value more than 0x1000 means upscale. If image is upscaled, this register must be set correctly. |

### VERTICAL UPSCALE FACTOR LOW BYTE REGISTER – 0X487

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | POS_VSCALE[7:0]<br><br>Channel position information upscale factor. 0x1000 means no upscale. Value more than 0x1000 means upscale. If image is upscaled, this register must be set correctly. |

### VERTICAL UPSCALE FACTOR HIGH BYTE REGISTER – 0X488

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x10 | POS_VSCALE[15:8]<br><br>Channel position information upscale factor. 0x1000 means no upscale. Value more than 0x1000 means upscale. If image is upscaled, this register must be set correctly. |

### POSITION UPSCALE ENABLE REGISTER – 0X489

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | R | 0 | Reserved |
| 0 | R/W | 0 | POS_UPS_EN<br><br>Use upscaled position information if this bit set to high. |

### OSD BLINK RATE REGISTER – 0X48A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | R | 0 | Reserved |
| 1:0 | R/W | 0x00 | OSD_BLINK_TIME: OSD blinking frequency control<br><br>00: blinking on each 32 frames<br>01: blinking on each 16 frames<br>10: blinking on each 8 frames<br>11: blinking on each 4 frames |

### MOTION BOX CURSOR VERTICAL POSITION REGISTER – 0X48B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0 | CUR1_VPOS[3:0]<br><br>Channel 1 Cursor Vertical Position |
| 3:0 | R/W | 0 | CUR0_VPOS[3:0]<br><br>Channel 0 Cursor Vertical Position |

**Similar registers are assigned to control channel 2 to 16 using register 0x48C – 0x492 respectively.**

### MOTION BOX OUT BOUNDARY RED COLOR REGISTER – 0X493

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | MDBOX_OBR[7:0]<br><br>Red color for motion box out boundary |

### MOTION BOX OUT BOUNDARY GREEN COLOR REGISTER – 0X494

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | MDBOX_OBG[7:0]<br><br>Green color for motion box out boundary |

### MOTION BOX OUT BOUNDARY BLUE COLOR REGISTER – 0X495

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | MDBOX_OBB[7:0]<br><br>Blue color for motion box out boundary |

### MOTION BOX INNER BOUNDARY RED COLOR REGISTER – 0X496

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | MDBOX_IBR[7:0]<br><br>Red color for motion box Inner boundary |

### MOTION BOX INNER BOUNDARY GREEN COLOR REGISTER – 0X497

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | MDBOX_IBG[7:0]<br><br>Green color for motion box Inner boundary |

### MOTION BOX INNER BOUNDARY BLUE COLOR REGISTER – 0X498

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | MDBOX_IBB[7:0]<br><br>Blue color for motion box Inner boundary |

### MOTION BOX MASK RED COLOR REGISTER – 0X499

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | MDBOX_MSKR[7:0]<br><br>Red color for motion box mask area |

### MOTION BOX MASK GREEN COLOR REGISTER – 0X49A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | MDBOX_MSKG[7:0]<br><br>Green color for motion box mask area |

### MOTION BOX MASK BLUE COLOR REGISTER – 0X49B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | MDBOX_MSKB[7:0]<br><br>Blue color for motion box mask area |

### MOTION BOX PLANE RED COLOR REGISTER – 0X49C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | MDBOX_PR[7:0]<br><br>Red color for motion box plane area |

### MOTION BOX PLANE GREEN COLOR REGISTER – 0X49D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | MDBOX_PG[7:0]<br><br>Green color for motion box plane area |

### MOTION BOX PLANE BLUE COLOR REGISTER – 0X49E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | MDBOX_PB[7:0]<br><br>Blue color for motion box plane area |

### HORIZONTAL START LOW BYTE REGISTER – 0X4A0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | HSTART[7:0]<br><br>Horizontal start in SDARM for read out. Unit is 4 pixels |

### HORIZONTAL START HIGH BYTE REGISTER – 0X4A1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | RO | - | Reserved |
| 1:0 | R/W | 0 | HSTART[9:8]<br><br>Horizontal start in SDARM for read out. Unit is 4 pixels |

### VERTICAL START LOW BYTE REGISTER – 0X4A2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | VSTART[7:0]<br><br>Vertical start in SDARM for read out. Unit is 1 line. |

### VERTICAL START HIGH BYTE REGISTER – 0X4A3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RO | - | Reserved |
| 3:0 | R/W | 0 | VSTART[11:8]<br><br>Vertical start in SDARM for read out. Unit is 1 line. |

### RTI WNDOW REGISTER – 0X4A4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xFF | VTT_WIN[7:0]<br><br>Vertical Total can be changed according to input video frame rate. If difference is more than this number, VTT will not be adjusted. Unit is 1 line. |

### VERTICAL TOTAL ADJUST MODE REGISTER – 0X4A5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | RO | - | Reserved |
| 2 | R/W | 0x00 | VTT_ADJUST<br><br>It is used for manual mode. After set this bit, VTT will be changed according to current input video frame rate. |
| 1 | R/W | 0 | VTT_ADJUST_MODE<br><br>1: auto mode, adjust VTT each frame<br>0: manual mode, adjust VTT after set VTT_ADJUST |
| 0 | R/W | 0 | VTT_ADJUST_EN<br><br>Enable VTT adjustment. If set to 0, use register setting. If set to 1, use input video frame rate. |

### VERTICAL TOTAL FROM INPUT VIDEO LOW BYTE REGISTER – 0X4A6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RO | - | NEW_VTT_RGBW[7:0]<br><br>It is calculated from input video. The number is for two fields. |

### VERTICAL TOTAL FROM INPUT VIDEO HIGH BYTE REGISTER – 0X4A7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RO | - | Reserved |
| 3:0 | RO | - | NEW_VTT_RGBW[11:8]<br><br>It is calculated from input video. The number is for two fields. |

### VERTICAL TOTAL AFTER ADJUSTMENT BYTE REGISTER – 0X4A8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RO | - | NEW_VTT[7:0]<br><br>It is new VTT which is used in output timing. The number is for one field. |

### VERTICAL TOTAL AFTER ADJUSTMENT HIGH REGISTER – 0X4A9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | RO | - | Reserved |
| 2:0 | RO | - | NEW_VTT[10:8]<br><br>It is new VTT which is used in output timing. The number is for one field. |

**BOUNDARY ENABLE REGISTER – 0X4AA**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0xF | **BND_EN_1[3:0]: Boundary Enable for Channel 1**<br><br>[0]: left side boundary enable<br>[1]: right side boundary enable<br>[2]: top side boundary enable<br>[3]: bottom side boundary enable |
| 3:0 | R/W | 0xF | **BND_EN_0[3:0]: Boundary Enable for Channel 0**<br><br>[0]: left side boundary enable<br>[1]: right side boundary enable<br>[2]: top side boundary enable<br>[3]: bottom side boundary enable |

**Similar register 0x4AB – 0X4B3 are assigned to control channel 12 – 19 respectively.**

**PS2 MOUSE CONTROL REGISTER – 0X4B4**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xFF | **PS2CON[7:0]**<br><br>[0]: 1 = PS2 enable<br>[1]: 1 = Data write enable<br>[3:2]: MaxByte<br>[7:4]: MaxNoSig |

**PS2 MOUSE WRITE DATA REGISTER – 0X4B5**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xFF | **PS2DATA_IN**<br><br>The data written to PS2 mouse |

**PS2 MOUSE WRITE DATA REGISTER 1– 0X4B6**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | - | **PS2DATA_OUT[7:0]**<br><br>The data read from PS2 mouse |

**PS2 MOUSE WRITE DATA REGISTER 2– 0X4B7**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | - | **PS2DATA_OUT[15:8]**<br><br>The data read from PS2 mouse |

**PS2 MOUSE WRITE DATA REGISTER 3– 0X4B8**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | - | **PS2DATA_OUT[23:16]**<br><br>The data read from PS2 mouse |

### PS2 MOUSE WRITE DATA REGISTER 4– 0X4B9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | - | PS2DATA_OUT[31:24]<br><br>The data read from PS2 mouse |

### PS2 INTERRUPT REGISTER– 0X4BA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | RO | - | Reserved |
| 0 | R/W | - | PS2_INT_REG<br><br>1 = PS2 interrupt. Clear by write |

### EXTERNAL OSD CONTROL REGISTER– 0X4BB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | RO | - | Reserved |
| 4 | R/W | 0 | EOSD_SEL<br><br>0: select external OSD<br>1: select internal pure color |
| 3 | R/W | 0 | EOSD_EN<br><br>1 = External OSD enable |
| 2 | R/W | 0 | EOSD_MODE<br><br>0: ex_osd_en high active<br>1: ex_osd_en low active |
| 1 | R/W | 0 | EOSD_EN_DIS<br><br>0: use ex_osd_en<br>1: always enable, no transparent |
| 0 | R/W | 0 | EOSD_ALPHA_EN<br><br>1 = External OSD alpha blending enable |

### EXTERNAL OSD ALPHA REGISTER– 0X4BC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RO | - | Reserved |
| 3:0 | R/W | 0 | EOSD_ALPHA<br><br>Alpha blending alpha number for external OSD layer.<br>The output image will be: Video_data * alpha + eosd_data * (1 – alpha).<br>Here alpha = OSD_ALPHA / 16. Maximum is 15. |

### EXTERNAL OSD COLOR R REGISTER– 0X4BD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | EOSD_R<br><br>Red Color for external OSD. It is used when EOSD_SEL is set to high. |

### EXTERNAL OSD COLOR G REGISTER– 0X4BE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | EOSD_G<br><br>Green Color for external OSD. It is used when EOSD_SEL is set to high. |

### EXTERNAL OSD COLOR B REGISTER– 0X4BF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | EOSD_B<br><br>Blue Color for external OSD. It is used when EOSD_SEL is set to high. |

### INTERRUPT MASK REGISTER– 0X4C0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RO | - | Reserved |
| 3 | RW | 1 | INT_MASK_OSG_VBLANK<br><br>1 = OSG vblank interrupt mask enable |
| 2 | R/W | 1 | INT_MASK_PS2<br><br>1 = PS2 interrupt mask enable |
| 1 | R/W | 1 | INT_MASK_OSG_BUSY<br><br>1 = OSG busy interrupt mask enable |
| 0 | R/W | 1 | INT_MASK_OSG_WAIT<br><br>1 = OSG wait interrupt mask enable |

### INTERRUPT STATUS REGISTER– 0X4C1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RO | - | Reserved |
| 3 | RO | - | INT_OSG_VBLANK<br><br>1 = OSG vertial blank toggle interrupt, write clear |
| 2 | RO | - | INT_PS2<br><br>1 = PS2 interrupt, write clear |
| 1 | RO | - | INT_OSG_BUSY<br><br>1 = OSG busy interrupt, write clear |
| 0 | RO | - | INT_OSG_WAIT<br><br>1 = OSG wait interrupt, write clear. When OSG write buffer (1KB) is almost empty, interrupt will be high. |

### FRAME RATE CONTROL DEBUG REGISTER– 0X4C2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | rd_buf_id_toggle<br><br>1: read buffer ID always toggle between odd and even<br>0: odd or even can be repeat |
| 6 | R/W | 0 | Rd_buf_inc<br><br>1: read buffer ID free running<br>0: increase read buffer ID according to write buffer ID |
| 5 | R/W | 0 | WR_PAGE_EN<br><br>Manual set write page enable, 1 = enable |
| 4 | R/W | 0 | RD_PAGE_EN<br><br>Manual set read page enable |
| 3:2 | R/W | 0 | WR_PAGE<br><br>Used in manual mode |
| 1:0 | R/W | 0 | RD_PAGE<br><br>Used in manual mode |

### MISC CONTROL REGISTER – 0X4C3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R/W | 3 | WR_BUF_OFST<br><br>This register should be set to 3. |
| 4 | R/W | 0 | INTERLACE_DBG<br><br>Interlaced mode debugging use. |
| 3 | R/W | 0 | VTT_INV<br><br>It is used in interlaced mode, it should be set to 0 |
| 2 | R/W | 0 | EVEN_REPEAT<br><br>0: repeat odd or even when needed<br>1: repeat even only |
| 1 | R/W | 0 | DMON_VS_EN<br><br>0: not use dual monitor vsync<br>1: use dual monitor vsync. This is used when main display data goes to dual monitor display. |
| 0 | R/W | 0 | INTERLACE<br><br>0: progressive mode<br>1: interlaced mode |

### BOUNDARY ENABLE REGISTER 2 – 0X4C4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0xF | **BND_EN_21[3:0]: Boundary Enable for Channel 21**<br><br>[0]: left side boundary enable<br>[1]: right side boundary enable<br>[2]: top side boundary enable<br>[3]: bottom side boundary enable |
| 3:0 | R/W | 0xF | **BND_EN_20[3:0]: Boundary Enable for Channel 20**<br><br>[0]: left side boundary enable<br>[1]: right side boundary enable<br>[2]: top side boundary enable<br>[3]: bottom side boundary enable |

**Similar register 0x4C5 – 0X4C9 are assigned to control channel 22 – 31 respectively.**

### IMAGE CONTROL REGISTER – 0X4CE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RO | - | **Reserved** |
| 6 | R/W | 0 | **OSD_CH_EN_SEL:** Simple OSD channel enable select<br><br>0: use rgb_interface channel enable<br>1: use another register set, [0x4F0] to [0x4F3] |
| 5 | R/W | 0 | **NOVID_SWITCH_EN**<br><br>In normal case, this bit set to 0. No video color will show up. If this bit set to 1, SDRAM data will show up. |
| 4 | R/W | 0 | **BND_CH_EN_SEL:** Boundary channel enable select.<br><br>0: use rgb_interface channel enable<br>1: use another register set, [0x4F0] to [0x4F3] |
| 3 | R/W | 1 | **PEAK_C_DUP:** In peak module, Cb/Cr using average or using duplicate data<br><br>0: average<br>1: duplicate |
| 2 | R/W | 0 | **GAIN_EN:** 1 = RGB digital gain enable |
| 1 | R/W | 0 | **PEAK_EN:** 1 = Peak enable |
| 0 | R | 0 | **Reserved** |

### PEAK CONTROL REGISTER – 0X4D0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RO | - | **Reserved** |
| 6 | R/W | 0 | **PEAK_HD_EN**<br><br>1 = enable |
| 5 | R/W | 1 | **PEAK_V_EN**<br><br>Peak vertical enable, 1 = enable |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4 | R/W | 1 | **PEAK_H_EN**<br><br>Peak horizontal enable, 1 = enable |
| 3:2 | RO | - | **Reserved** |
| 1 | R/W | | **PEAK_DIS_V_CON2,** 1 = disable |
| 0 | R/W | | **PEAK_DIS_V_CON1,** 1 = disable |

### DIGITAL R GAIN REGISTER – 0X4DA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x40 | **R_GAIN:** R digital gain. 0x40 is no gain. More than 0x40 means gain more than 1. |

### DIGITAL G GAIN REGISTER – 0X4DB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x40 | **G_GAIN:** G digital gain. 0x40 is no gain. More than 0x40 means gain more than 1. |

### DIGITAL B GAIN REGISTER – 0X4DC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x40 | **B_GAIN:** B digital gain. 0x40 is no gain. More than 0x40 means gain more than 1. |

### R OFFSET REGISTER – 0X4DD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **R_OFST[7:0]**<br><br>R offset. Rout = Rin * R_GAIN + R_OFST. This value is 2's complement value. R_OFST can be negative data. |

### G OFFSET REGISTER – 0X4DE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **G_OFST[7:0]**<br><br>G offset. Rout = Rin * R_GAIN + R_OFST. This value is 2's complement value. R_OFST can be negative data. |

### B OFFSET REGISTER – 0X4DF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **B_OFST[7:0]**<br><br>B offset. Rout = Rin * R_GAIN + R_OFST. This value is 2's complement value. R_OFST can be negative data. |

### BOUNDARY CHANNEL ENABLE REGISTER 1 – 0X4F0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BND_CH_EN[7:0]:** Control channel 7 - 0<br><br>Boundary channel enable. This register is only used when BND_CH_EN_SEL is high. This is only used when user want to show boundary when channel is not enable. |

### BOUNDARY CHANNEL ENABLE REGISTER 2 – 0X4F1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BND_CH_EN[15:8]:** Control live channel 15 - 8<br><br>Boundary channel enable. This register is only used when BND_CH_EN_SEL is high. This is only used when user want to show boundary when channel is not enable. |

### BOUNDARY CHANNEL ENABLE REGISTER 3 – 0X4F2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BND_CH_EN[23:16]:** Control live channel 23 - 16<br><br>Boundary channel enable. This register is only used when BND_CH_EN_SEL is high. This is only used when user want to show boundary when channel is not enable. |

### BOUNDARY CHANNEL ENABLE REGISTER 4 – 0X4F3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BND_CH_EN[31:24]:** Control live channel 31 - 24<br><br>Boundary channel enable. This register is only used when BND_CH_EN_SEL is high. This is only used when user want to show boundary when channel is not enable. |

### FRAME RATE CONTROL ENABLE REGISTER 1 – 0X4F4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xFF | **FRSC_CH_EN[7:0]:** Control live channel 7 - 0<br><br>This register and the next three forms a group. These 32 bits will determine which channel are included in the frame rate control decision. This is only for debugging purpose. In default, all channels will be included. |

### FRAME RATE CONTROL ENABLE REGISTER 2 – 0X4F5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xFF | **FRSC_CH_EN[15:8]:** Control live channel 15 - 8 |

### FRAME RATE CONTROL ENABLE REGISTER 3 – 0X4F6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xFF | **FRSC_CH_EN[23:16]:** Control PB channel 7 - 0 |

### FRAME RATE CONTROL ENABLE REGISTER 4 – 0X4F7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xFF | **FRSC_CH_EN[31:24]:** Control PB channel 15 - 8 |

### RED COLOR FOR NO VIDEO CHANNEL – 0X4F8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **NOVID_R**<br><br>This color will show up when the channel boundary is enable, but channel has no video. |

### GREEN COLOR FOR NO VIDEO CHANNEL – 0X4F9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **NOVID_G**<br><br>This color will show up when the channel boundary is enable, but channel has no video. |

### BLUE COLOR FOR NO VIDEO CHANNEL – 0X4FA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xFF | **NOVID_B**<br><br>This color will show up when the channel boundary is enable, but channel has no video. |

### HORIZONTAL TOTAL REGISTER 1 – 0X500

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x2F | **HTT[7:0]**<br><br>Pixel count of one horizontal line including blanking. This value should minus 1 by real horizontal total. In interlace mode, if down scaler and TV encoder is used, the vclk is set to 108MHz, so HTT will be set to 4*1716-1=6863 |

### HORIZONTAL TOTAL REGISTER 2 – 0X501

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R | - | **Reserved** |
| 3:0 | R/W | 0x07 | **HTT[12:8]**<br><br>Pixel count of one horizontal line including blanking. This value should minus 1 by real horizontal total. In interlace mode, if down scaler and TV encoder is used, the vclk is set to 108MHz, so HTT will be set to 4*1716-1=6863 |

### VERTICAL TOTAL REGISTER 1 – 0X502

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x37 | VTT[7:0]<br><br>Line count of one display frame including blanking. This value should minus 1 by real vertical total. In interlace mode, this value is set to lines of odd filed. Even field is this value plus 1. |

### VERTICAL TOTAL REGISTER 2 – 0X503

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | - | Reserved |
| 2:0 | R/W | 0x04 | VTT[10:8]<br><br>Line count of one display frame including blanking. This value should minus 1 by real vertical total. In interlace mode, this value is set to lines of odd filed. Even field is this value plus 1. |

### HORIZONTAL DISPLAY END REGISTER 1 – 0X504

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x8F | HDE[7:0]<br><br>Active display pixel count of one horizontal line. This value should minus 1 by real value. |

### HORIZONTAL DISPLAY END REGISTER 2 – 0X505

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | PHSYNC[9] |
| 6:4 | R | - | Reserved |
| 3:0 | R/W | 0x06 | HDE[10:8]<br><br>Active display pixel count of one horizontal line. This value should minus 1 by real value. |

### VERTICAL DISPLAY END REGISTER 1 – 0X506

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x19 | VDE[7:0]<br><br>Line count of active display frame. This value should minus 1 by real value. This value is set to active lines in one field. |

### VERTICAL DISPLAY END REGISTER 2 – 0X507

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | - | Reserved |
| 2:0 | R/W | 0x04 | VDE[10:8]<br><br>Line count of active display frame. This value should minus 1 by real value. This value is set to active lines in one field. |

### HORIZONTAL SYNC STARTING POSITION REGISTER – 0X508

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x30 | PHSYNC[8:1]<br><br>Horizontal sync pulse starting position after HDE (in two pixels increment) |

### VERTICAL SYNC STARTING POSITION REGISTER – 0X509

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x02 | PVSYNC[7:0]<br><br>Vertical sync pulse starting position after VDE |

### HORIZONTAL SYNC WIDTH REGISTER – 0X50A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x20 | HSPW[7:0]<br><br>Horizontal sync pulse width |

### VERTICAL SYNC WIDTH REGISTER – 0X50B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x06 | VSPW[7:0]<br><br>Vertical sync pulse width |

### SBOX0 CONTROL REGISTER – 0X50C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R/W | 0x0 | Reserved |
| 4 | R/W | 0x0 | Boundary Line enable<br><br>0 = disable<br>1 = enable |
| 3 | R/W | 0x0 | Box plane enable<br><br>0 = disable<br>1 = enable |
| 2 | R/W | 0x0 | Blinking enable<br><br>0 = disable<br>1 = enable |
| 1:0 | R/W | 0x0 | Mixing control<br><br>00 = 75% original pixel value / 25% plane (boundary) color mix<br>01 = 50% original pixel value / 50% plane (boundary) color mix<br>10 = 25% original pixel value / 75% plane (boundary) color mix<br>11 = plane (boundary) color |

### SBOX LINE WIDTH CONTROL REGISTER 1 – 0X511

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | Box3 Vertical line width control<br><br>00 = 1 line<br>01 = 2 line<br>10 = 3 line<br>11 = 4 line |
| 5:4 | R/W | 0x0 | Box3 Horizontal line width control<br><br>00 = 1 line<br>01 = 2 line<br>10 = 3 line<br>11 = 4 line |
| 3:2 | R/W | 0x0 | Box2 Vertical line width control<br><br>00 = 1 line<br>01 = 2 line<br>10 = 3 line<br>11 = 4 line |
| 1:0 | R/W | 0x0 | Box2 Horizontal line width control<br><br>00 = 1 line<br>01 = 2 line<br>10 = 3 line<br>11 = 4 line |

### SBOX LINE WIDTH CONTROL REGISTER 2 – 0X512

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | Box1 Vertical line width control<br><br>00 = 1 line<br>01 = 2 line<br>10 = 3 line<br>11 = 4 line |
| 5:4 | R/W | 0x0 | Box1 Horizontal line width control<br><br>00 = 1 line<br>01 = 2 line<br>10 = 3 line<br>11 = 4 line |
| 3:2 | R/W | 0x0 | Box0 Horizontal line width control<br><br>00 = 1 line<br>01 = 2 line<br>10 = 3 line<br>11 = 4 line |
| 1:0 | R/W | 0x0 | Box0 Horizontal line width control<br><br>00 = 1 line<br>01 = 2 line<br>10 = 3 line<br>11 = 4 line |

### SBOX0 HL CONTROL REGISTER 1 – 0X513

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Box0 Left Horizontal point SBOX0_HL[7:0], unit is pixel |

### SBOX0 HL CONTROL REGISTER 2 – 0X514

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | Box0 Left Horizontal point SBOX0_HL[10:8], unit is pixel |

### SBOX0 HR CONTROL REGISTER 1 – 0X51B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Box0 Right Horizontal point SBOX0_HR[7:0], unit is pixel |

### SBOX0 HR CONTROL REGISTER 2 – 0X51C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | Box0 Right Horizontal point SBOX0_HR[10:8], unit is pixel |

### SBOX0 VT CONTROL REGISTER 1 – 0X523

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Box0 Top Vertical point SBOX0_VT[7:0], unit is line |

### SBOX0 VT CONTROL REGISTER 2 – 0X524

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | Box0 Top Vertical point SBOX0_VT[10:8], unit is line |

### SBOX0 VB CONTROL REGISTER 1 – 0X52B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Box0 Bottom Vertical point SBOX0_VB[7:0], unit is line |

### SBOX0 VB CONTROL REGISTER 2 – 0X52C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | Box0 Bottom Vertical point SBOX0_VB[10:8], unit is line |

### BOX BOUNDARY COLOR R BYTE – 0X533

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color red byte[7:0] |

### BOX BOUNDARY COLOR G BYTE – 0X534

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color green byte[7:0] |

### BOX BOUNDARY COLOR B BYTE – 0X535

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color blue byte[7:0] |

### BOX PLANE COLOR R BYTE – 0X536

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color red byte[7:0] |

### BOX PLANE COLOR G BYTE – 0X537

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color green byte[7:0] |

### BOX PLANE COLOR B BYTE – 0X538

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color blue byte[7:0] |

### SCREEN BACKGROUND COLOR R BYTE – 0X539

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color red byte[7:0] |

### SCREEN BACKGROUND COLOR G BYTE – 0X53A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color green byte[7:0] |

### SCREEN BACKGROUND COLOR B BYTE – 0X53B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color blue byte[7:0] |

### MOUSE0 HORIZONTAL POSITION REGISTER 1 – 0X53C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Mouse0 Horizontal position MOUSE0_HPOS[7:0] |

### MOUSE0 HORIZONTAL POSITION REGISTER 2 – 0X53D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | Mouse0 Horizontal position MOUSE0_HPOS[10:8] |

### MOUSE0 VERTICAL POSITION REGISTER 1 – 0X53E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Mouse0 Vertical position MOUSE0_VPOS[7:0] |

### MOUSE0 VERTICAL POSITION REGISTER 2 – 0X53F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | Mouse0 Vertical position MOUSE0_VPOS[10:8] |

### MOUSE1 HORIZONTAL POSITION REGISTER 1 – 0X540

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Mouse0 Horizontal position MOUSE1_HPOS[7:0] |

### MOUSE1 HORIZONTAL POSITION REGISTER 2 – 0X541

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | Mouse0 Horizontal position MOUSE1_HPOS[10:8] |

### MOUSE1 VERTICAL POSITION REGISTER 1 – 0X542

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Mouse0 Vertical position MOUSE1_VPOS[7:0] |

### MOUSE1 VERTICAL POSITION REGISTER 2 – 0X543

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | Mouse0 Vertical position MOUSE1_VPOS[10:8] |

### MOUSE CONTROL REGISTER – 0X544

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | Reserved |
| 6 | R/W | 0x0 | Mouse1 Enable<br><br>0 = disable<br>1= enable |
| 5:4 | R/W | 0x0 | Mouse1 Mixing control<br><br>00 = 75% original pixel value / 25% mouse color mix<br>01 = 50% original pixel value / 50% mouse color mix<br>10 = 25% original pixel value / 75% mouse color mix<br>11 = mouse color |
| 3 | R | 0x0 | Reserved |
| 2 | R/W | 0x0 | Mouse0 Enable<br><br>0 = disable<br>1= enable |
| 1:0 | R/W | 0x0 | Mouse0 Mixing control<br><br>00 = 75% original pixel value / 25% mouse color mix<br>01 = 50% original pixel value / 50% mouse color mix<br>10 = 25% original pixel value / 75% mouse color mix<br>11 = mouse color |

### MOUSE BACKGROUND COLOR R BYTE – 0X545

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color red byte[7:0] |

### MOUSE BACKGROUND COLOR G BYTE – 0X546

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color green byte[7:0] |

### MOUSE BACKGROUND COLOR B BYTE – 0X547

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color blue byte[7:0] |

### MOUSE FOREGROUND COLOR R BYTE – 0X548

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color red byte[7:0] |

### MOUSE FOREGROUND COLOR G BYTE – 0X549

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color green byte[7:0] |

### MOUSE FOREGROUND COLOR B BYTE – 0X54A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Color blue byte[7:0] |

### DISPLAY MODE CONTROL REGISTER – 0X54B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | DM_MD_SET: Dual monitor motion box enable (motion box, cursor)<br><br>1 : dual monitor motion box register set<br>0 : Main motion box register set |
| 6:3 | R | 0x0 | Reserved |
| 2 | R/W | 0x0 | LCD_YC_SWAP<br><br>Control luma and chroma position for BT1120 output |
| 1 | R/W | 0x0 | DIS_VIDEO<br><br>1: disable video<br>0: enable video |
| 0 | R/W | 0x0 | DE_INTLACE<br><br>0: weave<br>1: 2D or 3D |

### MOUSE RAM ADDRESS REGISTER – 0X54C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MOUSE_WR_LOC[7:0] |

### MOUSE RAM DATA REGISTER – 0X54D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MOUSE_WR_DATA |

### MOUSE REGISTER UPDATE ENABLE REGISTER – 0X54E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x3 | Repeat frame expected value |
| 3 | R/W | 0x0 | Enable |
| 2 | R/W | 0x1 | Disable |
| 1 | R/W | 0x0 | Masking Bit |
| 0 | R/W | 0x1 | MOUSE_REG_UPDATE<br><br>0: not update mouse related registers.<br>1: update mouse related registers<br><br>Before set mouse position registers, set this bit to 0. After set done, set this bit to 1. |

### MOUSE RAM WRITE ENABLE REGISTER – 0X54F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | R/W | 0x0 | MOUSE_WR_EN |

### MD BOX CONTROL REGISTER 0 – 0X550

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | MD Box Enable |
| 6 | R/W | 0x0 | MD Box operation mode select<br><br>0 = table mode<br>1 = motion display mode |
| 5 | R/W | 0x0 | Cursor Cell Enable<br><br>0 = disable<br>1 = enable |
| 4 | R/W | 0x0 | Out Boundary Cell Enable<br><br>0 = disable<br>1 = enable |
| 3 | R/W | 0x0 | Masking Plane Enable<br><br>0 = disable<br>1 = enable |
| 2 | R/W | 0x0 | Detection Plane Enable<br><br>0 = disable<br>1 = enable |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | R/W | 0x0 | Mixing control<br><br>00 = 75% original pixel value / 25% plane (boundary) color mix<br>01 = 50% original pixel value / 50% plane (boundary) color mix<br>10 = 25% original pixel value / 75% plane (boundary) color mix<br>11 = plane (boundary) color |

**This register is for channel 1. Similar register 0x551 – 0x55F is for channel 2 – channel 16.**

### MD BOX LINE WIDTH CONTROL REGISTER 0 – 0X560

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | Box1 Vertical line width control<br><br>00 = 1 line<br>01 = 2 line<br>10 = 3 line<br>11 = 4 line |
| 5:4 | R/W | 0x0 | Box1 Horizontal line width control<br><br>00 = 1 line<br>01 = 2 line<br>10 = 3 line<br>11 = 4 line |
| 3:2 | R/W | 0x0 | Box0 Vertical line width control<br><br>00 = 1 line<br>01 = 2 line<br>10 = 3 line<br>11 = 4 line |
| 1:0 | R/W | 0x0 | Box0 Horizontal line width control<br><br>00 = 1 line<br>01 = 2 line<br>10 = 3 line<br>11 = 4 line |

**This register is for box0, 1. Similar register 0x561 – 0x567 is for box 2 – box 16.**

### MD BOX 0 HORIZONTAL LEFT POSITION REGISTER 1 – 0X568

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Box 0 Horizontal Left Position<br><br>MDBOX0_HL[7:0] |

### MD BOX 0 HORIZONTAL LEFT POSITION REGISTER 2 – 0X569

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | Box 0 Horizontal Left Position<br><br>MDBOX0_HL[10:8] |

This register is for box0. Similar register 0x56A – 0x587 is for box 1 – box 16.

### MD BOX 0 VERTICAL TOP POSITION REGISTER 1 – 0X588

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Box 0 Vertical Top Position<br><br>MDBOX0_VT[7:0] |

### MD BOX 0 VERTICAL TOP POSITION REGISTER 2 – 0X589

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | Box 0 Vertical Top Position<br><br>MDBOX0_VT[10:8] |

This register is for box0. Similar register 0x58A – 0x5A7 is for box 1 – box 16.

### MD BOX 0 HORIZONTAL SIZE REGISTER 1 – 0X5A8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Box 0 Horizontal Size Position<br><br>MDBOX0_HS[7:0] |

### MD BOX 0 HORIZONTAL SIZE REGISTER 2 – 0X5A9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | Box 0 Horizontal Size Position<br><br>MDBOX0_HS[10:8] |

This register is for box0. Similar register 0x5AA – 0x5C7 is for box 2 – box 16.

### MD BOX 0 VERTICAL SIZE REGISTER 1 – 0X5C8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Box 0 Vertical Size Position<br><br>MDBOX0_VS[7:0] |

### MD BOX 0 VERTICAL SIZE REGISTER 2 – 0X5C9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x0 | Reserved |
| 2:0 | R/W | 0x0 | Box 0 Vertical Size Position<br><br>MDBOX0_VS[10:8] |

This register is for box0. Similar register 0x5CA – 0x5E7 is for box 2 – box 16.

### MD BOX HCELL REGISTER 0 – 0X5E8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | Box 1 Horizontal Cell Number<br><br>MDBOX1_HNUM[3:0] |
| 3:0 | R/W | 0x0 | Box 0 Horizontal Cell Number<br><br>MDBOX0_HNUM[3:0] |

This register is for box0. Similar register 0x5E9 – 0x5EF is for box 2 – box 16.

### MD BOX VCELL REGISTER 0 – 0X5F0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | Box 1 Vertical Cell Number<br><br>MDBOX1_VNUM[3:0] |
| 3:0 | R/W | 0x0 | Box 0 Vertical Cell Number<br><br>MDBOX0_VNUM[3:0] |

This register is for box0, 1. Similar register 0x5F1 – 0x5F7 is for box 2 – box 16.

### MD BOX CURSOR HORIZONTAL POSITION REGISTER 0 – 0X5F8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | Box 1 Horizontal Cursor Position<br><br>CUR1_HPOS[3:0] |
| 3:0 | R/W | 0x0 | Box 0 Horizontal Cursor Position<br><br>CUR0_HPOS[3:0] |

This register is for box0, 1. Similar register 0x5F8 – 0x5FF is for box 2 – box 16.

# HDMI Controller

## Introduction

**TW2880's HDMI Transmitter Controller supports HDMI 1.2 (High Definition Multimedia Interface) specification. This fully-compliant device provides a simple, low cost method of sending protected digital audio and video data stream to the compatible devices providing end users with a truly all-digital experience. TW2880 HDMI controller is backward compatible with DVI 1.0 which allows TW2880 to connect to any DVI capable display (for example, DTV, PDP, LCD TV or PC monitor).**

**TW2880's HDMI Transmitter Controller incorporates a flexible audio and video interface. An industry standard S/PDIF input accepts PCM encoded data as well as Dolby Digital, DTS, and other formats capable of being sent over S/PDIF. Four I²S inputs support DVD-Audio.**

**The controller consists of a transmitting Link block and a PHY block. The Link block supports video, audio input interfaces, I²C interface and DDC interface. The PHY Block is serializer / driver including three data channels and one clock channel. The signals of these channels are output to TMDS link.**

- Monitor Detection supported through Hot Plug and Receiver Detection
- Flexible power management

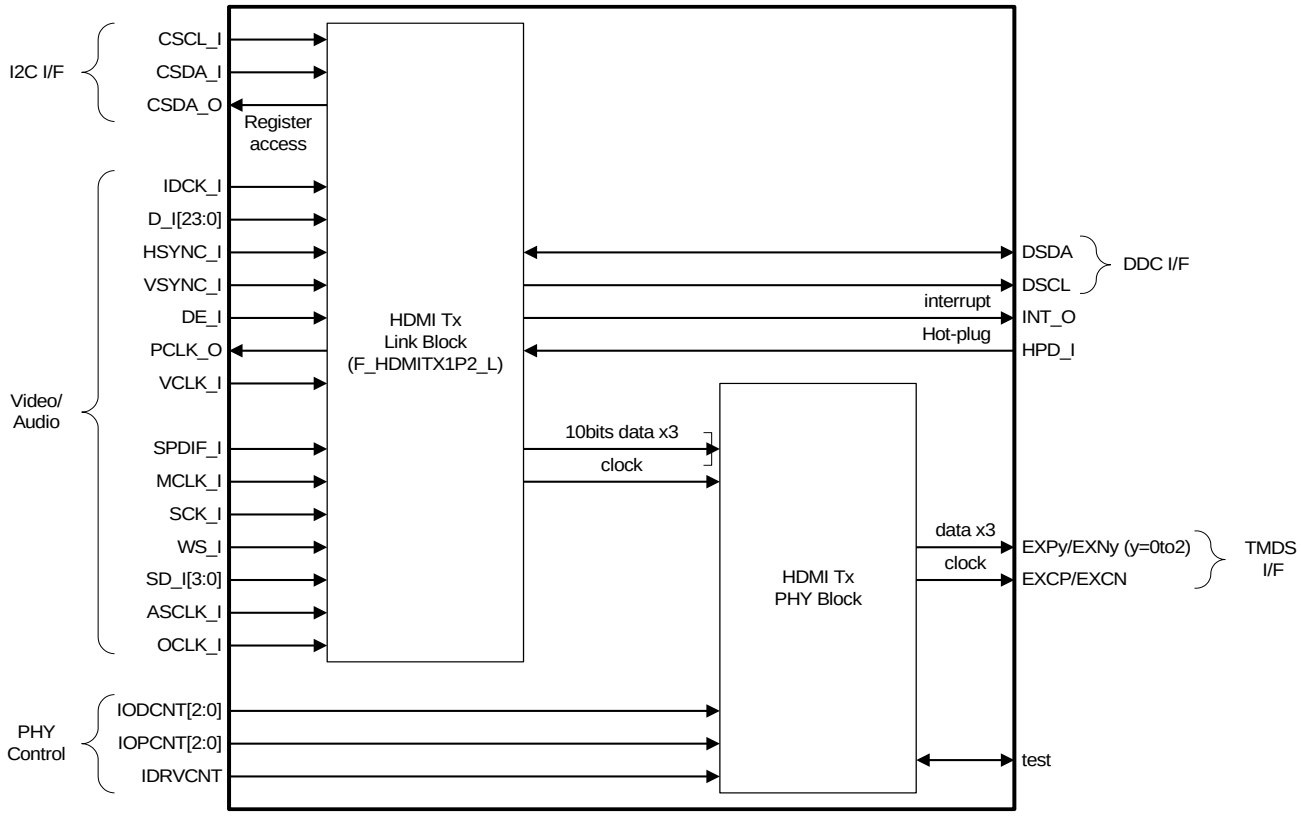
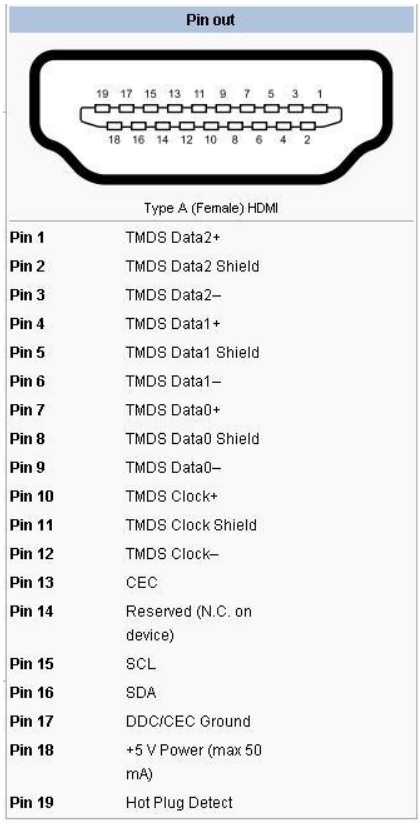


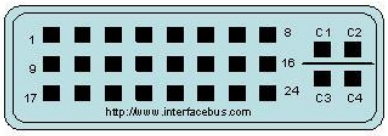

**FIGURE 6. HDMI FUNCTIONAL BLOCK DIAGRAM**

## Physical Port

**Two kinds of output connector are supported: DVI and HDMI. DVI uses a 24+4 pin socket and TW2880 can support DVI-D, DVI-I and DVI-A if tapping the output from traditional HD15 VGA analog connector. The pinout is listed below. For HDMI we can support the type A style connector. Both output method support DDC monitor feedback capabilities.**




| | |
|---|---|
| **Pin 1** | TMDS Data2+ |
| **Pin 2** | TMDS Data2 Shield |
| **Pin 3** | TMDS Data2– |
| **Pin 4** | TMDS Data1+ |
| **Pin 5** | TMDS Data1 Shield |
| **Pin 6** | TMDS Data1– |
| **Pin 7** | TMDS Data0+ |
| **Pin 8** | TMDS Data0 Shield |
| **Pin 9** | TMDS Data0– |
| **Pin 10** | TMDS Clock+ |
| **Pin 11** | TMDS Clock Shield |
| **Pin 12** | TMDS Clock– |
| **Pin 13** | CEC |
| **Pin 14** | Reserved (N.C. on device) |
| **Pin 15** | SCL |
| **Pin 16** | SDA |
| **Pin 17** | DDC/CEC Ground |
| **Pin 18** | +5 V Power (max 50 mA) |
| **Pin 19** | Hot Plug Detect |




29 pin DVI Connector PinOut and Signal Names

| Pin # | Signal Name | Pin # | Signal Name | Pin # | Signal Name |
|---|---|---|---|---|---|
| 1 | TMDS Data2- | 9 | TMDS Data1- | 17 | TMDS Data0- |
| 2 | TMDS Data2+ | 10 | TMDS Data1+ | 18 | TMDSData0+ |
| 3 | TMDS Data2/4 Shield | 11 | TMDS Data1/3 Shield | 19 | TMDS Data0/5 Shield |
| 4 | TMDS Data4- | 12 | TMDS Data3- | 20 | TMDS Data5- |
| 5 | TMDS Data4+ | 13 | TMDS Data3+ | 21 | TMDS Data5+ |
| 6 | DDC Clock [SCL] | 14 | +5 V Power | 22 | TMDS Clock Shield |
| 7 | DDC Data [SDA] | 15 | Ground (for +5 V) | 23 | TMDS Clock + |
| 8 | Analog vertical sync | 16 | Hot Plug Detect | 24 | TMDS Clock - |
| C1 | Analog Red | -- | -- | -- | -- |
| C2 | Analog Green | -- | -- | -- | -- |
| C3 | Analog Blue | -- | -- | -- | -- |
| C4 | Analog Horizontal Sync | -- | -- | -- | -- |
| C5 | Analog GND Return: (analog R, G, B) | -- | -- | -- | -- |

## Control and Configuration

**All functions of HDMI Transmitter are controlled and observed by internal I²C registers. TW2880 host interface has an I²C master interface which will translate parallel bus operation into serial I²C operation. If serial interface is chosen as the host interface, The user is actually accessing both TW2880 core and HDMI transmitter using I²C bus. The device addresses are 0x72 for the first device and 0x7A for the second device.**

**Register addresses range from 0x00 to 0xFF on each page in I²C protocol. Because there are more than 255 bytes of registers in the HDMI Transmitter, the device is accessible at one of two I²C device addresses. All references to device address in this document use the default values of 0x72 and 0x7A.**

### REGISTERS/CONFIGURATION LOGIC

**The register/configuration logic block incorporates all the registers required for configuring and managing this HDMI Transmitter. These registers are used to perform, audio/video format processing, CEA-861B info-packet formatting, and power-down control.**

### SLAVE I²C INTERFACE

**The controller I²C interface on this HDMI Transmitter (ports CSCL_I, CSDA_I and CSDA_O) is a slave interface capable of running up to 400 kHz. This bus is used to configure this HDMI Transmitter by reading/writing to necessary registers.**

## Audio Data Capture Logic

TW2880 HDMI Transmitter has a high-End digital audio interface which can accept digital audio over S/PDIF, four I²S inputs and eight one-bit audio inputs. 4x I²S inputs accept Dolby Digital, DVD-Audio input (2-cahnnel 192kHz, 8-channel 96kHz). S/PDIF input supports PCM, Dolby Digital, DTS digital audio transmission (32-192kHz Fs sample rate) IEC60958 or IEC61937 compatible. Flexible, programmable I²S channel mapping. Master I²C interface for DDC connection simplifies board layout and lower cost

### S/PDIF

The S/PDIF stream can carry 2-channel uncompressed PCM data (IEC 60958) or a compressed bit stream for multi-channel (IEC 61937) formats. The audio data capture logic forms the audio data into packets in accordance with the HDMI specification. The S/PDIF input supports audio sampling rates from 32 to 192 kHz. A separate master clock input (MCLK), coherent with the S/PDIF input, is required for time-stamping purposes. Coherent means that the MCLK and S/PDIF must have been created from the same clock source. This is typically done by using the original MCLK to strobe out the S/PDIF from the sourcing chip. There is no setup or hold timing requirement on an input with respect to MCLK.

### I²S

Four I²S inputs allow transmission of DVD-Audio or decoded Dolby Digital to A/V Receivers and high-end displays. The interface supports up to 8-channel 192 kHz. The I²S ports must also be coherent with MCLK. The audio data can be down-sampled by one half or one-fourth with register control. This allows this HDMI Transmitter to share the audio bus with a high sample rate audio DAC, while down-sampling audio for an attached display which supports only lower rates. Conversions from 192 to 48 kHz, 176.4 to 44.1 kHz, 96 to 48 kHz, and 88.2 to 44.1 kHz are supported. Audio data can only be down-sampled on 2-ch. audio. The appropriate registers must be configured to describe the format of audio being input into this HDMI Transmitter. This information is passed over the HDMI link in the CEA-861B Audio Info (AI) packets. MCLK frequencies support various audio sample rates as shown in Table 18.

TABLE 18. SUPPORTED MCLK FREQUENCIES

| | Audio Sample Rate, Fs | | | | | | |
|---|---|---|---|---|---|---|---|
| | $I^2S$ and S/PDIF Supported Rates | | | | | | |
| Multiple of Fs | 32kHz | 44.1kHz | 48kHz | 88.2kHz | 96kHz | 176.4kHz | 192kHz |
| 128 | 4.096 MHz | 5.645 MHz | 6.144 MHz | 11.29 MHz | 12.288 MHz | 22.579 MHz | 24.576 MHz |
| 192 | 6.144 MHz | 8.467 MHz | 9.216 MHz | 16.934 MHz | 18.432 MHz | 33.868 MHz | 36.864 MHz |
| 256 | 8.192 MHz | 11.29 MHz | 12.288 MHz | 22.579 MHz | 24.576 MHz | 45.158 MHz | 49.152 MHz |
| 384 | 12.288 MHz | 16.934 MHz | 18.432 MHz | 33.864 MHz | 36.864 MHz | 67.737 MHz | 73.728 MHz |
| 512 | 16.384 MHz | 22.579 MHz | 24.576 MHz | 45.158 MHz | 49.152 MHz | | |
| 768 | 24.576 MHz | 33.869 MHz | 36.864 MHz | 67.738 MHz | 73.728 MHz | | |
| 1024 | 32.768 MHz | 45.158MHz | 49.152 MHz | | | | |
| 1152 | 36.864 MHz | 50.803 MHz | 55.296 MHz | | | | |

## TMDS Digital Core

The TMDS digital core performs 8-to-10-bit TMDS encoding on the audio/video/aux data received from the HDMI Link layer. This data is output onto three TMDS differential data lines along with a TMDS differential clock. A resistor tied to the EXT_SWING pin is used to control the TMDS swing amplitude.

## DDC Master I²C Interface

This HDMI Transmitter includes a master I²C port for direct connection to the HDMI cable. DDC reads and writes are executed by reading and writing registers in this HDMI Transmitter.

The Master DDC block supports I²C transactions specified by VESA Enhanced Display Data Channel Standard (Section 3.1.2) and supports an I²C write transaction. The Master DDC block complies with the Standard Mode timing of the I²C specification (100 kHz) and supports slave clock stretching as required by EDDC.

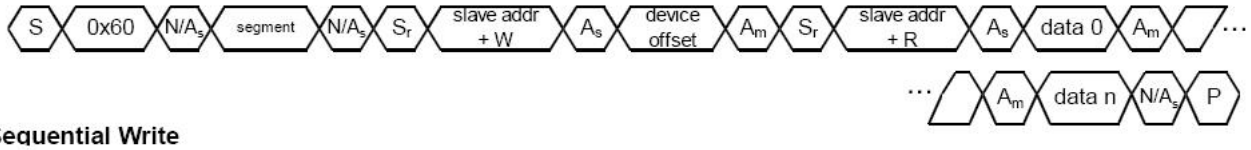
**Current Read**



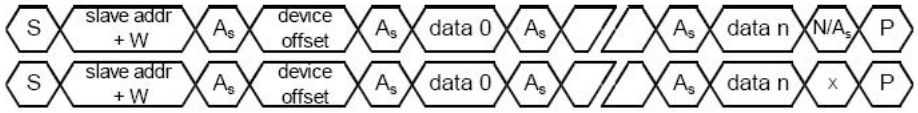

**Sequential Read**




**Enhanced DDC Read**




**Sequential Write**




S = start
$S_r$ = restart
$A_s$ = slave acknowledge
$A_m$ = master acknowledge
N = no ack
P = stop

**Supported Master $I^2C$ Transactions**

## Register Grouping

**The registers in this document are described in groups according to their function. Certain registers in each address range are reserved for future use. The reserved registers have no detailed definitions in this document. Register addresses that are not described in this document should not be written. Modifications to undocumented registers may cause unintended errors in the chip function. Register bits marked as reserved should be written as zeros and will be read as zeros unless otherwise noted.**

TABLE 19 REGISTER ADDRESS GROUPS

| I²C Address | Address Range | Group Name | Purpose | Section |
|---|---|---|---|---|
| 0x72 | 0x00-0x0E | Base | Device identification and general programming. | 188 |
| | 0x0F-0x31 | Reserved | Reserved | Reserved |
| | 0x32-0x6F | Video | DE, Sync Decoder and Encoder. | 191 |
| | 0x70-0x7F | Interrupt | Interrupt processing. | 199 |
| | | | | |
| | 0xEC-0xFF | DDC | Mastering DDC Bus. | 205 |
| | 0xF8-0xFF | Reserved | Reserved | 207 |
| 0x7A | 0x00-0x3D | Audio | Audio features and translations. | 208 |
| | 0x3E-0xFE | CEA-861B | Support for InfoFrame Packets. | 216 |

## Usages and Conventions

| | |
|---|---|
| Bit N | Bits are numbered in little-endian format: the least-significant bit of a byte or word is referred to as bit 0. |
| 0xNN | Hexadecimal representation of base-16 numbers are represented using C language notation, preceded by 0x. |
| 0bNN | Binary (base-2) numbers are represented using C language notation, preceded by 0b. |
| NN | Decimal (base-10) numbers are represented using no additional prefixes or suffixes. |
| | Reserved register bits are shaded in the register description. |
| RSVD0 | A bit in a register which is reserved and read-only, and returns a zero value |
| RSVD1 | A bit in a register which is reserved and read-only, and returns a one value. |
| RSVDRW | A bit in a register which is reserved and read-write, returning the value written to it.<br>RSVDRW0 implies a default of 0 and<br>RSVDRW1 implies a default of 1.<br>RSVDRW implies a default of 0 unless other specified. |
| X | A register bit defaulting to X has no defined state after hardware reset. |

# Base Register Set

### VENDOR ID REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x00 | VND_IDL | Vendor ID Low Byte | | | | | | | |
| 0x72 | 0x01 | VND_IDH | Vendor ID High Byte | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 15:0 | VND_ID | R | Provides unique vendor identification through I²C. | 0x01(Low) 0x00 (High) |

### DEVICE ID REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x02 | DEV_IDL | Device ID Low Byte | | | | | | | |
| 0x72 | 0x03 | DEV_IDH | Device ID High Byte | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 15:0 | DEV_ID | R | Provides unique device type identification through I²C. | 0x32 (Low) 0x91 (High) |

### DEVICE REVISION REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x04 | DEV_REV | Device Revision Byte | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7:0 | DEV_REV | R | Allows distinction between revisions of same device. | 0x01 |

### SOFTWARE RESET REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x05 | SRST | RSVD0 | | | | | | FIFORST | SWRST |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 1 | FIFORST | RW | Audio FIFO reset:<br>0 = Normal operation<br>1 = Reset (flush) audio FIFO | 0 |
| 0 | SWRST | R/W | Software reset:<br>0 = Normal Operation<br>1 = Reset all sections, including audio FIFO, but *not* writable registers | 0 |

### SYSTEM CONTROL REGISTER #1

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x08 | SYS_CTRL1 | RSVDRW0 | VSYNC | VEN | HEN | RSVDRW0 | RSVDRW1 | EDGE | PD# |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 6 | VSYNC | R | The current status of the VSYNC input pin. Refer to the INTR2 register (0x72:0x72), described on page 199, for an interrupt tied to VSYNC active edge. | X |
| 5 | VEN | R/W | VSYNC enable:<br>0 = Fixed LOW<br>1 = Follow VSYNC input | 1 |
| 4 | HEN | R/W | HSYNC enable:<br>0 = Fixed LOW<br>1 = Follow HSYNC input | 1 |
| 1 | EDGE | R/W | Edge select:<br>0 = Latch Input on Falling Edge<br>1 = Latch Input on Rising Edge | 0 |
| 0 | PD# | R/W | Power down mode: HIGH is normal operation. When LOW, TMDS core is powered down and interrupts are in power-down mode. Most other register values are not affected by assertion of the PD# bit. For exceptions, see page .259 | 0 |

### SYSTEM STATUS REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x09 | SYS_STAT | VLOW | RSVDRW0 | | | | RSEN | HPD | P_STABLE |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7 | VLOW | R | VREF mode. Always HIGH. | 1 |
| 2 | RSEN | R | Receiver Sense (works in DC-coupled systems only): 0 = No Receiver connected 1 = Receiver is connected and powered on RSEN is active when the TMDS link is terminated, usually into a powered-on TMDS Receiver chip. An active RSEN implies an active Hot Plug Detect, as the link must also be physically connected. | X |
| 1 | HPD | R | Hot Plug Detect: The state of the Hot Plug Detect pin can be read here. | X |
| 0 | P_STABLE | R | IDCK to TMDS Clock Stable. A change to the IDCK sets this bit LOW. After a subsequent LOW to HIGH transition, indicating a stable input clock, a software reset is recommended. | 0 |

### LEGACY REGISTERS

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x0A | SYS_CTRL3 | RSVDRW0 | | | | | CTL SVD0 | | RSVDRW0 |
| 0x72 | 0x0B | LEGACY1 | RSVDRW0 | | | | | | | |
| 0x72 | 0x0E | LEGACY3 | RSVDRW0 | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 2:1 | CTL | R/W | The states of these control bits are transmitted across the TMDS link during blanking times for DVI 1.0 mode only. | 0b00 |

**DATA CONTROL REGISTER**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x0D | DCTL | RSVDRW0 | | | RSVD0 | | RSVDRW0 | AUD_MUTE | Reserved |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 1 | AUD_MUTE | R/W | 0 = Do not send zeros in audio packet<br>1 = Send zeros in audio packet | 0 |

## Automatic Ri Check

**The auto-synchronous Ri check compares the HDMI Transmitter's Ri value to the HDMI Receiver's Ri' value on the 0 and 127th frame. When reading the HDMI Receiver's Ri' value, the automatic Ri check uses the DDC output port. Use bit 0 in the RI_STAT register (0x72.0x26), described below, for proper handshaking between the automatic Ri check and the Master DDC functionality. The automatic Ri check can also be enabled or disabled (it is disabled by default). Interrupts can be configured to trigger on an HDMI Transmitter Ri and Receiver Ri' mismatch condition.**

**RI STATUS REGISTER**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x26 | RI_STAT | RSVD0 | | | | | | | Ri_Started |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 0 | Ri_Started | R | Ri check started status.<br><br>This signal is used for handshaking between the firmware and the hardware. After the Ri check is enabled, the hardware waits for the DDC master to finish the current transaction before taking control. After this bit is set, the firmware loses the ability to use the DDC Master, unless it disables Ri Check and this bit resets to 0. | 0 |

**RI COMMAND REGISTER**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x27 | RI_Cmd | RSVDRW0 | | | | | | | Ri_En |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 0 | Ri_En | R/W | Enable Ri check. Check bit 0 of the Ri_STAT register (0x72:0x26) for firmware and hardware DDC control handshaking.<br>1 = Enabled<br>0 = Disabled | 0 |

**RI LINE START REGISTER**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x28 | RI_START | Ri_Line_Start | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7:0 | Ri_Line_Start | R/W | Indicates at what line within frame 127 or 0 to start the Ri Check. 2 LSB are 0. | 0x04 |

**RI FROM RX REGISTER**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x29 | RI_RX_L | Ri_RX[7:0] | | | | | | | |
| 0x72 | 0x2A | RI_RX_H | Ri_RX[15:8] | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 15:0 | Ri_RX | R | This value represents the HDMI Receiver's Ri value if any of the Ri Check errors occurred. | 0 |

## DE Generator Register Set

**The HDMI Transmitter provides an internal Data Enable (DE) generator for use when the attached video source does not provide a DE signal with the other video signals. The DE signal is needed for encoding the TMDS output of the HDMI Transmitter. A DE signal is generated for use by the Transmitter, based on the values in the DE Generator registers and the arrival times of the HSYNC and VSYNC pulses from the video source. This DE signal is included in the control data sent to the HDMI Receiver across the link. The DE Generator registers are used only for DE generation. The registers are shown diagrammatically in Figure 7. The vertical sync pulse (VSYNC) occurs in the top area of the frame, before the active video area. The active (leading) edge is shown with an arrow. The horizontal sync pulse (HSYNC) occurs every line before the active video area during the DE_DLY time. The active (leading) edge of HSYNC is shown with an arrow.**

**Note: The VSYNC and HSYNC widths are not shown to scale.**

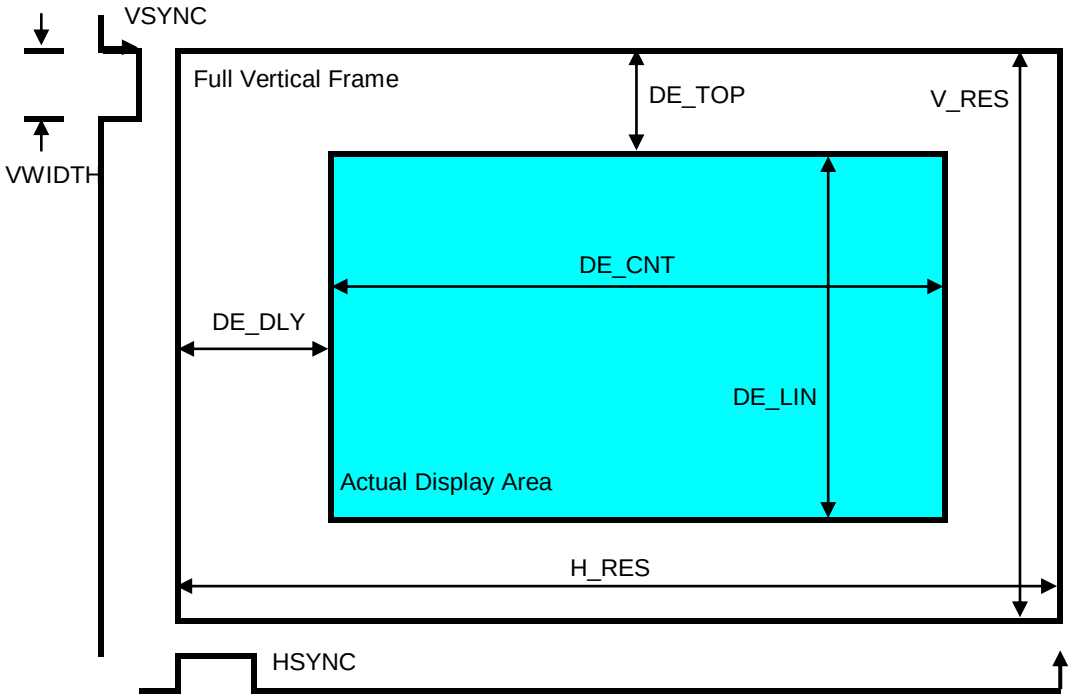



FIGURE 7. DE GENERATOR MEASUREMENTS

**In the following definitions, *pixels* mean unique pixels. The counts in the DE Generator registers, if expressed in pixels, are counted according to the original input clock even if that original input clock is multiplied within the chip. For example, a 480i field contains 720 unique pixels per line in the active video area even when the clock is multiplied to 1440 clock cycles per active video time.**

**Register 0x72:0x3F records the detected polarity of VSYNC and HSYNC. The output polarities are set by register 0x72:0x33.**

**Note: When using the Sync Decoding module (see page 195), the DE input signal should be tied LOW.**

### VIDEO DE DELAY REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x32 | DE_DLY | DE_DLY[7:0] | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7:0 | DE_DLY[7:0] | R/W | Width of the area to the left of the active display. The unit of measure is pixels. This register should be set to the sum of (HSYNC width) + (horizontal back porch) + (horizontal left border), and is used only for DE generation.<br><br>**Note**: This 12-bit value includes two bits from register 0x72:0x33. The valid range is 1-4095. 0 is invalid. | 0x00 |

### VIDEO DE CONTROL REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x33 | DE_CTRL | RSVD0 | DE_GEN | VS_POL# | HS_POL# | DE_DLY[11:8] | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 6 | DE_GEN | R/W | Generate DE signal: 0 = Disabled 1 = Enabled | 0 |
| 5 | VS_POL# | R/W | VSYNC polarity:<br>0 = Positive polarity (leading edge rises).<br>1 = Negative polarity (leading edge falls).<br><br>This bit should be set to the input VSYNC polarity for the source that provides VSYNC. For embedded syncs, this bit should be set to the desired VSYNC polarity that is generated from the embedded sync codes. | 0 |
| 4 | HS_POL# | R/W | HSYNC polarity:<br>0 = Positive polarity (leading edge rises).<br>1 = Negative polarity (leading edge falls).<br><br>This bit should be set to the input HSYNC polarity for the source that provides HSYNC. For embedded syncs, this bit should be set to the desired HSYNC polarity that is generated from the embedded sync codes. | 0 |
| 3:0 | DE_DLY[11:8] | R/W | Bits 11:8 of the DE_DLY value (see register 0x72:0x32). | 0b0000 |

**Note:** If both DE signal generation and sync decoding are enabled, VS_POL# and HS_POL# define the polarities of VSYNC and HSYNC from the sync decoder. The DE Generator may be used in combination with sync decoding because the encoded syncs do not carry polarity information. To set the transmitted HSYNC and VSYNC to HDMI-compliant states, the DE Generator may be needed. See the VID_MODE register (0x72:0x4A), described on page 198 and the diagram on 198.

### VIDEO DE TOP REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x34 | DE_TOP | RSVD0 | DE_TOP | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 6:0 | DE_TOP | R/W | Defines the height of the area above the active display. The unit of measure is lines (HSYNC pulses). This register should be set to the sum of (VSYNC width) + (vertical back porch) + (vertical top border). The valid range is 1-127. 0 is invalid. | 0b0000000 |

### VIDEO DE COUNT REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x36 | DE_CNTL | DE_CNT[7:0] | | | | | | | |
| 0x72 | 0x37 | DE_CNTH | RSVD0 | | | | DE_CNT[11:8] | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 11:0 | DE_CNT | R/W | Defines the width of the active display. The unit of measure is pixels. This register should be set to the desired horizontal resolution. The valid range is 1-4095. 0 is invalid. | 0 |

**Note**: Values measured in pixels (DE_CNT, and so on) count the total number of unique pixels on a line. If the input clock is a multiple of the pixel rate (see the DEMUX bit, described on page 198 and the ICLK bit, described on page 197), the registers indicate pixel count, which is not the same as clock count.

### VIDEO DE LINE REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x38 | DE_LINL | DE_LIN[7:0] | | | | | | | |
| 0x72 | 0x39 | DE_LINH | RSVD0 | | | | | DE_LIN[10:8] | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 10:0 | DE_LIN | R/W | Defines the height of the active display. The unit of measure is lines (HSYNC pulses). This register should be set to the desired vertical resolution. For interlaced modes, this register should be programmed to the number of lines per field, which is half the overall vertical resolution. The valid range is 1-2047. 0 is invalid. | 0 |

### VIDEO H RESOLUTION REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x3A | HRES_L | H_RES[7:0] | | | | | | | |
| 0x72 | 0x3B | HRES_H | RSVD0 | | | H_RES[12:8] | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 12:0 | H_RES | R | Measures the time between two HSYNC active edges. The unit of measure is pixels. | 0 |

### VIDEO V REFRESH REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x3C | VRES_L | V_RES[7:0] | | | | | | | |
| 0x72 | 0x3D | VRES_H | RSVD0 | | | | | V_RES[10:8] | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 10:0 | V_RES | R | Measures the time between two VSYNC active edges. The unit of measure is lines. | 0 |

**The values in the DE Generator read-only registers are maintained until input HSYNC and VSYNC pulses are stopped. If an input IDCK continues, the DE Generator counters overflow and the registers store zero until HSYNC and VSYNC are active again. The values in these registers are accurate only when there are active HSYNC and VSYNC inputs, or active SAV/EAV sequences (for embedded sync input).**

# Video Embedded Sync Decoding Registers

**When decoding syncs from the embedded sync stream (see register 0x72:0x4A, on page 198), the DE Generator block should be disabled (see register 0x72:0x33[6], on page 193). Also, the DE input signal should be tied LOW.**

**See the diagram on page 229 and the explanation for handling interlaced video on page 231.**

### VIDEO INTERFACE ADJUSTMENT REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x3E | IADJUST | RSVD0 | | | | | DE_ADJ# | F2VADJ | F2VOFST |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 2 | DE_ADJ# | R/W | 0 = Enable VSYNC<br>1 = Disable VSYNC<br><br>Setting this bit HIGH disables VSYNC adjustments and sets the DE generator to be more compatible with the existing Transmitters. Clearing this bit enables detection circuits to locate the position of VSYNC relative to HSYNC and only include HSYNC edges that are greater than 3/4 lines from VSYNC in the line count for DE_TOP. | 1 |
| 1 | F2VADJ | R/W | If this bit is set, the VBIT_TO_VSYNC value (register 0x72:0x46) is adjusted during field 2 of an interlace frame according to the setting of the F2VOFST bit.<br>This bit defaults to 0. | 0 |
| 0 | F2VOFST | R/W | 0 = If F2VADJ and this bit are cleared, VBIT_TO_VSYNC (register 0x72:0x46) is decremented by one during field 2 of an interlace frame.<br>1 = If F2VADJ and this bit are set, VBIT_TO_VSYNC is incremented by one during field 2 of an interlace frame. | 0 |

### VIDEO SYNC POLARITY DETECTION REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x3F | POL_DETECT | RSVD0 | | | | | I_DET | VPOL_DET# | HPOL_DET# |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 2 | I_DET | R | Interlace detect:<br>0 = Non-interlaced video<br>1 = Interlaced video<br><br>This bit is set by checking for a varying VSYNC timing characteristic of interlaced modes. | 0 |
| 1 | VPOL_DET# | R | Detected input VSYNC polarity, using internal circuit:<br>0 = Active high (leading edge rises)<br>1 = Active low (leading edge falls)<br><br>**Note**: The polarity of VPOL_DET# is opposite the polarity of the detect bits in the receiver. | 1 |
| 0 | HPOL_DET# | R | Detected input HSYNC polarity, using internal circuit:<br>0 = Active HIGH (leading edge rises)<br>1 = Active LOW (leading edge falls)<br><br>**Note**: The polarity of HPOL_DET# is opposite the polarity of the detect bits in the receiver. | 1 |

### VIDEO HBIT TO HSYNC REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x40 | HBIT_2HSYNC1 | HBIT_TO_HSYNC[7:0] | | | | | | | |
| 0x72 | 0x41 | HBIT_2HSYNC2 | RSVD0 | | | | | | | [9:8] |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 9:0 | HBIT_TO_HSYNC | R/W | Creates HSYNC pulses. Set this register to the delay from the detection of an EAV sequence (H bit change from 1 to 0) to the active edge of HSYNC. The unit of measure is pixels. The valid range is 1-1023. 0 is invalid. | 0 |

**Note**: Registers 0x72:0x40 and 0x72:0x41 are useful only when the input video uses 656 encoded syncs.

### VIDEO FIELD2 HSYNC OFFSET REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x42 | FLD2_HS_OFSTL | FIELD2_OFST[7:0] | | | | | | | |
| 0x72 | 0x43 | FLD2_HS_OFSTH | RSVD0 | | | | FIELD2_OFST[11:8] | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 11:0 | FIELD2_OFST | R/W | Determines VSYNC pixel offset for the odd field of an interlaced source. Set this to half the number of pixels/line. The valid range is 1-4095. 0 is invalid. | 0 |

### VIDEO HSYNC LENGTH REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x44 | HWIDTH1 | HWIDTH[7:0] | | | | | | | |
| 0x72 | 0x45 | HWIDTH2 | RSVD0 | | | | | | [9:8] | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 9:0 | HWIDTH | R/W | Sets the width of the HSYNC pulses. Set this register to the desired HSYNC pulse width. The unit of measure is pixels. The valid range is 1-1023. 0 is invalid. | 0 |

### VIDEO VBIT TO VSYNC REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x46 | VBIT_TO_VSYNC | RSVD0 | | VBIT_TO_VSYNC | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 5:0 | VBIT_TO_VSYNC | R/W | Sets the delay from the detection of V bit changing from 1 to 0 in an EAV sequence, to the asserting edge of VSYNC. The unit of measure is lines. The valid range is 1-63. 0 is invalid. | 0b000000 |

**Note**: Registers 0x72:0x42-0x46 are useful only when the input video uses 656 encoded syncs.

### VIDEO VSYNC LENGTH REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x47 | VWIDTH | RSVD0 | | VWIDTH | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 5:0 | VWIDTH | R/W | Sets the width of VSYNC pulse. The unit of measure is lines. The valid range is 1-63. 0 is invalid. | 0b000000 |

**Figure 7 shows the HWIDTH and VWIDTH dimensions relative to the complete frame time.**

### VIDEO CONTROL REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x48 | VID_CTRL | IFPOL | RSVDRW | EXTN | CSCSEL | RSVD0 | | ICLK | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7 | IFPOL | R/W | Invert field polarity:<br>0 = Do not invert field bit<br>1 = Invert field bit<br><br>This bit is used when the 656 Flag bit is opposite the standard polarity for Field1 and Field2. Inverting the field polarity causes the sync extraction to format HSYNC and VSYNC properly based on the F bit. In embedded sync mode, the HDMI Transmitter does not detect *even* from *odd* field, except based on the setting of the F bit. With explicit syncs, the HDMI Transmitter encodes HSYNC and VSYNC across the HDMI/TMDS link regardless of field sequence. | 0 |
| 6 | RSVDRW | R/W | Do not write this bit to 1. | 0 |
| 5 | EXTN | R/W | Extended Bit mode:<br>0 = All 8-bit input modes<br>1 = All 12-bit 4:2:2 input modes<br><br>For 4:2:2 inputs wider than 8 bits but less than 12 bits, the unused bits should be set to 0. | 0 |
| 4 | CSCSEL | R/W | Color Space Conversion Standard select:<br>0 = BT.601 conversion<br>1 = BT.709 conversion | 0 |
| 1:0 | ICLK | R/W | Clock mode:<br>0b00 = Pixel data is not replicated<br>0b01 = Pixels are replicated once (each sent twice)<br>0b10 = RSVD<br>0b11 = Pixels are replicated 4 times (each sent four times)<br><br>Refer to page 233. for examples of coding ICLK and the related field TCLKSEL for various video+audio modes.<br><br>**Note**: If the DEMUX bit (0x72:0x4A[1]) is set to 0, the value of ICLK must be the same as the value for the pixel replication field of AVI v2 data byte 5. If the DEMUX bit is set to 1, the value of ICLK must be the next highest entry. | 0b00 |

**VIDEO MODE REGISTER**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x4A | VID_MODE | RSVDRW0 | | DITHER | RANGE | CSC | UPSMP | DEMUX | SYNCEXT |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 5 | DITHER | R/W | 0 = Dither disabled; the video output is truncated to the output width specified in DITHER_MODE [7:6]<br>1 = Dither enabled; the video output is dithered to the output width specified in DITHER_MODE [7:6] | 0 |
| 4 | RANGE | R/W | Data Range 16-to-235 to 0-to-255 expansion:<br>0 = Disabled<br>1 = Enabled When this bit is set, the HDMI Transmitter expands the range of pixel data values from 16-235 into the full 8-bit range of 0-255. This is suitable for translating input YCbCr data into output RGB data in PC modes that use the complete range. The HDMI 1.0 Specification allows one non-CEA-861B mode in the first and only 18-byte descriptor of the Sink's EDID 1.3. This is the native resolution of the Sink, which may be RGB. It may be a standard PC resolution (XGA, SXGA, WXGA, and so on), or a specific native resolution. In these cases (or for a Sink with the Type B HDMI connector, which allows multiple PC modes), when the HDMI Transmitter receives YCbCr data, the data must be expanded to full range for outputting RGB full-range modes. See the HDMI 1.0 Specification, Sections 4.1.2.1 and 4.1.2.2. | 0 |
| 3 | CSC | R/W | YCbCr to RGB Color Space Conversion:<br>0 = Disabled<br>1 = Enabled | 0 |
| 2 | UPSMP | R/W | Up sampling 4:2:2 to 4:4:4:<br>0 = Disabled<br>1 = Enabled | 0 |
| 1 | DEMUX | R/W | One- to Two-Data-Channel Demux:<br>0 = Disabled<br>1 = Enabled | 0 |
| 0 | SYNCEXT | R/W | Embedded Sync Extraction:<br>0 = Disabled<br>1 = Enabled | 0 |

# Interrupt Registers

**The interrupt registers are used to coordinate enabling and recognition of a variety of interrupts with the HDMI Transmitter.**

### INTERRUPT STATE REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x70 | INTR_STATE | RSVD0 | | | | | | | INTR |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 0 | INTR | R | Interrupt State. When an interrupt is asserted, this bit is set to 1. The polarity of the INT output signal is set using this bit and the POLARITY# bit in the INT_CTRL register (0x72:0x79). Only INTR1, INTR2, and INTR3 bits with matching set bits in INT_UNMASK can contribute to setting the INTR bit. | 0 |

### INTERRUPT SOURCE REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x71 | INTR1 | SOFT | HPD | RSEN | DROP_SAMPLE | BI_PHASE_ERR | RI_128 | OVER_RUN | UNDER_RUN |
| 0x72 | 0x72 | INTR2 | RSVD0 | SPDIF_PAR | ENC_DIS | PREAM_ERR | CTS_CHG | ACR_OVR | TCLK_STBL | VSYNC_REC |
| 0x72 | 0x73 | INTR3 | RI ERR #3 | RI ERR #2 | RI ERR #1 | RI ERR #0 | DDC Cmd Done | DDC FIFO Half | DDC FIFO Full | DDC FIFO Empty |

INTR1

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7 | SOFT | R | Software Induced Interrupt - allows the firmware to generate an interrupt directly. | 0 |
| 6 | HPD | R | Monitor Detect Interrupt - asserted if Hot Plug Detect has changed state. The HDMI Transmitter signals a change in the connectivity to a Sink, either unplug or plug. HDMI specifies that Hot Plug be active only when the Sink's EDID is ready to be read and that Hot Plug be toggled any time there is a change in connectivity downstream of an attached Repeater. | 0 |
| 5 | RSEN | R | Receiver Sense Interrupt asserted if RSEN has changed. This interrupt is set whenever VCC is applied to, or removed from, the attached HDMI Receiver chip. A Receiver with multiple input ports can also disconnect the TMDS termination to the unused port, which triggers this RSEN interrupt. | 0 |
| 4 | DROP_SAMPLE | R | New preamble forced to drop sample (S/PDIF input only). If the HDMI Transmitter detects an 8-bit preamble in the S/PDIF input stream before the sub-frame has been captured, this interrupt is set. A S/PDIF input that stops signaling or a flat line condition can create such a premature preamble. | 0 |
| 3 | BIP_HASE_ERR | R | Input S/PDIF stream has bi-phase error. This can occur when there is noise or an Fs rate change on the S/PDIF input. | 0 |
| 2 | RI_128 | R | Input counted past frame count threshold set in RI_128_COMP register. This interrupt occurs when the count written to register 0x72:0x24 is matched by the VSYNC (frame) counter in the HDMI Transmitter. It should trigger the firmware to perform a link integrity check. Such a match occurs every 128 frames. | 0 |

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | OVER_RUN | | R | Audio FIFO Overflow. This interrupt occurs if the audio FIFO overflows when more samples are written into it than are drawn out across the HDMI link. Such a condition can occur from a transient change in the Fs or pixel clock rate. | | | | | | 0 |
| 0 | UNDER_RUN | | R | Audio FIFO Underflow. Similar to OVER_RUN. This interrupt occurs when the audio FIFO empties. | | | | | | 0 |

INTR2

| | | | | |
|---|---|---|---|---|
| 6 | SPDIF_PAR | R | S/PDIF Parity Error.<br><br>The S/PDIF stream includes a parity (P) bit at the end of each sub frame. An interrupt occurs if the calculated parity does not match the state of this bit. | 0 |
| 5 | ENC_DIS | R | The ENC_EN bit (0x72:0x0F[0]) changed from 1 to 0.<br>This interrupt occurs if encryption is turned off. | 0 |
| 4 | PREAM_ERR | R | The condition is the opposite of the condition that causes DROP_SAMPLE (0x72:0x71[4]). This interrupt occurs if a preamble is expected but not found when decoding the S/PDIF stream. | 0 |
| 3 | CTS_CHG | R | Change In ACR CTS Value.<br><br>This interrupt occurs when the change is of an unexpected magnitude. Such an interrupt should be expected when changing Fs or pixel clock frequency. | 0 |
| 2 | ACR_OVR | R | ACR Packet Overwrite.<br><br>This interrupt occurs if the HDMI Transmitter puts a NCTS packet into the queue before the previous NCTS packet has been sent. This can happen if very long active data times do not allow for sufficient NCTS packet bandwidth. For all CEA-861B modes, no ACR_OVR interrupt should occur. | 0 |
| 1 | TCLK_STBL | R | TCLK_STABLE (register 0x72:0x09[0] ) changes state.<br><br>Whenever IDCK changes, there is a temporary instability in the internal clocking. This interrupt is set when the internal clocking has stabilized. | 0 |
| 0 | VSYNC_REC | R | Asserted when VSYNC active edge is recognized. It is useful for triggering firmware actions that occur during vertical blanking. | 0 |

INTR3

| | | | | |
|---|---|---|---|---|
| 7 | Ri Err #3 | R | Ri not read within one frame. | 0 |
| 6 | Ri Err #2 | R | Ri did not change between frame #127 and #0. | 0 |
| 5 | Ri Err #1 | R | Ri and Ri' do not match during frame #0 (ICNT) | 0 |
| 4 | Ri Err #0 | R | Ri and Ri' do not match during frame #127 (ICNT-1) | 0 |
| 3 | DCC Cmd Done | R | DDC command is complete. | 0 |
| 2 | DDC FIFO Half | R | DDC FIFO is half full. | 0 |
| 1 | DDC FIFO Full | R | DDC FIFO is full. | 0 |
| 0 | DDC FIFO Empty | R | DDC FIFO is empty. | 1 |

| | | | | |
|---|---|---|---|---|
| 7:0 | For each interrupt bit... | R | 1 = Interrupt asserted<br>0 = No interrupt occurred | |
| 7:0 | | W | Write any bit to 1 to clear the bit and the interrupt | |

**When the device is powered-down (PD#=0) the INT output pin is driven with RSEN. No other condition generates an interrupt when the HDMI Transmitter is powered-down with PD#. The Source device must use other means to monitor the hot plug state, if necessary, such as polling the System Status Register (see page 189). The RSEN interrupt bit is set *only* when the HDMI Receiver sense changes from 0 to 1, *not* on a change from 1 to 0.**

### INTERRUPT UNMASK REGISTER

**Interrupts are set in the INTR1, INTR2, and INTR3 registers as they occur, but only those interrupts with a corresponding set bit in INT_UNMASK1 through INT_UNMASK3 are also logically ORed into the INT output pin signal. The state of all interrupts can be checked at any time by reading the INTR1, INTR2, and INTR3 registers directly.**

**All bits marked RSVD in the interrupt registers should have the corresponding bit in the INT_UNMASK register cleared to zero.**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x75 | INT_UNMASK1 | | | | | | | | |
| 0x72 | 0x76 | INT_UNMASK2 | | | | | | | | |
| 0x72 | 0x77 | INT_UNMASK3 | | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 23:0 | INT_UNMASK | R/W | Each bit corresponds to one bit in INTR1, INTR2, or INTR3:<br>0 = Disable corresponding interrupt from INT output<br>1 = Enable corresponding interrupt to INT output | 0 |

### INTERRUPT CONTROL REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x79 | INT_CTRL | RSVDR0 | | | | SOFT_INTR | RSVDRW0 | POLARITY# | RSVDR0 |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 3 | SOFT_INTR | R/W | Set software interrupt:<br>0 = Clear interrupt<br>1 = Set interrupt | 0 |
| 1 | POLARITY# | R/W | INT pin assertion level:<br>0 = Assert HIGH<br>1 = Assert LOW | 1 |

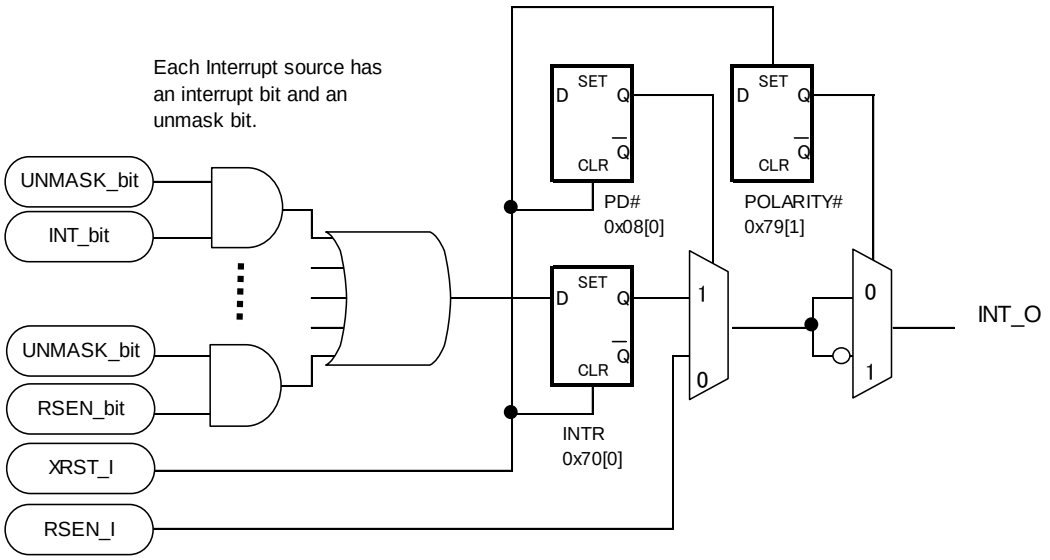



**FIGURE 8. INTERRUPT PIN CONTROL**

The representation in Figure 8 shows the control of the INT output pin. Each interrupt source has a bit (shown as INT_bit) and an unmask (shown as UNMASK_bit). These are logically ANDed, then ORed, and latched with the active output clock. During RESET#=LOW, PD# is reset to zero. In power-down mode (PD# asserted), the active level from the HDMI Receiver sense logic (RSEN) is output as the INT signal. The POLARITY# control affects INT in all modes. RESET#=LOW loads PD#=0 and POLARITY#=1 as a default. Therefore, the INT output will be driven LOW when a powered-on TMDS Receiver is attached (RSEN=1).

## TMDS Control Register Set

The TMDS registers control TMDS (Transition-Minimized Differential Signaling).

**TMDS CONTROL REGISTER**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0x82 | TMDS_CTRL | RSVDRW0 | TCLKSEL | | RSVDRW0 | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 6:5 | TCLKSEL | R/W | Selects FPLL multiple of the IDCK:<br>0b00 = FPLL is 0.5*IDCK<br>0b01 = FPLL is 1.0*IDCK<br>0b10 = FPLL is 2.0*IDCK<br>0b11 = FPLL is 4.0*IDCK | 01 |

For certain combinations of video input clock frequency and audio sampling rate, it is necessary for the HDMI Transmitter to use a higher multiple of the input pixel clock when sampling the S/PDIF input.

You must set ICLK to reflect the pixel replication factor of the input data stream so that it is properly decoded. TCLKSEL indicates the factor by which the input clock (IDCK) must be multiplied to yield the output clock frequency. These settings ensure that the output clock can provide sufficient bandwidth for multi-channel audio data on an HDMI link (25MHz at a minimum). The pixel replication count bits in the AVI InfoFrame Packet must be accurate.

| 480p mode (pixel clock = 27MHz) | | | | | | | |
|---|---|---|---|---|---|---|---|
| Input Clock (IDCK)[1] | Audio Mode & max $f_s$ | ICLK 0x48[1:0] Input Pixel Replication[4] | DEMUX 0x4A[1] | TCLKSEL[3] 0x82[6:5] | Output (Link) Clock[1] | Output Pixel Replication[10] | AVI InfoFrame Packet Byte 5bits PR3:PR0[10] |
| 54MHz | 8ch, 96kHz | 0b01 (2x) | 0 | 0b01 (1.0) | 54MHz[6] | 2x[7] | 0b0001 |
| 54MHz | 2ch, 192kHz | 0b01 (2x) | 0 | 0b00 (0.5) | 27MHz | 1x | 0b0000 |
| 54MHz | 8ch, 96kHz | 0b00 (1x) | 1[5] | 0b01 | 54MHz[6] | 2x[7] | 0b0001 |
| 54MHz | 2ch, 192kHz | 0b00 (1x) | 1[5] | 0b00 | 27MHz | 1x | 0b0000 |
| 27MHz | 8ch, 96kHz | 0b00 (1x) | 0 | 0b10 (2.0) | 54MHz[6] | 2x[7] | 0b0001 |
| 27MHz | 2ch, 192kHz | 0b00 (1x) | 0 | 0b11 (4.0) | 108MHz | 4x[8] | 0b0011 |
| 27MHz | 8ch, 48kHz | 0b00 (1x) | 0 | 0b01 | 27MHz | 1x[9] | 0b0000 |

**NOTES:**

**Input Clock (IDCK) and Output Clock must be within the mix/max range for the HDMI Transmitter.**

**For proper decoding, you must set ICLK to reflect the pixel replication factor of the input data stream.**

**Factor by which input clock must be multiplied to give output clock frequency.**

**There is only one pixel per 27 MHz clock cycle, so each must be replicated.**

**When YCbCr 4:2:2 data is multiplexed onto a single channel, the input clock must be doubled.**

**54 MHz is necessary so the blanking intervals have sufficient bandwidth to carry the 8-channel audio data sampled at frequencies up to 96 kHz.**

**Because the output clock has been doubled, pixels must be replicated.**

**Illustrates 4x pixel replication on output.**

**27 MHz input clock provides sufficient bandwidth for 8-channel audio data sampled at frequencies 48 kHz and below. Refer to the *HDMI Specification*.**

**Bits PR0:PR3 of Byte 5 of the AVI InfoFrame packet indicate to the HDMI Sink how many repetitions of each unique pixel are transmitted. Refer to Table 14 in the *CEA-861B Specification*.**

**These settings ensure that the output clock provides sufficient bandwidth for multi-channel audio data on a HDMI link (25MHz minimum).**

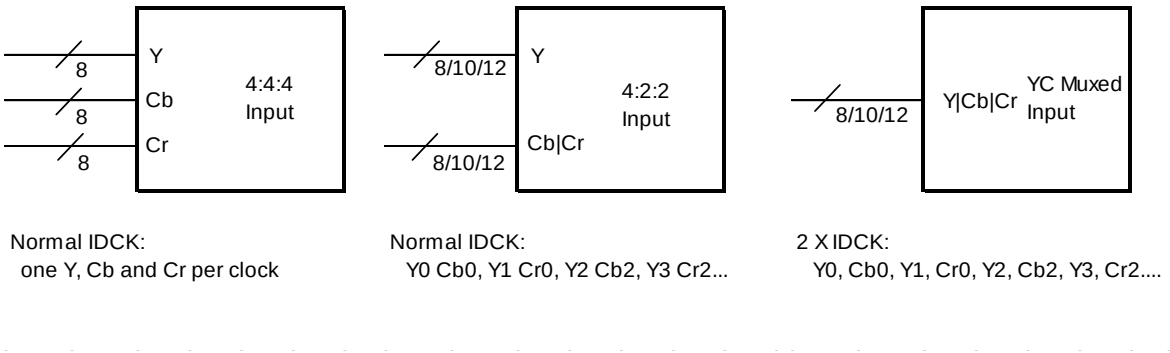



**FIGURE 9. INPUT BUS DIAGRAM FOR DIFFERENT FORMATS**

**Note: All three input bus formats can use 656 encoded syncs.**

# DDC Master Registers

**The following registers control the DDC output port. A description of this feature begins on page 255. The speed of the Master DDC clock is determined by an internal oscillator in the HDMI Transmitter, with a maximum of 100kbps. There is no requirement for an active input pixel clock to the HDMI Transmitter and the DDC SCL speed is not affected by the pixel clock frequency.**

**The DDC output port is also used by the auto-synchronous Ri check. For proper handshaking, refer to the Ri_Started bit in the RI_STAT register (0x72.0x26[0]), described on page 190.**

**Note: The DDC_CMD, DDC_DATA and DDC_STATUS registers are not accessible if PDOSC#=0 or**

**PDTOT#=0.**

### DDC I²C MANUAL REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0xEC | DDC_MAN | RSVDRW0 | | | | | | IO_SCL | IO_SDA |
| **Bit** | **Label** | **R/W** | **Description** | | | | | | **Default** | |
| 1 | IO_SDA | R | DDC SDA input state. | | | | | | 0 | |
| 0 | IO_SCL | R | DDC SCL input state. | | | | | | 0 | |

**Note:** The default values of IO_SCL and IO_SDA depend on the condition of DDC bus.

### DDC I²C TARGET SLAVE ADDRESS REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0xED | DDC_ADDR | DDC_ADDR | | | | | | | RSVD0 |
| **Bit** | **Label** | **R/W** | **Description** | | | | | | **Default** | |
| 7:1 | DDC_ADDR | R/W | DDC device address. | | | | | | 0b0000000 | |

### DDC I²C TARGET SEGMENT ADDRESS REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0xEE | DDC_SEGM | DDC_SEGM | | | | | | | |
| **Bit** | **Label** | **R/W** | **Description** | | | | | | **Default** | |
| 7:0 | DDC_ SEGM | R/W | DDC segment address. | | | | | | 0x00 | |

### DDC I²C TARGET OFFSET ADDRESS REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0xEF | DDC_OFFSET | DDC_OFFSET | | | | | | | |
| **Bit** | **Label** | **R/W** | **Description** | | | | | | **Default** | |
| 7:0 | DDC_OFFSET | R/W | DDC offset address. | | | | | | 0x00 | |

### DDC I²C DATA COUNT REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0xF0 | DDC_COUNT1 | DDC_COUNT[7:0] | | | | | | | |
| 0x72 | 0xF1 | DDC_COUNT2 | RSVDRW0 | | | | | | [9:8] | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 9:0 | DDC_COUNT | R/W | The total number of bytes to be read from the slave or written to the slave before a *Stop* bit is sent on the DDC bus. | 0 |

### DDC I²C STATUS REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0xF2 | DDC_STATUS | RSVDR0 | BUS_LOW | NO_ACK | IN_PROG | FIFO_FULL | FIFO_EMP | FRD_USE | FWT_USE |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 6 | BUS_LOW | R | 1 = I²C transaction did not start because I²C bus is pulled LOW by an external device | 0 |
| 5 | NO_ACK | R | 1 = HDMI Transmitter did not receive an ACK from slave device during address or data write | 0 |
| 4 | IN_PROG | R | 1 = DDC operation in progress | 0 |
| 3 | FIFO_FULL | R | 1 = DDC FIFO Full | 0 |
| 2 | FIFO_EMP | R | 1 = DDC FIFO Empty | 1 |
| 1 | FRD_USE | R | 1 = DDC FIFO Read In Use | 0 |
| 0 | FWT_USE | R | 1 = DDC FIFO Write In Use | 0 |

The DDC master feature recognizes and supports clock stretching by an I²C slave device. Clock stretching is described in the I²C Specification, Section 13.2.

### DDC I²C COMMAND REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0xF3 | DDC_CMD | RSVD0 | | DDC_FLT_EN | SDA_DEL_EN | DDC_CMD | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 5 | DDC_FLT_EN | R/W | Enable the DDC delay:<br>0 = Enabled<br>1 = Disabled<br><br>A DDC delay is inserted into the SDA line to create a 300ns delay for the falling edge of the DDC SDA signal to avoid an erroneous I²C START condition. The real start condition must have a setup time of 600ns so that this delay of 300ns does not remove the real START condition. Filtering is done using a Ring Oscillator. | 0 |
| 4 | SDA_DEL_EN | R/W | Enable 3ns glitch filtering on the DDC clock and data line:<br>0 = Enabled<br>1 = Disabled<br><br>Filtering is done using a Ring Oscillator. | 0 |

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 3:0 | DDC_CMD | R/W | DDC command:<br>0b1111 = Abort Transaction<br>0b1001 = Clear FIFO<br>0b1010 = Clock SCL<br>0b0000 = Current Address Read with no ACK on last byte<br>0b0010 = Sequential Read with no ACK on last byte<br>0b0100 = Enhanced DDC Read with no ACK on last byte<br>0b0110 = Sequential Write ignoring ACK on last byte<br>0b0111 = Sequential Write requiring ACK on last byte Writing to this register immediately initiates the I²C transaction on the DDC bus.<br><br>**Note**: The Clear FIFO command resets the FIFO read and write pointers to zero. Data formerly loaded into the FIFO cannot be re-read after a Clear FIFO, as the FIFO will be empty. Other command codes are reserved and may cause the DDC bus to hang if used. The Clock SCL command resets any I²C devices on the DDC lines. This should be initiated once before initiating the DDC commands. | | | | | | | 0b0000 |

### DDC I²C DATA REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0xF4 | DDC_DATA | DDC_DATA | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7:0 | DDC_DATA | R | DDC data input. | 0xXX |

**The FIFO supports multi-byte sequential read commands from the controller. Such a command is diagrammed in Figure 18 on page 255. Up to 16 bytes can be read in one local I²C command. Data bytes continue to be put into the FIFO as long as it is not full.**

### DDC I²C FIFO COUNT REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x72 | 0xF5 | DDC_FIFOCNT | RSVDRW0 | | | DDC_FIFOCNT | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 4:0 | DDC_FIFOCNT | R | FIFO data byte count (the number of bytes in the FIFO). The DDC FIFO size is 16. The maximum value for DDC_FIFOCNT is 0x10. | 0x00 |

## Audio Registers

**The HDMI link does *not* transport an explicit audio master clock, it encodes the frequency of that clock in N/CTS Packets sent during command times. The ratio of the current pixel clock frequency to the desired MCLK frequency is defined by a numerator, N and a denominator, CTS. Whenever the pixel clock frequency changes (the video mode changes), or the audio clock changes (the audio sampling rate changes), the N value must also be updated.**

### ACR CONTROL REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x01 | ACR_CTRL | RSVD0 | | | | | | NCTSPKT_EN | CTS_SEL |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 1 | NCTSPKT_EN | R/W | CTS Request Enable:<br>0 = N/CTS packet disabled<br>1 = N/CTS packet enabled | 1 |
| 0 | CTS_SEL | R/W | CTS Source Select:<br>0 = Send HW-updated CTS value in N/CTS packet (recommended)<br>1 = Send SW-updated CTS value in N/CTS packet (for diagnostic use)<br><br>**Note:** FUJITSU recommends that you do *not* set this bit to 1. | 0 |

### ACR AUDIO FREQUENCY REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x02 | FREQ_SVAL | RSVD0 | | | | | MCLK_CONF | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 2:0 | MCLK_CONF | R/W | MCLK input mode:<br>0b000 = MCLK is 128*Fs<br>0b001 = MCLK is 256*Fs<br>0b010 = MCLK is 384*Fs<br>0b011 = MCLK is 512*Fs<br>0b100 = MCLK is 768*Fs<br>0b101 = MCLK is 1024*Fs<br>0b110 = MCLK is 1152*Fs<br>0b111 = MCLK is 192*Fs<br><br>The HDMI Transmitter uses these bits to divide the MCLK input to produce CTS values according to the 128*Fs formula. The MCLK to Fs ratio is for the input Fs, not the down-sampled output Fs (see the ASRC register (0x7A:0x23), on page 213). | 0b001 |

### ACR N SOFTWARE VALUE REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x03 | N_SVAL1 | N_SVAL[7:0] | | | | | | | |
| 0x7A | 0x04 | N_SVAL2 | N_SVAL[15:8] | | | | | | | |
| 0x7A | 0x05 | N_SVAL3 | RSVD0 | | | | N_SVAL[19:16] | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 19:0 | N_SVAL | R/W | N Value for audio clock regeneration method; a 20-bit value. This must be written to the registers to create the correct divisor for audio clock regeneration. Only values greater than 0 are valid. This register must be written after a hardware reset. | 0 |

### ACR CTS SOFTWARE VALUE REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x06 | CTS_SVAL1 | CTS_SVAL[7:0] | | | | | | | |
| 0x7A | 0x07 | CTS_SVAL2 | CTS_SVAL[15:8] | | | | | | | |
| 0x7A | 0x08 | CTS_SVAL3 | RSVD0 | | | | CTS_SVAL[19:16] | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 19:0 | CTS_SVAL | R/W | CTS Value for audio clock regeneration method; a 20-bit value. Diagnostic use and applied only when the CTS_SEL bit (0x7A:0x01[0]) is set to 1. | 0 |

### ACR CTS HARDWARE VALUE REGISTERS

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x09 | CTS_HVAL1 | CTS_HVAL[7:0] | | | | | | | |
| 0x7A | 0x0A | CTS_HVAL2 | CTS_HVAL[15:8] | | | | | | | |
| 0x7A | 0x0B | CTS_HVAL3 | RSVD0 | | | | CTS_HVAL[19:16] | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 19:0 | CTS_HVAL | R | CTS Value for audio clock regeneration method; a 20-bit value. This value is measured and stored here by the hardware when MCLK is active and N is valid, after 128Fs/N cycles of MCLK. | X |

### AUDIO IN MODE REGISTER

| Dev | Add | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x14 | AUD_MOD E | SD3_EN | SD2_EN | SD1_EN | SD0_EN | RSVDRW0 | RSVDRW0 | SPDIF_EN | AUD_EN |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7 | SD3_EN | R/W | I²S input channel #3 enable. | 0 |
| 6 | SD2_EN | R/W | I²S input channel #2 enable. | 0 |
| 5 | SD1_EN | R/W | I²S input channel #1 enable. | 0 |
| 4 | SD0_EN | R/W | I²S input channel #0 enable. | 0 |
| 1 | SPDIF_EN | R/W | S/PDIF input stream enable:<br>0 = Disabled<br>1 = Enabled | 0 |
| 0 | AUD_EN | R/W | Audio input stream enable:<br>0 = Disabled<br>1 = Enabled | 0 |

**Audio input data is selected from either the S/PDIF input or the I²S inputs. Audio input data can be disabled by clearing the AUD_EN bit. See Figure 10 on page 224.**

**See the note on the HDMI Transmitter's limitation in assigning the I²S channels to audio FIFOs, on page 211.**

### AUDIO IN S/PDIF CONTROL REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x15 | SPDIF_CTRL | RSVDRW0 | | | | NOAUDIO | RSVDRW0 | FS_OVERRIDE | RSVDRW0 |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 3 | NOAUDIO | R | No S/PDIF audio:<br>1 = No change detected on the S/PDIF input<br>0 = Detected change on the S/PDIF input | 1 |
| 1 | FS_OVERRIDE | R/W | S/PDIF input stream override:<br>0 = Use input S/PDIF stream's detected FS<br>1 = Use software FS in I2S_CHST4 register (0x7A:0x21) | 0 |

### AUDIO IN S/PDIF EXTRACTED FS AND LENGTH REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x18 | HW_SPDIF_FS | HW_SPDIF_LEN | | | HW_MAXLEN | HW_SPDIF_FS | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7:5 | HW_SPDIF_LEN | R | Channel status bits 33 to 35 (bit 33=LSB, bit 35=MSB)<br>Combines with HW_MAXLEN to indicate sample size: | 110 |
| 4 | HW_MAXLEN | R | Maximum sample length (channel status bit 32):<br>1 = Maximum sample length is 24 bits<br>0 = Maximum sample length is 20 bits | 1 |
| 3:0 | HW_SPDIF_FS | R | Set to the FS extracted from the S/PDIF input channel status bits 24-27. | 0000 |

| Bits | Max 24 | Max 20 |
|---|---|---|
| 000 | not indicated | |
| 001 | 20 bits | 16 bits |
| 010 | 22 bits | 18 bits |
| 100 | 23 bits | 19 bits |
| 101 | 24 bits | 20 bits |
| 110 | 21 bits | 17 bits |

### AUDIO IN I²S CHANNEL SWAP REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x19 | SWAP_I2S | SWCH3 | SWCH2 | SWCH1 | SWCH0 | RSVDRW | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7 | SWCH3 | R/W | Swap left-right channels for I²S Channel 3:<br>0 = Do not swap left and right<br>1 = Swap left and right | 0 |
| 6 | SWCH2 | R/W | Swap left-right channels for I²S Channel 2. | 0 |
| 5 | SWCH1 | R/W | Swap left-right channels for I²S Channel 1. | 0 |
| 4 | SWCH0 | R/W | Swap left-right channels for I²S Channel 0. | 0 |
| 3:0 | RSVDRW | R/W | Reserved – do not modify. | 0x9 |

**Note:** Each SWCH[3:0] bit is active *only* when the corresponding I²S input channel is enabled with register 0x7A:0x14.

### AUDIO ERROR THRESHOLD REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x1B | SPDIF_ERTH | RSVDRW0 | RSVDRW1 | AUD_ERR_THRESH | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 5:0 | AUD_ERR_THRESH | R/W | Specifies the error threshold level. The frame is marked as invalid if the number of bi-phase mark encoding errors in the audio stream exceeds this threshold level during frame decoding. | 0b001000 |

### AUDIO IN I²S DATA IN MAP REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x1C | I2S_IN_MAP | FIFO3_MAP | | FIFO2_MAP | | FIFO1_MAP | | FIFO0_MAP | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7:6 | FIFO3_MAP | R/W | Channel map to FIFO #3 (for HDMI Layout 1):<br>0b00 = Map SD0 to FIFO #3<br>0b01 = Map SD1 to FIFO #3<br>0b10 = Map SD2 to FIFO #3<br>0b11 = Map SD3 to FIFO #3 | 0b11 |
| 5:4 | FIFO2_MAP | R/W | Channel map to FIFO #2 (for HDMI Layout 1):<br>0b00 = Map SD0 to FIFO #2<br>0b01 = Map SD1 to FIFO #2<br>0b10 = Map SD2 to FIFO #2<br>0b11 = Map SD3 to FIFO #2 | 0b10 |
| 3:2 | FIFO1_MAP | R/W | Channel map to FIFO #1 (for HDMI Layout 1):<br>0b00 = Map SD0 to FIFO #1<br>0b01 = Map SD1 to FIFO #1<br>0b10 = Map SD2 to FIFO #1<br>0b11 = Map SD3 to FIFO #1 | 0b01 |
| 1:0 | FIFO0_MAP | R/W | Channel map to FIFO #0 (for HDMI Layout 0 or 1):<br>0b00 = Map SD0 to FIFO #0<br>0b01 = Map SD1 to FIFO #0<br>0b10 = Map SD2 to FIFO #0<br>0b11 = Map SD3 to FIFO #0 | 0b00 |

**HDMI allows for up to eight channels of audio content. Two channels pass through each of the four FIFOs listed in the description for register 0x7A:0x1C. HDMI does not restrict the Source to use any specific subset of the FIFOs. For example, 4-channel content can use many combinations of two FIFOs and two fields in the Audio Packets (indicated by each packet's B.X and SP.X bits, see HDMI 1.1, Sections 5.3.4 and 7.6), limited only by the channel assignment choices in EIA/CEA-861B Section 8.3.2. The B and PR bits are set automatically in each Audio InfoFrame packet by the HDMI Transmitter logic, using both the 0x1C register settings and the I²S channel enables in the 0x7A:0x14 register.**

**Some HDMI Receiver chips may not make the arriving B and PR bit information accessible to the Sink's firmware. Therefore, the only indicator of *used audio channels* is in the Audio InfoFrame packet's Data Byte 4.**

### AUDIO IN I²S CONTROL REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x1D | I2S_IN_CTRL | RSVD0 | SCK_EDGE | RSVD0 | | I2S_WS | I2S_JUST | I2S_DIR | I2S_SHIFT |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 6 | SCK_EDGE | R/W | SCK sample edge:<br>0 = Sample edge is falling; SD3-SD0 and WS source should change state on the rising edge of SCK<br>1 = Sample clock is rising; SD3-SD0 and WS source should change state on the falling edge of SCK | 1 |
| 3 | I2S_WS | R/W | WS polarity:<br>0 = Left polarity when WS is LOW<br>1 = Left polarity when WS is HIGH | 0 |
| 2 | I2S_JUST | R/W | SD justify:<br>0 = Data is left-justified<br>1 = Data is right-justified | 1 |
| 1 | I2S_DIR | R/W | SD direction:<br>0 = MSB shifted first<br>1 = LSB shifted first | 0 |
| 0 | I2S_SHIFT | R/W | WS to SD first bit shift:<br>0 = First bit shift (refer to the Philips Specification)<br>1 = No shift | 1 |

### AUDIO IN I²S CHANNEL STATUS REGISTERS

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x1E | I2S_CHST1 | cbit7 | cbit6 | cbit5 | cbit4 | cbit3 | cbit2 | cbit1 | cbit0 |
| 0x7A | 0x1F | I2S_CHST2 | cbit15 | cbit14 | cbit13 | cbit12 | cbit11 | cbit10 | cbit9 | cbit8 |
| 0x7A | 0x20 | I2S_CHST3 | I2S_CHAN_NUM | | | | I2S_SRC_NUM | | | |
| 0x7A | 0x21 | I2S_CHST4 | CLK_ACCUR | | | | SW_SPDIF_FS | | | |
| 0x7A | 0x22 | I2S_CHST5 | FS_ORIG | | | | I2S_LEN | | | I2S_MAXLEN |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7:0 | I2S_CHST1 | R/W | Channel Status Byte #0 | 0x00 |
| 7:0 | I2S_CHST2 | R/W | Channel Status Byte #1: Category Code | 0x00 |
| 7:4 | I2S_CHAN_NUM | R/W | Channel Status Byte #2: Source Number | 0b0000 |
| 3:0 | I2S_SRC_NUM | R/W | Channel Status Byte #3: Source Number | 0b0000 |
| 7:4 | CLK_ACCUR | R/W | Clock Accuracy | 0b0000 |
| 3:0 | SW_SPDIF_FS | R/W | Sampling frequency as set by software. If S/PDIF source is connected, this value will be inserted into the S/PDIF stream if FS_OVERRIDE is enabled. See note below. | 0b1111 |
| 7:4 | FS_ORIG | R/W | Original Fs | 0b0000 |
| 3:1 | I2S_LEN | R/W | Audio sample word length: Defined in bits with I2S_MAXLEN:<br>I2S LEN / I2S_MAXLEN=0 / I2S_MAXLEN=1<br>0b000 / not indicated / not indicated<br>0b001 / 16 / 20<br>0b010 / 18 / 22<br>0b100 / 19 / 23<br>0b101 / 20 / 24<br>0b110 / 17 / 21 | 0b000 |
| 0 | I2S_MAXLEN | R/W | Maximum audio sample word length<br>0 = 20 bits<br>1 = 24 bits | 1 |

**Note: The word length bits [3:0] should always match the word length of the audio samples coming *into* the HDMI Transmitter, even if the audio is down-sampled in the Transmitter. The word length of the input I²S stream is set in**

the 0x7A:0x24 register, described on page 214. Audio stream down sampling is enabled in the 0x7A:0x23 register, described on page 213.

HDMI 1.0 requires that accurate channel status information be transmitted in the Audio InfoFrames. When the audio is supplied to the HDMI Transmitter at the S/PDIF input, the channel status is extracted from the S/PDIF stream. When the audio is supplied by I²S inputs, the firmware must write accurate values into the channel status registers. Refer to IEC 60958-3 for detailed definitions of the channel status bits. The channel status information is used in the HDMI Receiver to reconstruct the audio into I²S format.

When down-sampling the audio stream (see the ASRC register on page 213), both CHST5 and CHST4 should be loaded with the original (input) Fs rate. The chip sends the original Fs in the audio packet channel status bits corresponding to CHST5, divides the original Fs rate (according to register ASRC), and sends that slower Fs in the bits corresponding to CHST4. Always write the input Fs into CHST4 and CHST5 when connected to an I²S source. A S/PDIF source loads both bytes automatically.

If FS_OVERRIDE is set to 0 (0x7A:0x15[1]), the sampling frequency is extracted from the S/PDIF input stream. It is encoded in that stream's Channel Status bits as shown in Table 20.

TABLE 20 ENCODED AUDIO SAMPLING FREQUENCY

| CH_ST | | | | Fs Sampling |
|---|---|---|---|---|
| 3 | 2 | 1 | 0 | Frequency |
| 0 | 0 | 0 | 0 | 44.1kHz |
| 1 | 0 | 0 | 0 | 88.2kHz |
| 1 | 1 | 0 | 0 | 176.4kHz |
| 0 | 0 | 1 | 0 | 48kHz |
| 1 | 0 | 1 | 0 | 96kHz |
| 1 | 1 | 1 | 0 | 192kHz |
| 0 | 0 | 1 | 1 | 32kHz |
| 0 | 0 | 0 | 1 | no indicated |

Note: The value in FS_OVERRIDE is provided to assist with source systems that do not provide correct FS information in the raw S/PDIF stream. The HDMI Specification requires the transmitted FS value to be correct in the channel status bits of the audio sample packets (see Section 7.3). By setting FS_OVERRIDE to 1, the HDMI Transmitter can be programmed (in SW_SPDIF_FS) with the correct FS value and can ignore the value in the input S/PDIF stream. *Values not listed in Table 20 are reserved*.

### AUDIO SAMPLE RATE CONVERSION REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x23 | ASRC | RSVDRW0 | | | | | | RATIO | SRC_EN |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 1 | RATIO | R/W | Sample rate down-conversion ratio:<br>0 = Down-sample 2-to-1 when SRC_EN is set to 1<br>1 = Down-sample 4-to-1 when SRC_EN is set to 1 | 0 |
| 0 | SRC_EN | R/W | Audio sample rate conversion:<br>0 = Disabled<br>1 = Enabled | 0 |

Sample rate conversion is applied only to 2-channel audio, either from the S/PDIF input or from the I²S Channel 0 input. Setting register ASRC to 0x01, down-samples 96KHz audio to 48KHz and 88.2KHz to 44.1KHz. Setting ASRC to 0x02, down-samples 192KHz to 48KHz and 174.4KHz to 44.1KHz. This conversion is performed after selecting S/PDIF or I²S input paths to the HDMI Transmitter. The CHST5 bits written in register 0x7A:0x22 should always indicate the sample rate before any down-sampling (see page 212).

If an audio source is connected to both the HDMI Transmitter and an audio DAC, the audio source may output 196KHz (or 176.4KHz) audio to drive the DAC and the Transmitter. The DAC uses the higher sample rate, while the

**HDMI Transmitter simultaneously down-samples that stream to the 48KHz (or 44.1KHz) sample rate. 48KHz and 44.1KHz are the default audio rates for HDMI.**

**The N value and the I²S channel status registers should be set per input audio configuration. The N value is affected by the down-sampling so that the outgoing N value in the N/CTS packets represents the video-to-audio ratio for the down-sampled audio stream. For example, to input 96kHz audio with 74.25MHz video input and to output 48kHz audio, the N register value should be set to 12288. This N will be divided by the down-sample RATIO, so that the HDMI Transmitter outputs packets with N=6144.**

**Important: Sample rate conversion works only for 2-channel PCM audio; you must set LAYOUT to 0 in register 0x7A:0x2F.**

### AUDIO I²S INPUT LENGTH REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x24 | I2S_IN_LEN | RSVDRW0 | | | | IN_LENGTH | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 3:0 | IN_LENGTH | R/W | Number of valid bits in the input I²S stream. Used for the extraction of the I²S data from the input stream.<br>0b1111 – 0b1110 = N/A<br>0b1101 = 21 bit<br>0b1100 = 17 bit<br>0b1011 = 24 bit<br>0b1010 = 20 bit<br>0b1001 = 23 bit<br>0b1000 = 19 bit<br>0b0111 – 0b0110 = N/A<br>0b0101 = 22 bit<br>0b0100 = 18 bit<br>0b0011 = N/A<br>0b0010 = 16 bit<br>0b0001 – 0b0000 = N/A | 0b1011 |

### HDMI CONTROL REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x2F | HDMI_CTRL | RSVD0 | RSVDRW | PACKET_MODE | | | LAYOUT | | HDMI_MODE |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 5:3 | PACKET_MODE | R/W | Specifies the number of bits per pixel sent to the packetizer:<br>0b000 = Reserved<br>0b001 = Reserved<br>0b010 = Reserved<br>0b011 = Reserved<br>0b100 = 24 bits per pixel (8 bits per pixel; no packing)<br>0b101 = 30 bits per pixel (10 bits per pixel pack to 8 bits)<br>0b110 = 36 bits per pixel (12 bits per pixel pack to 8 bits)<br>0b111 = 48 bits per pixel (16 bits per pixel; no packing)<br><br>**Note**: The firmware must program 24 bits per pixel (8 bits per pixel; no packing) for initialization. | 0b000 |
| 2:1 | LAYOUT | R/W | Audio packet header layout indicator:<br>0b00 = Layout 0 (2-channel) | 0b00 |

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | 0b01 = Layout 1 (up to 8 channels)<br>0b1x = Reserved | | | | | | | |
| 0 | HDMI_MODE | R/W | HDMI mode:<br>0 = Disabled<br>1 = Enabled | | | | | | | 0 |

**AUDIO PATH STATUS REGISTER**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x30 | AUDO_TXSTAT | RSVD0 | | | | | MUTE | RSVD0 | |
| **Bit** | **Label** | **R/W** | **Description** | | | | | | | **Default** |
| 2 | MUTE | R | General Control Packet mute status:<br>0 = No packet with SETAVM=1 has been sent.<br>1 = A packet with SETAVM=1 has been sent.<br><br>The MUTE bit is equal to the SETAVM bit in register 0x7A.0xDF, described on page 221. MUTE is not set immediately when the SETAVM bit in register 0xDF is written. After writing SETAVM to 1, MUTE is set after Control Packet is transmitted in HDMI mode. In DVI mode, MUTE is set at the start of VSYNC. MUTE is cleared when a Control Packet with CLRAVM=1 is sent, or (in DVI mode) at the start of VSYNC after CLRAVM has been written to 1.<br><br>**Note:** The combinations {SETAVM, CLRAVM}={0,0} and {1,1} are not supported by *HDMI 1.0 Specification*, Section 5.3.6. For more details see page 222. | | | | | | | |

**DIAGNOSTIC POWER DOWN REGISTER**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x3D | DPD | RSVDRW0 | | | | TCLKPHZ# | PDIDCK# | PDOSC# | PDTOT# |
| **Bit** | **Label** | **R/W** | **Description** | | | | | | | **Default** |
| 3 | TCLKPHZ# | R/W | Selects the TCLK phase:<br>0 = Invert TCLK; change the phase 180 degrees<br>1 = Default phase; the same as TMDS core | | | | | | | 1 |
| 2 | PDIDCK# | R/W | Power down IDCK input:<br>0 = Power down; gate off IDCK signal to disable all IDCK-based logic<br>1 = Normal operation | | | | | | | 1 |
| 1 | PDOSC# | R/W | Power down Internal Oscillator. This disables the I²C port to the internal ROM (disabling loading), halts all interrupt updates, and disables the Master DDC block.<br>0 = Power down<br>1 = Normal operation | | | | | | | 1 |
| 0 | PDTOT# | R/W | Power down total:<br>0 = Power down everything; INT source is RSEN.<br>1 = Normal operation | | | | | | | 1 |

**Refer to the page 259 for details on the restrictions for using these power-down bits.**

## Packet Registers

For all the packets described in the following registers, the packet transmission is controlled by two bits that allow the packet to be sent once or repeatedly. To send a packet one time, set the *enable* bit only after the corresponding packet data has been written into the appropriate registers. Read the enable bit to determine if the packet has actually been transmitted across the link. When cleared to zero, the packet has been transmitted.

If you want to send the packet repeatedly, write the *repeat* bit to 1 at the same time as enable is set to 1. This sends the corresponding packet during every VBLANK period.

To disable the repeated transmission, clear the repeat and enable bits simultaneously.

To guarantee that the HDMI Receiver remains in HDMI mode, at least one HDMI packet must be transmitted every two VSYNC periods (see the HDMI Specification, Section 5.2.3.2). This is accomplished automatically whenever audio is being transmitted, but if audio has not yet been enabled or if the HDMI Transmitter is not sending audio for some other reason, it is recommended that a Null Packet (all zeroes) be transmitted during every VBLANK. This can be accomplished by setting the contents of the Generic Packet buffer to all zeroes and then setting both enable and repeat bits for that packet.

The ID, TYPE, checksum, and length fields in each HDMI InfoFrame must be loaded by the microcontroller. There is no internal logic for calculating the checksum automatically, or any preset length value.

Note: The enable bits in registers 0x7A:0x3E-0x3F can only be set or cleared when the HDMI Transmitter is *not* in a powered-down state. The following bits must not be set: PD# (0x72:0x08[0]) and PDTOT# (0x7A:0x3D[0]). See page 159 for more details on power-down bits. Also, IDCK must be active. The data bytes in each InfoFrame, as well as the SETAVM/CLRAVM bits in the Control Packet and the repeat bit for each packet type can be written when the HDMI Transmitter is powered-down. Therefore, the firmware can write all necessary registers except the enable bits, then de-assert any power-down bits, and finally write any necessary enable bits.

Note: The enable and repeat bits for the various packet types are not affected by the state of the HDMI_MODE bit (register 0x7A:0x2F[0], described on page 214). Although packets cannot be transmitted in DVI mode (when HDMI_MODE is set to 0), and the HDMI Transmitter ignores the states of registers 0x3E-0x3F when HDMI_MODE is set to 0, the firmware should clear the packet enable and repeat bits whenever switching to DVI mode so that the status of packet sending, when read back from registers 0x3E-0x3F, is consistent with the link mode. All packet enables and repeats are set to their default values after hardware reset.

### PACKET BUFFER CONTROL #1 REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x3E | PB_CTRL1 | MPEG_EN | MPEG_RPT | AUD_EN | AUD_RPT | SPD_EN | SPD_RPT | AVI_EN | AVI_RPT |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7 | MPEG_EN | R/W | Enable MPEG InfoFrame transmission:<br>0 = Disabled<br>1 = Enabled | 0 |
| 6 | MPEG_RPT | R/W | Repeat MPEG InfoFrame transmission:<br>0 = Disabled (send once after *enable* bit is set)<br>1 = Enabled (send in every VBLANK period) | 0 |
| 5 | AUD_EN | R/W | Enable Audio InfoFrame transmission:<br>0 = Disabled<br>1 = Enabled | 0 |
| 4 | AUD_RPT | R/W | Repeat Audio InfoFrame transmission:<br>0 = Disabled (send once after *enable* bit is set)<br>1 = Enabled (send in every VBLANK period) | 0 |
| 3 | SPD_EN | R/W | Enable SPD InfoFrame transmission:<br>0 = Disabled<br>1 = Enabled | 0 |
| 2 | SPD_RPT | R/W | Repeat SPD InfoFrame transmission:<br>0 = Disabled (send once after *enable* bit is set)<br>1 = Enabled (send in every VBLANK period) | 0 |
| 1 | AVI_EN | R/W | Enable AVI InfoFrame transmission:<br>0 = Disabled<br>1 = Enabled | 0 |
| 0 | AVI_RPT | R/W | Repeat AVI InfoFrame transmission:<br>0 = Disabled (send once after *enable* bit is set)<br>1 = Enabled (send in every VBLANK period) | 0 |

### PACKET BUFFER CONTROL #2 REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x3F | PB_CTRL2 | RSVD0 | | GEN2_EN | GEN2_RPT | CP_EN | CP_RPT | GEN_EN | GEN_RPT |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 5 | GEN2_EN | R/W | Enable Generic #2 Packet transmission:<br>0 = Disabled<br>1 = Enabled | 0 |
| 4 | GEN2_RPT | R/W | Repeat Generic #2 Packet transmission:<br>0 = Disabled (send once after *enable* bit is set)<br>1 = Enabled (send in every VBLANK period) | 0 |
| 3 | CP_EN | R/W | Enable General Control Packet transmission:<br>0 = Disabled<br>1 = Enabled | 0 |
| 2 | CP_RPT | R/W | Repeat General Control Packet transmission:<br>0 = Disabled (send once after enable bit is set)<br>1 = Enabled (send in every VBLANK period) | 0 |
| 1 | GEN_EN | R/W | Enable Generic Packet transmission:<br>0 = Disabled<br>1 = Enabled | 0 |
| 0 | GEN_RPT | R/W | Repeat Generic Packet transmission:<br>0 = Disable (send once after enable bit is set)<br>1= Enable (send in every VBLANK period) | 0 |

**Note:** The PB_CTRL1 and PB_CTRL2 registers cannot be written when PD#=0, PDTOT#=0, or PDIDCK#=0; or if IDCK is inactive.

### PACKET REGISTERS

Refer to the HDMI and CEA-861B Specifications for a detailed explanation of these register fields.

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x40 | AVI_TYPE | AVI_HDR[7:0] | | | | | | | |
| 0x7A | 0x41 | AVI_VERS | AVI_HDR[15:8] | | | | | | | |
| 0x7A | 0x42 | AVI_LEN | AVI_HDR[23:16] | | | | | | | |
| 0x7A | 0x43 | AVI_CHSUM | AVI_HDR[31:24] | | | | | | | |
| 0x7A | 0x44 | AVI_DBYTE1 | AVI_DATA | | | | | | | |
| 0x7A | 0x45 | AVI_DBYTE2 | | | | | | | | |
| 0x7A | 0x46 | AVI_DBYTE3 | | | | | | | | |
| 0x7A | 0x47 | AVI_DBYTE4 | | | | | | | | |
| 0x7A | 0x48 | AVI_DBYTE5 | | | | | | | | |
| 0x7A | 0x49 | AVI_DBYTE6 | | | | | | | | |
| 0x7A | 0x4A | AVI_DBYTE7 | | | | | | | | |
| 0x7A | 0x4B | AVI_DBYTE8 | | | | | | | | |
| 0x7A | 0x4C | AVI_DBYTE9 | | | | | | | | |
| 0x7A | 0x4D | AVI_DBYTE10 | | | | | | | | |
| 0x7A | 0x4E | AVI_DBYTE11 | | | | | | | | |
| 0x7A | 0x4F | AVI_DBYTE12 | | | | | | | | |
| 0x7A | 0x50 | AVI_DBYTE13 | | | | | | | | |
| 0x7A | 0x51 | AVI_DBYTE14 | | | | | | | | |
| 0x7A | 0x52 | AVI_DBYTE15 | | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7:0 | AVI_TYPE | R/W | AVI InfoFrame Type Code. | 0x00 |
| 7:0 | AVI_VERS | R/W | AVI InfoFrame Version Code. | 0x00 |
| 7:0 | AVI_LEN | R/W | AVI InfoFrame Length. | 0x00 |
| 7:0 | AVI_CHSUM | R/W | AVI InfoFrame Checksum. | 0x00 |
| | AVI_DATA | R/W | AVI InfoFrame Data Bytes. | |

Refer to the page 252 for more details on the fields and valid settings in the AVI InfoFrame.

### SPD INFOFRAME REGISTERS

Refer to the CEA-861B Specification for a detailed explanation of these register fields.

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x60 | SPD_TYPE | SPD_HDR[7:0] | | | | | | | |
| 0x7A | 0x61 | SPD_VERS | SPD_HDR[15:8] | | | | | | | |
| 0x7A | 0x62 | SPD_LEN | SPD_HDR[23:16] | | | | | | | |
| 0x7A | 0x63 | SPD_CHSUM | SPD_HDR[31:24] | | | | | | | |
| 0x7A | 0x64 | SPD_DBYTE1 | SPD_DATA | | | | | | | |
| 0x7A | 0x65 | SPD_DBYTE2 | | | | | | | | |
| 0x7A | 0x66 | SPD_DBYTE3 | | | | | | | | |
| 0x7A | 0x67 | SPD_DBYTE4 | | | | | | | | |
| 0x7A | 0x68 | SPD_DBYTE5 | | | | | | | | |
| 0x7A | 0x69 | SPD_DBYTE6 | | | | | | | | |
| 0x7A | 0x6A | SPD_DBYTE7 | | | | | | | | |
| 0x7A | 0x6B | SPD_DBYTE8 | | | | | | | | |

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x6C | SPD_DBYTE9 | | | | | | | | |
| 0x7A | 0x6D | SPD_DBYTE10 | | | | | | | | |
| 0x7A | 0x6E | SPD_DBYTE11 | | | | | | | | |
| 0x7A | 0x6F | SPD_DBYTE12 | | | | | | | | |
| 0x7A | 0x70 | SPD_DBYTE13 | | | | | | | | |
| 0x7A | 0x71 | SPD_DBYTE14 | | | | | | | | |
| 0x7A | 0x72 | SPD_DBYTE15 | | | | | | | | |
| 0x7A | 0x73 | SPD_DBYTE16 | | | | | | | | |
| 0x7A | 0x74 | SPD_DBYTE17 | | | | | | | | |
| 0x7A | 0x75 | SPD_DBYTE18 | | | | | | | | |
| 0x7A | 0x76 | SPD_DBYTE19 | | | | | | | | |
| 0x7A | 0x77 | SPD_DBYTE20 | | | | | | | | |
| 0x7A | 0x78 | SPD_DBYTE21 | | | | | | | | |
| 0x7A | 0x79 | SPD_DBYTE22 | | | | | | | | |
| 0x7A | 0x7A | SPD_DBYTE23 | | | | | | | | |
| 0x7A | 0x7B | SPD_DBYTE24 | | | | | | | | |
| 0x7A | 0x7C | SPD_DBYTE25 | | | | | | | | |
| 0x7A | 0x7D | SPD_DBYTE26 | | | | | | | | |
| 0x7A | 0x7E | SPD_DBYTE27 | | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7:0 | SPD_TYPE | R/W | SPD InfoFrame Type Code. | 0x00 |
| 7:0 | SPD_VERS | R/W | SPD InfoFrame Version Code. | 0x00 |
| 7:0 | SPD_LEN | R/W | SPD InfoFrame Length. | 0x00 |
| 7:0 | SPD_CHSUM | R/W | SPD InfoFrame Checksum. | 0x00 |
| | SPD_DATA | R/W | SPD InfoFrame Data Bytes. | |

**AUDIO INFOFRAME REGISTERS**

**Refer to the HDMI and CEA-861B Specifications for a detailed explanation of these register fields.**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0x80 | AUDIO_TYPE | AUDIO_HDR[7:0] | | | | | | | |
| 0x7A | 0x81 | AUDIO_VERS | AUDIO_HDR[15:8] | | | | | | | |
| 0x7A | 0x82 | AUDIO_LEN | AUDIO_HDR[23:16] | | | | | | | |
| 0x7A | 0x83 | AUDIO_CHSUM | AUDIO_HDR[31:24] | | | | | | | |
| 0x7A | 0x84 | AUDIO_DBYTE1 | AUDIO_DATA | | | | | | | |
| 0x7A | 0x85 | AUDIO_DBYTE2 | | | | | | | | |
| 0x7A | 0x86 | AUDIO_DBYTE3 | | | | | | | | |
| 0x7A | 0x87 | AUDIO_DBYTE4 | | | | | | | | |
| 0x7A | 0x88 | AUDIO_DBYTE5 | | | | | | | | |
| 0x7A | 0x89 | AUDIO_DBYTE6 | | | | | | | | |
| 0x7A | 0x8A | AUDIO_DBYTE7 | | | | | | | | |
| 0x7A | 0x8B | AUDIO_DBYTE8 | | | | | | | | |
| 0x7A | 0x8C | AUDIO_DBYTE9 | | | | | | | | |
| 0x7A | 0x8D | AUDIO_DBYTE10 | | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7:0 | AUDIO_TYPE | R/W | AUDIO InfoFrame Type Code. | 0x00 |

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 7:0 | AUDIO_VERS | R/W | AUDIO InfoFrame Version Code. | | | | | | 0x00 | |
| 7:0 | AUDIO_LEN | R/W | AUDIO InfoFrame Length. | | | | | | 0x00 | |
| 7:0 | AUDIO_CHSUM | R/W | AUDIO InfoFrame Checksum. | | | | | | 0x00 | |
| | AUDIO_DATA | R/W | AUDIO InfoFrame Data Bytes. | | | | | | | |

**MPEG INFOFRAME REGISTERS**

**Refer to the CEA-861B Specification for a detailed explanation of these register fields.**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0xA0 | MPEG_TYPE | MPEG_HDR[7:0] | | | | | | | |
| 0x7A | 0xA1 | MPEG_VERS | MPEG_HDR[15:8] | | | | | | | |
| 0x7A | 0xA2 | MPEG_LEN | MPEG_HDR[23:16] | | | | | | | |
| 0x7A | 0xA3 | MPEG_CHSUM | MPEG_HDR[31:24] | | | | | | | |
| 0x7A | 0xA4 | MPEG_DBYTE1 | MPEG_DATA | | | | | | | |
| 0x7A | 0xA5 | MPEG_DBYTE2 | | | | | | | | |
| 0x7A | 0xA6 | MPEG_DBYTE3 | | | | | | | | |
| 0x7A | 0xA7 | MPEG_DBYTE4 | | | | | | | | |
| 0x7A | 0xA8 | MPEG_DBYTE5 | | | | | | | | |
| 0x7A | 0xA9 | MPEG_DBYTE6 | | | | | | | | |
| 0x7A | 0xAA | MPEG_DBYTE7 | | | | | | | | |
| 0x7A | 0xAB | MPEG_DBYTE8 | | | | | | | | |
| 0x7A | 0xAC | MPEG_DBYTE9 | | | | | | | | |
| 0x7A | 0xAD | MPEG_DBYTE10 | | | | | | | | |
| 0x7A | 0xAE | MPEG_DBYTE11 | | | | | | | | |
| 0x7A | 0xAF | MPEG_DBYTE12 | | | | | | | | |
| 0x7A | 0xB0 | MPEG_DBYTE13 | | | | | | | | |
| 0x7A | 0xB1 | MPEG_DBYTE14 | | | | | | | | |
| 0x7A | 0xB2 | MPEG_DBYTE15 | | | | | | | | |
| 0x7A | 0xB3 | MPEG_DBYTE16 | | | | | | | | |
| 0x7A | 0xB4 | MPEG_DBYTE17 | | | | | | | | |
| 0x7A | 0xB5 | MPEG_DBYTE18 | | | | | | | | |
| 0x7A | 0xB6 | MPEG_DBYTE19 | | | | | | | | |
| 0x7A | 0xB7 | MPEG_DBYTE20 | | | | | | | | |
| 0x7A | 0xB8 | MPEG_DBYTE21 | | | | | | | | |
| 0x7A | 0xB9 | MPEG_DBYTE22 | | | | | | | | |
| 0x7A | 0xBA | MPEG_DBYTE23 | | | | | | | | |
| 0x7A | 0xBB | MPEG_DBYTE24 | | | | | | | | |
| 0x7A | 0xBC | MPEG_DBYTE25 | | | | | | | | |
| 0x7A | 0xBD | MPEG_DBYTE26 | | | | | | | | |
| 0x7A | 0xBE | MPEG_DBYTE27 | | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 7:0 | MPEG_TYPE | R/W | MPEG InfoFrame Type Code. | 0x00 |
| 7:0 | MPEG_VERS | R/W | MPEG InfoFrame Version Code. | 0x00 |
| 7:0 | MPEG_LEN | R/W | MPEG InfoFrame Length. | 0x00 |
| 7:0 | MPEG_CHSUM | R/W | MPEG InfoFrame Checksum. | 0x00 |
| | MPEG_DATA | R/W | MPEG InfoFrame Data Bytes. | |

### GENERIC PACKET REGISTERS

**These registers may be used to transmit any type of packet.**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0xC0 | GEN_DBYTE1 | GEN_DATA | | | | | | | |
| 0x7A | 0xC1 | GEN_DBYTE2 | | | | | | | | |
| 0x7A | 0xC2 | GEN_DBYTE3 | | | | | | | | |
| 0x7A | 0xC3 | GEN_DBYTE4 | | | | | | | | |
| 0x7A | 0xC4 | GEN_DBYTE5 | | | | | | | | |
| 0x7A | 0xC5 | GEN_DBYTE6 | | | | | | | | |
| 0x7A | 0xC6 | GEN_DBYTE7 | | | | | | | | |
| 0x7A | 0xC7 | GEN_DBYTE8 | | | | | | | | |
| 0x7A | 0xC8 | GEN_DBYTE9 | | | | | | | | |
| 0x7A | 0xC9 | GEN_DBYTE10 | | | | | | | | |
| 0x7A | 0xCA | GEN_DBYTE11 | | | | | | | | |
| 0x7A | 0xCB | GEN_DBYTE12 | | | | | | | | |
| 0x7A | 0xCC | GEN_DBYTE13 | | | | | | | | |
| 0x7A | 0xCD | GEN_DBYTE14 | | | | | | | | |
| 0x7A | 0xCE | GEN_DBYTE15 | | | | | | | | |
| 0x7A | 0xCF | GEN_DBYTE16 | | | | | | | | |
| 0x7A | 0xD0 | GEN_DBYTE17 | | | | | | | | |
| 0x7A | 0xD1 | GEN_DBYTE18 | | | | | | | | |
| 0x7A | 0xD2 | GEN_DBYTE19 | | | | | | | | |
| 0x7A | 0xD3 | GEN_DBYTE20 | | | | | | | | |
| 0x7A | 0xD4 | GEN_DBYTE21 | | | | | | | | |
| 0x7A | 0xD5 | GEN_DBYTE22 | | | | | | | | |
| 0x7A | 0xD6 | GEN_DBYTE23 | | | | | | | | |
| 0x7A | 0xD7 | GEN_DBYTE24 | | | | | | | | |
| 0x7A | 0xD8 | GEN_DBYTE25 | | | | | | | | |
| 0x7A | 0xD9 | GEN_DBYTE26 | | | | | | | | |
| 0x7A | 0xDA | GEN_DBYTE27 | | | | | | | | |
| 0x7A | 0xDB | GEN_DBYTE28 | | | | | | | | |
| 0x7A | 0xDC | GEN_DBYTE29 | | | | | | | | |
| 0x7A | 0xDD | GEN_DBYTE30 | | | | | | | | |
| 0x7A | 0xDE | GEN_DBYTE31 | | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| | GEN_DATA | R/W | Generic Packet Data Bytes. | |

### GENERAL CONTROL PACKET REGISTER

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0xDF | CP_BYTE1 | RSVD0 | | | CLRAVM | RSVD0 | | | SETAVM |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| 4 | CLRAVM | R/W | Clear AV Mute flag. | 0 |
| 0 | SETAVM | R/W | Set AV Mute flag. | 0 |

**Note: Before enabling Control Packet transmission (CP_EN, CP_RPT), the firmware must write 0x10 or 0x01 to this register. The default value of 0x00 is not valid in a transmitted Control Packet.**

### GENERIC PACKET #2 REGISTERS

**These registers may be used to transmit any type of packet.**

| Dev | Addr | Name | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0x7A | 0xE0 | GEN2_DBYTE1 | GEN2_DATA | | | | | | | |
| 0x7A | 0xE1 | GEN2_DBYTE2 | | | | | | | | |
| 0x7A | 0xE2 | GEN2_DBYTE3 | | | | | | | | |
| 0x7A | 0xE3 | GEN2_DBYTE4 | | | | | | | | |
| 0x7A | 0xE4 | GEN2_DBYTE5 | | | | | | | | |
| 0x7A | 0xE5 | GEN2_DBYTE6 | | | | | | | | |
| 0x7A | 0xE6 | GEN2_DBYTE7 | | | | | | | | |
| 0x7A | 0xE7 | GEN2_DBYTE8 | | | | | | | | |
| 0x7A | 0xE8 | GEN2_DBYTE9 | | | | | | | | |
| 0x7A | 0xE9 | GEN2_DBYTE10 | | | | | | | | |
| 0x7A | 0xEA | GEN2_DBYTE11 | | | | | | | | |
| 0x7A | 0xEB | GEN2_DBYTE12 | | | | | | | | |
| 0x7A | 0xEC | GEN2_DBYTE13 | | | | | | | | |
| 0x7A | 0xED | GEN2_DBYTE14 | | | | | | | | |
| 0x7A | 0xEE | GEN2_DBYTE15 | | | | | | | | |
| 0x7A | 0xEF | GEN2_DBYTE16 | | | | | | | | |
| 0x7A | 0xF0 | GEN2_DBYTE17 | | | | | | | | |
| 0x7A | 0xF1 | GEN2_DBYTE18 | | | | | | | | |
| 0x7A | 0xF2 | GEN2_DBYTE19 | | | | | | | | |
| 0x7A | 0xF3 | GEN2_DBYTE20 | | | | | | | | |
| 0x7A | 0xF4 | GEN2_DBYTE21 | | | | | | | | |
| 0x7A | 0xF5 | GEN2_DBYTE22 | | | | | | | | |
| 0x7A | 0xF6 | GEN2_DBYTE23 | | | | | | | | |
| 0x7A | 0xF7 | GEN2_DBYTE24 | | | | | | | | |
| 0x7A | 0xF8 | GEN2_DBYTE25 | | | | | | | | |
| 0x7A | 0xF9 | GEN2_DBYTE26 | | | | | | | | |
| 0x7A | 0xFA | GEN2_DBYTE27 | | | | | | | | |
| 0x7A | 0xFB | GEN2_DBYTE28 | | | | | | | | |
| 0x7A | 0xFC | GEN2_DBYTE29 | | | | | | | | |
| 0x7A | 0xFD | GEN2_DBYTE30 | | | | | | | | |
| 0x7A | 0xFE | GEN2_DBYTE31 | | | | | | | | |

| Bit | Label | R/W | Description | Default |
|---|---|---|---|---|
| | GEN2_DATA | R/W | Generic Packet #2 Data Bytes. | |

# Appendices

The following sections describe how to use the HDMI Transmitter's functional block.

| Area | Page | Topic |
|---|---|---|
| Audio | 224 | Handling Audio |
| | 226 | Handling DVD Audio |
| Video | 228 | Programming Video Input Mode and Video Output Mode |
| Control | 252 | Handling InfoFrames |
| | 255 | Operating DDC Master |

# Handling Audio

### ENABLING AUDIO INPUTS

**Audio input data is accepted from either the S/PDIF input or one or more I²S channels. The AUD_EN bit (register 0x7A:0x14) is logically ANDed with each channel enable bit (S/PDIF and I²S in register 0x7A:0x14). You can stop audio processing by writing 0 to AUD_EN. This stops the decoding of input samples so that no samples are written into the Audio FIFOs. When the FIFOs are emptied by HDMI formatting, the HDMI Transmitter stops sending audio packets on the HDMI link. NCTS packets continue to be sent as long as the Transmitter is in HDMI mode.**

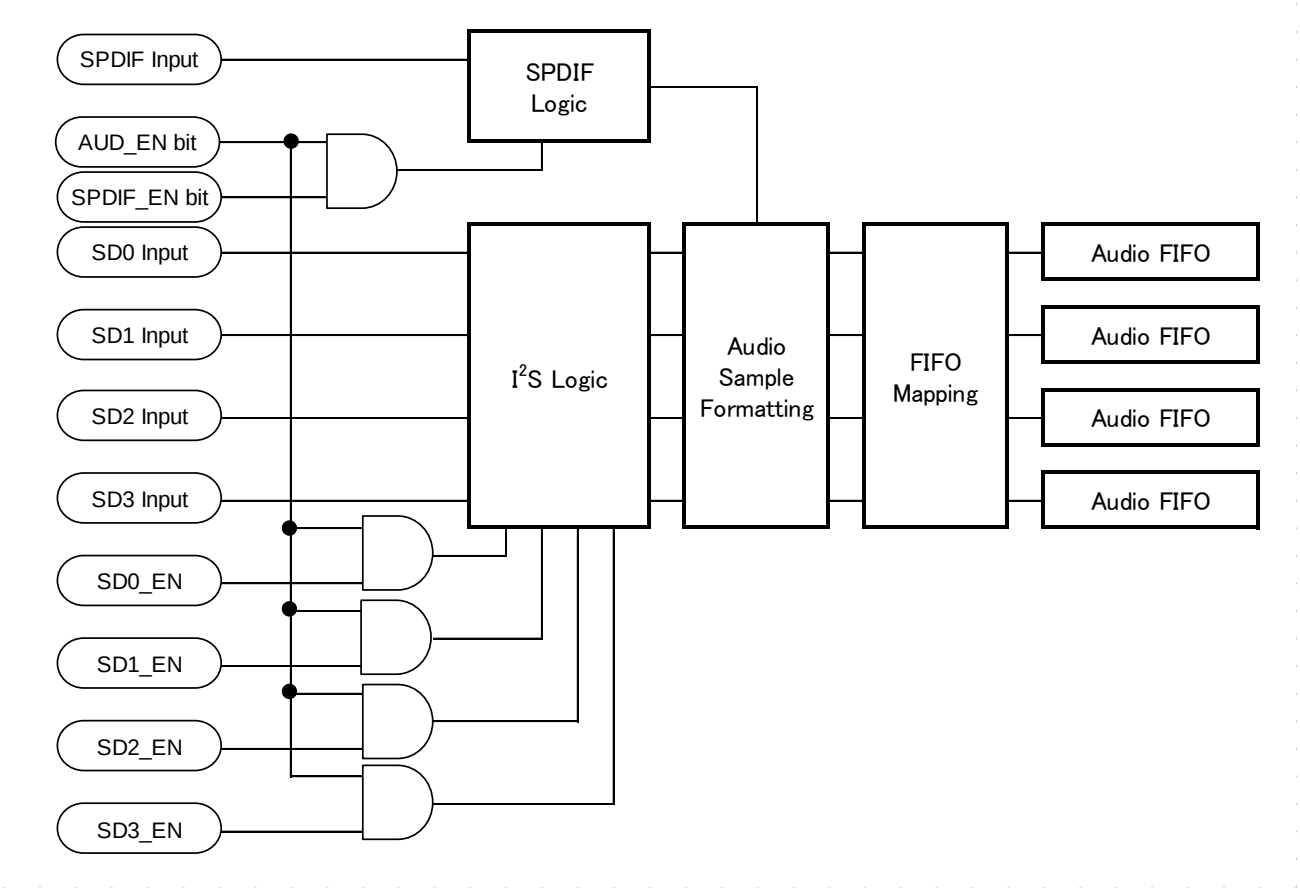



**FIGURE 10. AUDIO INPUT CONTROL**

## ENCODING AUDIO ON HDMI

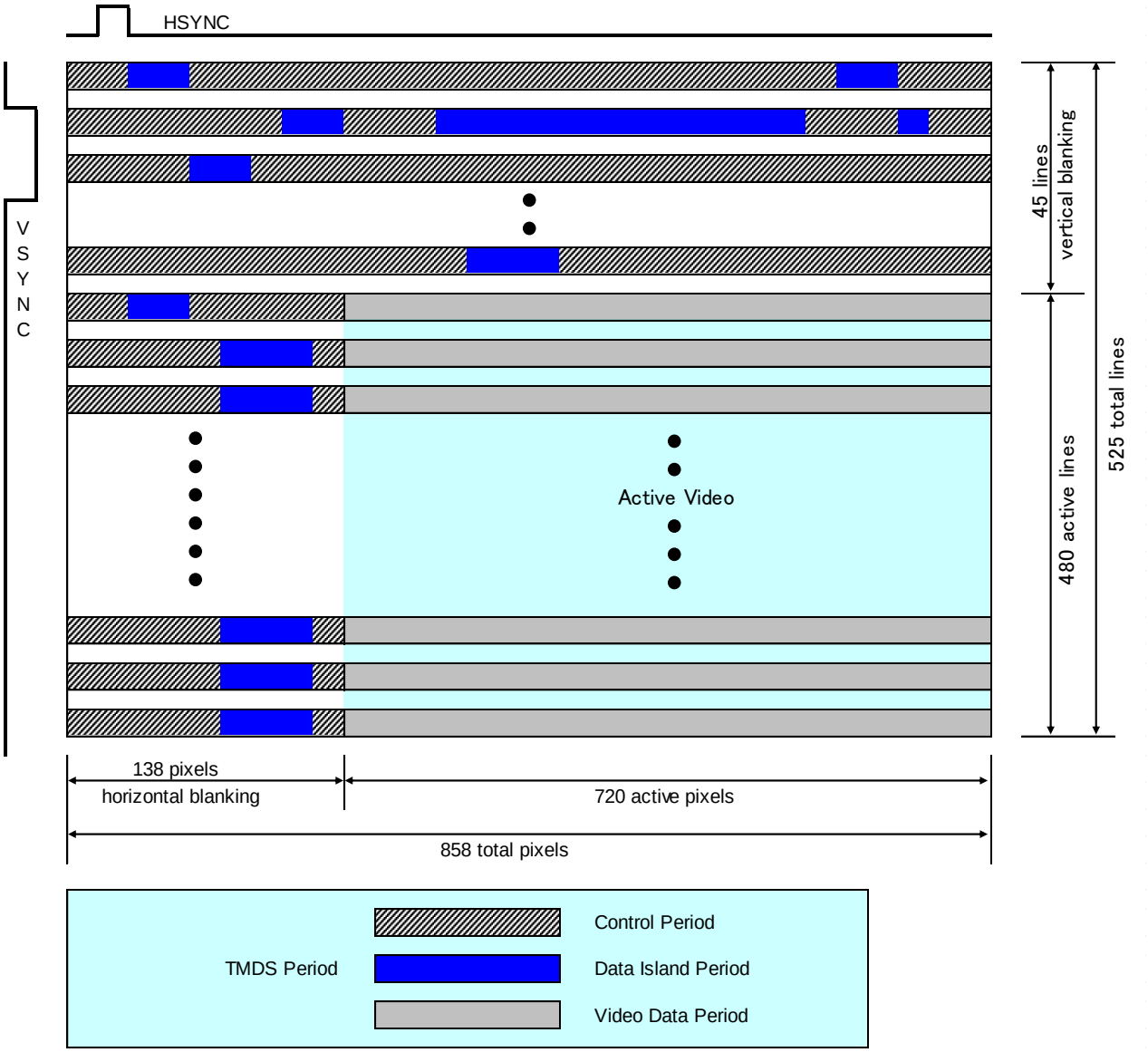



**FIGURE 11. OVERVIEW OF HDMI OPERATING MODES IN 480P STREAM**

# Handling DVD Audio

### SINK SUPPORT FOR AUDIO CONTENT PROTECTION

HDMI 1.1 describes three new packet types to support content protection for various audio formats: ACP, ISRC1, and ISRC2. The Source device transmits these packets, like other packets, as data islands during the vertical blanking time. The Source transmits these packets only if it recognizes the attached Sink or Repeater device as capable of receiving these packets as indicated in the EDID VSDB. The formats of the ACP, ISRC1, and ISRC2 packets are defined in HDMI 1.1, Sections 5.3.7 and 5.3.8.

The Sink and Repeater devices indicate support for these packets with a Vendor-Specific Data Block (VSDB) of length greater than 5 bytes in their E-EDID. This longer VSDB must be in the first 861B EDID timing extension, and includes bit fields new to HDMI 1.1. Older HDMI devices with 5-byte HDMI VSDBs indicate non-acceptance of ACP, ISRC1, or ISRC2 packets.

### TRANSMITTING ACP PACKETS

The HDMI Transmitter allocates register space for four packets with a full-size payload of 31 data bytes. Each includes a 4-byte header with TYPE, VERSION, LENGTH, and CHECKSUM fields. The SPD and MPEG InfoFrames are defined in CEA-861B, although their lengths are less than 31 bytes. The HDMI Transmitter expands the register space for those InfoFrame Packets to accommodate any type of packet with 31 total bytes. In addition, the HDMI Transmitter provides for a *Generic Packet* and *Generic Packet #2*. With these four packets, the HDMI Transmitter can set up and transmit any four of the five defined packet types: SPD, MPEG, ACP, ISRC1, and ISRC2.

ISRC packets are used together to transmit International Standard Recording Code (ISRC) data from the Source to the Sink. When this information extends beyond 16 bytes, the ISRC_CONT field in the ISRC1 packet is set in the header and the remaining bytes are sent in the ISRC2 packet. The requirements for handling ISRC data are defined in the *DVD Specifications for Read-Only Disc, Part 4 (Version 1.0, March 1999, Annex B)*.

The Source device is required by HDMI 1.1 to transmit ACP packets within 300 milliseconds of changing to audio content, which requires transmission of content protection information and to repeat transmission of ACP packets at least every 300 milliseconds. If a Sink does not detect ACP packets for 600 milliseconds, it assumes that there is no content protection information needed. Details on transmission timing requirements for ACP and related packets are described in *HDMI 1.1 Section 9.3*.

To control transmission of ACP and related packets with the HDMI Transmitter, the packet registers must be loaded and enabled. If, in addition to ACP, ISRC1, and ISRC2, the Source also needs to send MPEG and SPD InfoFrames, then one set of packet registers must alternate between one of those packet payloads and the ACP or related payload. One possible flowchart for this process is shown in Figure 12. If all five types are sent, the last register set alternates between sending SPD and MPEG InfoFrames. See page 216 for register details on enabling and disabling packet transmission.

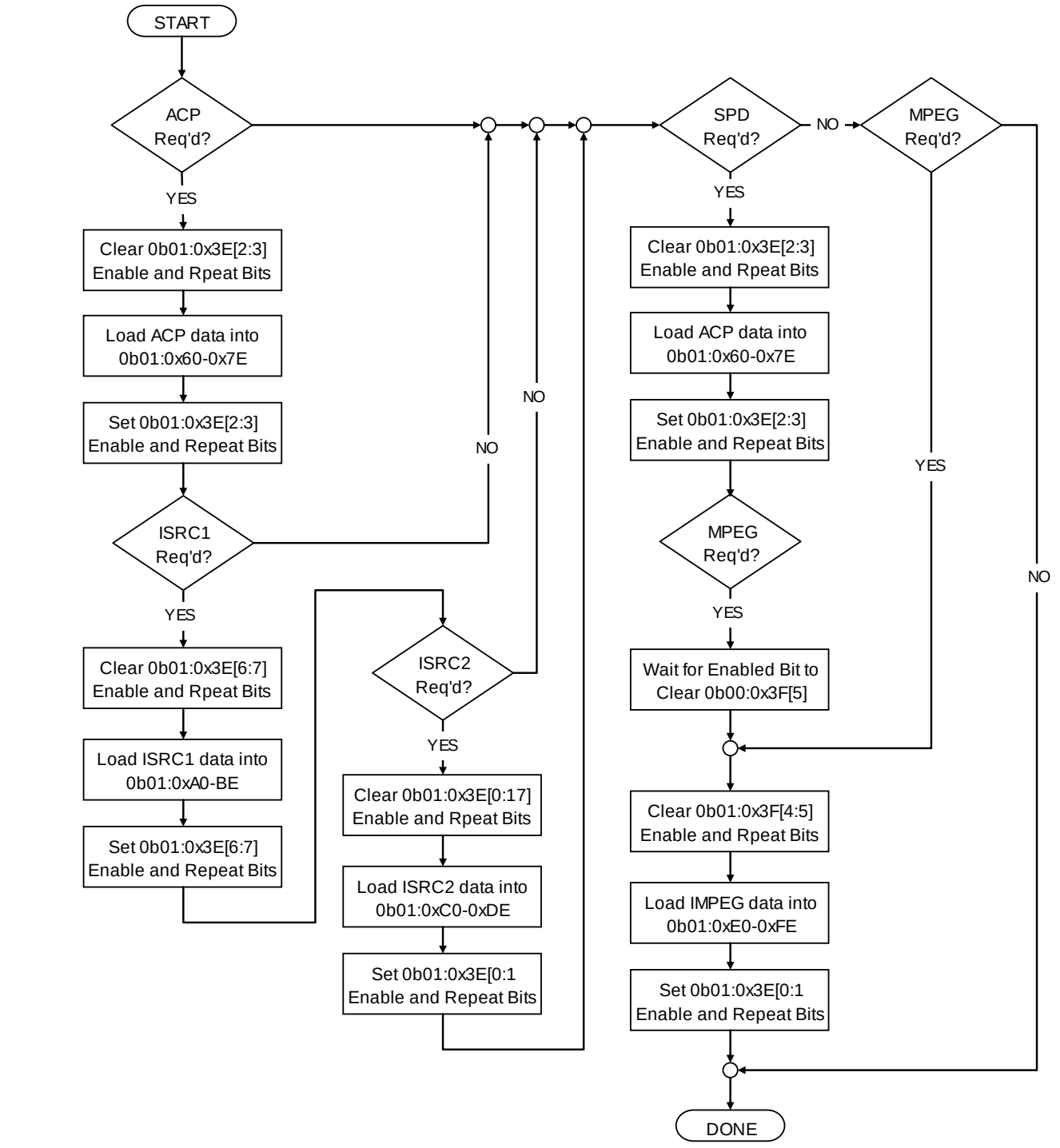



**FIGURE 12. ACP PACKET CONTROL FLOWCHART**

**Before loading new data into a packet buffer, the firmware must check that the previous packet was sent.**

# Handling Video

### PROGRAMMING VIDEO INPUT MODE AND VIDEO OUTPUT MODE

Specific registers in the HDMI Transmitter must be programmed according to the selection of input video bus mode and output video format.

The following notes apply to all tables on pages 233-247:

The DE parameters need to be programmed only when the DE Generator is enabled (0x72:0x33[6]).

The timings are based on EIA/CEA-861B specifications.

The parameters are application dependent. Consult the timing requirements of the Source device.

Set to 1 only when Y and C channels are each 12 bits wide. If these bus widths are less than 12 bits, set this bit to 0 and tie all unused pins to GND.

Shaded registers need not be programmed in this mode.

HBIT_TO_HSYNC, FIELD2_OFST, and VBIT_TO_VSYNC may differ depending on the porch timings in the input stream.

Set RANGE to 1 whenever converting YCbCr data and outputting full-range (0-255) RGB (PC mode) data across HDMI. When outputting limited-range (16-235) RGB (CE mode) data across the link, clear RANGE to 0.

Set WIDE_BUS to the number of bits per *input* video channel.

Set DITHER_MODE to the number of bits per *output* video channel supported by the Sink. By default, the HDMI Transmitter dithers the input (if DITHER 0x72:0x4A[5] is enabled) or truncates the input (if DITHER is disabled) to 8 bits.

Figure 13 diagrams the processing of video input data, leading to the TMDS encoding.

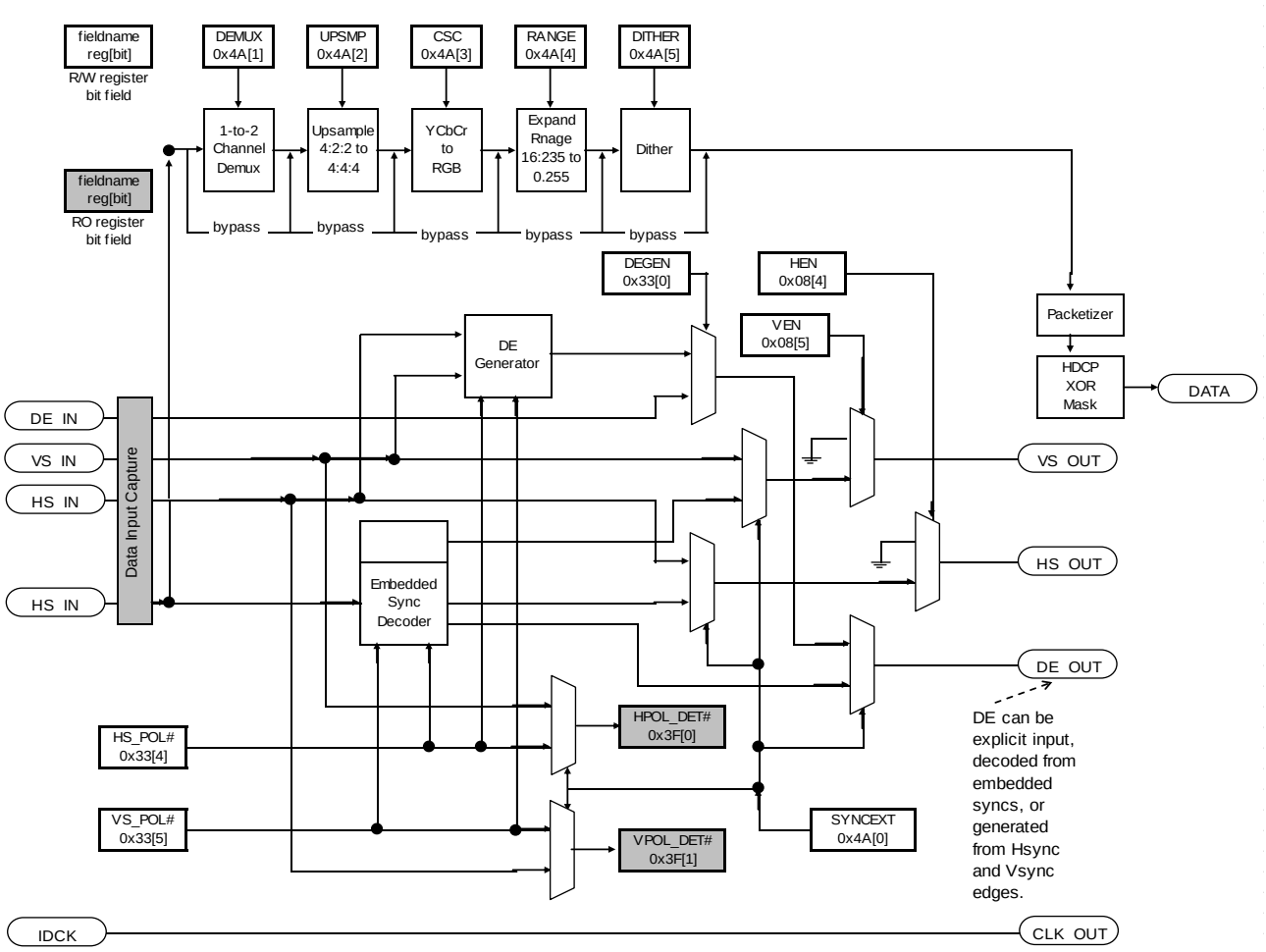



**FIGURE 13. TRANSMITTER VIDEO DATA PROCESSING PATH**

**If Range Expand is enabled (0x72:0x4A[4]) and Color Space Conversion from YCbCr to RGB is enabled (0x72:0x4A[3]), then all 3 color channels are expanded by a ratio of 1.164.**

**If DITHER is enabled (0x72:0x4A[5]), the video output is *dithered* to the output width specified in DITHER_MODE (0x72:0x4A[7:6]). If DITHER is disabled, the video output is *truncated* to the output width specified in DITHER_MODE.**

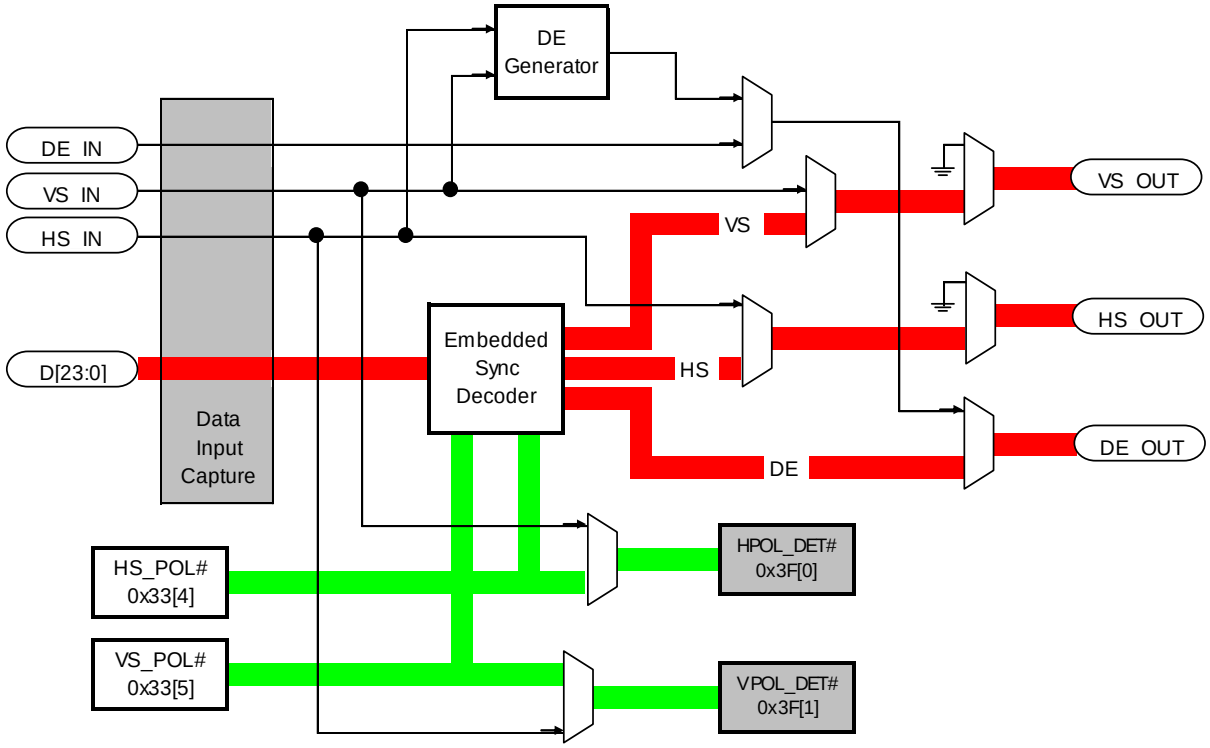



**FIGURE 14. TRANSMITTER VIDEO DATA WITH SYNC DECODING**

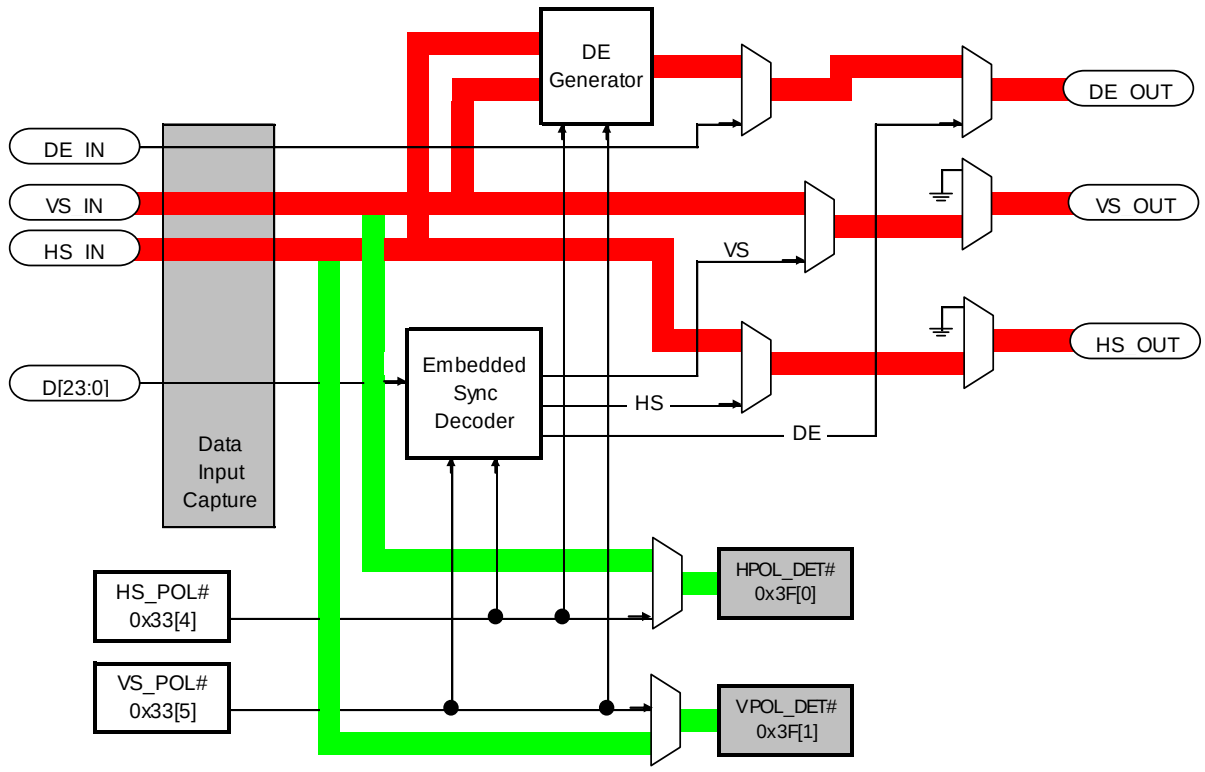



**FIGURE 15. TRANSMITTER VIDEO DATA WITH DE GENERATOR FROM VS, HS**

In Figure 14 and Figure 15, TMDS cores are generated using the signal paths colored red. The state of the VSYNC and HSYNC signals sent across the link can be read from VPOL_DET# and HPOL_DET#, as shown by the signal paths colored green.

### HANDLING INTERLACED VIDEO

Interlaced video is more challenging for the HDMI Transmitter because the timing from VSYNC to pixel data changes from even to odd fields. Also, many MPEG sources do not provide standard timings in interlaced modes.

This must be corrected by the HDMI Transmitter so that the video timing across the HDMI link is compliant with CEA-861B. The registers listed in Table 21 are involved in decoding embedded syncs and generating DE for interlaced modes.

TABLE 21 REGISTERS USED TO HANDLE INTERFACED VIDEO

| Field | Register | Address | Use |
|---|---|---|---|
| DE_ADJ# | IADJUST | 0x72:0x3E | Video Interface Adjustment. |
| F2VADJ | | | |
| F2VOFST | | | |
| I_DET | POL_DETECT | 0x72:0x3F | Video SYNC Polarity Detection. |
| VPOL_DET# | | | |
| HPOL_DET# | | | |
| HBIT_2H_SYNC | HBIT_2HSYNC | 0x72:0x40-0x41 | Video Hbit to HSYNC. |
| FIELD2_OFST | FLD2_HS_OFST | 0x72:0x42-0x43 | Video Field to HSYNC Offset. |
| HWIDTH | HWIDTH | 0x72:0x44-0x45 | Video HSYNC Length. |
| VBIT_TO_VSYNC | VBIT_TO_VSYNC | 0x72:0x46 | Video Vbit to VSYNC. |
| VWIDTH | VWIDTH | 0x72:0x47 | Video VSYNC Length. |

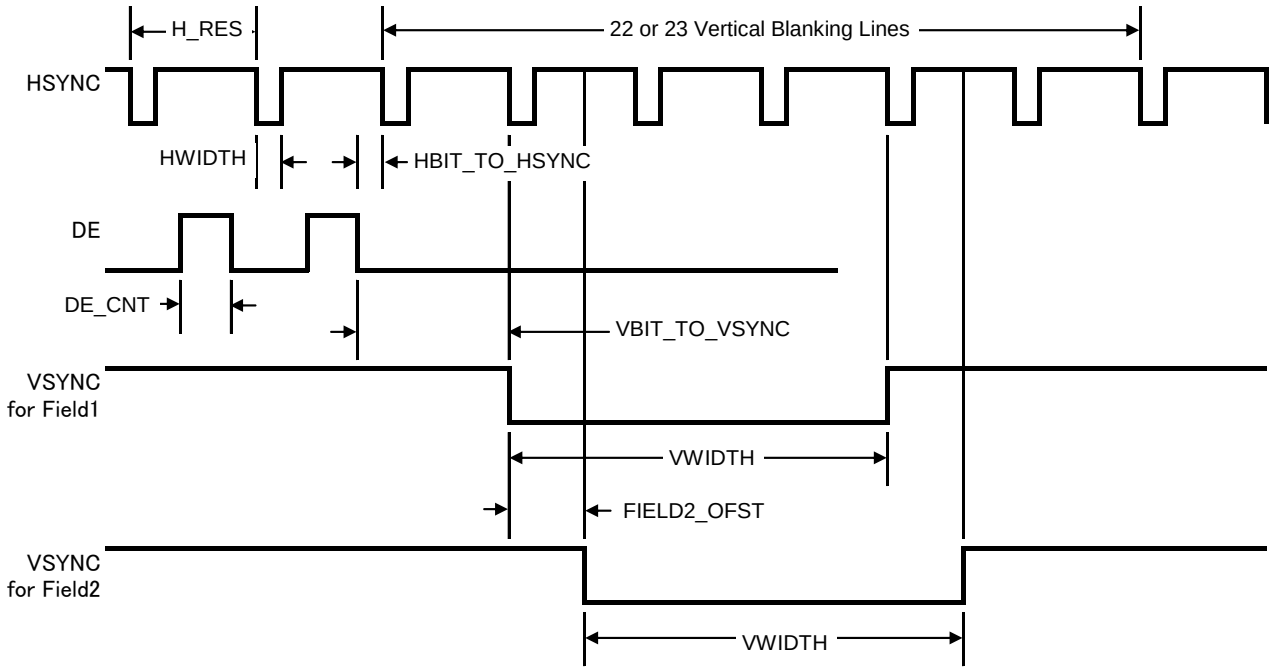



FIGURE 16. 480I EXAMPLE FOR HANDLING SYNCS

Note: Registers 0x72:0x40-0x46 are useful only when the input video uses 656 encoded syncs.

## Summary of Video Processing Path Options

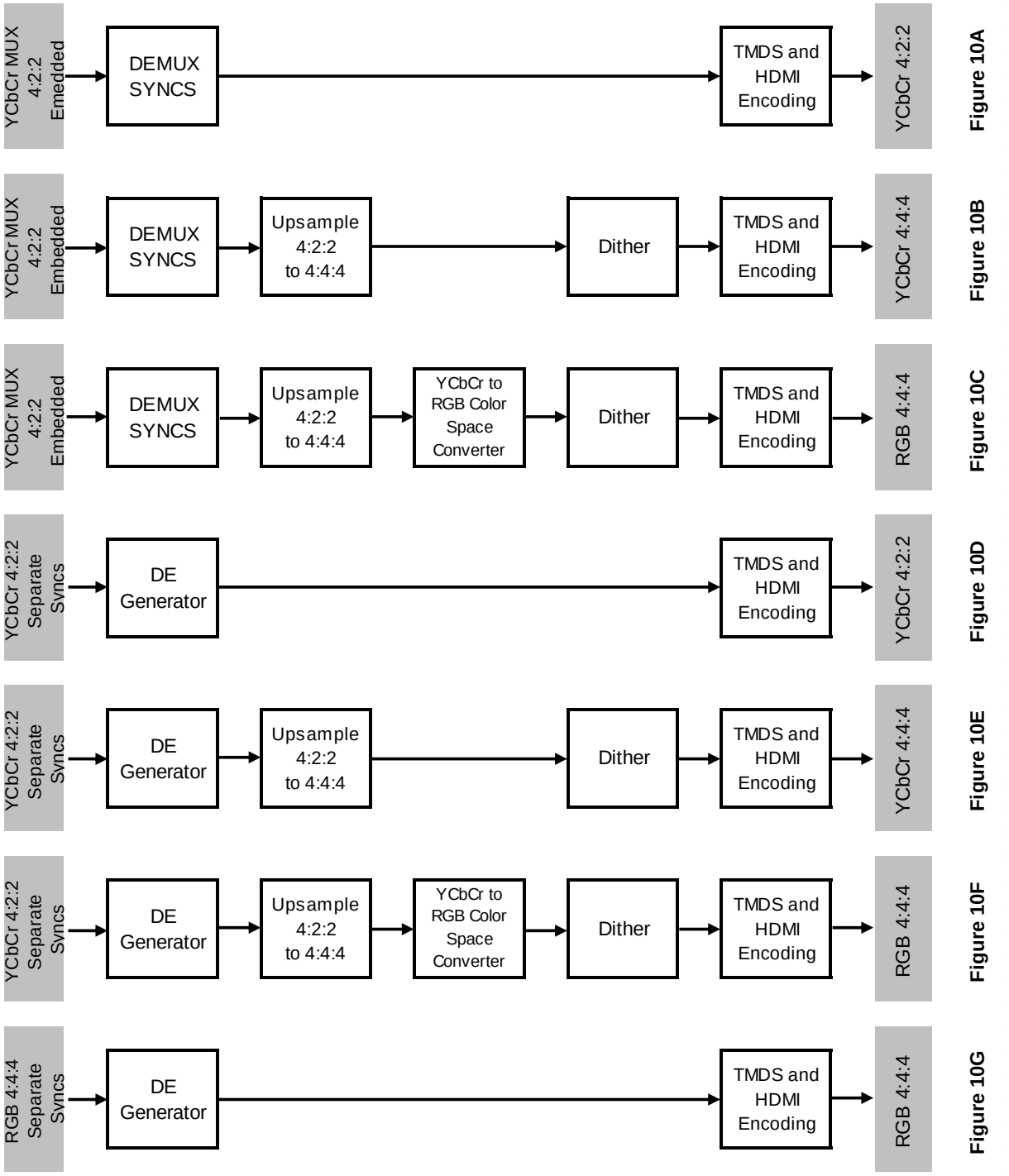



**FIGURE 17. VIDEO INPUT TO VIDEO OUTPUT DATA FLOW**

# 480i Input

### 480I MUXED YCBCR 4:2:2 EMBEDDED SYNC INPUT

| Input Mode | | | Muxed YCbCr 4:2:2 Embedded Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33 0x32 | 3:0 7:0 | | | | DE Delay | 1,2,3 | 192 |
| DE_TOP | 0x34 | 6:0 | | | | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37 0x36 | 3:0 7:0 | | | | DE Count | 1,2 | 193 |
| DE_LIN | 0x39 0x38 | 2:0 7:0 | | | | DE Lines | 1,2 | 194 |
| HS_POL# | 0x33 | 4 | 1 | 1 | 1 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 1 | 1 | 1 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 0 | 0 | 0 | DE_GEN Enable | | 193 |
| HBIT_TO_HSYNC | 0x41 0x40 | 1:0 7:0 | 00 0x13 | 00 0x13 | 00 0x13 | HBit to HSYNC Delay | 2 | 196 |
| FIELD2_OFST | 0x43 0x42 | 3:0 7:0 | 001 0xAD | 001 0xAD | 001 0xAD | Odd Field Offset | 2 | 196 |
| HWIDTH | 0x45 0x44 | 1:0 7:0 | 000 0x3E | 000 0x3E | 000 0x3E | HSYNC Pulse Width | 2 | 196 |
| VBIT_TO_VSYNC | 0x46 | 5:0 | 0x04 | 0x04 | 0x04 | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | 0x03 | 0x03 | 0x03 | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 00 | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 0 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 1 | 1 | 1 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 1 | 1 | 1 | 1-to-2 Chan Demux | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Upsampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | A | B | C | | | 232 |

**The video reconstructed by the HDMI Transmitter may not be compliant with CEA-861B timings, depending on the sync timings received from the Source device. See page 234 for register settings that enable both the sync decoder and the DE generator.**

**Refer to page 228 for additional notes.**

### 480I MUXED YCBCR 4:2:2 EMBEDDED SYNC INPUT – FIXING FOR CEA-861B

| Input Mode | | | Muxed YCbCr 4:2:2 Embedded Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33 | 3:0 | 0x0 | 0x0 | 0x0 | DE Delay | 1,2,3 | 192 |
| | 0x32 | 7:0 | 0x77 | 0x77 | 0x77 | | | |
| DE_TOP | 0x34 | 6:0 | 0x12 | 0x12 | 0x12 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37 | 3:0 | 0x2 | 0x2 | 0x2 | DE Count | 1,2 | 193 |
| | 0x36 | 7:0 | 0xD0 | 0xD0 | 0xD0 | | | |
| DE_LINC | 0x39 | 2:0 | 0x0 | 0x0 | 0x0 | DE Lines | 1,2 | 194 |
| | 0x38 | 7:0 | 0xF0 | 0xF0 | 0xF0 | | | |
| HS_POL# | 0x33 | 4 | 1 | 1 | 1 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 1 | 1 | 1 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 1 | 1 | 1 | DE_GEN Enable | | 193 |
| HBIT_TO_HSYNC | 0x41 | 1:0 | 00 | 00 | 00 | HBit to HSYNC Delay | 2, 6 | 196 |
| | 0x40 | 7:0 | 0x13 | 0x13 | 0x13 | | | |
| FIELD2_OFST | 0x43 | 3:0 | 001 | 001 | 001 | Odd Field Offset | 2, 6 | 196 |
| | 0x42 | 7:0 | 0xAD | 0xAD | 0xAD | | | |
| HWIDTH | 0x45 | 1:0 | 000 | 000 | 000 | HSYNC Pulse Width | 2 | 196 |
| | 0x44 | 7:0 | 0x3E | 0x3E | 0x3E | | | |
| VBIT_TO_VSYNC | 0x46 | 5:0 | 0x04 | 0x04 | 0x04 | VBit to VSYNC Delay | 2, 6 | 196 |
| VWIDTH | 0x47 | 5:0 | 0x03 | 0x03 | 0x03 | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 00 | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 0 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 1 | 1 | 1 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 1 | 1 | 1 | 1-to-2 Chan Demux | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Up sampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | A | B | C | | | 232 |

**The video reconstructed by the HDMI Transmitter may not be compliant with CEA-861B timings, depending on the sync timings received from the Source device. This table shows programming of both the sync decoder and the DE generator to correct porch timings to be CEA-861B compliant. See page 233 for register settings that enable only the sync decoder when the Source device provides compliant SAV/EAV timings.**

**Refer to page 228 for additional notes.**

### 480I YCBCR 4:2:2 MUX YC SEPARATE SYNC INPUT

| Input Mode | | | YCbCr 4:2:2 Mux YC Separate Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33 | 3:0 | 0x0 | 0x0 | 0x0 | DE Delay | 1,2,3 | 192 |
| | 0x32 | 7:0 | 0x77 | 0x77 | 0x77 | | | |
| DE_TOP | 0x34 | 6:0 | 0x12 | 0x12 | 0x12 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37 | 3:0 | 0x2 | 0x2 | 0x2 | DE Count | 1,2 | 193 |
| | 0x36 | 7:0 | 0xD0 | 0xD0 | 0xD0 | | | |
| DE_LIN | 0x39 | 2:0 | 0x0 | 0x0 | 0x0 | DE Lines | 1,2 | 194 |
| | 0x38 | 7:0 | 0xF0 | 0xF0 | 0xF0 | | | |
| HS_POL# | 0x33 | 4 | 1 | 1 | 1 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 1 | 1 | 1 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 1 | 1 | 1 | DE_GEN Enable | | 193 |
| HBIT_TO_HSYNC | 0x41 | 1:0 | | | | HBit to HSYNC Delay | 2 | 196 |
| | 0x40 | 7:0 | | | | | | |
| FIELD2_OFST | 0x43 | 3:0 | | | | Odd Field Offset | 2 | 196 |
| | 0x42 | 7:0 | | | | | | |
| HWIDTH | 0x45 | 1:0 | | | | HSYNC Pulse Width | 2 | 196 |
| | 0x44 | 7:0 | | | | | | |
| VBIT_TO_VSYNC | 0x46 | 5:0 | | | | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | | | | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 00 | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 0 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 0 | 0 | 0 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 1 | 1 | 1 | | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Up sampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | D | E | F | | | 232 |

Refer to page 228 for notes.

# 576i Input

### 576I YCBCR 4:2:2 EMBEDDED SYNC INPUT

| Input Mode | | | YCbCr 4:2:2 Embedded Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33<br>0x32 | 3:0<br>7:0 | | | | DE Delay | 1,2,3 | 193 |
| DE_TOP | 0x34 | 6:0 | | | | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37<br>0x36 | 3:0<br>7:0 | | | | DE Count | 1,2 | 193 |
| DE_LIN | 0x39<br>0x38 | 2:0<br>7:0 | | | | DE Lines | 1,2 | 194 |
| HS_POL# | 0x33 | 4 | 1 | 1 | 1 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 1 | 1 | 1 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 0 | 0 | 0 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41<br>0x40 | 1:0<br>7:0 | 00<br>0x0C | 00<br>0x0C | 00<br>0x0C | HBit to HSYNC Delay | 2 | 196 |
| FIELD2_OFST | 0x43<br>0x42 | 3:0<br>7:0 | 001<br>0xB0 | 001<br>0xB0 | 001<br>0xB0 | Odd Field Offset | 2 | 196 |
| HWIDTH | 0x45<br>0x44 | 1:0<br>7:0 | 000<br>0x3F | 000<br>0x3F | 000<br>0x3F | HSYNC Pulse Width | 2 | 196 |
| VBIT_2_VSYNC | 0x46 | 5:0 | 0x02 | 0x02 | 0x02 | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | 0x03 | 0x03 | 0x03 | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 01 | 01 | 01 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 0 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 1 | 1 | 1 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 1 | 1 | 1 | 1-to-2 Chan Demux | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Up sampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | A | B | C | | | 232 |

**The video reconstructed by the HDMI Transmitter may not be compliant with CEA-861B timings, depending on the sync timings received from the Source device. See page 237 for register settings that enable both the sync decoder and the DE generator.**

**Refer to page 228 for additional notes.**

### 576I YCBCR 4:2:2 EMBEDDED SYNC INPUT – FIXING FOR CEA-861B

| Input Mode | | | YCbCr 4:2:2 Embedded Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33<br>0x32 | 3:0<br>7:0 | 0x1<br>0x08 | 0x1<br>0x08 | 0x1<br>0x08 | DE Delay | 1,2,3 | 193 |
| DE_TOP | 0x34 | 6:0 | 0x16 | 0x16 | 0x16 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37<br>0x36 | 3:0<br>7:0 | 0x5<br>0xA0 | 0x5<br>0xA0 | 0x5<br>0xA0 | DE Count | 1,2 | 193 |
| DE_LIN | 0x39<br>0x38 | 2:0<br>7:0 | 0x1<br>0x20 | 0x1<br>0x20 | 0x1<br>0x20 | DE Lines | 1,2 | 194 |
| HS_POL# | 0x33 | 4 | 1 | 1 | 1 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 1 | 1 | 1 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 1 | 1 | 1 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41<br>0x40 | 1:0<br>7:0 | 00<br>0x0C | 00<br>0x0C | 00<br>0x0C | HBit to HSYNC Delay | 2, 6 | 196 |
| FIELD2_OFST | 0x43<br>0x42 | 3:0<br>7:0 | 001<br>0xB0 | 001<br>0xB0 | 001<br>0xB0 | Odd Field Offset | 2, 6 | 196 |
| HWIDTH | 0x45<br>0x44 | 1:0<br>7:0 | 000<br>0x3F | 000<br>0x3F | 000<br>0x3F | HSYNC Pulse Width | 2 | 196 |
| VBIT_2_VSYNC | 0x46 | 5:0 | 0x02 | 0x02 | 0x02 | VBit to VSYNC Delay | 2, 6 | 196 |
| VWIDTH | 0x47 | 5:0 | 0x03 | 0x03 | 0x03 | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 01 | 01 | 01 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 0 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 1 | 1 | 1 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 1 | 1 | 1 | 1-to-2 Chan Demux | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Up sampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | A | B | C | | | 232 |

**The video reconstructed by the HDMI Transmitter may not be compliant with CEA-861B timings, depending on the sync timings received from the Source device. This table shows programming of both the sync decoder and the DE generator to correct porch timings to be CEA-861B compliant. See page 236 for register settings that enable only the sync decoder when the Source device provides compliant SAV/EAV timings.**

**Refer to page 228 for additional notes.**

# 780p Input

### 720P RGB INPUT

| Input Mode | | | RGB | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33 | 3:0 | | | 0x1 | DE Delay | 1,2,3 | 193 |
| | 0x32 | 7:0 | | | 0x04 | | | |
| DE_TOP | 0x34 | 6:0 | | | 0x19 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37 | 3:0 | | | 0x5 | DE Count | 1,2 | 193 |
| | 0x36 | 7:0 | | | 0x00 | | | |
| DE_LIN | 0x39 | 2:0 | | | 0x2 | DE Lines | 1,2 | 194 |
| | 0x38 | 7:0 | | | 0xD0 | | | |
| HS_POL# | 0x33 | 4 | | | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | | | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | | | 1 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41 | 1:0 | | | | HBit to HSYNC Delay | 2 | 196 |
| | 0x40 | 7:0 | | | | | | |
| FIELD2_OFST | 0x43 | 3:0 | | | | Odd Field Offset | 2 | 196 |
| | 0x42 | 7:0 | | | | | | |
| HWIDTH | 0x45 | 1:0 | | | | HSYNC Pulse Width | 2 | 196 |
| | 0x44 | 7:0 | | | | | | |
| VBIT_2_VSYNC | 0x46 | 5:0 | | | | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | | | | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | | | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | | | 0 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | | | 0 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | | | 0 | | | 198 |
| UPSMP | 0x4A | 2 | | | 0 | Up sampling | | 198 |
| CSC | 0x4A | 3 | | | 0 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | | Range Select | | 198 |
| DITHER_MODE | 0x4A | 7:6 | | | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | | | 0 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | | | G | | | 232 |

**Refer to page 228 for notes.**

### 720P YCBCR 4:4:4 SEPARATE SYNC INPUT

| Input Mode | | | YCbCr 4:4:4 Separate Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33 | 3:0 | | 0x1 | 0x1 | DE Delay | 1,2,3 | 193 |
| | 0x32 | 7:0 | | 0x04 | 0x04 | | | |
| DE_TOP | 0x34 | 6:0 | | 0x19 | 0x19 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37 | 3:0 | | 0x5 | 0x5 | DE Count | 1,2 | 193 |
| | 0x36 | 7:0 | | 0x00 | 0x00 | | | |
| DE_LIN | 0x39 | 2:0 | | 0x2 | 0x2 | DE Lines | 1,2 | 194 |
| | 0x38 | 7:0 | | 0xD0 | 0xD0 | | | |
| HS_POL# | 0x33 | 4 | | 0 | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | | 0 | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | | 1 | 1 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41 | 1:0 | | | | HBit to HSYNC Delay | 2 | 196 |
| | 0x40 | 7:0 | | | | | | |
| FIELD2_OFST | 0x43 | 3:0 | | | | Odd Field Offset | 2 | 196 |
| | 0x42 | 7:0 | | | | | | |
| HWIDTH | 0x45 | 1:0 | | | | HSYNC Pulse Width | 2 | 196 |
| | 0x44 | 7:0 | | | | | | |
| VBIT_2_VSYNC | 0x46 | 5:0 | | | | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | | | | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 1 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | | 0 | 0 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | | 0 | 0 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | | 0 | 0 | | | 198 |
| UPSMP | 0x4A | 2 | | 0 | 0 | Up sampling | | 198 |
| CSC | 0x4A | 3 | | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | only | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | | 0 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | D | E | F | | | 232 |

**Refer to page 228 for notes.**

### 720P YCBCR 4:2:2 SEPARATE SYNC INPUT

| Input Mode | | | YCbCr 4:2:2 Separate Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33 | 3:0 | 0x1 | 0x1 | 0x1 | DE Delay | 1,2,3 | 193 |
| | 0x32 | 7:0 | 0x04 | 0x04 | 0x04 | | | |
| DE_TOP | 0x34 | 6:0 | 0x19 | 0x19 | 0x19 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37 | 3:0 | 0x5 | 0x5 | 0x5 | DE Count | 1,2 | 193 |
| | 0x36 | 7:0 | 0x00 | 0x00 | 0x00 | | | |
| DE_LIN | 0x39 | 2:0 | 0x2 | 0x2 | 0x2 | DE Lines | 1,2 | 194 |
| | 0x38 | 7:0 | 0xD0 | 0xD0 | 0xD0 | | | |
| HS_POL# | 0x33 | 4 | 0 | 0 | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 0 | 0 | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 1 | 1 | 1 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41 | 1:0 | | | | HBit to HSYNC Delay | 2 | 196 |
| | 0x40 | 7:0 | | | | | | |
| FIELD2_OFST | 0x43 | 3:0 | | | | Odd Field Offset | 2 | 196 |
| | 0x42 | 7:0 | | | | | | |
| HWIDTH | 0x45 | 1:0 | | | | HSYNC Pulse Width | 2 | 196 |
| | 0x44 | 7:0 | | | | | | |
| VBIT_2_VSYNC | 0x46 | 5:0 | | | | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | | | | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 00 | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 1 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 0 | 0 | 0 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 0 | 0 | 0 | | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Up sampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | only | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | D | E | F | | | 232 |

**Refer to page 228 for notes.**

### 720P YCBCR 4:2:2 EMBEDDED SYNC INPUT

| Input Mode | | | YCbCr 4:2:2 Embedded Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Note | Page |
| DE_DLY | 0x33<br>0x32 | 3:0<br>7:0 | | | | DE Delay | 1,2,3 | 193 |
| DE_TOP | 0x34 | 6:0 | | | | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37<br>0x36 | 3:0<br>7:0 | | | | DE Count | 1,2 | 193 |
| DE_LIN | 0x39<br>0x38 | 2:0<br>7:0 | | | | DE Lines | 1,2 | 194 |
| HS_POL# | 0x33 | 4 | 0 | 0 | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 0 | 0 | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 0 | 0 | 0 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41<br>0x40 | 1:0<br>7:0 | 00<br>0x6E | 00<br>0x6E | 00<br>0x6E | HBit to HSYNC Delay | 2 | 196 |
| FIELD2_OFST | 0x43<br>0x42 | 3:0<br>7:0 | | | | Odd Field Offset | 2 | 196 |
| HWIDTH | 0x45<br>0x44 | 1:0<br>7:0 | 0x0<br>0x28 | 0x0<br>0x28 | 0x0<br>0x28 | HSYNC Pulse Width | 2 | 196 |
| VBIT_2_VSYNC | 0x46 | 5:0 | 0x05 | 0x05 | 0x05 | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | 0x05 | 0x05 | 0x05 | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 00 | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 1 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 1 | 1 | 1 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 0 | 0 | 0 | | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Up sampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | only | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | A | B | C | | | 232 |

**The video reconstructed by the HDMI Transmitter may not be compliant with CEA-861B timings, depending on the sync timings received from the Source device. See page 242 for register settings that enable both the sync decoder and the DE generator.**

**Refer to page 228 for additional notes.**

### 720P YCBCR 4:2:2 EMBEDDED SYNC INPUT – FIXING FOR CEA-861B

| Input Mode | | | YCbCr 4:2:2 Embedded Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33 | 3:0 | 0x1 | 0x1 | 0x1 | DE Delay | 1,2,3 | 193 |
| | 0x32 | 7:0 | 0x04 | 0x04 | 0x04 | | | |
| DE_TOP | 0x34 | 6:0 | 0x19 | 0x19 | 0x19 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37 | 3:0 | 0x5 | 0x5 | 0x5 | DE Count | 1,2 | 193 |
| | 0x36 | 7:0 | 0x00 | 0x00 | 0x00 | | | |
| DE_LIN | 0x39 | 2:0 | 0x2 | 0x2 | 0x2 | DE Lines. | 1,2 | 194 |
| | 0x38 | 7:0 | 0xD0 | 0xD0 | 0xD0 | | | |
| HS_POL# | 0x33 | 4 | 0 | 0 | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 0 | 0 | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 1 | 1 | 1 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41 | 1:0 | 00 | 00 | 00 | HBit to HSYNC Delay | 2, 6 | 196 |
| | 0x40 | 7:0 | 0x6E | 0x6E | 0x6E | | | |
| FIELD2_OFST | 0x43 | 3:0 | | | | Odd Field Offset | 2, 6 | 196 |
| | 0x42 | 7:0 | | | | | | |
| HWIDTH | 0x45 | 1:0 | 0x0 | 0x0 | 0x0 | HSYNC Pulse Width | 2 | 196 |
| | 0x44 | 7:0 | 0x28 | 0x28 | 0x28 | | | |
| VBIT_2_VSYNC | 0x46 | 5:0 | 0x05 | 0x05 | 0x05 | VBit to VSYNC Delay | 2, 6 | 196 |
| VWIDTH | 0x47 | 5:0 | 0x05 | 0x05 | 0x05 | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 00 | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 1 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 1 | 1 | 1 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 0 | 0 | 0 | | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Up sampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | only | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | A | B | C | | | 232 |

**The video reconstructed by the HDMI Transmitter may not be compliant with CEA-861B timings, depending on the sync timings received from the Source device. This table shows programming of both the sync decoder and the DE generator to correct porch timings to be CEA-861B compliant. See page 241 for register settings that enable only the sync decoder when the Source device provides compliant SAV/EAV timings.**

**Refer to page 228 for additional notes.**

# 1080i Input

### 1080I RGB INPUT

| Input Mode | | | RGB | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33 | 3:0 | | | 0x0 | DE Delay | 1,2,3 | 193 |
| | 0x32 | 7:0 | | | 0xC0 | | | |
| DE_TOP | 0x34 | 6:0 | | | 0x14 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37 | 3:0 | | | 0x7 | DE Count | 1,2 | 193 |
| | 0x36 | 7:0 | | | 0x80 | | | |
| DE_LIN | 0x39 | 2:0 | | | 0x2 | DE Lines | 1,2 | 194 |
| | 0x38 | 7:0 | | | 0x1C | | | |
| HS_POL# | 0x33 | 4 | | | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | | | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | | | 1 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41 | 1:0 | | | | HBit to HSYNC Delay | 2 | 196 |
| | 0x40 | 7:0 | | | | | | |
| FIELD2_OFST | 0x43 | 3:0 | | | | Odd Field Offset | 2 | 196 |
| | 0x42 | 7:0 | | | | | | |
| HWIDTH | 0x45 | 1:0 | | | | HSYNC Pulse Width | 2 | 196 |
| | 0x44 | 7:0 | | | | Fujitsu for | | |
| VBIT_2_VSYNC | 0x46 | 5:0 | | | | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | | | | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | | | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | | | 0 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | | | 0 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | | | 0 | | | 198 |
| UPSMP | 0x4A | 2 | | | 0 | Up sampling | | 198 |
| CSC | 0x4A | 3 | | | 0 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | | | 0 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | | | G | | | 232 |

**Refer to page 228 for notes.**

### 1080I YCBCR 4:4:4 SEPARATE SYNC INPUT

| Input Mode | | | YCbCr 4:4:4 Separate Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33 | 3:0 | | 0x0 | 0x0 | DE Delay | 1,2,3 | 193 |
| | 0x32 | 7:0 | | 0xC0 | 0xC0 | | | |
| DE_TOP | 0x34 | 6:0 | | 0x14 | 0x14 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37 | 3:0 | | 0x7 | 0x7 | DE Count | 1,2 | 193 |
| | 0x36 | 7:0 | | 0x80 | 0x80 | | | |
| DE_LIN | 0x39 | 2:0 | | 0x2 | 0x2 | DE Lines | 1,2 | 194 |
| | 0x38 | 7:0 | | 0x1C | 0x1C | | | |
| HS_POL# | 0x33 | 4 | | 0 | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | | 0 | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | | 1 | 1 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41 | 1:0 | | | | HBit to HSYNC Delay | 2 | 196 |
| | 0x40 | 7:0 | | | | | | |
| FIELD2_OFST | 0x43 | 3:0 | | | | Odd Field Offset | 2 | 196 |
| | 0x42 | 7:0 | | | | | | |
| HWIDTH | 0x45 | 1:0 | | | | HSYNC Pulse Width | 2 | 196 |
| | 0x44 | 7:0 | | | | | | |
| VBIT_2_VSYNC | 0x46 | 5:0 | | | | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | | | | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 1 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | | 0 | 0 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | | 0 | 0 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | | 0 | 0 | | | 198 |
| UPSMP | 0x4A | 2 | | 0 | 0 | Up sampling | | 198 |
| CSC | 0x4A | 3 | | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | only | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | | 0 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | D | E | F | | | 232 |

**Refer to page 228 for notes.**

### 1080I YCBCR 4:2:2 SEPARATE SYNC INPUT

| Input Mode | | | YCbCr 4:2:2 Separate Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33 | 3:0 | 0x0 | 0x0 | 0x0 | DE Delay | 1,2,3 | 193 |
| | 0x32 | 7:0 | 0xC0 | 0xC0 | 0xC0 | | | |
| DE_TOP | 0x34 | 6:0 | 0x14 | 0x14 | 0x14 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37 | 3:0 | 0x7 | 0x7 | 0x7 | DE Count | 1,2 | 193 |
| | 0x36 | 7:0 | 0x80 | 0x80 | 0x80 | | | |
| DE_LIN | 0x39 | 2:0 | 0x2 | 0x2 | 0x2 | DE Lines | 1,2 | 194 |
| | 0x38 | 7:0 | 0x1C | 0x1C | 0x1C | | | |
| HS_POL# | 0x33 | 4 | 0 | 0 | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 0 | 0 | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 1 | 1 | 1 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41 | 1:0 | | | | HBit to HSYNC Delay | 2 | 196 |
| | 0x40 | 7:0 | | | | | | |
| FIELD2_OFST | 0x43 | 3:0 | | | | Odd Field Offset | 2 | 196 |
| | 0x42 | 7:0 | | | | | | |
| HWIDTH | 0x45 | 1:0 | | | | HSYNC Pulse Width | 2 | 196 |
| | 0x44 | 7:0 | | | | | | |
| VBIT_2_VSYNC | 0x46 | 5:0 | | | | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | | | | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 00 | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 1 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 0 | 0 | 0 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 0 | 0 | 0 | | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Up sampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | only | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | D | E | F | | | 232 |

**Refer to page 228 for notes.**

### 1080I YCBCR 4:2:2 EMBEDDED SYNC INPUT

| Input Mode | | | YCbCr 4:2:2 Embedded Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33<br>0x32 | 3:0<br>7:0 | | | | DE Delay | 1,2,3 | 193 |
| DE_TOP | 0x34 | 6:0 | | | | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37<br>0x36 | 3:0<br>7:0 | | | | DE Count | 1,2 | 193 |
| DE_LIN | 0x39<br>0x38 | 2:0<br>7:0 | | | | DE Lines Confidential. | 1,2 | 194 |
| HS_POL# | 0x33 | 4 | 0 | 0 | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 0 | 0 | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 0 | 0 | 0 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41<br>0x40 | 1:0<br>7:0 | 00<br>0x58 | 00<br>0x58 | 00<br>0x58 | HBit to HSYNC Delay<br>Provided to | 2 | 196 |
| FIELD2_OFST | 0x43<br>0x42 | 3:0<br>7:0 | 0x4<br>0x4C | 0x4<br>0x4C | 0x4<br>0x4C | Odd Field Offset | 2 | 196 |
| HWIDTH | 0x45<br>0x44 | 1:0<br>7:0 | 0x0<br>0x2C | 0x0<br>0x2C | 0x0<br>0x2C | HSYNC Pulse Width | 2 | 196 |
| VBIT_2_VSYNC | 0x46 | 5:0 | 0x02 | 0x02 | 0x02 | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | 0x05 | 0x05 | 0x05 | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 00 | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 1 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 1 | 1 | 1 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 0 | 0 | 0 | | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Up sampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | only | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | A | B | C | | | 232 |

**The video reconstructed by the HDMI Transmitter may not be compliant with CEA-861B timings, depending on the sync timings received from the Source device. See page 247 for register settings that enable both the sync decoder and the DE generator.**

**Refer to page 228 for additional notes.**

### 1080I YCBCR 4:2:2 EMBEDDED SYNC INPUT – FIXING FOR CEA-861B

| Input Mode | | | YCbCr 4:2:2 Embedded Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33<br>0x32 | 3:0<br>7:0 | 0x0<br>0xC0 | 0x0<br>0xC0 | 0x0<br>0xC0 | DE Delay | 1,2,3 | 193 |
| DE_TOP | 0x34 | 6:0 | 0x14 | 0x14 | 0x14 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37<br>0x36 | 3:0<br>7:0 | 0x7<br>0x80 | 0x7<br>0x80 | 0x7<br>0x80 | DE Count | 1,2 | 193 |
| DE_LIN | 0x39<br>0x38 | 2:0<br>7:0 | 0x2<br>0x1C | 0x2<br>0x1C | 0x2<br>0x1C | DE Lines | 1,2 | 194 |
| HS_POL# | 0x33 | 4 | 0 | 0 | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 0 | 0 | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 1 | 1 | 1 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41<br>0x40 | 1:0<br>7:0 | 00<br>0x58 | 00<br>0x58 | 00<br>0x58 | HBit to HSYNC Delay | 2, 6 | 196 |
| FIELD2_OFST | 0x43<br>0x42 | 3:0<br>7:0 | 0x4<br>0x4C | 0x4<br>0x4C | 0x4<br>0x4C | Odd Field Offset | 2, 6 | 196 |
| HWIDTH | 0x45<br>0x44 | 1:0<br>7:0 | 0x0<br>0x2C | 0x0<br>0x2C | 0x0<br>0x2C | HSYNC Pulse Width | 2 | 196 |
| VBIT_2_VSYNC | 0x46 | 5:0 | 0x02 | 0x02 | 0x02 | VBit to VSYNC Delay | 2, 6 | 196 |
| VWIDTH | 0x47 | 5:0 | 0x05 | 0x05 | 0x05 | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 00 | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 1 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 1 | 1 | 1 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 0 | 0 | 0 | | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Up sampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | only | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | A | B | C | | | 232 |

**The video reconstructed by the HDMI Transmitter may not be compliant with CEA-861B timings, depending on the sync timings received from the Source device. This table shows programming of both the sync decoder and the DE generator to correct porch timings to be CEA-861B compliant. See page 246 for register settings that enable only the sync decoder when the Source device provides compliant SAV/EAV timings.**

**Refer to page 228 for additional notes.**

### 1080P YCBCR 4:4:4 SEPARATE SYNC INPUT

| Input Mode | | | YCbCr 4:4:4 Separate Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33 | 3:0 | | 0x0 | 0x0 | DE Delay | 1,2,3 | 193 |
| | 0x32 | 7:0 | | 0xC0 | 0xC0 | | | |
| DE_TOP | 0x34 | 6:0 | | 0x29 | 0x29 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37 | 3:0 | | 0x7 | 0x7 | DE Count | 1,2 | 193 |
| | 0x36 | 7:0 | | 0x80 | 0x80 | | | |
| DE_LIN | 0x39 | 2:0 | | 0x4 | 0x4 | DE Lines | 1,2 | 194 |
| | 0x38 | 7:0 | | 0x38 | 0x38 | | | |
| HS_POL# | 0x33 | 4 | | 0 | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | | 0 | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | | 1 | 1 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41 | 1:0 | | | | HBit to HSYNC Delay | 2 | 196 |
| | 0x40 | 7:0 | | | | | | |
| FIELD2_OFST | 0x43 | 3:0 | | | | Odd Field Offset | 2 | 196 |
| | 0x42 | 7:0 | | | | | | |
| HWIDTH | 0x45 | 1:0 | | | | HSYNC Pulse Width | 2 | 196 |
| | 0x44 | 7:0 | | | | | | |
| VBIT_2_VSYNC | 0x46 | 5:0 | | | | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | | | | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 1 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | | 0 | 0 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | | 0 | 0 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | | 0 | 0 | | | 198 |
| UPSMP | 0x4A | 2 | | 0 | 0 | Up sampling | | 198 |
| CSC | 0x4A | 3 | | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | | 0 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | D | E | F | | | 232 |

**Refer to page 228 for notes.**

### 1080P YCBCR 4:2:2 SEPARATE SYNC INPUT

| Input Mode | | | YCbCr 4:2:2 Separate Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33<br>0x32 | 3:0<br>7:0 | 0x0<br>0xC0 | 0x0<br>0xC0 | 0x0<br>0xC0 | DE Delay | 1,2,3 | 193 |
| DE_TOP | 0x34 | 6:0 | 0x29 | 0x29 | 0x29 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37<br>0x36 | 3:0<br>7:0 | 0x7<br>0x80 | 0x7<br>0x80 | 0x7<br>0x80 | DE Count | 1,2 | 193 |
| DE_LIN | 0x39<br>0x38 | 2:0<br>7:0 | 0x4<br>0x38 | 0x4<br>0x38 | 0x4<br>0x38 | DE Lines | 1,2 | 194 |
| HS_POL# | 0x33 | 4 | 0 | 0 | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 0 | 0 | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 1 | 1 | 1 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41<br>0x40 | 1:0<br>7:0 | | | | HBit to HSYNC Delay | 2 | 196 |
| FIELD2_OFST | 0x43<br>0x42 | 3:0<br>7:0 | | | | Odd Field Offset | 2 | 196 |
| HWIDTH | 0x45<br>0x44 | 1:0<br>7:0 | | | | HSYNC Pulse Width | 2 | 196 |
| VBIT_2_VSYNC | 0x46 | 5:0 | | | | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | | | | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 00 | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 1 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 0 | 0 | 0 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 0 | 0 | 0 | | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Up sampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | D | E | F | | | 232 |

**Refer to page 228 for notes.**

### 1080P YCBCR 4:2:2 EMBEDDED SYNC INPUT

| Input Mode | | | YCbCr 4:2:2 Embedded Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33<br>0x32 | 3:0<br>7:0 | | | | DE Delay | 1,2,3 | 193 |
| DE_TOP | 0x34 | 6:0 | | | | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37<br>0x36 | 3:0<br>7:0 | | | | DE Count | 1,2 | 193 |
| DE_LIN | 0x39<br>0x38 | 2:0<br>7:0 | | | | DE Lines Confidential. | 1,2 | 194 |
| HS_POL# | 0x33 | 4 | 0 | 0 | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 0 | 0 | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 0 | 0 | 0 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41<br>0x40 | 1:0<br>7:0 | 00<br>0x58 | 00<br>0x58 | 00<br>0x58 | HBit to HSYNC Delay<br>Provided to | 2 | 196 |
| FIELD2_OFST | 0x43<br>0x42 | 3:0<br>7:0 | | | | Odd Field Offset | 2 | 196 |
| HWIDTH | 0x45<br>0x44 | 1:0<br>7:0 | 0x0<br>0x2C | 0x0<br>0x2C | 0x0<br>0x2C | HSYNC Pulse Width | 2 | 196 |
| VBIT_2_VSYNC | 0x46 | 5:0 | 0x04 | 0x04 | 0x04 | VBit to VSYNC Delay | 2 | 196 |
| VWIDTH | 0x47 | 5:0 | 0x05 | 0x05 | 0x05 | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 00 | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 1 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 1 | 1 | 1 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 0 | 0 | 0 | | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Up sampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | A | B | C | | | 232 |

**The video reconstructed by the HDMI Transmitter may not be compliant with CEA-861B timings, depending on the sync timings received from the Source device. See page 251 for register settings that enable both the sync decoder and the DE generator.**

**Refer to page 228 for additional notes.**

### 1080P YCBCR 4:2:2 EMBEDDED SYNC INPUT – FIXING FOR CEA-861B

| Input Mode | | | YCbCr 4:2:2 Embedded Syncs | | | | | |
|---|---|---|---|---|---|---|---|---|
| Output Mode | | | YCbCr 4:2:2 | YCbCr 4:4:4 | RGB | | | |
| Register | | | | | | | Notes | Page |
| DE_DLY | 0x33 | 3:0 | 0x0 | 0x0 | 0x0 | DE Delay | 1,2,3 | 193 |
| | 0x32 | 7:0 | 0xC0 | 0xC0 | 0xC0 | | | |
| DE_TOP | 0x34 | 6:0 | 0x29 | 0x29 | 0x29 | DE Top | 1,2,3 | 193 |
| DE_CNT | 0x37 | 3:0 | 0x7 | 0x7 | 0x7 | DE Count | 1,2 | 193 |
| | 0x36 | 7:0 | 0x80 | 0x80 | 0x80 | | | |
| DE_LIN | 0x39 | 2:0 | 0x4 | 0x4 | 0x4 | DE Lines | 1,2 | 194 |
| | 0x38 | 7:0 | 0x38 | 0x38 | 0x38 | | | |
| HS_POL# | 0x33 | 4 | 0 | 0 | 0 | HSYNC Polarity | 2,3 | 193 |
| VS_POL# | 0x33 | 5 | 0 | 0 | 0 | VSYNC Polarity | 2,3 | 193 |
| DE_GEN | 0x33 | 6 | 1 | 1 | 1 | DE_GEN Enable | | 193 |
| HBIT_2_HSYNC | 0x41 | 1:0 | 00 | 00 | 00 | HBit to HSYNC Delay | 2, 6 | 196 |
| | 0x40 | 7:0 | 0x58 | 0x58 | 0x58 | | | |
| FIELD2_OFST | 0x43 | 3:0 | | | | Odd Field Offset | 2, 6 | 196 |
| | 0x42 | 7:0 | | | | | | |
| HWIDTH | 0x45 | 1:0 | 0x0 | 0x0 | 0x0 | HSYNC Pulse Width | 2 | 196 |
| | 0x44 | 7:0 | 0x2C | 0x2C | 0x2C | | | |
| VBIT_2_VSYNC | 0x46 | 5:0 | 0x04 | 0x04 | 0x04 | VBit to VSYNC Delay | 2, 6 | 196 |
| VWIDTH | 0x47 | 5:0 | 0x05 | 0x05 | 0x05 | VSYNC Pulse Width | 2 | 197 |
| ICLK | 0x48 | 1:0 | 00 | 00 | 00 | Pixel Replication | | 197 |
| CSCSEL | 0x48 | 4 | | | 1 | Color Space Select | | 197 |
| EXTN | 0x48 | 5 | 1 | 1 | 1 | Extended Bit Mode | 4 | 197 |
| SYNCEXT | 0x4A | 0 | 1 | 1 | 1 | Sync Extraction | | 198 |
| DEMUX | 0x4A | 1 | 0 | 0 | 0 | | | 198 |
| UPSMP | 0x4A | 2 | 0 | 1 | 1 | Up sampling | | 198 |
| CSC | 0x4A | 3 | 0 | 0 | 1 | Color Space Convert | | 198 |
| RANGE | 0x4A | 4 | | | 1 | Range Select | 7 | 198 |
| DITHER_MODE | 0x4A | 7:6 | | | | Bits per Output Video Channel | 9 | 198 |
| DITHER | 0x4A | 5 | 0 | 1 | 1 | Dither Enable | | 198 |
| Data Flow Diagram (Figure 17) | | | A | B | C | | | 232 |

**The video reconstructed by the HDMI Transmitter may not be compliant with CEA-861B timings, depending on the sync timings received from the Source device. This table shows programming of both the sync decoder and the DE generator to correct porch timings to be CEA-861B compliant. See page 250 for register settings that enable only the sync decoder when the Source device provides compliant SAV/EAV timings.**

**Refer to page 228 for additional notes.**

# Handling InforFrames

### AVI INFORFRAMES

**The following descriptions are adapted from Section 6.1 of the *EIA/CEA-861B Specification* and from Section 8.2.1 of the *HDMI 1.0 Specification*. AVI InfoFrame packets are enabled by AVI_EN and AVI_RPT in the Packet Buffer Control #1 Register (0x7A:0x3E), described on page 217.**

| | | | Bit Number | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Reg. | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Notes |
| 0x40 | 0 | Type Code | 0x82 | | | | | | | | 1 |
| 0x41 | 1 | Version | 0x02 | | | | | | | | 2 |
| 0x42 | 2 | Length | 0x0D | | | | | | | | 3 |
| 0x43 | 3 | Checksum | | | | | | | | | 4 |
| 0x44 | 4 | Data Byte 1 | 0 | Y1 | Y0 | A0 | B1 | B0 | S1 | S0 | |
| 0x45 | 5 | Data Byte 2 | C1 | C0 | M1 | M0 | R3 | R2 | R1 | R0 | |
| 0x46 | 6 | Data Byte 3 | 0 | 0 | 0 | 0 | 0 | 0 | SC1 | SC0 | |
| 0x47 | 7 | Data Byte 4 | 0 | VIC6 | VIC5 | VIC4 | VIC3 | VIC2 | VIC1 | VIC0 | |
| 0x48 | 8 | Data Byte 5 | 0 | 0 | 0 | 0 | PR3 | PR2 | PR1 | PR0 | |
| 0x49 | 9 | Data Byte 6 | Bar Info for Letterbox and Curtained Transmissions | | | | | | | | 5,6 |
| 0x4A | 10 | Data Byte 7 | | | | | | | | | |
| 0x4B | 11 | Data Byte 8 | | | | | | | | | |
| 0x4C | 12 | Data Byte 9 | | | | | | | | | |
| 0x4D | 13 | Data Byte 10 | | | | | | | | | |
| 0x4E | 14 | Data Byte 11 | | | | | | | | | |
| 0x4F | 15 | Data Byte 12 | | | | | | | | | |
| 0x50 | 16 | Data Byte 13 | | | | | | | | | |
| 0x51 | 17 | Data Byte 14 | | | | | | | | | 7 |
| 0x52 | 18 | Data Byte 15 | | | | | | | | | |

**Notes:**

**Although CEA-861B defines TYPE as 0x02, the HDMI Specification defines TYPE as 0x82 with bit 7 set.**

**VERSION 0x02 defines this as a CEA-861B AVI InfoFrame.**

**LENGTH should be set to 13 (0x0D). Additional data bytes are ignored. See the HDMI Specification, Section 8.2.1.**

**CHECKSUM is calculated so that the modulo-256 sum of {TYPE + VERSION + LENGTH + CHECKSUM + Data Bytes 1 to LENGTH} is zero. See the HDMI Specification, Section 5.3.5.**

**Shaded bits are RESERVED and are not to be set to other values.**

**See CEA-861B, Sections 6.1.2-6.1.3 for more details on the labeled fields in Data Bytes 1 through 13. The fields are summarized in Table 22, below.**

1. CEA-861B defines only 13 data bytes. HDMI defines 27 data bytes. The HDMI Transmitter has 15 data byte registers. Load data byte 14 and 15 with 0x00.

TABLE 22 AVI INFORFRAME FIELD SETTINGS

| Field | | | | Definition |
|---|---|---|---|---|
| Y1 | | Y0 | | Color Space |
| 0 | | 0 | | RGB |
| 0 | | 1 | | YCbCr 4:2:2 |
| 1 | | 0 | | YCbCr 4:4:4 |
| 1 | | 1 | | Future |
| | | A0 | | Active Format Information Present |
| | | 0 | | No format data present. |
| | | 1 | | Active format identification data is present in the AVI InfoFrame. |
| B1 | | B0 | | Bar Info |
| 0 | | 0 | | Bar data not valid. |
| 0 | | 1 | | Vertical bar info valid. |
| 1 | | 0 | | Horizontal bar info valid. |
| 1 | | 1 | | Both vertical and horizontal bar info valid. |
| S1 | | S0 | | Scan Information |
| 0 | | 0 | | No data. |
| 0 | | 1 | | Over-scanned (television). Provided to |
| 1 | | 0 | | Under-scanned (computer). |
| 1 | | 1 | | Future |
| SC1 | | SC0 | | Non-Uniform Picture Scaling |
| 0 | | 0 | | No known non-uniform scaling. |
| 0 | | 1 | | Picture has been scaled horizontally. |
| 1 | | 0 | | Picture has been scaled vertically. |
| 1 | | 1 | | Picture has been scaled both horizontally and vertically. |
| C1 | | C0 | | Colorimetry |
| 0 | | 0 | | No data. |
| 0 | | 1 | | SMTPE 170M and ITU 601 (for standard definition TV) |
| 1 | | 0 | | ITU 709 (for advanced and high definition TV) |
| 1 | | 1 | | Future |
| M1 | | M0 | | Picture Aspect Ratio |
| 0 | | 0 | | No data. |
| 0 | | 1 | | 4:3 |
| 1 | | 0 | | 16:9 |
| 1 | | 1 | | Future |
| R3 | R2 | R1 | R0 | Active Format Aspect Ratio |
| 1 | 0 | 0 | 0 | Same as Picture Aspect Ratio (M1:M0 field). |
| 1 | 0 | 0 | 1 | 4:3 |
| 1 | 0 | 1 | 0 | 16:9 |
| 1 | 0 | 1 | 1 | 14:9 |
| other values | | | | Per DVB AFD active_format field. |

### GENERAL CONTROL PACKETS

General Control packets control the flow of video and audio data across the HDMI link. The Control Packet is defined in Section 5.3.6 of the HDMI Specification. This data byte is controlled in the HDMI Transmitter with the CP_BYTE1 register (0x7A:0xDF), described on page 221. The Control Packet is transmitted only during the vertical blanking period, after the active edge of VSYNC.

To change the content of the CP_BYTE1 register, CP_EN must be zero. The firmware should keep CP_EN cleared until the desired value is written into SETAVM and CLRAVM. Then CP_EN and CP_RPT should be set to 1. (A more extensive explanation for Enable and Repeat bits is found on page 216.) The SETAVM bit and CLRAVM bit cannot both be set at the same time (see page 221). Because the Control Packet is synchronized to VSYNC, the action from SETAVM and CLRAVM takes effect immediately after the next VSYNC pulse.

| SETAVM | CLRAVM | Action |
|---|---|---|
| 0 | 0 | Default setting after RESET#. Same action as SETAVM=0 with CLRAVM=1. |
| 0 | 1 | Allow pixel data and audio sample data to pass across the link. No muting. |
| 1 | 0 | Mute video and audio with 0x00 for all pixel data and send 0x00 for all audio packet data. |
| 1 | 1 | *NOT ALLOWED BY HDMI SPECIFICATION.* |

When transmitting in DVI mode, setting SETAVM to 1 changes the video content to 0x00. CTL3 pulses stop at the next frame, so the HDMI Receiver stops decrypting and outputs the 0x00 data as a blank screen.

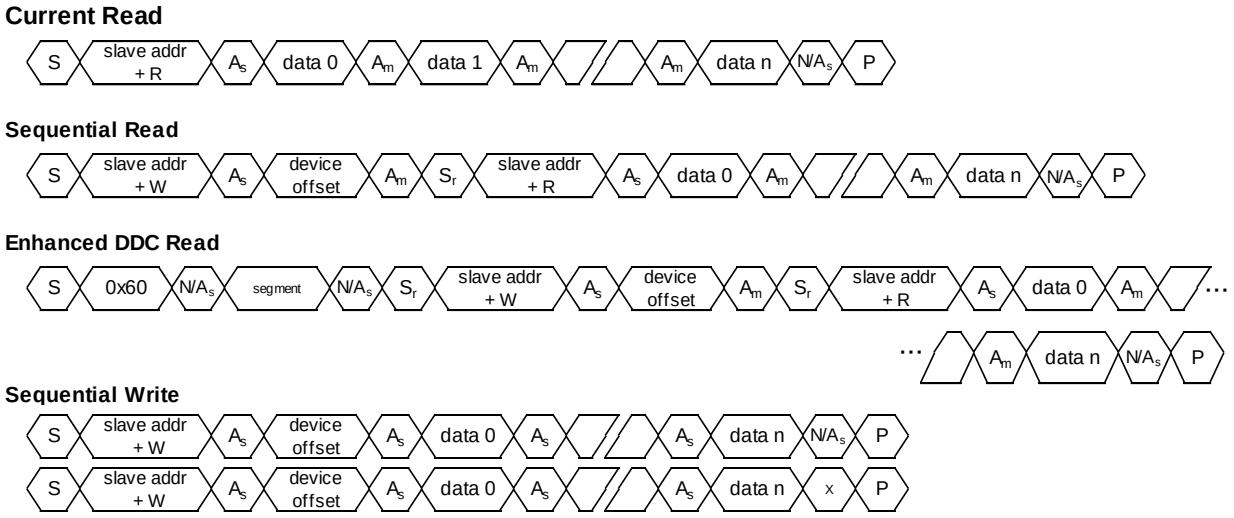
When a Link Integrity check fails in DVI mode, the HDMI Transmitter should stop the video signaling briefly before beginning a new authentication.

### HANDLING AUDIO CONTENT PROTECTION (ACP) PACKETS

HDMI 1.1 defines new packets for handling content protection for audio. Refer to the explanations beginning on page 226.

## Operating DDC Master

**The HDMI Transmitter includes a logic block to drive the E-DDC bus, which supports a variety of I²C commands. The individual registers are described beginning on page 205. The speed of the I²C clock is determined by an internal oscillator in the HDMI Transmitter and is not dependent on, or a function of, any input pixel clock. The I²C frequency does not exceed 100 kHz.**




S = start
$S_r$ = restart
$A_s$ = slave acknowledge
$A_m$ = master acknowledge
N = no ack
P = stop

FIGURE 18. SUPPORTED MASTER I2C TRANSACTIONS

**Current Address Read – Reads from the last offset address so that no offset needs to be written to the slave. Multiple bytes may be read from consecutive addresses, which auto-increment in the slave with each ACK from the master.**

**Sequential Read – Reads from a specific start address, which is sent as a write command to the slave. Multiple bytes may be read from consecutive addresses. Although the FIFO in the HDMI Transmitter can hold only 16 bytes, the sequential read command may be of any length up to the 10-bit value in the DDC_COUNT register (0x72:0xF0-0xF1). A *Stop* bit is sent by the DDC master only when the entire DDC_COUNT has been completed.**

**Enhanced DDC Read – A special command defined by the VESA E-DDC Specification, which writes a segment address to a separate I²C device address, then sends an offset address to the slave device, and finally reads one or more data bytes beginning from address 128*segment+offset. Multiple bytes within the same segment may be read, as the slave auto-increments the offset with each ACK from the master. The segment register in the slave is reset at the end of each command. A NACK or ACK is required from the slave device if the segment is not zero, but is ignored if the segment is zero. Refer to the E-DDC Specification.**

**Sequential Write – Similar to the sequential read, this command sends one or more bytes to the slave, beginning at the explicit offset address. Multiple bytes may be written, as the slave auto-increments the address until the master sends a stop bit. Two command op codes are available to either wait for ACK/NACK on last byte or to ignore the ACK/NACK on last byte.**

### DEVICE ADDRESSES

**The standardized device addresses for the E-DDC bus, as specified in the E-DDC standards, are listed in Table 23.**

TABLE 23 DDC STANDARD DEVICE ADDRESSES

| Device | Address |
|---|---|
| EDID PROM | 0xA0 |
| E-DDC Segment Address | 0x60 |
| Reserved | 0x74 |

**DDC READ OPERATION**

**The read command is setup by the micro when it loads values for the device address, the offset address, and the byte count. If the read command is to be an extended-DDC read, the segment address must also be loaded. After loading these values, the command register is loaded with the read command code.**

**The HDMI Transmitter uses a FIFO to hold data read across the DDC bus. Up to 16 bytes may be held in the FIFO, but as many as 1023 bytes may be read in one master DDC operation by setting the overall count in DDC_COUNT. The IN_PROG bit (0x72:0xF2[4]) is cleared when the last byte has been read. Figure 19 shows how to do a *short read* of 16 bytes or less. All *N* bytes can be read from the finished FIFO with one local I²C read command.**

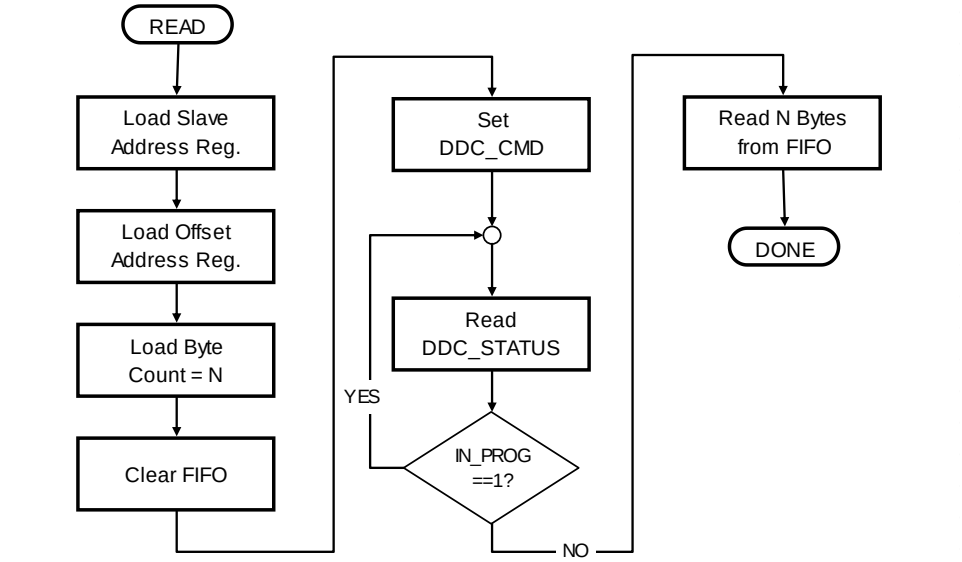



FIGURE 19. MASTER I2C READ COMMAND FLOWCHART #1

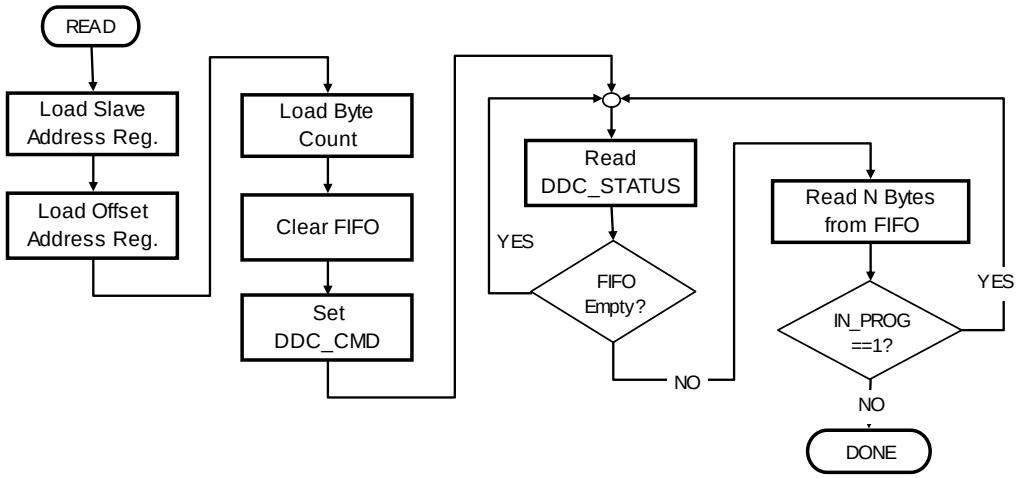
**There are at least three methods for performing *long reads*. Choose the method suited to the RAM available in the microcontroller.**

**Figure 20 shows three methods for reading byte blocks through the FIFO. As soon as a data byte is read from the FIFO to the micro, the HDMI Transmitter resumes its read operation across the DDC bus and puts the next data byte from the DDC slave device into the FIFO.**

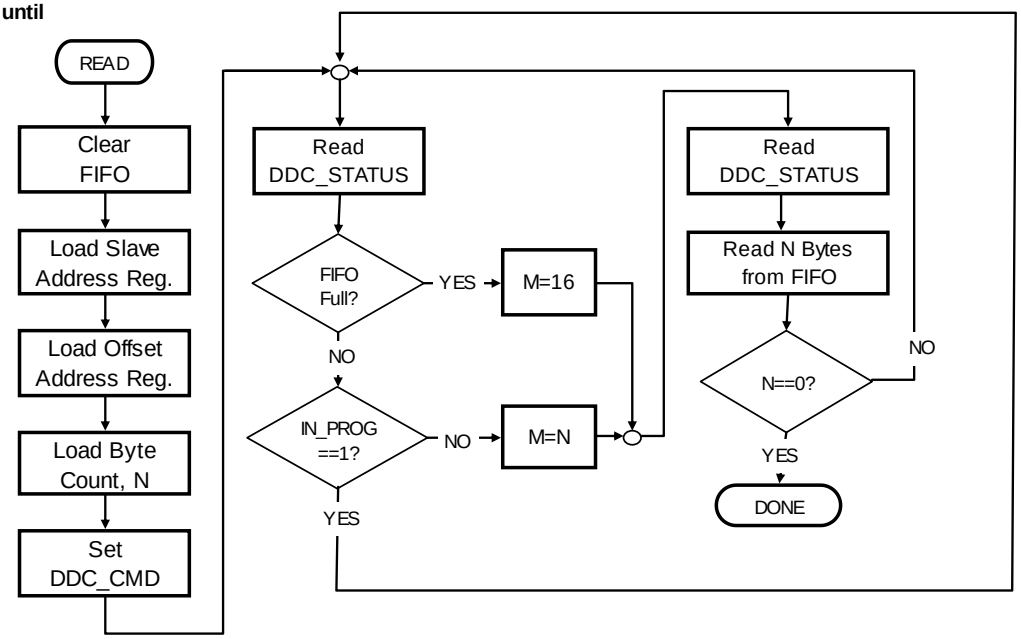
| | Advantages | Disadvantages |
|---|---|---|
| Method #1 | Requires least amount of RAM in micro. Shortest flowchart. | Slow execution when f{DDC} is faster than f{local I²C}, from high overhead of re-reading status and transferring one byte at a time. |
| Method #2 | More efficient than Method #1 when f{DDC} is slower than f{local I²C} by minimizing reads on local I²C bus. | Requires most amount of RAM in micro. Flowchart complicated by reading remainder. |
| Method #3 | Shorter flowchart than Method #2. Balances local I²C and DDC operations. Optimizes read commands on local I²C. | Requires more RAM than Method #2. No more efficient than Method #1 when f{local I²C} is much faster than f{DDC}. |

**An abort command is used if the master device needs to stop the read command.**

**Read Single Bytes until Command Complete**




**Read 16-Byte Blocks until Command Complete**




**Read FIFO Content until Command Complete**

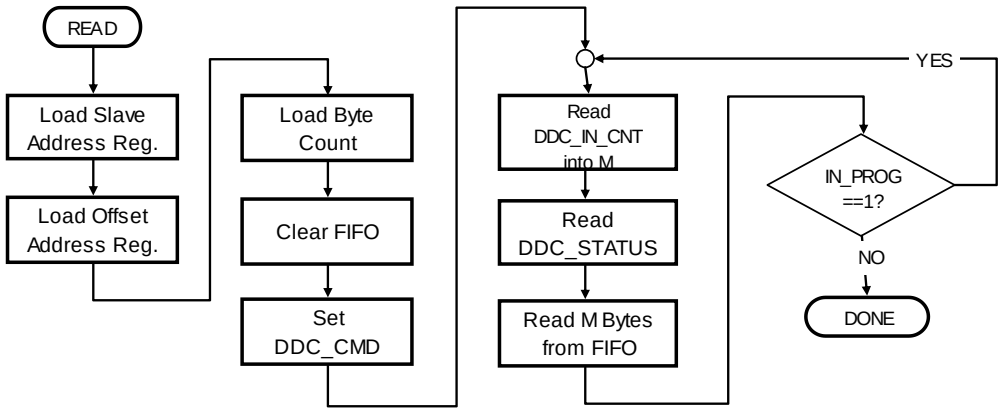



**FIGURE 20. MASTER I2C READ COMMAND FLOWCHART #2**

### WRITE COMMAND

Write commands are similar to read commands. The FIFO must be written to by the micro before initiating the command, after resetting the FIFO to empty as described above. If fewer than 17 bytes are to be written to the DDC channel, the micro must only load the bytes to the FIFO and then write the DDC_CMD register with the write command code. If more than 16 bytes are to be written to the DDC channel, the micro must fill the FIFO, wait for it to begin sending (and not be full), then put the remaining bytes into the FIFO without overflowing it. *Since HDMI does not require a write of more than 16 bytes, such an operation is not described in this Programmer's Reference.*

Each master I²C operation begins with writes to several registers in the HDMI Transmitter. For a write command, the destination device address is followed with the offset address and the byte count. The micro must also clear the FIFO to an empty state before writing data to it. When the byte count is completed, the HDMI Transmitter automatically sends the stop bit onto the DDC bus. Figure 21 shows how to write up to 16 bytes from the micro to the DDC bus. All the data fits into the FIFO and so a multi-byte I²C write can be used from the micro to the HDMI Transmitter. The WRITE command is then issued to the DDC_CMD register and all data bytes are transferred across the DDC bus. Some type of timeout should be used when checking that the Master DDC module is available, either before starting a new command or before returning from the subroutine after beginning a command.

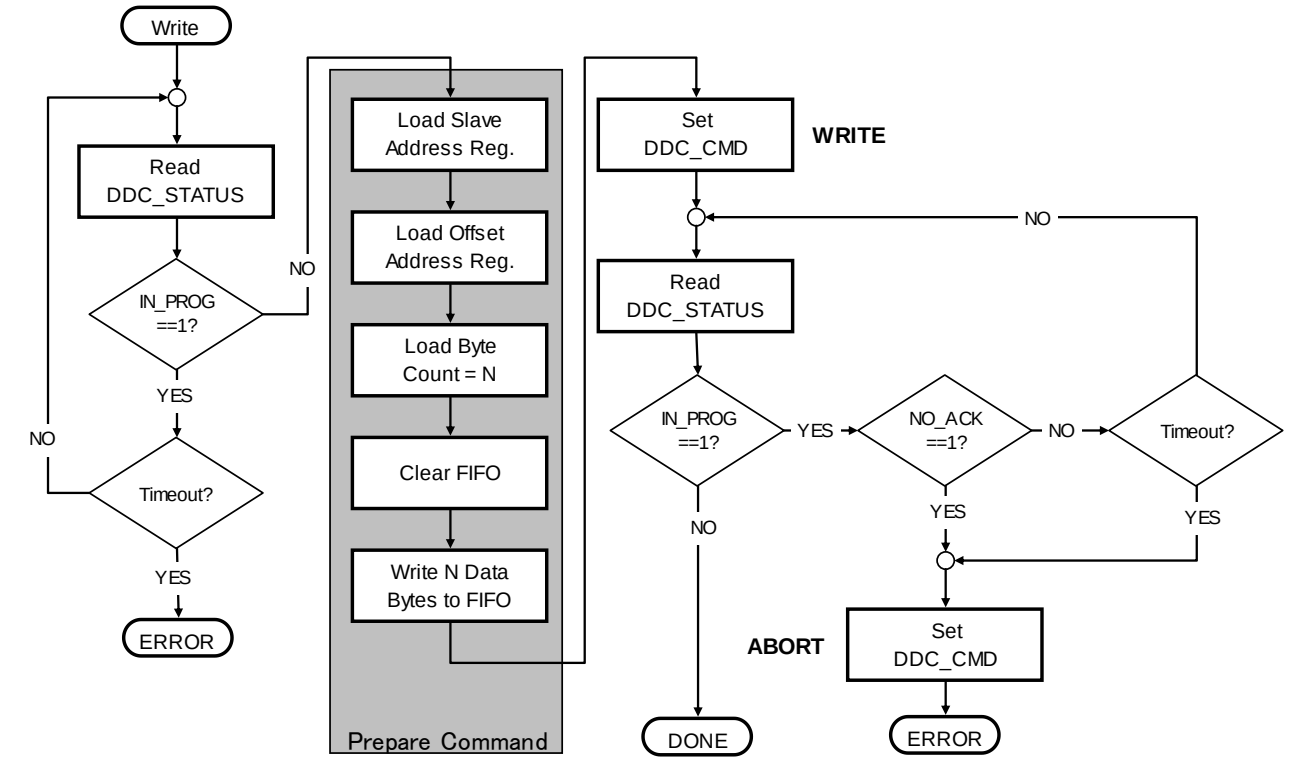



FIGURE 21. MASTER I²C WRITE COMMAND FLOWCHART #1

Figure 21 includes a check before beginning the write command to be certain that the Master DDC block is not already in use; and a check after triggering the write command to check for bus hangs. An abort command is used if the write command times out or fails to receive an ACK from the slave device.

### ABORT COMMAND

A read or write command can be interrupted by the micro at any time by writing an Abort command to the DDC_CMD register (0x72:0xF3). The abort command issues a STOP condition, followed by nine SCK clocks, but only if a previous command is incomplete. Issuing the abort command when no other command is in progress causes no action on the DDC link. Firmware should allow for DDC bus hangs and include a timeout in the loop that waits for the completion of a write or read command. This can be done by setting up a watchdog timer with an interrupt service routine and aborting the DDC command if the timer expires. It can also be done by polling the DDC_STATUS byte and deciding that the command is stalled, then issuing an abort command.

# Power Down Control

### LOGIC BLOCKS AFFECTED BY POWER DOWN

**The HDMI Transmitter provides various register bits to control power down of various sections of the chip. PD# is defined on page 189. PDIDCK#, PDOSC#, and PDTOT# are defined on page 215.**

TABLE 24 POWER DOWN CONTROL BIT EFFECTS

| PD# | PDIDCK# | PDOSC# | PDTOT# | Function | Note |
|---|---|---|---|---|---|
| X | X | X | 0 | Powers down everything. Equivalent to the combination of PD#=0 AND PDIDCK#=0 AND PDOSC#=0. | 1 |
| 0 | 1 | 1 | 1 | Powers down TMDS core and PLL. Digital logic is switching with an active IDCK. Registers are accessible via I²C with exceptions listed Table 25. | 2 |
| 1 | 0 | 1 | 1 | Powers down internal digital clock tree. | 3 |
| 1 | 1 | 0 | 1 | Powers down internal ring oscillator. Disables internal read of KSV. Disables Master DDC block. | 4,5 |

**Notes on Table 24:**

**This combination delivers the lowest power consumption if input signals are switching.**

**An attached HDMI Receiver sees no switching clock or data and should react by disabling that HDMI input port until switching is detected again.**

**A quiet internal clock tree significantly reduces power consumption.**

**Master DDC can be used when PD#=0 or PDIDCK#=0, such as for reading EDID when the attached HDMI Receiver is not powered on.**

# Registers Affected by Power Down

**The registers shown in Table 25 require PD#=1 AND PDIDCK#=1 AND PDOSC#=1 AND PDTOT#=1.**

TABLE 25 REGISTERS AFFECTED BY PD BITS

| Dev | Offset | Register |
|---|---|---|
| 0x7A | 0x3E | PB_CTRL1 |
| 0x7A | 0x3F | PB_CTRL2 |
| 0x72 | 0x0F | Reserved |
| 0x72 | 0x10 | BKSV1 |
| 0x72 | 0x11 | BKSV2 |
| 0x72 | 0x12 | BKSV3 |
| 0x72 | 0x13 | BKSV4 |
| 0x72 | 0x14 | BKSV5 |

| Dev | Offset | Register |
|---|---|---|
| 0x72 | 0x15 | AN1 |
| 0x72 | 0x16 | AN2 |
| 0x72 | 0x17 | AN3 |
| 0x72 | 0x18 | AN4 |
| 0x72 | 0x19 | AN5 |
| 0x72 | 0x1A | AN6 |
| 0x72 | 0x1B | AN7 |
| 0x72 | 0x1C | AN8 |

# OSG for Display Path

## Introduction

**TW2880 OSG controller supports triple bitmap windows with 16 bit-per-pixel mode. Each OSG display window can support 8 sub-windows. OSG display engine supports upscale function. The powerful OSG memory writing engine allows less firmware effort. 16 bit color uses RGB 565 format, and it does not need color look-up-table. The input graphics from CPU can be 2 or 16 bit per pixel to reduce the amount of data writing by the CPU. The OSG writing engine automatically extends 2 bit pixel to 16 bit pixel format before writing into OSG graphic buffer in SDRAM. This is used for objects only with two colors such as fonts. The powerful OSG memory writing engine supports CPU write, block transfer, and block fill mode. When writing to SDRAM, OSG engine also supports color conversion and selective color overwrite. Graphic data are save in the external SDRAM. The maximum size can be 8192x8192 pixels depends on SDRAM size.**

## Features

- Triple windows bitmap OSG
- Each OSG window has 8 sub-windows
- Window upscale
- Three write modes: written directly by host, block transfer and block fill by hardware
- Bitmap compression for reduced access bandwidth
- Color selective overwrite
- Bit extension from 2bit to 16 bit
- 4 Color conversion
- Blinking, transparent, alpha blending control when displaying on screen
- Hardware BitBlt unit
- 1K byte buffer for host write
- Host can write data using 8bit or 16bit
- Big endian and little endian mode

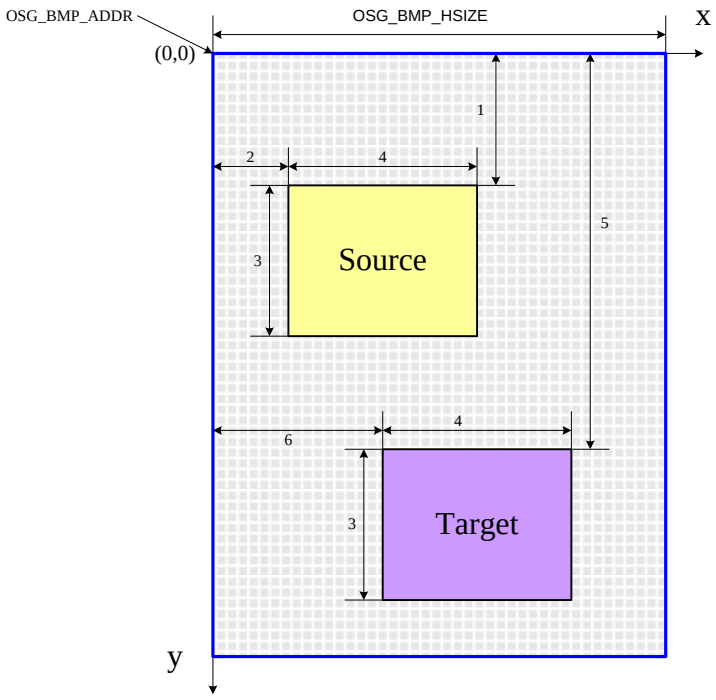
**For three write mode, the following table shows which features they can support.**

| MODE | DE-COMPRESS | LINEAR BUFFER | BIT EXTENSION | SELECTIVE OVERWRITE / BITBLT | COLOR CONVERSION |
|---|---|---|---|---|---|
| Host write | Y | Y | Y | Y | Y |
| Block transfer | N | Y | N | Y | Y |
| Block fill | N | N | N | N | N |

## OSG Graphic Buffer

### MEMORY DIAGRAM

**The following picture shows the OSG memory in SDRAM and some related registers.**



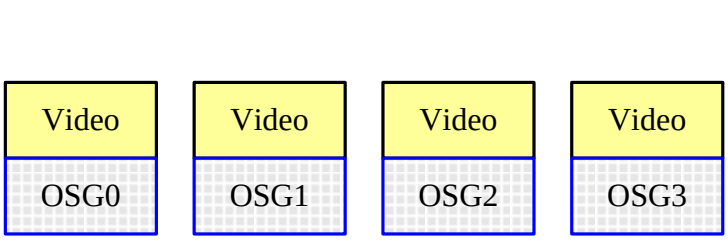

1. OSG_B1_VSTART
2. OSG_B1_HSTART
3. OSG_B1_VSIZE
4. OSG_B1_HSIZE
5. OSG_B2_VSTART
6. OSG_B2_HSTART

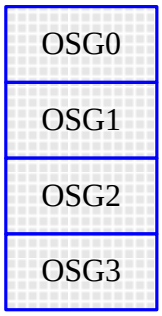
OSG Bitmap Buffer

**BITMAP BUFFER**

**TW2880 on screen display is a bitmapped based OSG controller. Each pixel in the OSG window corresponds to a pixel data in the bitmap buffer of 16 bit in external SDRAM memory. The bitmap buffer size supports up to a maximum size of 4096x4096 pixels. The horizontal pixel size is controlled by the OSG_BMP_HSIZE register. The bitmap buffer starting address in SDRAM is controlled by OSG_BMP_ADDR register.**

**Since video images are saved in upper side of SDRAM, Bitmap Buffer is not saved in continuous physical address in SDRAM. OSG_BMP_ADDR is the start address of each OSG bank in SDRAM. Firmware does not need to know the physical address of an object. Hardware will automatically be calculated physical address of bitmap buffer. Firmware can treat bitmap buffer is a continuous address space.**




OSG Bitmap Buffer

**WRITE MEMROY BY HOST**

With this method, CPU download original bitmap to bitmap buffer or linear buffer. To use this mode, first select mode by setting register OSG_WR_MODE to 2'b00 and other control register bits. If bitmap buffer is used, horizontal start address (OSG_B2_HSTART) and vertical start address (OSG_B2_VSTART) of target memory location need to be programmed. Both for bitmap buffer and linear buffer, horizontal size (OSG_B1_HSIZE) and vertical size (OSG_B1_VSIZE) need to be programmed. In this mode, data compression, color conversion, selective overwrite and pixel bit extension are all supported. After all registers are ready, host write OSG_WR_START command. After this command is triggered, host continuous write data to OSG_HOST_DATA. If internal memory is full, hardware will generate a wait signal to host. Host will stop running. Host can also read status register. The number of byte written need to be consistent with the setting.

**BLOCK TRANSFER**

CPU can move a rectangular area from one location to another location. Source can be in bitmap buffer or linear buffer. Target can only be bitmap buffer. It simply specifies the source / destination location and the object size, hardware performs the block transfer automatically. To use this mode, first select mode by setting register OSG_WR_MODE to 2'b01, and configure all the necessary registers such as source location and size and target location and size. After that, host write OSG_WR_START command. In this mode, compression and bit expanding are not supported, color conversion and selective overwrite are supported.

**BLOCK FILL**

In this mode, CPU specifies a command to fill a rectangular block with a single color and then the hardware automatically fill the specified area with a single color. This allows the CPU to easily fill the background color. To use this mode, first select the mode by setting register OSG_WR_MODE to 2'b10, and configures all the necessary registers such as destination register, then start block fill by setting register OSG_WR_START to 1. Data compression, color conversion, selective overwrite, pixel bit extension and linear buffer are not supported. The color is set by register OSG_FILL_COLOR.

**BITBLT**

TW2880 also supports BitBlt function. There are 256 operations in BitBlt. It is an extension for block transfer.

BitBlt function need three objects: source, destination and mask. The result is written to destination. It can be written as a function equation: destination = bitblit(mask, destination, source).

In this function, destination is one bit of destination pixel, source is the same bit of destination pixel, mask is same bit of destination pixel.

(mask, destination, source) generate one 3 bit data. For example if mask is 1, destination is 0 and source is 0. The (mask, destination, source) is 3'b100 which is 4. Then BitBlit bit 4 is the result. Three bit generate data from 0 to 7. Then BitBlt data bit 0 to 7 is the result because BitBlt operation code is 8 bit value.

For example, BitBlt operation code is 0xCA = 11001010b

We can get the look up table:

| MASK | DESTINATION | SOURCE | RESULT |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

You will find result is destination = mask ? destination : source.

**In our design, BitBlt function default operation is 0xCA.**

**If you want to generate another BitBlt function, you need to generate this look up table.**

**For example, if you want to get: destination = source & destination, look up table should like this:**

| MASK | DESTINATION | SOURCE | RESULT |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

**From the table, BitBlt operation code should be 0x80.**

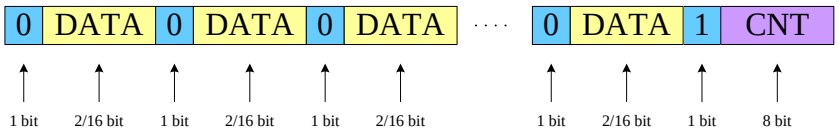
**In the equation, mask is determined by register OSG_OVWR_MODE (addr: 0x102 bit [2:1].**

| OSG_OVWR_MODE | MASK | FUNCTION |
|---|---|---|
| 00 | When source equals to selective overwrite color, mask is 1 | Selective overwrite determined by register |
| 01 | When source equals to 0xFFFF, mask is 1 | Selective overwrite determined by 0xFFFF |
| 10 | 0 | BitBlt with mask 0 |
| 11 | 0 | No BitBlt function, Destination equals to source |

### DE-COMPRESSION

**A compressed input bitmap can be used to reduce the bitmap size in CPU memory and also saves CPU access bandwidth. But this does not save SDRAM size. A hardware de-compression engine is integrated on chip to convert the compressed bitmap into an uncompressed bitmap. To enable de-compression, set OSG_RLC_EN to high. De-compression is only used in host write buffer mode.**

**Compressed data format is shown below:**




**The compressed bit stream is combined by "DATA" and "CNT". Before each data or count, there is a bit indicating data or count. "0" means data, "1" means count. "DATA" is uncompressed data. "CNT" is the count of the repeated "DATA". This is a run-length coding. "DATA" bit can be 2 or 16. It is set by OSG_PIXEL_BIT. "CNT" bit can be from 1 to 16.**

**For example, if data bit is 16 and count bit is 8. The uncompressed data stream is:**

0x0001, 0x0002, 0x0003, 0x0004, 0x0004, 0x0004, 0x0004, 0x0004, 0x0004.

Then compressed data string is:

0 0000000000000001 0 0000000000000010 0 0000000000000011 0 0000000000000100 1 00000101

The count number is 5 in this case. Count number is the repeated data number not include the first data.

If count number is 256 in 8-bit case, then count is set to “0”. If count number is more than 256, than put several count packets. For example, uncompressed data stream is:

0x0001, 0x0002, 0x0003, 0x0004, 0x0004, 0x0004, 0x0004, 0x0004, 0x0004, ..., 0x0004

In this case, total amount of 0x0004 is 400, then first count should be 0, second count should be 143. Or first count plus second count should be 399.

#### PIXEL BIT EXTENSION

Although the bitmap buffer is 16 bit per pixel, input pixel from CPU can be 2 bit or 16 bit. When 2 bit format is enabled, hardware will automatically convert 2 bit color to 16 bit color. Here “00” is converted to OSG_CON_TAR1, “01” is converted to OSG_CON_TAR2, “10” is converted to OSG_CON_TAR3, and “11” is converted to OSG_CON_TAR4,. Data written to SDRAM is always 16 bit. To enable this mode, set register OSG_PIXEL_BIT to high. This mode is used for simple color object like font. It will save a lot of CPU memory. It is only used in host write memory mode.

#### COLOR CONVERSION

Before data are written into SDRAM, color conversion can be enabled by setting bit OSG_COLCON_EN to high. This is used in direct host write and block transfer mode. Four color registers need to be programmed. Source color OSG_CON_SRC1 will be converted to target color OSG_CON_TAR1. Source color OSG_CON_SRC2 will be converted to target color OSG_CON_TAR2.

#### SELECTIVE OVERWRITE

When writing into the buffer through host write, block transfer. The pixels with color (OSG_OVWR_COLOR) can be specified as transparent pixels so that they will not be written into the buffer. There are four overwrite color can be used. For example, the font background will not be written and allows only the foreground pixels being written. To use this selective pixel writing, set register OSG_OVWR_MODE to 2’b00. This is one of the BitBlt operation.

#### PIXEL OPERATION SEQUENCE

The sequence of data operation is:

de-compression

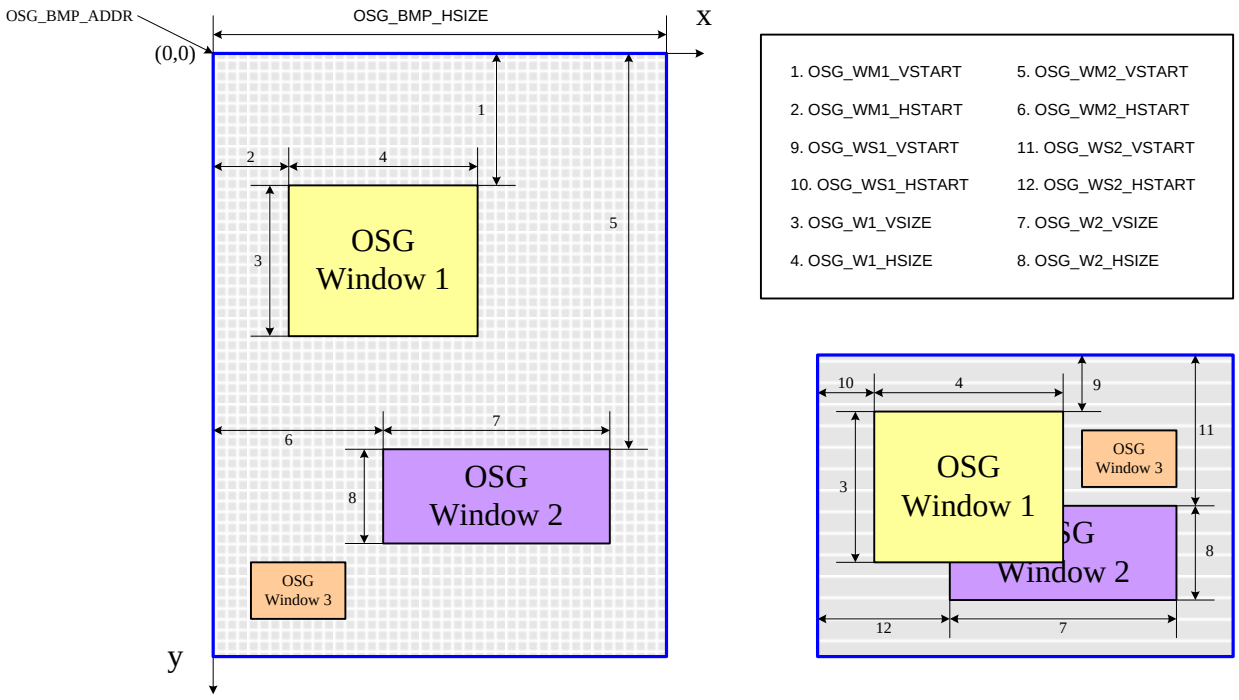
bit extension

color conversion

BitBlt (selective overwrite)

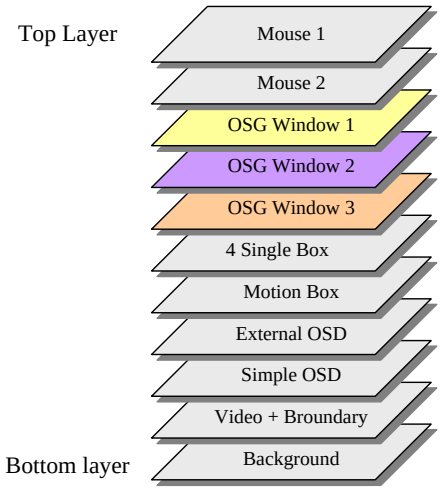
## Bitmap Buffer Display

TW2880 supports three OSG windows. The contents showed on screen are based on Bitmap Buffer. The following picture shows the bitmap buffer and related display registers.




OSG Bitmap Buffer LCD Screen

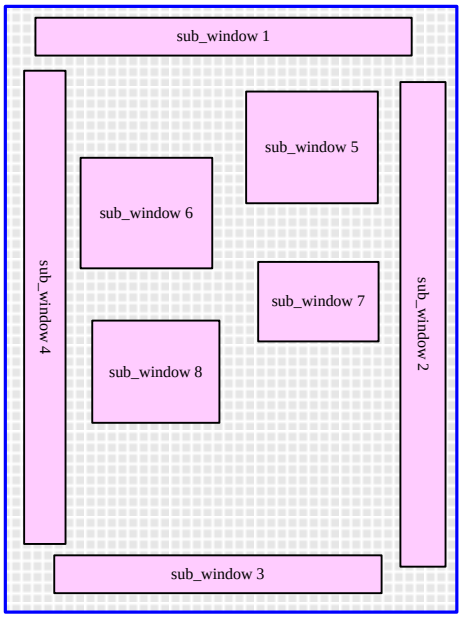
**When display, OSG window 3 will overlay lower video, OSG window 2 will overlay OSG window 3, and OSG window 1 will overlay OSG window 2. The sequence of display is:**




**Display modes of three windows are separate for transparent, alpha blending, blinking.**

### EIGHT SUB-WINDOWS

**In each OSG window, there are 8 sub-windows which can select from 8 different SDRAM contents. All eight windows can be different size and position, but cannot overlap.**




**Each window has its own position and size. To set different register, user must set register OSG_W1_SEL for window 1, OSG_W2_SEL or window 2, and OSG_W3_SEL for window 3. Each sub-window can be turn on or off by register OSG_W1_ENn or OSG_W2_ENn or OSG_W3_ENn. Here n is sub-window number. OSG_WM1_VSTARTn, OSG_WM1_HSTARTn, OSG_W1_VSIZEn, OSG_W1_HSIZEn, OSG_WS1_VSTARTn and OSG_WS1_HSTARTn share same address.**

### TRANSPARENT

**TW2880 has a fixed transparent color: 0xFFFF. To enable transparent, set bit OSG_TRANS_EN to high. The pixels with color 0xFFFF will not be display on screen. The lower layer image will be displayed. If OSG_TRANS_EN is set to low, all the pixels not in OSG window will display white.**

### ALPHA BLENDING

**Register 0x105[3:2], 0x106[3:2] and 0x163[3:2] are used blending mode control. It controls the alpha blending function of a OSG window. Mode "00" will disable blending. The other three modes enable alpha blending. When mode "01" enabled, pixels with color OSG_ALPHA_COLOR will be mixed with lower layer image, and the other pixels will overwrite lower layer image. When mode "10" enabled, pixels with color OSG_ALPHA_COLOR will overwrite lower layer image, and the other pixels will mixed with lower layer image. When mode "11" is set, all the pixels will mixed with lower layer image. The blending function will be: Video_data * alpha + osg_data * (1 – alpha). There are four big register to program alpha. Each window has two blending colors.**

### BLINKING

**When 0x105 bit [5:4] is other than 0x0, the pixel blinking feature is on. This feature allows the pixel to switch back and forth between a foreground color and a background color. Both colors can be programmed. If bit[5:4] equals 0x1, the pixels with color OSG_BLINK_COLOR will blink. If bit[5:4] equals 0x2, the pixels with colors not equals to OSG_BLINK_COLOR will blink. If bit[5:4] equals 0x3, all pixels will blink. The blink speed is determined by programming the OSG_BLINK_FRAME's value.**

### RGB FORMAT

**TW2880 Support for 16 bit RGB Format.**

| Mode | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | G | | | | | | R | | | | | B | | | | |
| 1 | B | G | | | | | R | | | | | B | | | | |
| 2 | A | G | | | | | R | | | | | B | | | | |
| 3 | B | A | G | | | | R | | | | | B | | | | |

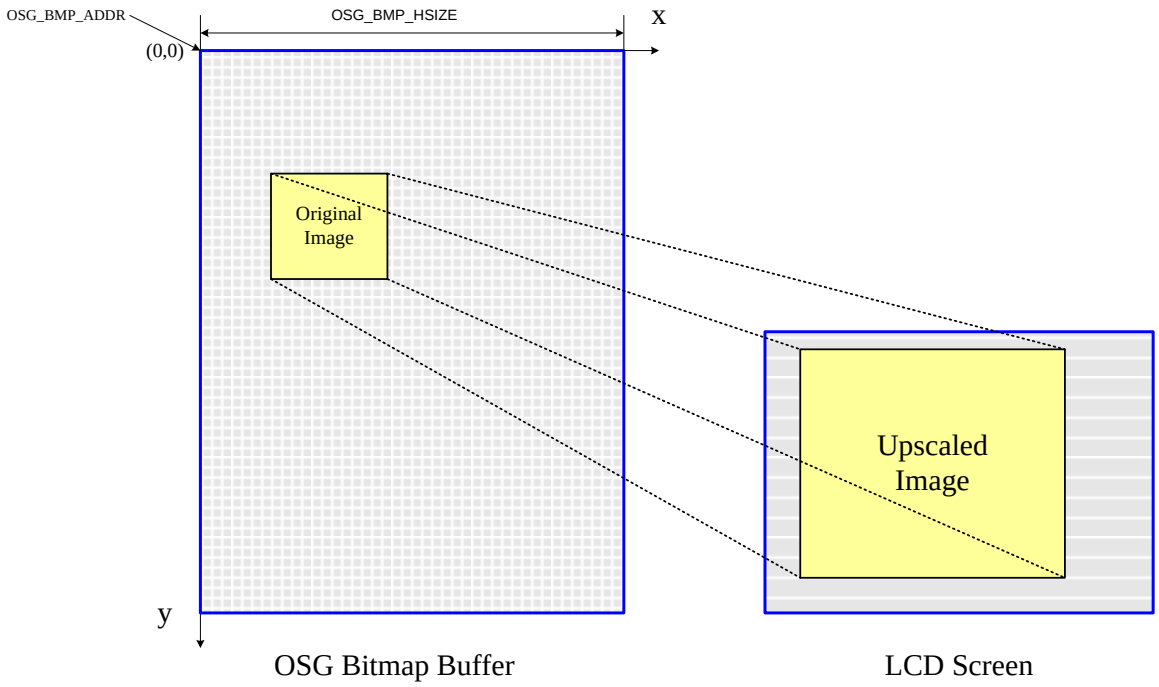
**In mode 0, blinking and blending can be done by color. In mode 1 to 3, blinking and blending can be done by pixel.**

**If OSG_FORMAT_RG is set to high, data in 16bit mode is like:**

| Mode | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | R | | | | | | G | | | | | B | | | | |
| 1 | B | R | | | | | G | | | | | B | | | | |
| 2 | A | R | | | | | G | | | | | B | | | | |
| 3 | B | A | R | | | | G | | | | | B | | | | |

## UPSCALE

**OSG can display upscaled bitmap image to screen. That means bitmap in SDRAM can be a small image, and display image can be upscaled image. Three windows can have different upscale ratios. Upscale ratio is determined by original image size to target image size. Original image size is determined by register [0x1C0] to [0x1CB]. Target image size is determined by LCD screen size. Upscale function can be turned on or off by register [0x105], [0x106] or [0x163] bit 6. OSG_WM_HSTART, OSG_WM_VSTART, OSG_WS_HSTART, OSG_WS_VSTART, OSG_W_HSIZE and OSG_W_VSIZE are setting for original size, not for target size.**

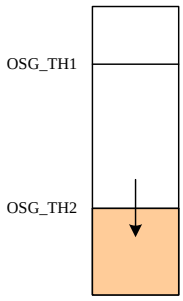



**Upscale function is used in bandwidth limitation case. User can use small size bitmap but display it in bigger LCD monitor. For example, original bitmap is 640x480, and LCD monitor is 1920x1080. Then OSG_VSIZE_DN should be 479, and OSG_HSIZE_DN should 639.**

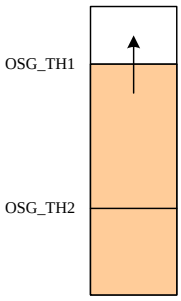
**HOST WRITE BUFFER THRESHOLD SETTING**

To make the transfer of OSG or menu data to the TW2880 become more efficient, there is a 1K Byte FIFO and some FIFO level control mechanism built in which can be used when the host try to write large amount of OSG data to TW2880.

In transferring OSG data to the off-screen memory, it usually involves consecutive writes the port 0x13A. Although there is a FIFO in the write path, user will still be worried about the operation because once the FIFO becomes full, the subsequent OSG writes will be lost and cannot recover. For this reason user needs to monitor OSG_WR_WAIT ([0x101] bit 1) closely because it represent the buffer state and is controlled by register OSG_TH1 and OSG_TH2. Following is a detailed explanation about the two registers.




buf_cnt < OSG_TH2 * 4,
then OSG_WR_WAIT becomes 0




buf_cnt >= OSG_TH1 * 4,
then OSG_WR_WAIT becomes 1

In the above picture, the two rectangles represent the FIFO data level. buf_cnt means data amount in the FIFO. The diagram shows the two criteria's of how the OSG_WR_WAIT change states. So if OSG_WR_WAIT is 0, you can write (1024- OSG_TH1*4) byte to the buffer without polling OSG_WR_WAIT state and the operation will still be right. This is very helpful to cut down the overall CPU overhead in the transfer process.

You must set OSG_TH1 >= OSG_TH2. If OSG_TH2*4 <= buf_cnt < OSG_TH1*4, OSG_WR_WAIT can be 0 or 1. If OSG_WR_WAIT is 0, that means buf_cnt < OSG_TH1*4, if OSG_WR_WAIT is 1, that means buf_cnt >= OSG_TH2*4.

If host does not want to make OSG_WR_WAIT toggle too much, you must set big difference between OSG_TH1 and OSG_TH2. If host does not care about OSG_WR_WAIT toggle too much, set OSG_TH1 = OSG_TH2. The P_WAIT_ST signal is the inverse of OSG_WR_WAIT register bit. Host can use this pin as a CPU halt signal in the total host write flow control.

Host can set a smaller OSG_TH1 value and write a lot of data to TW2880 and don't need to check the status. But if set the OSG_TH1 value become too small the host will need to wait a long time before OSG_WR_WAIT becomes 0. This will stop the transfer flow. The optimized setting is OSG_TH1 = OSG_TH2 = 128. If host want to read the status of OSG_WR_WAIT after writing 1 byte or 1 word, the best way to set OSG_TH1 = OSG_TH2 = 240 or some value close to 255. This is because OSG_WR_WAIT will be "0" most of the time. If carefully spacing the write pulse, host doesn't need to wait at all.

There are three examples for OSG_TH1 and OSG_TH2 setting.

| NO. | OSG_TH1 | OSG_TH2 | USED CASE |
|---|---|---|---|
| 1 | 240 | 16 | When wait_st pin is for interrupt, wait_st will not toggle too frequently |
| 2 | 128 | 128 | Host can write 512 byte data continuously without checking OSG_WR_WAIT register status |
| 3 | 240 | 240 | Host write one byte or one word data after checking OSG_WR_WAIT register status. Or Host use wait_st pin as CPU halt signal |

**HOST WRITE DATA SEQUENCE**

**Inside OSG design, the buffer uses 16-bit width. So the host needs to write even numbered data. The host can write in little endian sequence or big endian sequence. In default, big endian is used.**

**For example, if data is 0x12345678...., host write data sequence is:**

**8-bit mode, big endian, 0x12, 0x34, 0x56, 0x78...**

**8-bit mode, little endian, 0x34, 0x12, 0x78, 0x56...**

**16-bit mode, big endian, 0x1234, 0x5678...**

**16-bit mode, little endian, 0x3412, 0x7856...**

## Registers Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x100 | W | 0 | [0]: OSG_WR_START |
| 0x101 | RO | 0 | [3]: OSG_VACTIVE<br>[2]: OSG_VBLANK_TOGGLE<br>[1]: OSG_WR_WAIT<br>[0]: OSG_WR_BUSY |
| 0x102 | R/W | 0 | [7:6]: OSG_WR_MODE<br>[5]: OSG_RLC_EN<br>[4]: OSG_COLCON_EN<br>[3]: OSG_FORMAT_RG<br>[2:1]: OSG_OVWR_MODE<br>[0]: OSG_PIXEL_BIT |
| 0x103 | R/W | 0x80 | [7:4]: OSG_CNT_BIT<br>[3:2]: OSG_FORMAT<br>[1:0]: OSG_BLINK_FRAME |
| 0x104 | R/W | 0xCA | OSG_BITBLIT_OP |
| 0x105 | R/W | 0 | [6]: OSG_UPSCL_EN1<br>[5:4]: OSG_BLINK_MODE1<br>[3:2]: OSG_BLEND_MODE1<br>[1]: OSG_TRANS_EN1<br>[0]: OSG_W1_EN |
| 0x106 | R/W | 0 | [6]: OSG_UPSCL_EN2<br>[5:4]: OSG_BLINK_MODE2<br>[3:2]: OSG_BLEND_MODE2<br>[1]: OSG_TRANS_EN2<br>[0]: OSG_W2_EN |
| 0x107 | R/W | 0 | [7:4] OSG_ALPHA2<br>[3:0]: OSG_ALPHA1 |
| 0x108 | R/W | 0 | OSG_ALPHA_COLOR1 [7:0] |
| 0x109 | R/W | 0 | OSG_ALPHA_COLOR1 [15:8] |
| 0x10A | R/W | 0 | OSG_ALPHA_COLOR2 [7:0] |
| 0x10B | R/W | 0 | OSG_ALPHA_COLOR2 [15:8] |
| 0x10C | R/W | 0 | OSG_ALPHA_COLOR3 [7:0] |
| 0x10D | R/W | 0 | OSG_ALPHA_COLOR3 [15:8] |
| 0x10E | R/W | 0 | OSG_ALPHA_COLOR4 [7:0] |
| 0x10F | R/W | 0 | OSG_ALPHA_COLOR4 [15:8] |
| 0x110 | R/W | 0 | OSG_BLINK_COLOR1 [7:0] |
| 0x111 | R/W | 0 | OSG_BLINK_COLOR1 [15:8] |
| 0x112 | R/W | 0 | OSG_BLINK_COLOR2 [7:0] |
| 0x113 | R/W | 0 | OSG_BLINK_COLOR2 [15:8] |
| 0x114 | R/W | 0 | OSG_FILL_COLOR [7:0] |
| 0x115 | R/W | 0 | OSG_FILL_COLOR [15:8] |
| 0x116 | R/W | 0 | OSG_OVWR_COLOR [7:0] |
| 0x117 | R/W | 0 | OSG_OVWR_COLOR [15:8] |
| 0x118 | R/W | 0 | OSG_CON_SRC1 [7:0] |
| 0x119 | R/W | 0 | OSG_CON_SRC1 [15:8] |
| 0x11A | R/W | 0 | OSG_CON_TAR1 [7:0] |
| 0x11B | R/W | 0 | OSG_CON_TAR1 [15:8] |
| 0x11C | R/W | 0 | OSG_CON_SRC2 [7:0] |
| 0x11D | R/W | 0 | OSG_CON_SRC2 [15:8] |
| 0x11E | R/W | 0 | OSG_CON_TAR2 [7:0] |
| 0x11F | R/W | 0 | OSG_CON_TAR2 [15:8] |
| 0x120 | R/W | 0 | OSG_CON_SRC3 [7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x121 | R/W | 0 | OSG_CON_SRC3 [15:8] |
| 0x122 | R/W | 0 | OSG_CON_TAR3 [7:0] |
| 0x123 | R/W | 0 | OSG_CON_TAR3 [15:8] |
| 0x124 | R/W | 0 | OSG_CON_SRC4 [7:0] |
| 0x125 | R/W | 0 | OSG_CON_SRC4 [15:8] |
| 0x126 | R/W | 0 | OSG_CON_TAR4 [7:0] |
| 0x127 | R/W | 0 | OSG_CON_TAR4 [15:8] |
| 0x128 | R/W | 0 | OSG_BMP_HSIZE [7:0] |
| 0x129 | R/W | 0 | OSG_BMP_HSIZE [13:8] |
| 0x12A | R/W | 0 | OSG_B1_VSTART [7:0] |
| 0x12B | R/W | 0 | OSG_B1_VSTART [12:8] |
| 0x12C | R/W | 0 | OSG_B1_HSTART [7:0] |
| 0x12D | R/W | 0 | OSG_B1_HSTART [12:8] |
| 0x12E | R/W | 0 | OSG_B1_VSIZE [7:0] |
| 0x12F | R/W | 0 | OSG_B1_VSIZE [13:8] |
| 0x130 | R/W | 0 | OSG_B1_HSIZE [7:0] |
| 0x131 | R/W | 0 | OSG_B1_HSIZE [13:8] |
| 0x132 | R/W | 0 | OSG_B2_VSTART [7:0] |
| 0x133 | R/W | 0 | OSG_B2_VSTART [12:8] |
| 0x134 | R/W | 0 | OSG_B2_HSTART [7:0] |
| 0x135 | R/W | 0 | OSG_B2_HSTART [12:8] |
| 0x136 | R/W | 0 | OSG_B3_VSTART [7:0] |
| 0x137 | R/W | 0 | OSG_B3_VSTART [12:8] |
| 0x138 | R/W | 0 | OSG_B3_HSTART [7:0] |
| 0x139 | R/W | 0 | OSG_B3_HSTART [12:8] |
| 0x13A | R/W | 0 | OSG_HOST_DATA [7:0] / OSG_HOST_DATA [15:0] |
| 0x13B | R/W | 0 | OSG_BMP_ADDR [7:0] |
| 0x13C | R/W | 0 | OSG_BMP_ADDR [15:8] |
| 0x13D | R/W | 0 | OSG_BMP_ADDR [21:16] |
| 0x144 | R/W | 0 | OSG_WM1_VSTART [7:0] |
| 0x145 | R/W | 0 | OSG_WM1_VSTART [12:8] |
| 0x146 | R/W | 0 | OSG_WM1_HSTART [7:0] |
| 0x147 | R/W | 0 | OSG_WM1_HSTART [12:8] |
| 0x148 | R/W | 0 | OSG_W1_VSIZE [7:0] |
| 0x149 | R/W | 0 | OSG_W1_VSIZE [10:8] |
| 0x14A | R/W | 0 | OSG_W1_HSIZE [7:0] |
| 0x14B | R/W | 0 | OSG_W1_HSIZE [10:8] |
| 0x14C | R/W | 0 | OSG_WS1_VSTART [7:0] |
| 0x14D | R/W | 0 | OSG_WS1_VSTART [10:8] |
| 0x14E | R/W | 0 | OSG_WS1_HSTART [7:0] |
| 0x14F | R/W | 0 | OSG_WS1_HSTART [10:8] |
| 0x150 | R/W | 0 | OSG_WM2_VSTART [7:0] |
| 0x151 | R/W | 0 | OSG_WM2_VSTART [12:8] |
| 0x152 | R/W | 0 | OSG_WM2_HSTART [7:0] |
| 0x153 | R/W | 0 | OSG_WM2_HSTART [12:8] |
| 0x154 | R/W | 0 | OSG_W2_VSIZE [7:0] |
| 0x155 | R/W | 0 | OSG_W2_VSIZE [10:8] |
| 0x156 | R/W | 0 | OSG_W2_HSIZE [7:0] |
| 0x157 | R/W | 0 | OSG_W2_HSIZE [10:8] |
| 0x158 | R/W | 0 | OSG_WS2_VSTART [7:0] |
| 0x159 | R/W | 0 | OSG_WS2_VSTART [10:8] |
| 0x15A | R/W | 0 | OSG_WS2_HSTART [7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x15B | R/W | 0 | OSG_WS2_HSTART [10:8] |
| 0x15C | RW | 0 | OSG_OVWR_COLOR2 [7:0] |
| 0x15D | RW | 0 | OSG_OVWR_COLOR2 [15:8] |
| 0x15E | RW | 0 | OSG_OVWR_COLOR3 [7:0] |
| 0x15F | RW | 0 | OSG_OVWR_COLOR3 [15:8] |
| 0x160 | RW | 0 | OSG_OVWR_COLOR4 [7:0] |
| 0x161 | RW | 0 | OSG_OVWR_COLOR4 [15:8] |
| 0x162 | RW | 0 | [4]: OSG_WAIT_PINEN<br>[3:1]: OSG_VBLANK_SEL<br>[0]: OSG_VBLANK_EN |
| 0x163 | R/W | 0 | [6]: OSG_UPSCL_EN3<br>[5:4]: OSG_BLINK_MODE3<br>[3:2]: OSG_BLEND_MODE3<br>[1]: OSG_TRANS_EN3<br>[0]: OSG_W3_EN |
| 0x164 | R/W | 0 | [3:0]: OSG_ALPHA3 |
| 0x165 | R/W | 0 | OSG_ALPHA_COLOR5 [7:0] |
| 0x166 | R/W | 0 | OSG_ALPHA_COLOR5 [15:8] |
| 0x167 | R/W | 0 | OSG_ALPHA_COLOR6 [7:0] |
| 0x168 | R/W | 0 | OSG_ALPHA_COLOR6 [15:8] |
| 0x169 | R/W | 0 | OSG_BLINK_COLOR3 [7:0] |
| 0x16A | R/W | 0 | OSG_BLINK_COLOR3 [15:8] |
| 0x16B | R/W | 0x01 | [7]: OSG_W1_EN8<br>[6]: OSG_W1_EN7<br>[5]: OSG_W1_EN6<br>[4]: OSG_W1_EN5<br>[3]: OSG_W1_EN4<br>[2]: OSG_W1_EN3<br>[1]: OSG_W1_EN2<br>[0]: OSG_W1_EN1 |
| 0x16C | R/W | 0x01 | [7]: OSG_W2_EN8<br>[6]: OSG_W2_EN7<br>[5]: OSG_W2_EN6<br>[4]: OSG_W2_EN5<br>[3]: OSG_W2_EN4<br>[2]: OSG_W2_EN3<br>[1]: OSG_W2_EN2<br>[0]: OSG_W2_EN1 |
| 0x16D | R/W | 0x01 | [7]: OSG_W3_EN8<br>[6]: OSG_W3_EN7<br>[5]: OSG_W3_EN6<br>[4]: OSG_W3_EN5<br>[3]: OSG_W3_EN4<br>[2]: OSG_W3_EN3<br>[1]: OSG_W3_EN2<br>[0]: OSG_W3_EN1 |
| 0x16E | R/W | 0x00 | [6:4]: OSG_W2_SEL<br>[2:0]: OSG_W1_SEL |
| 0x16F | R/W | 0x00 | [2:0]: OSG_W3_SEL |
| 0x170 | R/W | 0 | OSG_WM3_VSTART [7:0] |
| 0x171 | R/W | 0 | OSG_WM3_VSTART [12:8] |
| 0x172 | R/W | 0 | OSG_WM3_HSTART [7:0] |
| 0x173 | R/W | 0 | OSG_WM3_HSTART [12:8] |
| 0x174 | R/W | 0 | OSG_W3_VSIZE [7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x175 | R/W | 0 | OSG_W3_VSIZE [10:8] |
| 0x176 | R/W | 0 | OSG_W3_HSIZE [7:0] |
| 0x177 | R/W | 0 | OSG_W3_HSIZE [10:8] |
| 0x178 | R/W | 0 | OSG_WS3_VSTART [7:0] |
| 0x179 | R/W | 0 | OSG_WS3_VSTART [10:8] |
| 0x17A | R/W | 0 | OSG_WS3_HSTART [7:0] |
| 0x17B | R/W | 0 | OSG_WS3_HSTART [10:8] |
| 0x17C | R/W | 0 | OSG_TH1 [7:0] |
| 0x17D | R/W | 0 | OSG_TH2 [7:0] |
| 0x17E | R/W | 0 | [2]: OSG_LIT_ENDIAN<br>[1]: OSG_FROM_DMA<br>[0]: OSG_FROM_DMA_OSG |
| 0x17F | R/W | 0x01 | [0]: OSG_REG_UPDATE |
| 0x1C0 | R/W | 0 | OSG_VSIZE_DN1 [7:0] |
| 0x1C1 | R/W | 0 | OSG_VSIZE_DN1 [10:8] |
| 0x1C2 | R/W | 0 | OSG_HSIZE_DN1 [7:0] |
| 0x1C3 | R/W | 0 | OSG_HSIZE_DN1 [10:8] |
| 0x1C4 | R/W | 0 | OSG_VSIZE_DN2 [7:0] |
| 0x1C5 | R/W | 0 | OSG_VSIZE_DN2 [10:8] |
| 0x1C6 | R/W | 0 | OSG_HSIZE_DN2 [7:0] |
| 0x1C7 | R/W | 0 | OSG_HSIZE_DN2 [10:8] |
| 0x1C8 | R/W | 0 | OSG_VSIZE_DN3 [7:0] |
| 0x1C9 | R/W | 0 | OSG_VSIZE_DN3 [10:8] |
| 0x1CA | R/W | 0 | OSG_HSIZE_DN3 [7:0] |
| 0x1CB | R/W | 0 | OSG_HSIZE_DN3 [10:8] |
| 0x1CC | R/W | 0 | OSG_BMP_HSIZE_2 [7:0] |
| 0x1CD | R/W | 0 | OSG_BMP_HSIZE_2 [13:8] |
| 0x1CE | R/W | 0 | OSG_BMP_ADDR_2 [7:0] |
| 0x1CF | R/W | 0 | OSG_BMP_ADDR_2 [15:8] |
| 0x1D0 | R/W | 0 | OSG_BMP_ADDR_2 [21:16] |

# Register Descriptions

### OSG CONTROL REGISTER – 0X100

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | W | 0 | **OSG_WR_START**<br><br>Block fill, transfer, BitBlt bitmap write mode start. The mode depends on bit [3:1] in this register<br>1: start<br>0: auto clear |

### OSG STATUS REGISTER 2 – 0X101

| Bit | R/W | Default | Description |
|---|---|---|---|
| 3 | RO | - | **OSG_VACTIVE**<br><br>OSG vertical active timing information. 1= video active |
| 2 | RO | - | **OSG_VBLANK_TOGGLE**<br><br>OSG display timing information. This signal toggles after vertical blank start. Read value can be 1 or zero. |
| 1 | RO | - | **OSG_WR_WAIT**<br><br>OSG buffer almost full. If high, internal data buffer is full, host cannot write data to hardware. If low, host can continue to write data to hardware. It's used for host write data to OSG memory. It is determined by OSG_TH1 and OSG_TH2<br>1: become full<br>0: become empty |
| 0 | RO | - | **OSG_WR_BUSY**<br><br>OSG write busy. If high, internal state is busy doing writing, host cannot send next command. If low, host can send next command.<br>1: busy<br>0: ready |

### OSG MODE REGISTER 1 – 0X102

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | **OSG_WR_MODE**<br><br>Block fill, transfer and OSG memory write mode<br>00: bitmap/linear memory write by host, through OSG_HOST_DATA<br>01: block transfer by hardware<br>10: block fill by hardware<br>11: BitBlt |
| 5 | RW | 0 | **OSG_RLC_EN**<br><br>OSG memory write run length compression enable<br>1: enable<br>0: disable |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4 | RW | 0 | **OSG_COLCON_EN**<br><br>OSG memory writing or block transfer color conversion enable<br>1: enable<br>0: disable |
| 3 | RW | 0 | **OSG_LNRMAP_EN**<br><br>OSG linear address mapping enable<br>1: enable<br>0: disable |
| 2:1 | RW | 0 | **OSG_OVWR_MODE**<br><br>OSG memory host writing or block transfer selective overwrite mode<br>00: selective overwrite, mask from registers<br>01: 0xFFFF will not be written to SDRAM<br>10: BitBlt with mask 0<br>11: no BitBlt function, all pixels written to SDRAM |
| 0 | R/W | 0 | **OSG_PIXEL_BIT**<br><br>When writing OSG bitmap buffer, input data width can be 2 or 16. When 2 bit is used, 2 bit will be expanded to 16 bit. In this case, 00 will be expanded to OSG_CON_TAR1, 01 will be expanded to OSG_CON_TAR2, 10 will be expanded to OSG_CON_TAR3, and 11 will be expanded to OSG_CON_TAR4.<br>1: 2 bit / pixel<br>0: 16 bit / pixel |

### OSG MODE REGISTER 2 – 0X103

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 8 | **OSG_CNT_BIT**<br><br>It is used for compressed data count bit number.<br>0 means 16 bits. |
| 3:2 | R/W | 0 | **OSG_FORMAT**<br><br>OSG 16 bit data format<br>00: RGB 565 format<br>01: Blinking bit + RGB 555 format<br>10: Alpha bit + RGB 555 format<br>11: Blinking and alpha bit + RGB 554 format |
| 1:0 | R/W | 0 | **OSG_BLINK_FRAME**<br><br>OSG blinking frequency control<br>00: blinking on each 32 frames<br>01: blinking on each 16 frames<br>10: blinking on each 8 frames<br>11: blinking on each 4 frames |

### OSG BITBLT OPERATION REGISTER – 0X104

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xCA | OSG_BITBLIT_OP<br><br>User need to fill in OPENGL style 3OP BitBlt opcode |

### OSG DISPLAY WINDOW 1 CONTROL REGISTER – 0X105

| Bit | R/W | Default | Description |
|---|---|---|---|
| 6 | R/W | 0 | OSG_UPSCL_EN1<br><br>Upscale enable<br>0: disable<br>1: enable |
| 5:4 | R/W | 0 | OSG_BLINK_MODE1<br><br>Blink enable. Blink frequency is determined by OSG_BLINK_FRAME<br>00: blink disable<br>01: blink in pixels with color OSG_BLINK_COLOR<br>10: blink in pixels with color not equals to OSG_BLINK_COLOR<br>11: blink in all pixels |
| 3:2 | R/W | 0 | OSG_BLEND_MODE1<br><br>00: alpha blending disable<br>01: alpha blending in pixels with color OSG_ALPHA_COLOR<br>10: alpha blending in pixels with color not equals to OSG_ALPHA_COLOR<br>11: alpha blending in all pixels |
| 1 | R/W | 0 | OSG_TRANS_EN1<br><br>OSG transparent enable<br>1: transparent when data is 0xFFFF<br>0: do not transparent |
| 0 | R/W | 0 | OSG_W1_EN<br><br>OSG first window enable<br>1: enable<br>0: disable |

### OSG DISPLAY WINDOW 2 CONTROL REGISTER – 0X106

| Bit | R/W | Default | Description |
|---|---|---|---|
| 6 | R/W | 0 | OSG_UPSCL_EN2<br><br>Upscale enable<br>0: disable<br>1: enable |
| 5:4 | R/W | 0 | OSG_BLINK_MODE2<br><br>Blink enable. Blink frequency is determined by OSG_BLINK_FRAME<br>00: blink disable<br>01: blink in pixels with color OSG_BLINK_COLOR<br>10: blink in pixels with color not equals to OSG_BLINK_COLOR<br>11: blink in all pixels |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 3:2 | R/W | 0 | OSG_BLEND_MODE2<br><br>00: alpha blending disable<br>01: alpha blending in pixels with color OSG_ALPHA_COLOR<br>10: alpha blending in pixels with color not equals to OSG_ALPHA_COLOR<br>11: alpha blending in all pixels |
| 1 | R/W | 0 | OSG_TRANS_EN2<br><br>OSG transparent enable<br>1: transparent when daa is 0xFFFF<br>0: do not transparent |
| 0 | R/W | 0 | OSG_W2_EN<br><br>OSG second window enable<br>1: enable<br>0: disable |

### OSG ALPHA BLENDING REGISTER – 0X107

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0 | OSG_ALPHA2<br><br>Alpha blending alpha number for OSG window 2.<br>The output will be: video_data2 * alpha2 + osg_data2 * (1 – alpha2).<br>Here alpha2 = OSG_ALPHA2 / 16. Maximum is 15. |
| 3:0 | R/W | 0 | OSG_ALPHA1<br><br>Alpha blending alpha number for OSG window 1.<br>The output will be: video_data1 * alpha1 + osg_data1 * (1 – alpha1).<br>Here alpha1 = OSG_ALPHA1 / 16. Maximum is 15. |

### OSG ALPHA BLENDING COLOR LOW BYTE REGISTER – 0X108

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_ALPHA_COLOR1 [7:0]<br><br>Alpha blending color for OSG window 1. RGB565 color |

### OSG ALPHA BLENDING COLOR HIGH BYTE REGISTER – 0X109

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_ALPHA_COLOR1 [15:8]<br><br>Alpha blending color for OSG window 1. RGB565 color |

### OSG ALPHA BLENDING COLOR LOW BYTE REGISTER – 0X10A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_ALPHA_COLOR2 [7:0]<br><br>Alpha blending color for OSG window 1. RGB565 color |

### OSG ALPHA BLENDING COLOR HIGH BYTE REGISTER – 0X10B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_ALPHA_COLOR2 [15:8]<br><br>Alpha blending color for OSG window 1. RGB565 color |

### OSG ALPHA BLENDING COLOR LOW BYTE REGISTER – 0X10C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_ALPHA_COLOR3 [7:0]<br><br>Alpha blending color for OSG window 2. RGB565 color |

### OSG ALPHA BLENDING COLOR HIGH BYTE REGISTER – 0X10D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_ALPHA_COLOR3 [15:8]<br><br>Alpha blending color for OSG window 2. RGB565 color |

### OSG ALPHA BLENDING COLOR LOW BYTE REGISTER – 0X10E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_ALPHA_COLOR4 [7:0]<br><br>Alpha blending color for OSG window 2. RGB565 color |

### OSG ALPHA BLENDING COLOR HIGH BYTE REGISTER – 0X10F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_ALPHA_COLOR4 [15:8]<br><br>Alpha blending color for OSG window 2. RGB565 color |

### OSG BLINKING FOREGOUND COLOR LOW BYTE REGISTER – 0X110

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_BLINK_COLOR1 [7:0]<br><br>Blinking foreground color for OSG window 1. RGB565 color |

### OSG BLINKING FOREGROUND COLOR HIGH BYTE REGISTER – 0X111

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_BLINK_COLOR1 [15:8]<br><br>Blinking foreground color for OSG window 1. RGB565 color |

### OSG BLINKING FOREGOUND COLOR LOW BYTE REGISTER – 0X112

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_BLINK_COLOR2 [7:0]<br><br>Blinking foreground color for OSG window 2. RGB565 color |

### OSG BLINKING FOREGROUND COLOR HIGH BYTE REGISTER – 0X113

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_BLINK_COLOR2 [15:8]<br><br>Blinking foreground color for OSG window 2. RGB565 color |

### OSG FILL COLOR LOW BYTE – 0X114

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_FILL_COLOR [7:0]<br><br>The auto filled pixel color in OSG bitmap buffer, RGB565 color |

### OSG FILL COLOR HIGH BYTE – 0X115

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_FILL_COLOR [15:8]<br><br>The auto filled pixel color in OSG bitmap buffer, RGB565 color |

### OSG OVERWRITE COLOR LOW BYTE – 0X116

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_OVWR_COLOR [7:0]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will not be written to the memory. |

### OSG OVERWRITE COLOR HIGH BYTE – 0X117

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_OVWR_COLOR [15:8]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will not be written to the memory. |

### OSG COLOR CONVERSION SOURCE COLOR 1 LOW BYTE – 0X118

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_SRC1 [7:0]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will be converted to color OSG_CON_TAR1 |

### OSG COLOR CONVERSION SOURCE COLOR 1 HIGH BYTE – 0X119

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_SRC1 [15:8]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will be converted to color OSG_CON_TAR1 |

### OSG COLOR CONVERSION TARGET COLOR 1 LOW BYTE – 0X11A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_TAR1 [7:0]<br><br>When OSG writing bitmap or block transfer, the pixels with color OSG_CON_SRC1 will be converted to this color |

### OSG COLOR CONVERSION TARGET COLOR 1 HIGH BYTE – 0X11B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_TAR1 [15:8]<br><br>When OSG writing bitmap or block transfer, the pixels with color OSG_CON_SRC1 will be converted to this color |

### OSG COLOR CONVERSION SOURCE COLOR 2 LOW BYTE – 0X11C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_SRC2 [7:0]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will be converted to color OSG_CON_TAR2 |

### OSG COLOR CONVERSION SOURCE COLOR 2 HIGH BYTE – 0X11D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_SRC2 [15:8]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will be converted to color OSG_CON_TAR2 |

### OSG COLOR CONVERSION TARGET COLOR 2 LOW BYTE – 0X11E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_TAR2 [7:0]<br><br>When OSG writing bitmap or block transfer, the pixels with color OSG_CON_SRC2 will be converted to this color |

### OSG COLOR CONVERSION TARGET COLOR 2 HIGH BYTE – 0X11F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_TAR2 [15:8]<br><br>When OSG writing bitmap or block transfer, the pixels with color OSG_CON_SRC2 will be converted to this color |

### OSG COLOR CONVERSION SOURCE COLOR 3 LOW BYTE – 0X120

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_SRC3 [7:0]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will be converted to color OSG_CON_TAR3 |

### OSG COLOR CONVERSION SOURCE COLOR 3 HIGH BYTE – 0X121

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_SRC3 [15:8]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will be converted to color OSG_CON_TAR3 |

### OSG COLOR CONVERSION TARGET COLOR 3 LOW BYTE – 0X122

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_TAR3 [7:0]<br><br>When OSG writing bitmap or block transfer, the pixels with color OSG_CON_SRC3 will be converted to this color |

### OSG COLOR CONVERSION TARGET COLOR 3 HIGH BYTE – 0X123

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_TAR3 [15:8]<br><br>When OSG writing bitmap or block transfer, the pixels with color OSG_CON_SRC3 will be converted to this color |

### OSG COLOR CONVERSION SOURCE COLOR 4 LOW BYTE – 0X124

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_SRC4 [7:0]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will be converted to color OSG_CON_TAR4 |

### OSG COLOR CONVERSION SOURCE COLOR 4 HIGH BYTE – 0X125

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_SRC4 [15:8]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will be converted to color OSG_CON_TAR4 |

### OSG COLOR CONVERSION TARGET COLOR 4 LOW BYTE – 0X126

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_TAR4 [7:0]<br><br>When OSG writing bitmap or block transfer, the pixels with color OSG_CON_SRC4 will be converted to this color |

### OSG COLOR CONVERSION TARGET COLOR 4 HIGH BYTE – 0X127

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_CON_TAR4 [15:8]<br><br>When OSG writing bitmap or block transfer, the pixels with color OSG_CON_SRC4 will be converted to this color |

### OSG BITMAP BUFFER HORIZONTAL SIZE LOW BYTE – 0X128

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_BMP_HSIZE [7:0]<br><br>OSG bitmap buffer total horizontal size for display. The maximum is 8192. Unit is 2 bytes / 1 pixel. The last 7 bit must be 0. OSG_BMP_HSIZE must be times of 128. |

### OSG BITMAP BUFFER HORIZONTAL SIZE HIGH BYTE – 0X129

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5:0 | R/W | 0x00 | OSG_BMP_HSIZE [13:8]<br><br>OSG bitmap buffer total horizontal size for display. The maximum is 8192. Unit is 2 bytes / 1 pixel. The last 7 bit must be 0. OSG_BMP_HSIZE must be times of 128. |

### OSG BITMAP BUFFER SOURCE WINDOW 1 VERTICAL START LOW BYTE – 0X12A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_B1_VSTART [7:0]<br><br>OSG block transfer memory source 1 vertical start location. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG BITMAP BUFFER SOURCE WINDOW 1 VERTICAL START HIGH BYTE – 0X12B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_B1_VSTART [12:8]<br><br>OSG block transfer memory source 1 vertical start location. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG BITMAP BUFFER SOURCE WINDOW 1 HORIZONTAL START LOW BYTE – 0X12C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_B1_HSTART [7:0]<br><br>OSG block transfer memory source 1 Horizontal start location. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG BITMAP BUFFER SOURCE WINDOW 1 HORIZONTAL START HIGH BYTE – 0X12D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_B1_HSTART [12:8]<br><br>OSG block transfer memory source 1 Horizontal start location. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG BITMAP BUFFER SOURCE WINDOW 1 VERTICAL SIZE LOW BYTE – 0X12E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_B1_VSIZE [7:0]<br><br>OSG block transfer memory source 1 vertical size. Unit is 1 pixel / 2 bytes. The maximum is 8192. This is also used for OSG bitmap write target size, block transfer / fill target size. |

### OSG BITMAP BUFFER SOURCE WINDOW 1 VERTICAL SIZE HIGH BYTE – 0X12F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5:0 | R/W | 0x00 | OSG_B1_VSIZE [13:8]<br><br>OSG block transfer memory source 1 vertical size. Unit is 1 pixel / 2 bytes. The maximum is 8192. This is also used for OSG bitmap write target size, block transfer / fill target size. |

### OSG BITMAP BUFFER SOURCE WINDOW 1 HORIZONTAL SIZE LOW BYTE – 0X130

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_B1_HSIZE [7:0]<br><br>OSG block transfer memory source 1 horizontal size. Unit is 1 pixel / 2 bytes. The maximum is 8192. This is also used for OSG bitmap write target size, block transfer / fill target size. |

### OSG BITMAP BUFFER SOURCE WINDOW 1 HORIZONTAL SIZE HIGH BYTE – 0X131

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5:0 | R/W | 0x00 | OSG_B1_HSIZE [13:8]<br><br>OSG block transfer memory source 1 horizontal size. Unit is 1 pixel / 2 bytes. The maximum is 8192. This is also used for OSG bitmap write target size, block transfer / fill target size. |

### OSG BITMAP BUFFER TARGET WINDOW VERTICAL START LOW BYTE – 0X132

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_B2_VSTART [7:0]<br><br>OSG bitmap write / block transfer / fill memory target vertical start location. Unit is 1 pixel / 2 bytes. The maximum is 4095. |

### OSG BITMAP BUFFER TARGET WINDOW VERTICAL START HIGH BYTE – 0X133

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_B2_VSTART [12:8]<br><br>OSG bitmap write / block transfer / fill memory target vertical start location. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG BITMAP BUFFER TARGET WINDOW HORIZONTAL START LOW BYTE – 0X134

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_B2_HSTART [7:0]<br><br>OSG bitmap write / block transfer / fill memory target horizontal start location. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG BITMAP BUFFER TARGET WINDOW HORIZONTAL START HIGH BYTE – 0X135

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_B2_HSTART [12:8]<br><br>OSG block transfer memory source Horizontal start location. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG BITMAP BUFFER SOURCE WINDOW 2 VERTICAL START LOW BYTE – 0X136

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_B3_VSTART [7:0]<br><br>OSG block transfer memory source 2 vertical start location. Unit is 1 pixel / 2 bytes. The maximum is 8191. It's used for BitBlt. |

### OSG BITMAP BUFFER SOURCE WINDOW 2 VERTICAL START HIGH BYTE – 0X137

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_B3_VSTART [12:8]<br><br>OSG block transfer memory source 2 vertical start location. Unit is 1 pixel / 2 bytes. The maximum is 8191. It's used for BitBlt. |

### OSG BITMAP BUFFER SOURCE WINDOW 2 HORIZONTAL START LOW BYTE – 0X138

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_B3_HSTART [7:0]<br><br>OSG block transfer memory source 2 Horizontal start location. Unit is 1 pixel / 2 bytes. The maximum is 8191. It's used for BitBlt. |

### BITMAP BUFFER SOURCE WINDOW 2 HORIZONTAL START HIGH BYTE – 0X139

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_B3_HSTART [12:8]<br><br>OSG block transfer memory source 2 Horizontal start location. Unit is 1 pixel / 2 bytes. The maximum is 8191. It's used for BitBlt. |

### OSG HOST WRITE DATA PORT – 0X13A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 15:8 | R/W | 0x00 | OSG_HOST_DATA [15:8]<br><br>High byte of the write date port. To use this portion of write port user needs to connect the high byte of the I/O pins and use EN16B signal to let host interface know a valid 16 bit transfer is needed. A 16 bit write will not write into 0x13B. |
| 7:0 | R/W | 0x00 | OSG_HOST_DATA [7:0]<br><br>The data register for OSG bitmap / linear buffer write. Once the write command is started, CPU keeps writing to this register for the byte count consistent with the setting. |

### OSG BITMAP BUFFER START ADDRESS LOW BYTE – 0X13B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_BMP_ADDR [7:0]<br><br>The bitmap buffer start address for display. Unit is 8 bytes / 4 pixels. Total is 21 bits. Start bank is fixed to 0. |

### OSG BITMAP BUFFER START ADDRESS MIDDLE BYTE – 0X13C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_BMP_ADDR [15:8]<br><br>The bitmap buffer start address for display. Unit is 8 bytes / 4 pixels. Total is 21 bits. Start bank is fixed to 0. |

### OSG BITMAP BUFFER START ADDRESS HIGH BYTE – 0X13D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_BMP_ADDR [23:16]<br><br>The bitmap buffer start address for display. Unit is 8 bytes / 4 pixels. Total is 24 bits. Start bank is fixed to 0 when bit [23] = 0. Start bank = bit [22:21] when bit [23] = 1 |

### OSG FORMAT CONTROL – 0X17E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4 | R/W | 0 | OSG_YcrCb2RGB<br><br>0: No conversion<br>1: Convert Y Cr Cb format to RGB |
| 2 | R/W | 0 | OSG_LIT_ENDIAN<br><br>0: big endian<br>1: little endian |
| 1 | R/W | 0 | OSG_FROM_DMA<br><br>0: Host DMA module write data directly to SDRAM<br>1: Host DMA module does not write data to SDRAM |
| 0 | R/W | 0 | OSG_FROM_DMA_OSG<br><br>0: OSG data is from CPU<br>1: OSG daa is from host DMA module |

### OSG BITMAP BUFFER START ADDRESS HIGH BYTE – 0X13D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_BMP_ADDR [21:16]<br><br>The bitmap buffer start address for display. Unit is 8 bytes / 4 pixels. Total is 22 bits. Start bank is fixed to 0. |

### OSG WINDOW 1 VERTICAL START IN BITMAP BUFFER LOW BYTE – 0X144

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_WM1_VSTART [7:0]<br><br>OSG first window vertical start in bitmap memory. Unit is 1 pixel / 2 bytes. The maximum is 4095. |

### OSG WINDOW 1 VERTICAL START IN BITMAP BUFFER HIGH BYTE – 0X145

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_WM1_VSTART [12:8]<br><br>OSG first window vertical start in bitmap memory. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG WINDOW 1 HORIZONTAL START IN BITMAP BUFFER LOW BYTE – 0X146

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_WM1_HSTART [7:0]<br><br>OSG first window horizontal start in bitmap memory. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG WINDOW 1 HORIZONTAL START IN BITMAP BUFFER HIGH BYTE – 0X147

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_WM1_HSTART [12:8]<br><br>OSG first window horizontal start in bitmap memory. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG WINDOW 1 VERTICAL SIZE LOW BYTE – 0X148

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_W1_VSIZE [7:0]<br><br>OSG first window vertical size. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 1 VERTICAL SIZE HIGH BYTE – 0X149

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_W1_VSIZE [10:8]<br><br>OSG first window vertical size. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 1 HORIZONTAL SIZE LOW BYTE – 0X14A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_W1_HSIZE [7:0]<br><br>OSG first window horizontal size. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 1 HORIZONTAL SIZE HIGH BYTE – 0X14B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_W1_HSIZE [10:8]<br><br>OSG first window horizontal size. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 1 VERTICAL START ON SCREEN LOW BYTE – 0X14C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_WS1_VSTART [7:0]<br><br>OSG first window vertical start on screen. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 1 VERTICAL START ON SCREEN HIGH BYTE – 0X14D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_WS1_VSTART [10:8]<br><br>OSG first window vertical start on screen. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 1 HORIZONTAL START ON SCREEN LOW BYTE – 0X14E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_WS1_HSTART [7:0]<br><br>OSG first window horizontal start on screen. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 1 HORIZONTAL START ON SCREEN HIGH BYTE – 0X14F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_WS1_HSTART [10:8]<br><br>OSG first window horizontal start on screen. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 2 VERTICAL START IN BITMAP BUFFER LOW BYTE – 0X150

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_WM2_VSTART [7:0]<br><br>OSG second window vertical start in bitmap memory. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG WINDOW 2 VERTICAL START IN BITMAP BUFFER HIGH BYTE – 0X151

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_WM2_VSTART [12:8]<br><br>OSG second window vertical start in bitmap memory. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG WINDOW 2 HORIZONTAL START IN BITMAP BUFFER LOW BYTE – 0X152

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_WM2_HSTART [7:0]<br><br>OSG second window horizontal start in bitmap memory. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG WINDOW 2 HORIZONTAL START IN BITMAP BUFFER HIGH BYTE – 0X153

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_WM2_HSTART [12:8]<br><br>OSG second window horizontal start in bitmap memory. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG WINDOW 2 VERTICAL SIZE LOW BYTE – 0X154

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_W2_VSIZE [7:0]<br><br>OSG second window vertical size. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 2 VERTICAL SIZE HIGH BYTE – 0X155

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_W2_VSIZE [10:8]<br><br>OSG second window vertical size. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 2 HORIZONTAL SIZE LOW BYTE – 0X156

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_W2_HSIZE [7:0]<br><br>OSG second window horizontal size. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 2 HORIZONTAL SIZE HIGH BYTE – 0X157

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_W2_HSIZE [10:8]<br><br>OSG second window horizontal size. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 2 VERTICAL START ON SCREEN LOW BYTE – 0X158

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_WS2_VSTART [7:0]<br><br>OSG second window vertical start on screen. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 2 VERTICAL START ON SCREEN HIGH BYTE – 0X159

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_WS2_VSTART [10:8]<br><br>OSG second window vertical start on screen. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 2 HORIZONTAL START ON SCREEN LOW BYTE – 0X15A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_WS2_HSTART [7:0]<br><br>OSG second window horizontal start on screen. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 2 HORIZONTAL START ON SCREEN HIGH BYTE – 0X15B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_WS2_HSTART [10:8]<br><br>OSG second window horizontal start on screen. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG OVERWRITE COLOR2 LOW BYTE – 0X15C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_OVWR_COLOR2 [7:0]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will not be written to the memory. |

### OSG OVERWRITE COLOR2 HIGH BYTE – 0X15D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_OVWR_COLOR2 [15:8]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will not be written to the memory. |

### OSG OVERWRITE COLOR3 LOW BYTE – 0X15E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_OVWR_COLOR3 [7:0]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will not be written to the memory. |

### OSG OVERWRITE COLOR3 HIGH BYTE – 0X15F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_OVWR_COLOR3 [15:8]<br><br>When OSG writing bitmap or block transfer, the pixels with this color will not be written to the memory. |

### OSG OVERWRITE COLOR4 LOW BYTE – 0X160

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **OSG_OVWR_COLOR4 [7:0]**<br><br>When OSG writing bitmap or block transfer, the pixels with this color will not be written to the memory. |

### OSG OVERWRITE COLOR4 HIGH BYTE – 0X161

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **OSG_OVWR_COLOR4 [15:8]**<br><br>When OSG writing bitmap or block transfer, the pixels with this color will not be written to the memory. |

### OSG VERTICAL BLANK REGISTER – 0X162

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | – | – | **Reserved** |
| 4 | R/W | 0 | **OSG_WAIT_PINEN**<br><br>Wait pin enable. If this bit is set to high, wait pin will be low when OSG_WR_WAIT is high. If this bit is set to low, wait pin is always high (open drain). |
| 3:1 | R/W | 0 | **OSG_VBLANK_SEL [1:0]**<br><br>000: osg1_vblank<br>001: osg1_vblank_window<br>010: osg2_vblank<br>011: osg2_vblank_window<br>100: osg3_vblank<br>101: osg3_vblank_window<br>Others: Reserved |
| 0 | R/W | 0 | **OSG_VBLANK_EN**<br><br>When this bit set to high, OSG write start begins from vertical blank start. Vertical blank is selected by OSG_VBLANK_SEL register. |

### OSG DISPLAY WINDOW 3 CONTROL REGISTER – 0X163

| Bit | R/W | Default | Description |
|---|---|---|---|
| 6 | R/W | 0 | **OSG_UPSCL_EN3:** Upscale enable<br><br>0: disable<br>1: enable |
| 5:4 | R/W | 0 | **OSG_BLINK_MODE3**<br><br>Blink enable. Blink frequency is determined by OSG_BLINK_FRAME<br>00: blink disable<br>01: blink in pixels with color OSG_BLINK_COLOR<br>10: blink in pixels with color not equals to OSG_BLINK_COLOR<br>11: blink in all pixels |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 3:2 | R/W | 0 | OSG_BLEND_MODE3<br><br>00: alpha blending disable<br>01: alpha blending in pixels with color OSG_ALPHA_COLOR<br>10: alpha blending in pixels with color not equals to OSG_ALPHA_COLOR<br>11: alpha blending in all pixels |
| 1 | R/W | 0 | OSG_TRANS_EN3<br><br>OSG transparent enable<br>1: transparent when daa is 0xFFFF<br>0: do not transparent |
| 0 | R/W | 0 | OSG_W3_EN<br><br>OSG third window enable<br>1: enable<br>0: disable |

### OSG ALPHA BLENDING REGISTER – 0X164

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | – | – | Reserved |
| 3:0 | R/W | 0 | OSG_ALPHA3<br><br>Alpha blending alpha number for OSG window 3.<br>The output will be: video_data3 * alpha3 + osg_data3 * (1 – alpha3).<br>Here alpha3 = OSG_ALPHA3 / 16. Maximum is 15. |

### OSG ALPHA BLENDING COLOR LOW BYTE REGISTER – 0X165

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_ALPHA_COLOR5 [7:0]<br><br>Alpha blending color for OSG window 3. 16 bit = RGB565 |

### OSG ALPHA BLENDING COLOR HIGH BYTE REGISTER – 0X166

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_ALPHA_COLOR5 [15:8]<br><br>Alpha blending color for OSG window 3. 16 bit = RGB565 |

### OSG ALPHA BLENDING COLOR LOW BYTE REGISTER – 0X167

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_ALPHA_COLOR6 [7:0]<br><br>Alpha blending color for OSG window 3. 16 bit = RGB565 |

### OSG ALPHA BLENDING COLOR HIGH BYTE REGISTER – 0X168

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_ALPHA_COLOR6 [15:8]<br><br>Alpha blending color for OSG window 3. 16 bit = RGB565 |

### OSG BLINKING FOREGOUND COLOR LOW BYTE REGISTER – 0X169

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_BLINK_COLOR3 [7:0]<br><br>Blinking foreground color for OSG window 3. 16 bit = RGB565 |

### OSG BLINKING FOREGROUND COLOR HIGH BYTE REGISTER – 0X16A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | OSG_BLINK_COLOR3 [15:8]<br><br>Blinking foreground color for OSG window 3. 16 bit = RGB565 |

### OSG DISPLAY WINDOW 1 ENABLE REGISTER – 0X16B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | OSG_W1_EN8 |
| 6 | R/W | 0 | OSG_W1_EN7 |
| 5 | R/W | 0 | OSG_W1_EN6 |
| 4 | R/W | 0 | OSG_W1_EN5 |
| 3 | R/W | 0 | OSG_W1_EN4 |
| 2 | R/W | 0 | OSG_W1_EN3 |
| 1 | R/W | 0 | OSG_W1_EN2 |
| 0 | R/W | 1 | OSG_W1_EN1<br><br>Sub-window enable bits for window 1, 1= enable |

### OSG DISPLAY WINDOW 2 ENABLE REGISTER – 0X16C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | OSG_W2_EN8 |
| 6 | R/W | 0 | OSG_W2_EN7 |
| 5 | R/W | 0 | OSG_W2_EN6 |
| 4 | R/W | 0 | OSG_W2_EN5 |
| 3 | R/W | 0 | OSG_W2_EN4 |
| 2 | R/W | 0 | OSG_W2_EN3 |
| 1 | R/W | 0 | OSG_W2_EN2 |
| 0 | R/W | 1 | OSG_W2_EN1<br><br>Sub-window enable bits for window 2, 1= enable |

### OSG DISPLAY WINDOW 3 ENABLE REGISTER – 0X16D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | OSG_W3_EN8 |
| 6 | R/W | 0 | OSG_W3_EN7 |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5 | R/W | 0 | OSG_W3_EN6 |
| 4 | R/W | 0 | OSG_W3_EN5 |
| 3 | R/W | 0 | OSG_W3_EN4 |
| 2 | R/W | 0 | OSG_W3_EN3 |
| 1 | R/W | 0 | OSG_W3_EN2 |
| 0 | R/W | 1 | OSG_W3_EN1<br><br>Sub-window enable bits for window 3, 1= enable |

### OSG DISPLAY WINDOW 1 AND 2 SELECT REGISTER – 0X16E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x00 | Reserved |
| 6:4 | R/W | 0x00 | OSG_W2_SEL<br><br>Sub-windows register selection for window 2 |
| 3 | R | 0x00 | Reserved |
| 2:0 | R/W | 0x00 | OSG_W1_SEL<br><br>Sub-windows register selection for window 1 |

### OSG DISPLAY WINDOW 3 SELECT REGISTER – 0X16F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0x00 | Reserved |
| 2:0 | R/W | 0x00 | OSG_W3_SEL<br><br>Sub-windows register selection for window 3 |

### OSG WINDOW 3 VERTICAL START IN BITMAP BUFFER LOW BYTE – 0X170

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_WM3_VSTART [7:0]<br><br>OSG third window vertical start in bitmap memory. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG WINDOW 3 VERTICAL START IN BITMAP BUFFER HIGH BYTE – 0X171

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_WM3_VSTART [12:8]<br><br>OSG third window vertical start in bitmap memory. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG WINDOW 3 HORIZONTAL START IN BITMAP BUFFER LOW BYTE – 0X172

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_WM3_HSTART [7:0]<br><br>OSG third window horizontal start in bitmap memory. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG WINDOW 3 HORIZONTAL START IN BITMAP BUFFER HIGH BYTE – 0X173

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_WM3_HSTART [12:8]<br><br>OSG third window horizontal start in bitmap memory. Unit is 1 pixel / 2 bytes. The maximum is 8191. |

### OSG WINDOW 3 VERTICAL SIZE LOW BYTE – 0X174

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_W3_VSIZE [7:0]<br><br>OSG third window vertical size. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 3 VERTICAL SIZE HIGH BYTE – 0X175

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_W3_VSIZE [10:8]<br><br>OSG third window vertical size. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 3 HORIZONTAL SIZE LOW BYTE – 0X176

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_W3_HSIZE [7:0]<br><br>OSG third window horizontal size. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 3 HORIZONTAL SIZE HIGH BYTE – 0X177

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_W3_HSIZE [10:8]<br><br>OSG third window horizontal size. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 3 VERTICAL START ON SCREEN LOW BYTE – 0X178

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_WS3_VSTART [7:0]<br><br>OSG third window vertical start on screen. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 3 VERTICAL START ON SCREEN HIGH BYTE – 0X179

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_WS3_VSTART [10:8]<br><br>OSG third window vertical start on screen. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 3 HORIZONTAL START ON SCREEN LOW BYTE – 0X17A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_WS3_HSTART [7:0]<br><br>OSG third window horizontal start on screen. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG WINDOW 3 HORIZONTAL START ON SCREEN HIGH BYTE – 0X17B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_WS3_HSTART [10:8]<br><br>OSG third window horizontal start on screen. Unit is 1 pixel / 2 bytes. The maximum is 2047. |

### OSG THRESHOLD 1 – 0X17C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x80 | OSG_TH1 [7:0]<br><br>OSG threshold 1. OSG_WR_WAIT will be high when data amount in the buffer is more than OSG_TH1*4. The buffer size is 1024 bytes.<br>OSG_TH1 >= OSG_TH2 |

### OSG THRESHOLD 2 – 0X17D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x80 | OSG_TH2 [7:0]<br><br>OSG threshold 2. OSG_WR_WAIT will be low when data amount in the buffer is less than OSG_TH2*4. The buffer size is 1024 bytes.<br>OSG_TH1 >= OSG_TH2 |

### OSG FORMAT CONTROL – 0X17E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2 | R/W | 0 | OSG_LIT_ENDIAN<br><br>0: big endian<br>1: little endian |
| 1 | R/W | 0 | OSG_FROM_DMA<br><br>0: Host DMA module write data directly to SDRAM<br>1: Host DMA module does not write data to SDRAM |
| 0 | R/W | 0 | OSG_FROM_DMA_OSG<br><br>0: OSG data is from CPU<br>1: OSG daa is from host DMA module |

### OSG FORMAT CONTROL – 0X17F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | R/W | 1 | OSG_REG_UPDATE<br><br>When host update OSG display related registers such as position and size, host must set this bit to 0 and then change those registers. After changing done, host set this bit to 1. |

### OSG ORIGINAL VERTICAL SIZE FOR WINDOW 1 LOW BYTE– 0X1C0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_VSIZE_DN1 [7:0]<br><br>OSG original vertical size minus 1. Unit is pixel. |

### OSG ORIGINAL VERTICAL SIZE FOR WINDOW 1 HIGH BYTE – 0X1C1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_VSIZE_DN1 [10:8]<br><br>OSG original vertical size minus 1. Unit is pixel. |

### OSG ORIGINAL HORIZONTAL SIZE FOR WINDOW 1 LOW BYTE– 0X1C2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_HSIZE_DN1 [7:0]<br><br>OSG original horizontal size minus 1. OSG_HSIZE_DN1+1 must be times of 4. |

### OSG ORIGINAL HORIZONTAL SIZE FOR WINDOW 1 HIGH BYTE – 0X1C3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | R/W | 0x00 | OSG_HSIZE_DN1 [10:8]<br><br>OSG original horizontal size minus 1. OSG_HSIZE_DN1+1 must be times of 4. |

Register 0x1C4 to 0x1CB are for window 2 to 3

### OSG BITMAP BUFFER HORIZONTAL SIZE LOW BYTE – 0X1CC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_BMP_HSIZE_2 [7:0]<br><br>OSG bitmap buffer total horizontal size for OSG write. The maximum is 8192. Unit is 2 bytes / 1 pixel. The last 7 bit must be 0. OSG_BMP_HSIZE must be times of 128. |

### OSG BITMAP BUFFER HORIZONTAL SIZE HIGH BYTE – 0X1CD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5:0 | R/W | 0x00 | OSG_BMP_HSIZE_2 [13:8]<br><br>OSG bitmap buffer total horizontal size for OSG write. The maximum is 8192. Unit is 2 bytes / 1 pixel. The last 7 bit must be 0. OSG_BMP_HSIZE must be times of 128. |

### OSG BITMAP BUFFER START ADDRESS LOW BYTE – 0X1CE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_BMP_ADDR_2 [7:0]<br><br>The bitmap buffer start address for OSG write. Unit is 8 bytes / 4 pixels. Total is 21 bits. Start bank is fixed to 0. |

### OSG BITMAP BUFFER START ADDRESS MIDDLE BYTE – 0X1CF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | OSG_BMP_ADDR_2 [15:8]<br><br>The bitmap buffer start address for OSG write. Unit is 8 bytes / 4 pixels. Total is 21 bits. Start bank is fixed to 0. |

### OSG BITMAP BUFFER START ADDRESS HIGH BYTE – 0X1D0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | R/W | 0x00 | OSG_BMP_ADDR_2 [21:16]<br><br>The bitmap buffer start address for OSG write. Unit is 8 bytes / 4 pixels. Total is 22 bits. Start bank is fixed to 0. |

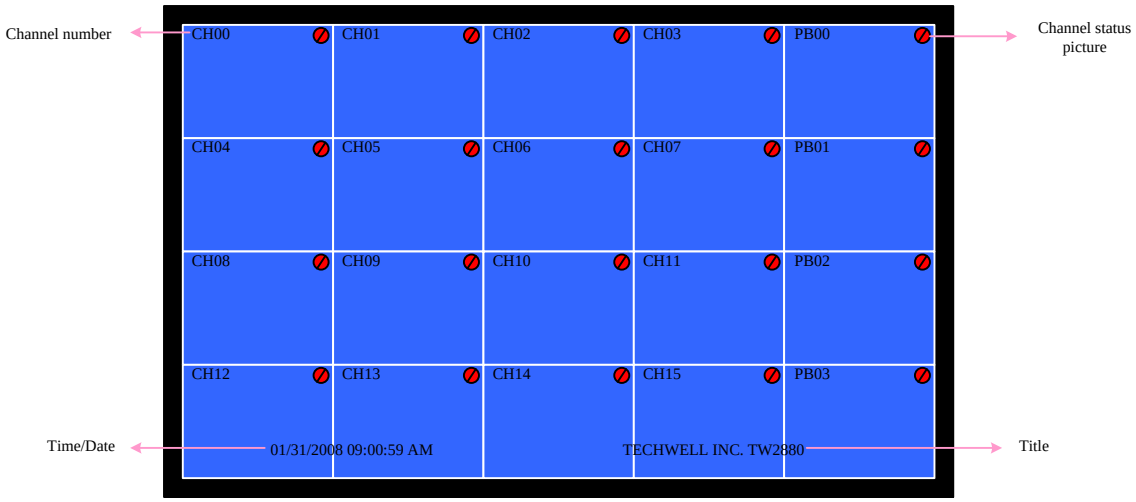
# OSD for Display Path

## Introduction

**TW2880 OSD controller for display path is a complementary to OSG. Since OSG need a lot of SDRAM bandwidth, this simple OSD controller does not need SDRAM bandwidth. All font tables are stored in local SRAM. It can display channel information for each channel. Channel information includes 8 characters and 1 picture. All information can be selected from 64 fonts and 4 pictures saved in SRAM. Font width can be selected from 6, 8, 12 or 16. Font height can be selected from 8, 10, 16 or 20. Picture size is fixed to 32x32. OSD can also display time and data with 32 characters, and title with 32 characters.**

## Features

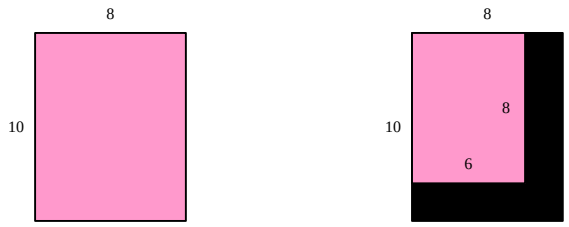
- Support 16 live channels and 16 playback channels
- 64 fonts table
- 4 pictures table
- 8 characters Channel information with one picture
- 32 characters title
- 32 characters time/date
- 2 bit color for font and picture




## Font and Picture Library

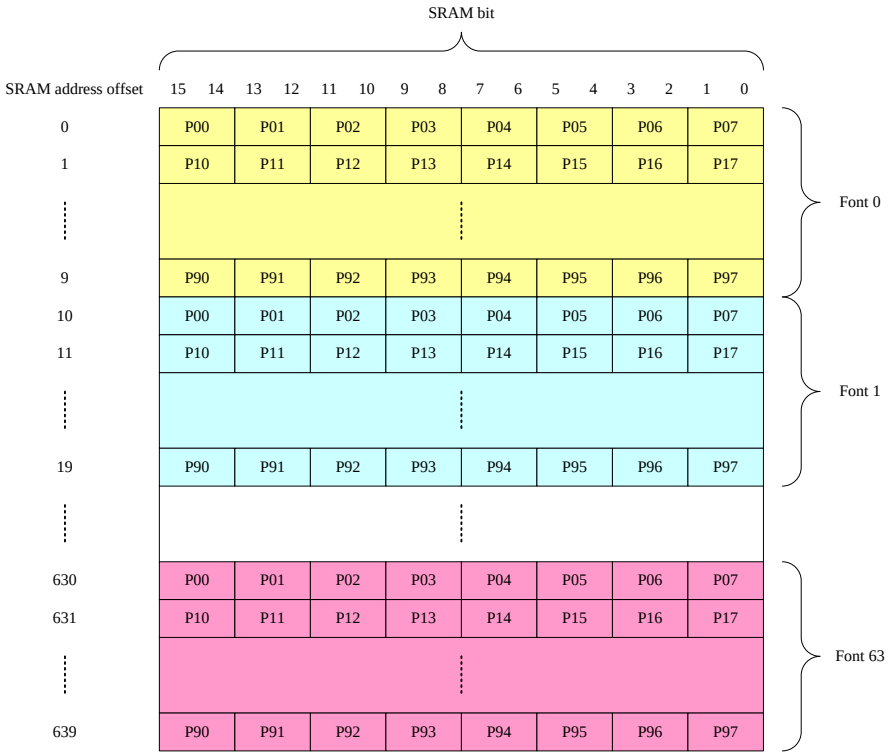
### FONT

**There are totally 64 fonts which can be saved in SRAM. The font size saved in SRAM is fixed to 8x10. But displayed font size can be changed. Horizontal can display four sizes: 6, 8, 12 or 16. Size 12 or 16 are doubled from size 6 or 8. If 6 or 12 are selected, fonts saved in SRAM must have small size. But additional two pixels must be saved in SRAM. Vertical can display four sizes: 8, 10, 16 or 20. Size 16 or 20 are doubled from size 8 or 10. If 8 or 16 are selected, fonts saved in SRAM must have small size. But additional two lines must be saved in SRAM. The following picture shows SRAM data. For 8x10 font, all SRAM data are valid. For 6x8 font, only 6x8 area are valid. Black area is for dummy data.**




**There are 2 bits for each pixel color. 00 means transparent, color 01, 10 and 11 can be set by registers: OSD_FONT_R1 (G1, B1), OSD_FONT_R2 (G2, B2) and OSD_FONT_R3 (G3, B3).**

**Data saved in SRAM is shown below. There are totally 64x8x10*2=640x16 bits in SRAM. Data are saved font by font. For each font, data is saved line by line. Pixel data in each line is in big endian.**



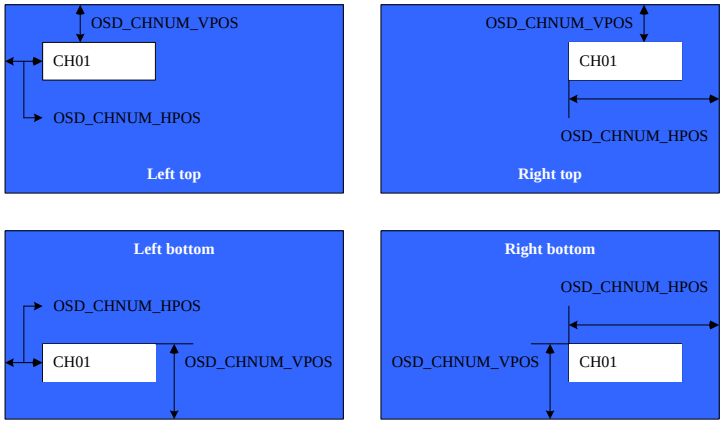

## PICTURE

**The picture is used for channel status. Each channel has picture display. The picture size in SRAM is fixed to 32x32. The display picture size is also fixed to 32x32. Same as font, it uses 2 bits for picture color. So there are four colors which can be set by registers: OSD_PIC_R0 (G0, B0), OSD_PIC_R1 (G1, B1), OSD_PIC_R2 (G2, B2) and OSD_PIC_R3 (G3, B3). There are four pictures saved in SRAM. Data in SRAM are same as fonts. Data are saved picture by picture. And in one picture, data are line by line. In one byte data, pixel data is stored in big endian. There are totally 32x32x4x2 = 512x16 bits in SRAM.**

# Display Information

### CHANNEL NUMBER

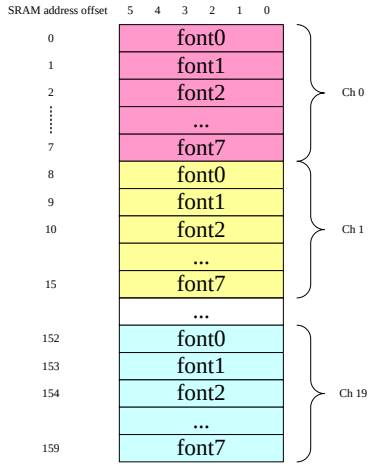
**For each channel, there is a 8-font channel number information. Each font is selected from 64-font table. So index for each font is 6 bit. Channel number can be enabled by setting register OSD_CHNUM_EN and OSD_EN to high. Channel number can be mixed with video data by setting register OSD_CHNUM_MIX to high. Mix percentage is 50% video plus 50% channel number. The positions for each channel are same. They can be in four positions: left top, right top, left bottom and right bottom. It is set by register OSD_CHNUM_POS. For each position, horizontal offset and vertical offset can be set by OSD_CHNUM_HPOS and OSD_CHNUM_VPOS. For each position, the meaning for HPOS and VPOS is different because each channel size may be different.**




**If channel number information has less than 8 fonts, you can set the remaining font to space. So you need to put space font in the 64-font table.**

**The font size can be changed according to register OSD_FONT_HSIZE and OSD_FONT_VSIZE. But remember the fonts saved in memory are always 8x10. If double size is selected, just repeat every pixel twice.**

**Channel information for display is saved in display SRAM. It contains 32x8x6=256x6. The sequence is channel by channel. In each channel, the sequence is font by font. SRAM is shown below:**




### CHANNEL PICTURE

**For each channel, there is a picture which can show channel status such as no camera. It can be enabled by setting register OSD_CHPIC_EN and OSD_EN to high. The picture can also be mixed with video by setting register OSD_CHPIC_MIX. The mix method is 50% video and 50% picture. If you want to make some pixels transparent, you can set OSD_CHPIC_TRANS to high. The pixels in color index "00" will be transparent. There is a blink option. If**

**OSD_CHPIC_BLINK is high, picture index 00 and 01 will not blink, and picture index "10" and "11" will blink. So if you put "no video" picture in picture index "10" or "11", these pictures will blink every 32 frames.**

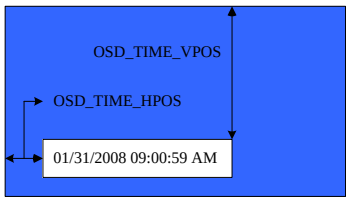
**Picture index information for each channel will be saved in SRAM. It needs totally 32x2=64 bits. But for design easily, we use 32x6 bits. Each 6 bit is for each channel, but only LSB 2 bits are used.**

**Picture position can also be programmed same as channel number. The register setting is shown below.**



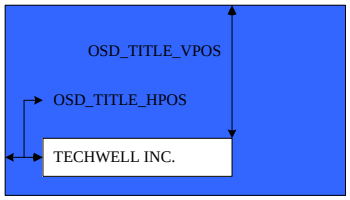

### DATE AND TIME

**Date and time are only display once on whole screen, not channel by channel. There are totally 32 fonts can be displayed including space. Like other display information, the position for date and time can be programmed. And it can also be disabled and be mixed with video. The font index is saved in display SRAM. It needs 32x6 bits.**




### TITLE

**Title is same as date and time. It has 32 fonts. It needs 32x6 SRAM size.**

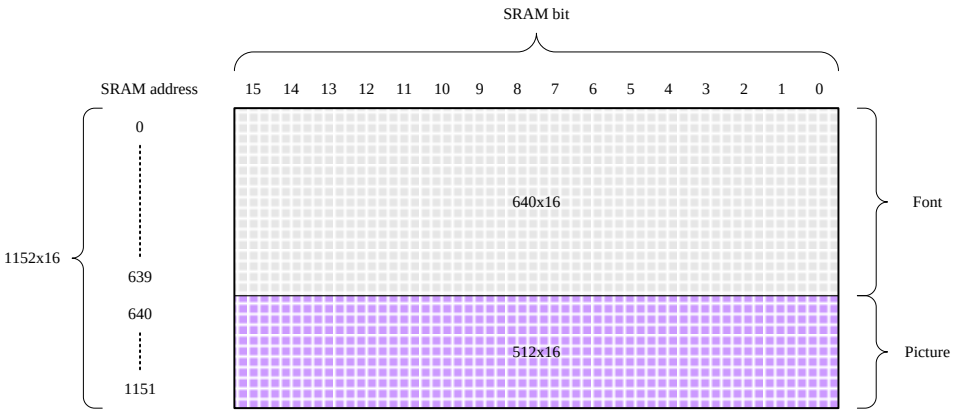



## SRAM Allocation

**There are two SRAM blocks in this design. One is for font and picture. The other is for display information.**

### FONT AND PICTURE SRAM

**Font and picture are saved in one SRAM block. Font needs 640x16 bits, and picture needs 512x16 bits. So the total SRAM size is 1152x16 bits.**




**During system initialization, host need to write font and picture data in this SRAM. The write sequence is:**

**1. OSD_FRAM_ADDR[7:0]**

**2. OSD_FRAM_ADDR[10:0]**

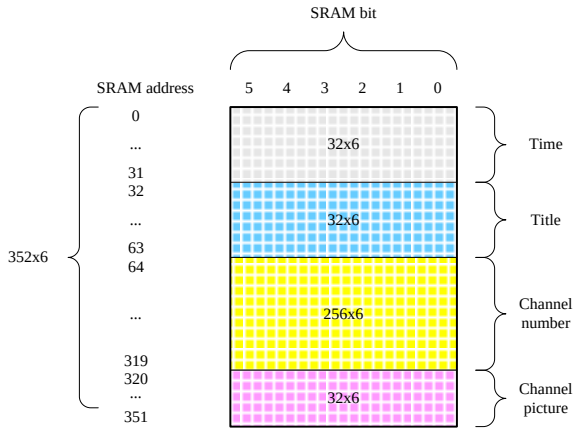
**3. OSD_FRAM_DATA[7:0]**

**4. OSD_FRAM_DATA[15:8]**

**OSD_FRAM_DATA[15:8] must be the last one.**

**DISPLAY SRAM**

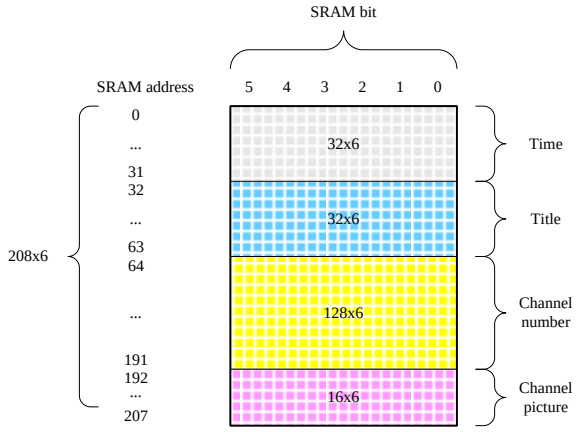
**Display SRAM includes channel number, channel picture, date/time and title information. Channel number needs 32x8x6=256 bits, channel picture needs 32x6 bits, date/time needs 32x6 bits and title needs 32x6 bits. So the total SRAM size is 352x6 bits.**




**During display, host need to write index data in this SRAM. The write sequence is:**

**Set OSD_DRAM_ADDR, and then set OSD_DRAM_DATA.**

**In SPOT simple OSD, display SRAM needs 208x6bits:**



## Registers Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x180 | R/W | 0 | [4]: OSD_TITLE_EN<br>[3]: OSD_TIME_EN<br>[2]: OSD_CHPIC_EN<br>[1]: OSD_CHNUM_EN<br>[0]: OSD_EN |
| 0x181 | R/W | 0 | [5]: OSD_CHPIC_BLINK<br>[4]: OSD_CHPIC_TRANS<br>[3]: OSD_TITLE_MIX<br>[2]: OSD_TIME_MIX<br>[1]: OSD_CHPIC_MIX<br>[0]: OSD_CHNUM_MIX |
| 0x182 | R/W | 0x4F | [7:6]: OSD_CHPIC_POS<br>[5:4]: OSD_CHNUM_POS<br>[3:2]: OSD_FONT_VSIZE<br>[1:0]: OSD_FONT_HSIZE |
| 0x183 | R/W | 0 | OSD_CHNUM_HPOS[7:0] |
| 0x184 | R/W | 0 | OSD_CHNUM_HPOS[10:8] |
| 0x185 | R/W | 0 | OSD_CHNUM_VPOS[7:0] |
| 0x186 | R/W | 0 | OSD_CHNUM_VPOS[10:8] |
| 0x187 | R/W | 0 | OSD_CHPIC_HPOS[7:0] |
| 0x188 | R/W | 0 | OSD_CHPIC_HPOS[10:8] |
| 0x189 | R/W | 0 | OSD_CHPIC_VPOS[7:0] |
| 0x18A | R/W | 0 | OSD_CHPIC_VPOS[10:8] |
| 0x18B | R/W | 0 | OSD_TIME_HPOS[7:0] |
| 0x18C | R/W | 0 | OSD_TIME_HPOS[10:8] |
| 0x18D | R/W | 0 | OSD_TIME_VPOS[7:0] |
| 0x18E | R/W | 0 | OSD_TIME_VPOS[10:8] |
| 0x18F | R/W | 0 | OSD_TITLE_HPOS[7:0] |
| 0x190 | R/W | 0 | OSD_TITLE_HPOS[10:8] |
| 0x191 | R/W | 0 | OSD_TITLE_VPOS[7:0] |
| 0x192 | R/W | 0 | OSD_TITLE_VPOS[10:8] |
| 0x193 | R/W | 0 | OSD_FONT_R1[7:0] |
| 0x194 | R/W | 0 | OSD_FONT_G1[7:0] |
| 0x195 | R/W | 0 | OSD_FONT_B1[7:0] |
| 0x196 | R/W | 0 | OSD_FONT_R2[7:0] |
| 0x197 | R/W | 0 | OSD_FONT_G2[7:0] |
| 0x198 | R/W | 0 | OSD_FONT_B2[7:0] |
| 0x199 | R/W | 0 | OSD_FONT_R3[7:0] |
| 0x19A | R/W | 0 | OSD_FONT_G3[7:0] |
| 0x19B | R/W | 0 | OSD_FONT_B3[7:0] |
| 0x19C | R/W | 0 | OSD_PIC_R0[7:0] |
| 0x19D | R/W | 0 | OSD_PIC_G0[7:0] |
| 0x19E | R/W | 0 | OSD_PIC_B0[7:0] |
| 0x19F | R/W | 0 | OSD_PIC_R1[7:0] |
| 0x1A0 | R/W | 0 | OSD_PIC_G1[7:0] |
| 0x1A1 | R/W | 0 | OSD_PIC_B1[7:0] |
| 0x1A2 | R/W | 0 | OSD_PIC_R2[7:0] |
| 0x1A3 | R/W | 0 | OSD_PIC_G2[7:0] |
| 0x1A4 | R/W | 0 | OSD_PIC_B2[7:0] |
| 0x1A5 | R/W | 0 | OSD_PIC_R3[7:0] |
| 0x1A6 | R/W | 0 | OSD_PIC_G3[7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x1A7 | R/W | 0 | OSD_PIC_B3[7:0] |
| 0x1A8 | W | 0 | OSD_FRAM_ADDR[7:0] |
| 0x1A9 | W | 0 | OSD_FRAM_ADDR[10:8] |
| 0x1AA | W | 0 | OSD_FRAM_DATA[7:0] |
| 0x1AB | W | 0 | OSD_FRAM_DATA[15:8] |
| 0x1AC | W | 0 | OSD_DRAM_ADDR[7:0] |
| 0x1AD | W | 0 | OSD_DRAM_ADDR[8] |
| 0x1AE | W | 0 | OSD_DRAM_DATA[5:0] |

# Registers Description

### OSD CONTROL REGISTER – 0X180

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4 | RW | 0 | **OSD_TITLE_EN**<br><br>Display title enable<br>1: enable<br>0: disable |
| 3 | RW | 0 | **OSD_TIME_EN**<br><br>Display time and date enable<br>1: enable<br>0: disable |
| 2 | RW | 0 | **OSD_CHPIC_EN**<br><br>Display picture for each channel<br>1: enable<br>0: disable |
| 1 | RW | 0 | **OSD_CHNUM_EN**<br><br>Display channel number for each channel<br>1: enable<br>0: disable |
| 0 | RW | 0 | **OSD_EN**<br><br>All OSD information enable or disable. If set to 1, each information is enabled by bit [4:1]. If set to 0, all OSD is disabled.<br>1: enable<br>0: disable |

### OSD MODE REGISTER – 0X181

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5 | RW | 0 | **OSD_CHPIC_BLINK**<br><br>If set to 1, picture for each channel blink in picture index 2 and 3. Picture 0 and 1 will not blink<br>1: enable<br>0: disable |
| 4 | RW | 0 | **OSD_CHPIC_TRANS**<br><br>If set to 1, picture in color index “00” will be transparent<br>1: enable<br>0: disable |
| 3 | RW | 0 | **OSD_TITLE_MIX**<br><br>Display title mix enable bit. If set to 1, title will be 50% blending with video<br>1: enable<br>0: disable |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2 | RW | 0 | **OSD_TIME_MIX**<br><br>Display time and date mix enable bit. If set to 1, time/date will be 50% blending with video<br>1: enable<br>0: disable |
| 1 | RW | 0 | **OSD_CHPIC_MIX**<br><br>Display picture mix enable bit. If set to 1, picture will be 50% blending with video<br>1: enable<br>0: disable |
| 0 | RW | 0 | **OSD_CHNUM_MIX**<br><br>Display channel number information mix enable bit. If set to 1, channel information will be 50% blending with video<br>1: enable<br>0: disable |

### OSD FONT SIZE REGISTER – 0X182

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0x1 | **OSD_CHPIC_POS:** Channel picture corner position.<br><br>00: left top<br>01: right top<br>10: left bottom<br>11: right bottom |
| 5:4 | RW | 0 | **OSD_CHNUM_POS:** Channel number corner position.<br><br>00: left top<br>01: right top<br>10: left bottom<br>11: right bottom |
| 3:2 | RW | 0x3 | **OSD_FONT_VSIZE:** Font vertical size for display. There are four choices.<br><br>00: 8<br>01: 10<br>10: 16, scale up from 8<br>11: 20, scale up from 10 |
| 1:0 | RW | 0x3 | **OSD_FONT_HSIZE:** Font horizontal size for display. There are four choices.<br><br>00: 6<br>01: 8<br>10: 12, scale up from 6<br>11: 16, scale up from 8 |

### OSD CHANNEL NUMBER HORIZONTAL POSITION LOW BYTE REGISTER – 0X183

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_CHNUM_HPOS[7:0]<br><br>Channel number information horizontal position offset to each channel horizontal start position. It is one pixel unit. |

### OSD CHANNEL NUMBER HORIZONTAL POSITION HIGH BYTE REGISTER – 0X184

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_CHNUM_HPOS[10:8]<br><br>Channel number information horizontal position offset to each channel horizontal start position. It is one pixel unit. |

### OSD CHANNEL NUMBER VERTICAL POSITION LOW BYTE REGISTER – 0X185

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_CHNUM_VPOS[7:0]<br><br>Channel number information vertical position offset to each channel vertical start position. It is one pixel unit. |

### OSD CHANNEL NUMBER HORIZONTAL POSITION HIGH BYTE REGISTER – 0X186

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_CHNUM_VPOS[10:8]<br><br>Channel number information vertical position offset to each channel vertical start position. It is one pixel unit. |

### OSD CHANNEL PICTURE HORIZONTAL POSITION LOW BYTE REGISTER – 0X187

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_CHPIC_HPOS[7:0]<br><br>Channel picture information horizontal position offset to each channel horizontal start position. It is one pixel unit. |

### OSD CHANNEL PICTURE HORIZONTAL POSITION HIGH BYTE REGISTER – 0X188

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_CHPIC_HPOS[10:8]<br><br>Channel picture information horizontal position offset to each channel horizontal start position. It is one pixel unit. |

### OSD CHANNEL PICTURE VERTICAL POSITION LOW BYTE REGISTER – 0X189

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_CHPIC_VPOS[7:0]<br><br>Channel picture information vertical position offset to each channel vertical start position. It is one pixel unit. |

### OSD CHANNEL PICTURE HORIZONTAL POSITION HIGH BYTE REGISTER – 0X18A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_CHPIC_VPOS[10:8]<br><br>Channel number information vertical position offset to each channel vertical start position. It is one pixel unit. |

### OSD TIME AND DATE HORIZONTAL POSITION LOW BYTE REGISTER – 0X18B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_TIME_HPOS[7:0]<br><br>Time and Date information horizontal position. It is one pixel unit. |

### OSD TIME AND DATE HORIZONTAL POSITION HIGH BYTE REGISTER – 0X18C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_TIME_HPOS[10:8]<br><br>Time and Date information horizontal position. It is one pixel unit. |

### OSD TIME AND DATE VERTICAL POSITION LOW BYTE REGISTER – 0X18D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_TIME_VPOS[7:0]<br><br>Time and Date information vertical position. It is one pixel unit. |

### OSD TIME AND DATE HORIZONTAL POSITION HIGH BYTE REGISTER – 0X18E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_TIME_VPOS[10:8]<br><br>Time and Date information vertical position. It is one pixel unit. |

### OSD TITLE HORIZONTAL POSITION LOW BYTE REGISTER – 0X18F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_TITLE_HPOS[7:0]<br><br>Title information horizontal position. It is one pixel unit. |

### OSD TITLE HORIZONTAL POSITION HIGH BYTE REGISTER – 0X190

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_TITLE_HPOS[10:8]<br><br>Title information horizontal position. It is one pixel unit. |

### OSD TITLE VERTICAL POSITION LOW BYTE REGISTER – 0X191

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_TITLE_VPOS[7:0]<br><br>Title information vertical position. It is one pixel unit. |

### OSD TITLE HORIZONTAL POSITION HIGH BYTE REGISTER – 0X192

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_TITLE_VPOS[10:8]<br><br>Title information vertical position. It is one pixel unit. |

### OSD FONT RED COLOR 1 REGISTER – 0X193

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_R1[7:0]<br><br>Font red color for color index 1. |

### OSD FONT GREEN COLOR 1 REGISTER – 0X194

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_G1[7:0]<br><br>Font green color for color index 1. |

### OSD FONT BLUE COLOR 1 REGISTER – 0X195

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_B1[7:0]<br><br>Font blue color for color index 1. |

### OSD FONT RED COLOR 2 REGISTER – 0X196

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_R2[7:0]<br><br>Font red color for color index 2. |

### OSD FONT GREEN COLOR 2 REGISTER – 0X197

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_G2[7:0]<br><br>Font green color for color index 2. |

### OSD FONT BLUE COLOR 2 REGISTER – 0X198

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_B2[7:0]<br><br>Font blue color for color index 2. |

### OSD FONT RED COLOR 3 REGISTER – 0X199

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_R3[7:0]<br><br>Font red color for color index 3. |

### OSD FONT GREEN COLOR 3 REGISTER – 0X19A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_G3[7:0]<br><br>Font green color for color index 3. |

### OSD FONT BLUE COLOR 3 REGISTER – 0X19B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_B3[7:0]<br><br>Font blue color for color index 3. |

### OSD PICTURE RED COLOR 0 REGISTER – 0X19C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_R0[7:0]<br><br>Picture red color for color index 0. |

### OSD PICTURE GREEN COLOR 0 REGISTER – 0X19D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_G0[7:0]<br><br>Picture green color for color index 0. |

### OSD PICTURE BLUE COLOR 0 REGISTER – 0X19E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_B0[7:0]<br><br>Picture blue color for color index 0. |

### OSD PICTURE RED COLOR 1 REGISTER – 0X19F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_R1[7:0]<br><br>Picture red color for color index 1. |

### OSD PICTURE GREEN COLOR 1 REGISTER – 0X1A0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_G1[7:0]<br><br>Picture green color for color index 1. |

### OSD PICTURE BLUE COLOR 1 REGISTER – 0X1A1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_B1[7:0]<br><br>Picture blue color for color index 1. |

### OSD PICTURE RED COLOR 2 REGISTER – 0X1A2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_R2[7:0]<br><br>Picture red color for color index 2. |

### OSD PICTURE GREEN COLOR 2 REGISTER – 0X1A3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_G2[7:0]<br><br>Picture green color for color index 2. |

### OSD PICTURE BLUE COLOR 2 REGISTER – 0X1A4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_B2[7:0]<br><br>Picture blue color for color index 2. |

### OSD PICTURE RED COLOR 3 REGISTER – 0X1A5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_R3[7:0]<br><br>Picture red color for color index 3. |

### OSD PICTURE GREEN COLOR 3 REGISTER – 0X1A6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_G3[7:0]<br><br>Picture green color for color index 3. |

### OSD PICTURE BLUE COLOR 3 REGISTER – 0X1A7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_B3[7:0]<br><br>Picture blue color for color index 3. |

### OSD FONT AND PICTURE RAM ADDRESS LOW BYTE REGISTER – 0X1A8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FRAM_ADDR[7:0]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### OSD FONT AND PICTURE RAM ADDRESS HIGH BYTE REGISTER – 0X1A9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_FRAM_ADDR[10:8]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### OSD FONT AND PICTURE RAM DATA LOW BYTE REGISTER – 0X1AA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FRAM_DATA[7:0]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### OSD FONT AND PICTURE RAM DATA HIGH BYTE REGISTER – 0X1AB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FRAM_DATA[15:8]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### OSD DISPLAY RAM ADDRESS REGISTER – 0X1AC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_DRAM_ADDR[7:0]<br><br>Display SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 244x6. Channel number size is 160x6, time/date size is 32x6, title size is 32x6 and picture index size is 20x6. |

### OSD DISPLAY RAM ADDRESS REGISTER – 0X1AD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_DRAM_ADDR[8]<br><br>Display SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 244x6. Channel number size is 160x6, time/date size is 32x6, title size is 32x6 and picture index size is 20x6. |

### OSD DISPLAY RAM DATA REGISTER – 0X1AE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_DRAM_Data[5:0]<br><br>Display SRAM Data. When host write data to this SRAM, address is written first and then data. SRAM size is 244x6. Channel number size is 160x6, time/date size is 32x6, title size is 32x6 and picture index size is 20x6. |

# Registers Description

### OSD CONTROL REGISTER – 0X180

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4 | RW | 0 | **OSD_TITLE_EN**<br><br>Display title enable<br>1: enable<br>0: disable |
| 3 | RW | 0 | **OSD_TIME_EN**<br><br>Display time and date enable<br>1: enable<br>0: disable |
| 2 | RW | 0 | **OSD_CHPIC_EN**<br><br>Display picture for each channel<br>1: enable<br>0: disable |
| 1 | RW | 0 | **OSD_CHNUM_EN**<br><br>Display channel number for each channel<br>1: enable<br>0: disable |
| 0 | RW | 0 | **OSD_EN**<br><br>All OSD information enable or disable. If set to 1, each information is enabled by bit [4:1]. If set to 0, all OSD is disabled.<br>1: enable<br>0: disable |

### OSD MODE REGISTER – 0X181

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5 | RW | 0 | **OSD_CHPIC_BLINK**<br><br>If set to 1, picture for each channel blink in picture index 2 and 3. Picture 0 and 1 will not blink<br>1: enable<br>0: disable |
| 4 | RW | 0 | **OSD_CHPIC_TRANS**<br><br>If set to 1, picture in color index “00” will be transparent<br>1: enable<br>0: disable |
| 3 | RW | 0 | **OSD_TITLE_MIX**<br><br>Display title mix enable bit. If set to 1, title will be 50% blending with video<br>1: enable<br>0: disable |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2 | RW | 0 | **OSD_TIME_MIX**<br><br>Display time and date mix enable bit. If set to 1, time/date will be 50% blending with video<br>1: enable<br>0: disable |
| 1 | RW | 0 | **OSD_CHPIC_MIX**<br><br>Display picture mix enable bit. If set to 1, picture will be 50% blending with video<br>1: enable<br>0: disable |
| 0 | RW | 0 | **OSD_CHNUM_MIX**<br><br>Display channel number information mix enable bit. If set to 1, channel information will be 50% blending with video<br>1: enable<br>0: disable |

**OSD FONT SIZE REGISTER – 0X182**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0x1 | **OSD_CHPIC_POS**<br><br>Channel picture corner position.<br>00: left top<br>01: right top<br>10: left bottom<br>11: right bottom |
| 5:4 | RW | 0 | **OSD_CHNUM_POS**<br><br>Channel number corner position.<br>00: left top<br>01: right top<br>10: left bottom<br>11: right bottom |
| 3:2 | RW | 0x3 | **OSD_FONT_VSIZE**<br><br>Font vertical size for display. There are four choices.<br>00: 8<br>01: 10<br>10: 16, scale up from 8<br>11: 20, scale up from 10 |
| 1:0 | RW | 0x3 | **OSD_FONT_HSIZE**<br><br>Font horizontal size for display. There are four choices.<br>00: 6<br>01: 8<br>10: 12, scale up from 6<br>11: 16, scale up from 8 |

### OSD CHANNEL NUMBER HORIZONTAL POSITION LOW BYTE REGISTER – 0X183

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_CHNUM_HPOS[7:0]<br><br>Channel number information horizontal position offset to each channel horizontal start position. It is one pixel unit. |

### OSD CHANNEL NUMBER HORIZONTAL POSITION HIGH BYTE REGISTER – 0X184

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_CHNUM_HPOS[10:8]<br><br>Channel number information horizontal position offset to each channel horizontal start position. It is one pixel unit. |

### OSD CHANNEL NUMBER VERTICAL POSITION LOW BYTE REGISTER – 0X185

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_CHNUM_VPOS[7:0]<br><br>Channel number information vertical position offset to each channel vertical start position. It is one pixel unit. |

### OSD CHANNEL NUMBER HORIZONTAL POSITION HIGH BYTE REGISTER – 0X186

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_CHNUM_VPOS[10:8]<br><br>Channel number information vertical position offset to each channel vertical start position. It is one pixel unit. |

### OSD CHANNEL PICTURE HORIZONTAL POSITION LOW BYTE REGISTER – 0X187

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_CHPIC_HPOS[7:0]<br><br>Channel picture information horizontal position offset to each channel horizontal start position. It is one pixel unit. |

### OSD CHANNEL PICTURE HORIZONTAL POSITION HIGH BYTE REGISTER – 0X188

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_CHPIC_HPOS[10:8]<br><br>Channel picture information horizontal position offset to each channel horizontal start position. It is one pixel unit. |

### OSD CHANNEL PICTURE VERTICAL POSITION LOW BYTE REGISTER – 0X189

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_CHPIC_VPOS[7:0]<br><br>Channel picture information vertical position offset to each channel vertical start position. It is one pixel unit. |

### OSD CHANNEL PICTURE HORIZONTAL POSITION HIGH BYTE REGISTER – 0X18A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_CHPIC_VPOS[10:8]<br><br>Channel number information vertical position offset to each channel vertical start position. It is one pixel unit. |

### OSD TIME AND DATE HORIZONTAL POSITION LOW BYTE REGISTER – 0X18B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_TIME_HPOS[7:0]<br><br>Time and Date information horizontal position. It is one pixel unit. |

### OSD TIME AND DATE HORIZONTAL POSITION HIGH BYTE REGISTER – 0X18C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_TIME_HPOS[10:8]<br><br>Time and Date information horizontal position. It is one pixel unit. |

### OSD TIME AND DATE VERTICAL POSITION LOW BYTE REGISTER – 0X18D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_TIME_VPOS[7:0]<br><br>Time and Date information vertical position. It is one pixel unit. |

### OSD TIME AND DATE HORIZONTAL POSITION HIGH BYTE REGISTER – 0X18E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_TIME_VPOS[10:8]<br><br>Time and Date information vertical position. It is one pixel unit. |

### OSD TITLE HORIZONTAL POSITION LOW BYTE REGISTER – 0X18F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_TITLE_HPOS[7:0]<br><br>Title information horizontal position. It is one pixel unit. |

### OSD TITLE HORIZONTAL POSITION HIGH BYTE REGISTER – 0X190

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_TITLE_HPOS[10:8]<br><br>Title information horizontal position. It is one pixel unit. |

### OSD TITLE VERTICAL POSITION LOW BYTE REGISTER – 0X191

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_TITLE_VPOS[7:0]<br><br>Title information vertical position. It is one pixel unit. |

### OSD TITLE HORIZONTAL POSITION HIGH BYTE REGISTER – 0X192

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_TITLE_VPOS[10:8]<br><br>Title information vertical position. It is one pixel unit. |

### OSD FONT RED COLOR 1 REGISTER – 0X193

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_R1[7:0]<br><br>Font red color for color index 1. |

### OSD FONT GREEN COLOR 1 REGISTER – 0X194

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_G1[7:0]<br><br>Font green color for color index 1. |

### OSD FONT BLUE COLOR 1 REGISTER – 0X195

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_B1[7:0]<br><br>Font blue color for color index 1. |

### OSD FONT RED COLOR 2 REGISTER – 0X196

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_R2[7:0]<br><br>Font red color for color index 2. |

### OSD FONT GREEN COLOR 2 REGISTER – 0X197

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_G2[7:0]<br><br>Font green color for color index 2. |

### OSD FONT BLUE COLOR 2 REGISTER – 0X198

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_B2[7:0]<br><br>Font blue color for color index 2. |

### OSD FONT RED COLOR 3 REGISTER – 0X199

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_R3[7:0]<br><br>Font red color for color index 3. |

### OSD FONT GREEN COLOR 3 REGISTER – 0X19A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_G3[7:0]<br><br>Font green color for color index 3. |

### OSD FONT BLUE COLOR 3 REGISTER – 0X19B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FONT_B3[7:0]<br><br>Font blue color for color index 3. |

### OSD PICTURE RED COLOR 0 REGISTER – 0X19C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_R0[7:0]<br><br>Picture red color for color index 0. |

### OSD PICTURE GREEN COLOR 0 REGISTER – 0X19D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_G0[7:0]<br><br>Picture green color for color index 0. |

### OSD PICTURE BLUE COLOR 0 REGISTER – 0X19E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_B0[7:0]<br><br>Picture blue color for color index 0. |

### OSD PICTURE RED COLOR 1 REGISTER – 0X19F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_R1[7:0]<br><br>Picture red color for color index 1. |

### OSD PICTURE GREEN COLOR 1 REGISTER – 0X1A0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_G1[7:0]<br><br>Picture green color for color index 1. |

### OSD PICTURE BLUE COLOR 1 REGISTER – 0X1A1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_B1[7:0]<br><br>Picture blue color for color index 1. |

### OSD PICTURE RED COLOR 2 REGISTER – 0X1A2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_R2[7:0]<br><br>Picture red color for color index 2. |

### OSD PICTURE GREEN COLOR 2 REGISTER – 0X1A3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_G2[7:0]<br><br>Picture green color for color index 2. |

### OSD PICTURE BLUE COLOR 2 REGISTER – 0X1A4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_B2[7:0]<br><br>Picture blue color for color index 2. |

### OSD PICTURE RED COLOR 3 REGISTER – 0X1A5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_R3[7:0]<br><br>Picture red color for color index 3. |

### OSD PICTURE GREEN COLOR 3 REGISTER – 0X1A6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_G3[7:0]<br><br>Picture green color for color index 3. |

### OSD PICTURE BLUE COLOR 3 REGISTER – 0X1A7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_PIC_B3[7:0]<br><br>Picture blue color for color index 3. |

### OSD FONT AND PICTURE RAM ADDRESS LOW BYTE REGISTER – 0X1A8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FRAM_ADDR[7:0]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### OSD FONT AND PICTURE RAM ADDRESS HIGH BYTE REGISTER – 0X1A9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | OSD_FRAM_ADDR[10:8]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### OSD FONT AND PICTURE RAM DATA LOW BYTE REGISTER – 0X1AA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FRAM_DATA[7:0]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### OSD FONT AND PICTURE RAM DATA HIGH BYTE REGISTER – 0X1AB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_FRAM_DATA[15:8]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### OSD DISPLAY RAM ADDRESS REGISTER – 0X1AC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_DRAM_ADDR[7:0]<br><br>Display SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 352x6. Channel number size is 256x6, time/date size is 32x6, title size is 32x6 and picture index size is 32x6. |

### OSD DISPLAY RAM ADDRESS REGISTER – 0X1AD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | RW | 0 | OSD_DRAM_ADDR[8]<br><br>Display SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 352x6. Channel number size is 256x6, time/date size is 32x6, title size is 32x6 and picture index size is 32x6. |

### OSD DISPLAY RAM DATA REGISTER – 0X1AE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | OSD_DRAM_Data[5:0]<br><br>Display SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 352x6. Channel number size is 256x6, time/date size is 32x6, title size is 32x6 and picture index size is 32x6. |

# Dualview Display Controller Unit

## Introduction

**In addition to the main display controller, TW2880 has a secondary display controller which can support display devices with interlaced or progressive timing. Using the integrated TV encoder and the DAC, this DualView display controller can drive traditional TV with CVBS or a progressive LCD display with VGA socket. With properly software registers setup and display memory planning between the two display controller, the sixteen input channels and sixteen playback channels can be displayed in many different resolutions and combinations between the two different monitors. (TW2880C cannot fully support main display is at 60 Hz and Dualview controller is set at NTSC snadard, some channels may have tearing effect)**

**Because the bandwidth limitation form the display memory controller, the resolution and frame rate in this secondary controller is limited. The DualView display controller is able to support all NTSC and PAL standards for analog output, which can be composite video or S-Video for display.**

## Features

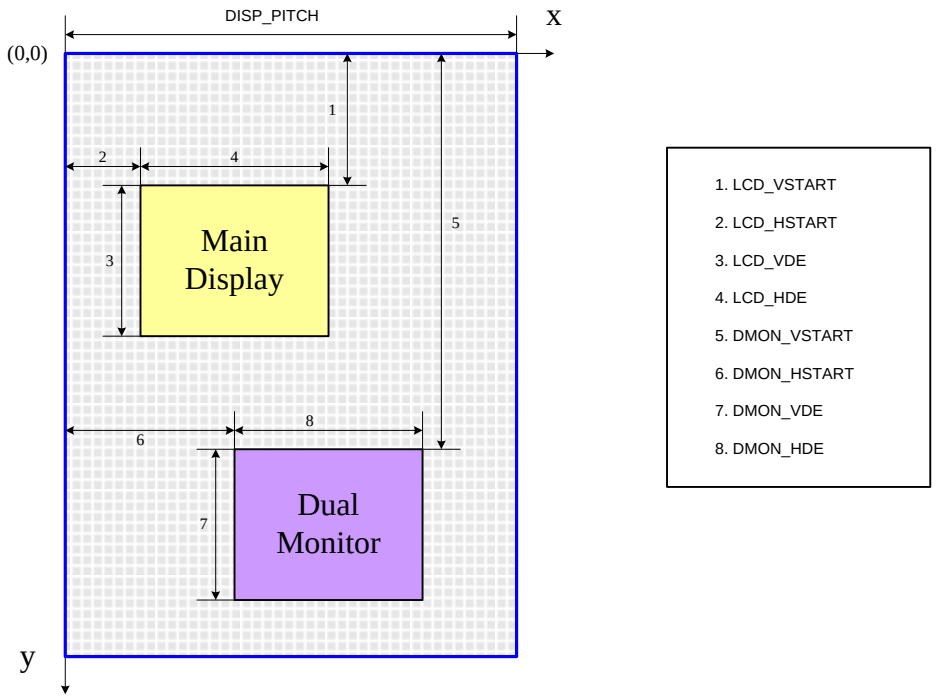
- Support NTSC / PAL standard TV monitor with integrated 10 bit DAC and TV encoder
- Support LCD monitor (up to 1080p)
- Down scaling from 1920x1080 resolution
- Support same or different video content with main display
- Two OSG layers, each layer has four sub-window
- Two OSG layers support different upscale ratio
- Support same or different OSG content with main display
- Two mouse layers on screen
- Support 16 mouse shape in SDRAM
- Four single boxes
- Support main display to CVBS output
- Support 16 live channels and 16 play back channels
- Support simple OSG for 32 channels

## Dual Monitor Controller Architecture

**For the controller to display the video channels on the main LCD screen and on the other CRT / LCD screen, the memory location need to be programmed for the controller to fetch data at the right area, Firmware needs to set the second monitor starting address, size, down scale factors, TV parameters control registers for the CRT monitor to display properly.**

### MEMORY DIAGRAM

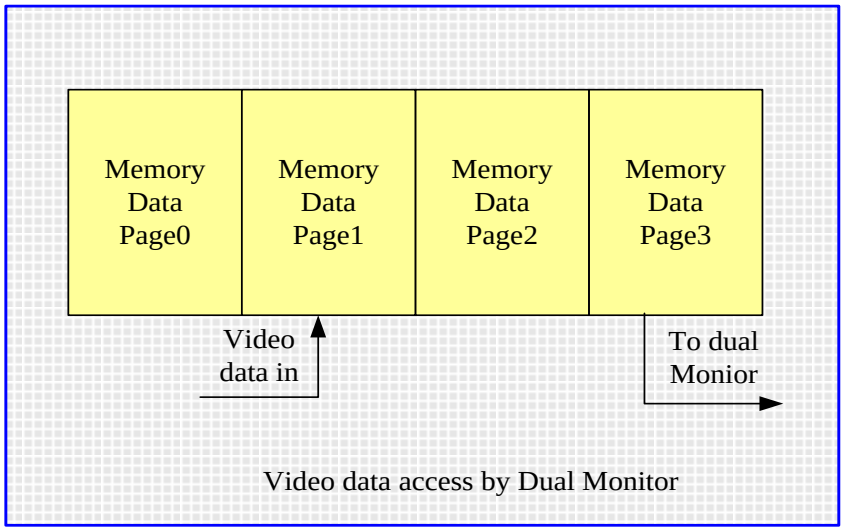
**The following picture shows the LCD and Dual monitor memory in SDRAM and some related registers. For the dual monitor start address, it can start anywhere by program the DM_HSTART and DM_VSTART address.**




Display Buffer

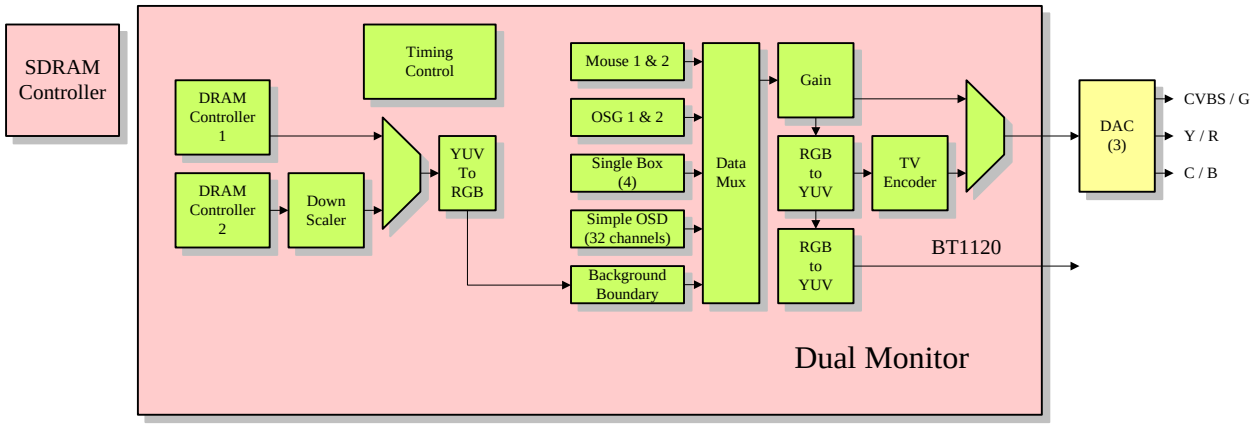
**FRAME DATA SYNCHONIZATION**

**To prevent the video tearing, the dual monitor controller takes use the frame rate synchronization circuit which will guarantee there are two frames difference between active display frame and active storing frame. If the frame rate of the Dual Monitor is higher than the incoming video pipe, frame repeating process is employed to make sure the two frame boundary is maintained.**




Video data access by Dual Monitor

## DUAL MONITOR CONTROLLER BLOCK DIAGRAM




## MEMORY INTERFACE

Based on the information of Horizontal and vertical timing, and under registers control, Memory Interface is a module uses to generate request to the DRAM controller and ask for the DRAM access to get the display data from the DRAM and send it the FIFO. There are two DRAM controller modules. One is for the case without down scaler. The other one is for the case with down scaler.

## FIFO INTERFACE

Due to the memory clock and video clock running in different frequency, there is a 128 deep and 64 bit wide FIFO to synchronize the data for cross clock domain operation.

## HV TIMING CONTROL

Base on the setting of horizontal and vertical registers and the inputs information of the TV Encoder, The HV timing control module generates all pixels, lines, and blanking signals and feed them to the Memory interface module to request data from DRAM.

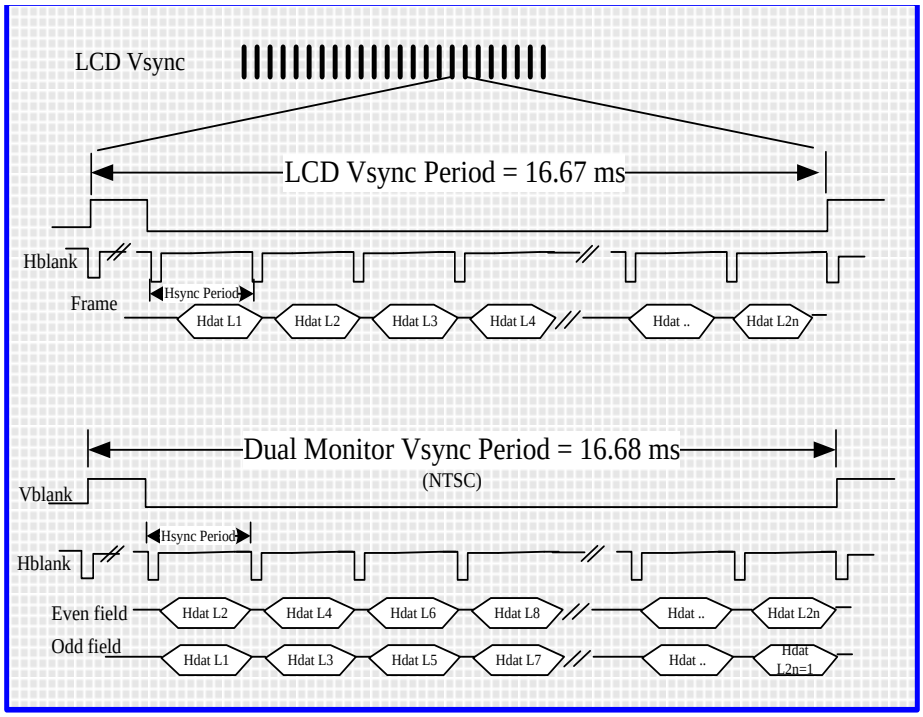
Depends on the Odd or Even field indicator output from the TV, The Dram Interface module will generate the memory address in that field and fetches data from the line memory, and send to the Down Scale controller and processes the image according the scale factor and send it to the Line Buffers.

Due to the different timing requirement of the LCD panel and the TV, once a while, we may encounter minor display delay between two displays as the calculation shown below.

The Field period in the dual monitor is 1/59.95 = 16.68 ms

The Frame period in the LCD display is 1/60 = 16.667 ms

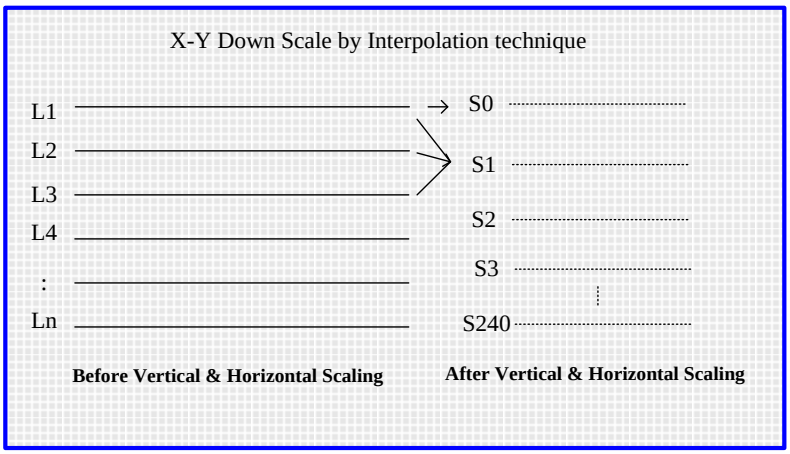
The different of the period is 0.05 mS that means every 20 seconds, there is a field data get from the same page in the memory and displays on the Dual Monitor.




**DOWN SCALAR**

**Down scalar is a module uses to scale down more channels on the display to fit the TV standard timing. So on the second monitor, we can pack as many channels as seen on the LCD and display them on a regular TV.**

**By using the linear interpolation technique, with independent setting of horizontal and vertical scale factors and others registers. We can scale the number of channels, the size and the location of the channels to the display's native resolution.**

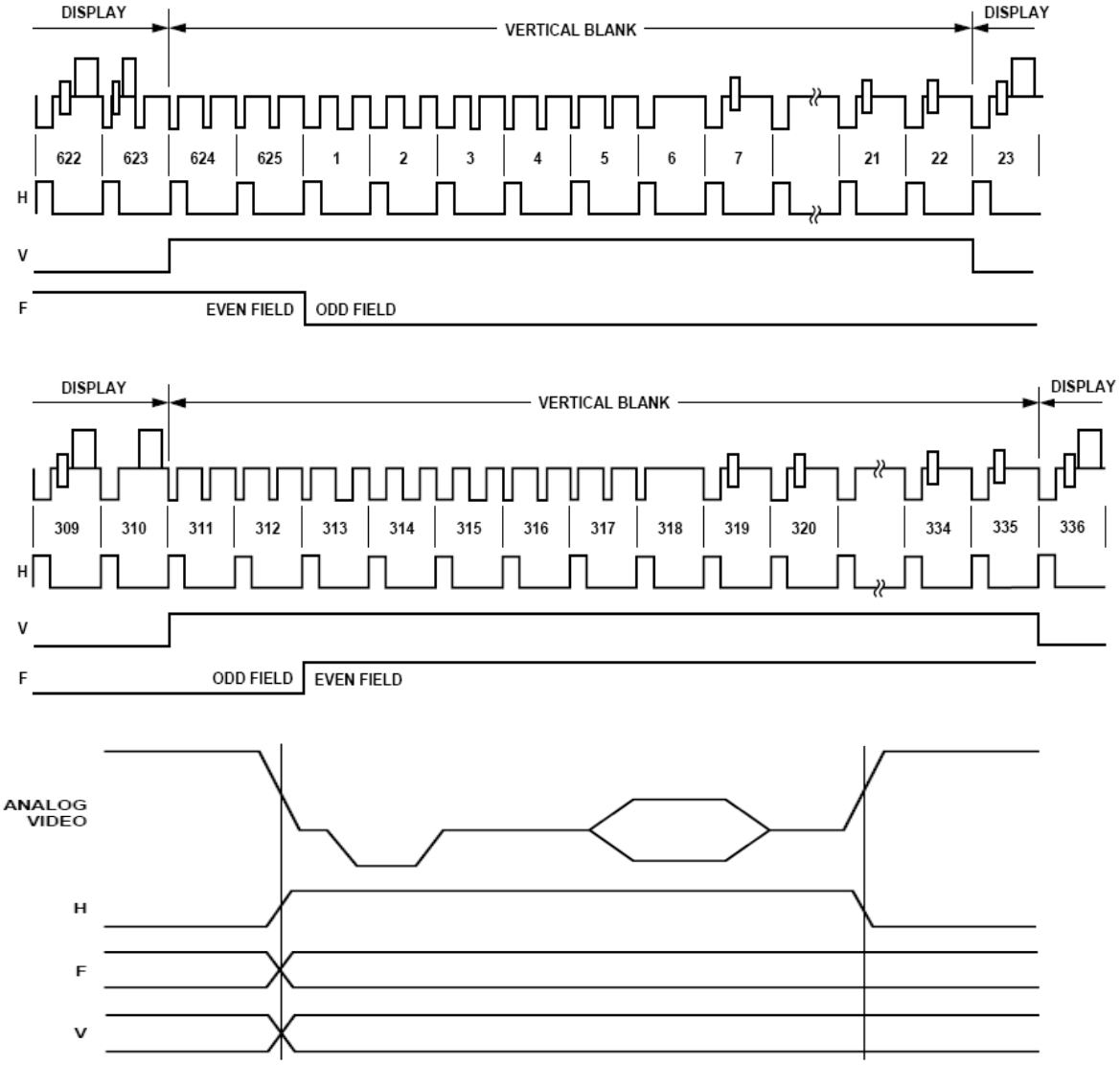



**LINE BUFFER CONTROL**

**Due to the clock rate difference of the Down Scalar and the TV Encoder, there are two line buffers size of 1920x16 to accommodate the timing of the TV Encoder.**

## TV ENCODER

**TV Encoder is the module converts all component data from scalar into a standard analog baseband television signal (CVBS) or S-Video signal which is compatible with worldwide standards. Follow is the PAL timing.**




## OSD CONTROL

**Dual monitor use the same architecture and design as the OSD in the LCD display path.**

**With all DM_OSD control registers in the Dual monitor page, Firmware can set up the DM_OSD display as the same source as the LCD Main Display Port if the Dual monitor displays the same channels.**

**If the LCD Main Display Port and Dual monitor show different channels on two displays, each OSD will display the channel correspondent to its own cameras. In other words, different channels will be mapped to different OSD source accordingly.**

**For all the operation and programming of the DM_OSD registers, please refer to the "OSD for Display Path" section beginning on page 299 where the detailed explanation and operation are fully documented.**

- 64 fonts table saved in SRAM
- Channel information table saved in SRAM
- Three lines channel information
- 32 characters date/time
- 16 characters channel title for each channel
- 16 characters channel status for each channel
- Font size can be changed in 6x8, 8x10, 12x16, 16x20

### CLOCK SETTING FOR DUAL MONITOR

**There are two clock for dual monitor need to be set correctly. One is dm_vclk, the other one is dm_fclk. "dm_vclk" is pixel clock. Dm_fclk is used for down scaler. When down scaler is used, dm_fclk must be bigger than dm_vclk. In worst case, dm_fclk = dm_vclk * horizontal down scaler ratio * vertical down scale ratio. But it also depends on horizontal blank time. If horizontal blank time is big, dm_fclk can be smaller.**

### THEORY OF OPERATION

**When the Dual monitor controller is enabling, the controller can access the whole memory map and display the content with the start address set by the HSTART and the VSTART registers. Of course, if the setting is out of bound, the display will show something on the screen with no meaning at all. So for the controller to operate properly, some registers need to set.**

**DM_HTT, DM_HDE, DM_VTT and DM_VDE are based on pixel clock.**

**In NTSC mode, dm_vclk must be set to 13.5MHz.**

**DM_HTT = 857**

**DM_HDE = 719**

**DM_VTT = 261**

**DM_VDE = 239**

**In 1280x1024 VGA mode, dm_vclk must be set to 108MHz.**

**DM_HTT = 1687**

**DM_HDE = 1279**

**DM_VTT = 1065**

**DM_VDE = 1023**

**In 1920x1080 VGA mode, dm_vclk must be set to 148.5MHz.**

**DM_HTT = 2199**

**DM_HDE = 1920**

**DM_VTT = 1124**

**DM_VDE = 1079**

**Dm_vclk can be selected from 13.5M, 27M, 54M and 108M, or it can be divided by faster clock from vpll.**

**TV ENCODER SETTING**

**In NTSC mode, user need to set:**

**[0x71A] = 0x00**

**[0x71B] = 0x01**

**[0x71C] = 0x08**

**In PAL mode, user need to set:**

**[0x71A] = 0x05**

**[0x71B] = 0x41**

**[0x71C] = 0x08**

## Register Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x700 | R/W | 0 | DM_HTT[7:0] |
| 0x701 | R/W | 0 | DM_HTT[12:8] |
| 0x702 | R/W | 0 | DM_VTT[7:0] |
| 0x703 | R/W | 0 | DM_VTT[10:8] |
| 0x704 | R/W | 0 | DM_HDE[7:0] |
| 0x705 | R/W | 0 | DM_HDE[10:8] |
| 0x706 | R/W | 0 | DM_VDE[7:0] |
| 0x707 | R/W | 0 | DM_VDE[10:8] |
| 0x708 | R/W | 0 | DM_PHSYNC[7:0] |
| 0x709 | R/W | 0 | DM_PHSYNC[8] |
| 0x70A | R/W | 0 | DM_PVSYNC[7:0] |
| 0x70B | R/W | 0 | DM_HSPW[7:0] |
| 0x70C | R/W | 0 | DM_VSPW[7:0] |
| 0x70D | R/W | 0 | [7]: MDBOX_EN<br>[6]: VTT_INV<br>[5]: DM_INT_CBAR<br>[4]: EVEN_REPEAT<br>[3:0]: VSYNC_DLY |
| 0x70E | R/W | 0 | TAR_W[7:0] |
| 0x70F | R/W | 0 | TAR_W[10:8] |
| 0x710 | R/W | 0 | TAR_H[7:0] |
| 0x711 | R/W | 0 | TAR_H[10:8] |
| 0x712 | R/W | 0 | [6]: IN_16_MODE<br>[5]: DM_YC_SWAP<br>[4]: DM_MAIN<br>[3]: DM_ENA<br>[2]: DM_LCD<br>[1]: DM_VSPOL<br>[0]: DM_HSPOL |
| 0x713 | R/W | 0 | DM_HSTART[7:0] |
| 0x714 | R/W | 0 | DM_VSTART[7:0] |
| 0x715 | R/W | 0 | [5:4]: DM_HSTART[9:8]<br>[3:0]: DM_VSTART[11:8] |
| 0x716 | R/W | 0 | SRC_W[7:0] |
| 0x717 | R/W | 0 | SRC_W[10:8] |
| 0x718 | R/W | 0 | SRC_H[7:0] |
| 0x719 | R/W | 0 | SRC_H[10:8] |
| 0x71A | R/W | 0 | [7:6]: TVENC_FSC_MODE<br>[5]: TVENC_SAT_LEVEL<br>[4:3]: TVENC_CCIR_IN_SEL<br>[2:1]: TVENC_IN_SEL<br>[0]: TVENC_PALNT |
| 0x71B | R/W | 0 | [7]: TVENC_VSPOL<br>[6]: TVENC_HSENCP<br>[5]: TVENC_VSDET<br>[4]: TVENC_PED<br>[3]: TVENC_PDRST<br>[2]: TVENC_PHALT<br>[1]: TVENC_F_INV<br>[0]: INTERLACE |
| 0x71C | R/W | 0x10 | [6:5]: TVENC_YC_INV<br>[4:2]: TVENC_OSENC |

| Address | R/W | Default | Description |
|---|---|---|---|
| | | | [1]: TVENC_FLDENCP<br>[0]: TVENC_FLDDET |
| 0x71D | R/W | 0 | TVENC_HSENC[7:0] |
| 0x71E | R/W | 0 | TVENC_HSENC[9:8] |
| 0x71F | R/W | 0 | TVENC_VSENC[5:0] |
| 0x720 | R/W | 0 | TVENC_LINE_OS[4:0] |
| 0x721 | R/W | 0 | TVENC_PIXEL_OS[5:0] |
| 0x722 | R/W | 0 | [6]: TVENC_C_OFF_D<br>[5:4]: TVENC_CBW_D<br>[3:2]: TVENC_YBW_D<br>[1:0]: TVENC_VIS_FLD_OS |
| 0x723 | R/W | 0 | [7]: TVENC_TST_FSC_FREE<br>[6]: TVENC_T_CCIR_TIM<br>[5]: TVENC_T_656_STD<br>[4:0]: TVENC_VIS_LINE_OS |
| 0x724 | R/W | 0 | [4]: OSD_CH_EN_SEL<br>[3]: NOVID_SWITCH_EN<br>[2]: BND_CH_EN_SEL<br>[1]: NO_DNS<br>[0]: VGA_DIS |
| 0x725 | R/W | 0 | [2]: DM_PD_R<br>[1]: DM_PD_G<br>[0]: DM_PD_B |
| 0X726 | R/W | 0 | [0]: MOUSE_UPDATE_EN / MOUSE_UPDATE_BUSY |
| 0x727 | R/W | 0x20 | DNS_HBLANK[7:0] |
| 0X728 | R/W | 0 | BND_CH_EN[7:0] |
| 0X729 | R/W | 0 | BND_CH_EN[15:8] |
| 0X72A | R/W | 0 | BND_CH_EN[23:16] |
| 0X72B | R/W | 0 | BND_CH_EN[31:24] |
| 0X72C | R/W | 0 | NOVID_R[7:0] |
| 0X72D | R/W | 0 | NOVID_G[7:0] |
| 0X72E | R/W | 0xFF | NOVID_B[7:0] |
| 0X72F | R/W | 0 | [4]: MOUSE_BUF<br>[3:0]: MOUSE_INDEX |
| 0x730 | R/W | 0 | [4]: OSD_TITLE_EN<br>[3]: OSD_TIME_EN<br>[2]: OSD_CHPIC_EN<br>[1]: OSD_CHNUM_EN<br>[0]: OSD_EN |
| 0x731 | R/W | 0 | [5]: OSD_CHPIC_BLINK<br>[4]: OSD_CHPIC_TRANS<br>[3]: OSD_TITLE_MIX<br>[2]: OSD_TIME_MIX<br>[1]: OSD_CHPIC_MIX<br>[0]: OSD_CHNUM_MIX |
| 0x732 | R/W | 0x4F | [7:6]: OSD_CHPIC_POS<br>[5:4]: OSD_CHNUM_POS<br>[3:2]: OSD_FONT_VSIZE<br>[1:0]: OSD_FONT_HSIZE |
| 0x733 | R/W | 0 | OSD_CHNUM_HPOS[7:0] |
| 0x734 | R/W | 0 | OSD_CHNUM_HPOS[10:8] |
| 0x735 | R/W | 0 | OSD_CHNUM_VPOS[7:0] |
| 0x736 | R/W | 0 | OSD_CHNUM_VPOS[10:8] |
| 0x737 | R/W | 0 | OSD_CHPIC_HPOS[7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x738 | R/W | 0 | OSD_CHPIC_HPOS[10:8] |
| 0x739 | R/W | 0 | OSD_CHPIC_VPOS[7:0] |
| 0x73A | R/W | 0 | OSD_CHPIC_VPOS[10:8] |
| 0x73B | R/W | 0 | OSD_TIME_HPOS[7:0] |
| 0x73C | R/W | 0 | OSD_TIME_HPOS[10:8] |
| 0x73D | R/W | 0 | OSD_TIME_VPOS[7:0] |
| 0x73E | R/W | 0 | OSD_TIME_VPOS[10:8] |
| 0x73F | R/W | 0 | OSD_TITLE_HPOS[7:0] |
| 0x740 | R/W | 0 | OSD_TITLE_HPOS[10:8] |
| 0x741 | R/W | 0 | OSD_TITLE_VPOS[7:0] |
| 0x742 | R/W | 0 | OSD_TITLE_VPOS[10:8] |
| 0x743 | R/W | 0 | OSD_FONT_R1[7:0] |
| 0x744 | R/W | 0 | OSD_FONT_G1[7:0] |
| 0x745 | R/W | 0 | OSD_FONT_B1[7:0] |
| 0x746 | R/W | 0 | OSD_FONT_R2[7:0] |
| 0x747 | R/W | 0 | OSD_FONT_G2[7:0] |
| 0x748 | R/W | 0 | OSD_FONT_B2[7:0] |
| 0x749 | R/W | 0 | OSD_FONT_R3[7:0] |
| 0x74A | R/W | 0 | OSD_FONT_G3[7:0] |
| 0x74B | R/W | 0 | OSD_FONT_B3[7:0] |
| 0x74C | R/W | 0 | OSD_PIC_R0[7:0] |
| 0x74D | R/W | 0 | OSD_PIC_G0[7:0] |
| 0x74E | R/W | 0 | OSD_PIC_B0[7:0] |
| 0x74F | R/W | 0 | OSD_PIC_R1[7:0] |
| 0x750 | R/W | 0 | OSD_PIC_G1[7:0] |
| 0x751 | R/W | 0 | OSD_PIC_B1[7:0] |
| 0x752 | R/W | 0 | OSD_PIC_R2[7:0] |
| 0x753 | R/W | 0 | OSD_PIC_G2[7:0] |
| 0x754 | R/W | 0 | OSD_PIC_B2[7:0] |
| 0x755 | R/W | 0 | OSD_PIC_R3[7:0] |
| 0x756 | R/W | 0 | OSD_PIC_G3[7:0] |
| 0x757 | R/W | 0 | OSD_PIC_B3[7:0] |
| 0x758 | W | 0 | OSD_FRAM_ADDR[7:0] |
| 0x759 | W | 0 | OSD_FRAM_ADDR[10:8] |
| 0x75A | W | 0 | OSD_FRAM_DATA[7:0] |
| 0x75B | W | 0 | OSD_FRAM_DATA[15:8] |
| 0x75C | W | 0 | OSD_DRAM_ADDR[7:0] |
| 0x75D | W | 0 | OSD_DRAM_ADDR[8] |
| 0x75E | W | 0 | OSD_DRAM_DATA[5:0] |
| 0x75F | R/W | 0x03 | [1]: MOUSE_REG_UPDATE<br>[0]: OSG_REG_UPDATE |
| 0x760 | R/W | 0 | BAR[7:0] |
| 0x761 | R/W | 0 | BAG[7:0] |
| 0x762 | R/W | 0 | BAB[7:0] |
| 0x763 | R/W | 0 | [7]: SWITCH_EN<br>[6]: COLBAR_EN<br>[5]: GAIN_EN<br>[4]: BND_EN<br>[3:0]: BND_WIDTH |
| 0x764 | R/W | 0 | BND_R[7:0] |
| 0x765 | R/W | 0 | BND_G[7:0] |
| 0x766 | R/W | 0 | BND_B[7:0] |
| 0x767 | R/W | 0 | VTT_WIN[7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0X768 | R/W | 0 | [2]: VTT_ADJUST<br>[1]: VTT_ADJUST_MODE<br>[0]: VTT_ADJUST_EN |
| 0X769 | RO | - | NEW_VTT_RGBW[7:0] |
| 0X76A | RO | - | NEW_VTT_RGBW[11:8] |
| 0X76B | RO | - | NEW_VTT[7:0] |
| 0X76C | R/W | 0 | NEW_VTT[10:8] |
| 0x76D | R/W | RO | [0]: VACTIVE |
| 0x780 | R/W | 0x00 | SBOX0_CTRL[4:0] |
| 0x781 | R/W | 0x00 | SBOX1_CTRL[4:0] |
| 0x782 | R/W | 0x00 | SBOX2_CTRL[4:0] |
| 0x783 | R/W | 0x00 | SBOX3_CTRL[4:0] |
| 0x784 | R/W | 0x00 | [7:6]: SBOX3_V_LINE<br>[5:4]: SBOX3_H_LINE<br>[3:2]: SBOX2_V_LINE<br>[1:0]: SBOX2_H_LINE |
| 0x785 | R/W | 0x00 | [7:6]: SBOX1_V_LINE<br>[5:4]: SBOX1_H_LINE<br>[3:2]: SBOX0_V_LINE<br>[1:0]: SBOX0_H_LINE |
| 0x786 | R/W | 0x00 | SBOX0_HL[7:0] |
| 0x787 | R/W | 0x00 | SBOX0_HL[10:8] |
| 0x788 | R/W | 0x00 | SBOX1_HL[7:0] |
| 0x789 | R/W | 0x00 | SBOX1_HL[10:8] |
| 0x78A | R/W | 0x00 | SBOX2_HL[7:0] |
| 0x78B | R/W | 0x00 | SBOX2_HL[10:8] |
| 0x78C | R/W | 0x00 | SBOX3_HL[7:0] |
| 0x78D | R/W | 0x00 | SBOX3_HL[10:8] |
| 0x78E | R/W | 0x00 | SBOX0_HR[7:0] |
| 0x78F | R/W | 0x00 | SBOX0_HR[10:8] |
| 0x790 | R/W | 0x00 | SBOX1_HR[7:0] |
| 0x791 | R/W | 0x00 | SBOX1_HR[10:8] |
| 0x792 | R/W | 0x00 | SBOX2_HR[7:0] |
| 0x793 | R/W | 0x00 | SBOX2_HR[10:8] |
| 0x794 | R/W | 0x00 | SBOX3_HR[7:0] |
| 0x795 | R/W | 0x00 | SBOX3_HR[10:8] |
| 0x796 | R/W | 0x00 | SBOX0_VT[7:0] |
| 0x797 | R/W | 0x00 | SBOX0_VT[10:8] |
| 0x798 | R/W | 0x00 | SBOX1_VT[7:0] |
| 0x799 | R/W | 0x00 | SBOX1_VT[10:8] |
| 0x79A | R/W | 0x00 | SBOX2_VT[7:0] |
| 0x79B | R/W | 0x00 | SBOX2_VT[10:8] |
| 0x79C | R/W | 0x00 | SBOX3_VT[7:0] |
| 0x79D | R/W | 0x00 | SBOX3_VT[10:8] |
| 0x79E | R/W | 0x00 | SBOX0_VB[7:0] |
| 0x79F | R/W | 0x00 | SBOX0_VB[10:8] |
| 0x7A0 | R/W | 0x00 | SBOX1_VB[7:0] |
| 0x7A1 | R/W | 0x00 | SBOX1_VB[10:8] |
| 0x7A2 | R/W | 0x00 | SBOX2_VB[7:0] |
| 0x7A3 | R/W | 0x00 | SBOX2_VB[10:8] |
| 0x7A4 | R/W | 0x00 | SBOX3_VB[7:0] |
| 0x7A5 | R/W | 0x00 | SBOX3_VB[10:8] |
| 0x7A6 | R/W | 0x00 | BBR: Box border R |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x7A7 | R/W | 0x00 | BBG: Box border G |
| 0x7A8 | R/W | 0x00 | BBB: Box border B |
| 0x7A9 | R/W | 0x00 | BPR: Box plane R |
| 0x7AA | R/W | 0x00 | BPG: Box plane G |
| 0x7AB | R/W | 0x00 | BPB: Box plane B |
| 0x7AC | R/W | 0x00 | MOUSE0_HPOS[7:0] |
| 0x7AD | R/W | 0x00 | MOUSE0_HPOS[10:8] |
| 0x7AE | R/W | 0x00 | MOUSE0_VPOS[7:0] |
| 0x7AF | R/W | 0x00 | MOUSE0_VPOS[10:8] |
| 0x7B0 | R/W | 0x00 | MOUSE1_HPOS[7:0] |
| 0x7B1 | R/W | 0x00 | MOUSE1_HPOS[10:8] |
| 0x7B2 | R/W | 0x00 | MOUSE1_VPOS[7:0] |
| 0x7B3 | R/W | 0x00 | MOUSE1_VPOS[10:8] |
| 0x7B4 | R/W | 0x00 | [6:4]: MOUSE1_CTRL<br>[2:0]: MOUSE0_CTRL |
| 0x7B5 | R/W | 0x00 | BR[7:0]: mouse background R |
| 0x7B6 | R/W | 0x00 | BG[7:0]: mouse background G |
| 0x7B7 | R/W | 0x00 | BB[7:0]: mouse background B |
| 0x7B8 | R/W | 0x00 | FR[7:0]: mouse foreground R |
| 0x7B9 | R/W | 0x00 | FG[7:0]: mouse foreground G |
| 0x7BA | R/W | 0x00 | FB[7:0]: mouse foreground B |
| 0x7BB | R/W | 0x00 | MOUSE_WR_LOC[7:0] |
| 0x7BC | WO | 0x00 | MOUSE_WR_DATA[7:0] |
| 0x7BD | WO | 0x00 | MOUSE_WR_EN |
| 0x7BE | R/W | 0x00 | POS_HSCALE[7:0] |
| 0x7BF | R/W | 0x10 | POS_HSCALE[15:8] |
| 0x7C0 | R/W | 0x00 | POS_VSCALE[7:0] |
| 0x7C1 | R/W | 0x10 | POS_VSCALE[15:8] |
| 0x7C2 | R/W | 0 | [0]: POS_DNS_EN |
| 0X7C3 | R/W | 0 | OSD_BLINK_TIME[1:0] |
| 0x7C4 | R/W | 0 | [6]: OSG_UPSCL_EN1<br>[5:4]: OSG_BLINK_MODE1<br>[3:2]: OSG_BLEND_MODE1<br>[1]: OSG_TRANS_EN1<br>[0]: OSG_W1_EN |
| 0x7C5 | R/W | 0 | [6]: OSG_UPSCL_EN2<br>[5:4]: OSG_BLINK_MODE2<br>[3:2]: OSG_BLEND_MODE2<br>[1]: OSG_TRANS_EN2<br>[0]: OSG_W2_EN |
| 0x7C6 | R/W | 0 | [7:4] OSG_ALPHA2<br>[3:0]: OSG_ALPHA1 |
| 0X7C7 | R/W | 0 | [4]: OSG_FORMAT_RG<br>[3:2]: OSG_FORMAT<br>[1:0]: OSG_BLINK_FRAME |
| 0x7C8 | R/W | 0 | OSG_ALPHA_COLOR1 [7:0] |
| 0x7C9 | R/W | 0 | OSG_ALPHA_COLOR1 [15:8] |
| 0x7CA | R/W | 0 | OSG_ALPHA_COLOR2 [7:0] |
| 0x7CB | R/W | 0 | OSG_ALPHA_COLOR2 [15:8] |
| 0x7CC | R/W | 0 | OSG_ALPHA_COLOR3 [7:0] |
| 0x7CD | R/W | 0 | OSG_ALPHA_COLOR3 [15:8] |
| 0x7CE | R/W | 0 | OSG_ALPHA_COLOR4 [7:0] |
| 0x7CF | R/W | 0 | OSG_ALPHA_COLOR4 [15:8] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x7D0 | R/W | 0 | OSG_BLINK_COLOR1 [7:0] |
| 0x7D1 | R/W | 0 | OSG_BLINK_COLOR1 [15:8] |
| 0x7D2 | R/W | 0 | OSG_BLINK_COLOR2 [7:0] |
| 0x7D3 | R/W | 0 | OSG_BLINK_COLOR2 [15:8] |
| 0x7D4 | R/W | 0 | OSG_WM1_VSTART [7:0] |
| 0x7D5 | R/W | 0 | OSG_WM1_VSTART [12:8] |
| 0x7D6 | R/W | 0 | OSG_WM1_HSTART [7:0] |
| 0x7D7 | R/W | 0 | OSG_WM1_HSTART [12:8] |
| 0x7D8 | R/W | 0 | OSG_W1_VSIZE [7:0] |
| 0x7D9 | R/W | 0 | OSG_W1_VSIZE [10:8] |
| 0x7DA | R/W | 0 | OSG_W1_HSIZE [7:0] |
| 0x7DB | R/W | 0 | OSG_W1_HSIZE [10:8] |
| 0x7DC | R/W | 0 | OSG_WS1_VSTART [7:0] |
| 0x7DD | R/W | 0 | OSG_WS1_VSTART [10:8] |
| 0x7DE | R/W | 0 | OSG_WS1_HSTART [7:0] |
| 0x7DF | R/W | 0 | OSG_WS1_HSTART [10:8] |
| 0x7E0 | R/W | 0 | OSG_WM2_VSTART [7:0] |
| 0x7E1 | R/W | 0 | OSG_WM2_VSTART [12:8] |
| 0x7E2 | R/W | 0 | OSG_WM2_HSTART [7:0] |
| 0x7E3 | R/W | 0 | OSG_WM2_HSTART [12:8] |
| 0x7E4 | R/W | 0 | OSG_W2_VSIZE [7:0] |
| 0x7E5 | R/W | 0 | OSG_W2_VSIZE [10:8] |
| 0x7E6 | R/W | 0 | OSG_W2_HSIZE [7:0] |
| 0x7E7 | R/W | 0 | OSG_W2_HSIZE [10:8] |
| 0x7E8 | R/W | 0 | OSG_WS2_VSTART [7:0] |
| 0x7E9 | R/W | 0 | OSG_WS2_VSTART [10:8] |
| 0x7EA | R/W | 0 | OSG_WS2_HSTART [7:0] |
| 0x7EB | R/W | 0 | OSG_WS2_HSTART [10:8] |
| 0x7EC | R/W | 0x11 | [7]: OSG_W2_EN4<br>[6]: OSG_W2_EN3<br>[5]: OSG_W2_EN2<br>[4]: OSG_W2_EN1<br>[3]: OSG_W1_EN4<br>[2]: OSG_W1_EN3<br>[1]: OSG_W1_EN2<br>[0]: OSG_W1_EN1 |
| 0x7ED | R/W | 0x00 | [3:2]: OSG_W2_SEL<br>[1:0]: OSG_W1_SEL |
| 0x7EE | R/W | 0 | OSG_VSIZE_DN1 [7:0] |
| 0x7EF | R/W | 0 | OSG_VSIZE_DN1 [10:8] |
| 0x7F0 | R/W | 0 | OSG_HSIZE_DN1 [7:0] |
| 0x7F1 | R/W | 0 | OSG_HSIZE_DN1 [10:8] |
| 0x7F2 | R/W | 0 | OSG_VSIZE_DN2 [7:0] |
| 0x7F3 | R/W | 0 | OSG_VSIZE_DN2 [10:8] |
| 0x7F4 | R/W | 0 | OSG_HSIZE_DN2 [7:0] |
| 0x7F5 | R/W | 0 | OSG_HSIZE_DN2 [10:8] |
| 0x7F6 | R/W | 0x40 | R_GAIN[7:0] |
| 0x7F7 | R/W | 0x40 | G_GAIN[7:0] |
| 0x7F8 | R/W | 0x40 | B_GAIN[7:0] |
| 0x7F9 | R/W | 0 | R_OFST[7:0] |
| 0x7FA | R/W | 0 | G_OFST[7:0] |
| 0x7FB | R/W | 0 | B_OFST[7:0] |

# Registers

### DUAL MONITOR HORIZONTAL TOTAOL LOW BYTE REGISTER – 0X700

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | DM_HTT[7:0]<br><br>Dual monitor horizontal total register. Lower 8 bits of the DM_HTT, it should be set to VESA standard. And it is real value minus 1 |

### DUAL MONITOR HORIZONTAL TOTAL HIGH BYTE REGISTER – 0X701

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | N/A | - | Reserved |
| 4:0 | RW | 0 | DM_HTT[12:8]<br><br>Dual monitor horizontal total register, upper 5 bits of the DM_HTT |

### DUAL MONITOR VERTICAL TOTAL LOW BYTE REGISTER – 0X702

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | DM_VTT [7:0]<br><br>Dual monitor vertical total register, it should be set to VESA standard. It is real value minus 1. |

### DUAL MONITOR VERTICAL TOTAL HIGH BYTE REGISTER – 0X703

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | N/A | - | Reserved |
| 2:0 | RW | 0 | DM_VTT[10:8]<br><br>Dual monitor vertical total register, it should be set to VESA standard. It is real value minus 1. In interlaced mode, it is set to top field vertical total value minus 1. |

### DUAL MONITOR HORIZONTAL WIDTH LOW BYTE REGISTER – 0X704

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | DM_HDE[7:0]<br><br>Dual monitor horizontal active data width. Lower 8 bits of the DM_HDE. This register must be set to read value minus 1. |

### DUAL MONITOR HORIZONTAL WIDTH HIGH BYTE REGISTER – 0X705

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | N/A | - | Reserved |
| 2:0 | RW | 0 | DM_HDE [10:8]<br><br>Dual monitor horizontal active data width. Higher 3 bits of the DM_HDE. This register must be set to read value minus 1. |

### DUAL MONITOR VERTICAL HEIGHT LOW BYTE REGISTER – 0X706

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | DM_VDE[7:0]<br><br>Dual monitor vertical display height. Lower 8 bits of the DM_VDE. This register must be set to real value minus 1. In interlaced mode, it is set to height of one field minus 1. |

### DUAL MONITOR VERTICAL HEIGHT HIGH BYTE REGISTER – 0X707

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | N/A | - | Reserved |
| 4:0 | RW | 0 | DM_VDE[10:8]<br><br>Dual monitor vertical display height. Higher 3 bits of the DM_VDE. This register must be set to real value minus 1. In interlaced mode, it is set to height of one field minus 1. |

### DUAL MONITOR HSYNC FRONT PORCH LOW BYTE REGISTER – 0X708

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | DM_PHSYNC[7:0]<br><br>Dual monitor hsync front porch register. Lower 8 bits of DM_PHSYNC. |

### DUAL MONITOR HSYNC FRONT PORCH HIGH BYTE REGISTER – 0X709

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | N/A | - | Reserved |
| 0 | RW | 0 | DM_PHSYNC [8]<br><br>Dual monitor hsync front porch register. MSB of DM_PHSYNC. |

### DUAL MONITOR VSYNC FRONT PORCH REGISTER – 0X70A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | DM_PVSYNC[7:0]<br><br>Dual monitor vsync front porch register. |

### DUAL MONITOR HSYNC WIDTH REGISTER – 0X70B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | DM_HSPW [7:0]<br><br>Dual monitor hsync width register. It uses one pixel unit. |

### DUAL MONITOR VSYNC WIDTH REGISTER – 0X70C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | DM_VSPW [7:0]<br><br>Dual monitor vsync width register. It uses one line unit. |

### DUAL MONITOR CONTROL REGISTER – 0X70D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **MDBOX_EN:** Dual monitor motion box control<br><br>0: off<br>1: on |
| 6 | RW | 0 | **VTT_INV**<br><br>Vertical total exchange between top field and bottom field. In interlace mode, top field has VTT+1 lines, even field has VTT+2 lines. If this bit set to high, top field has VTT+2 lines, and even field has VTT+1 lines. This bit is not used in normal case. It is only used in debug mode. |
| 5 | RW | 0 | **DM_INT_CBAR**<br><br>Dual monitor internal color bar test on in TV encoder |
| 4 | RW | 0 | **EVEN_REPEAT**<br><br>0: repeat odd or even when needed<br>1: repeat even field only |
| 3:0 | RW | 0 | **VSYNC_DLY**<br><br>The time between hsync and vsync edge. |

### DUAL MONITOR TARGET WIDTH FOR DOWN SCALER LOW BYTE REGISTER – 0X70E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **TAR_W[7:0]**<br><br>Dual monitor down scaler target image width. One pixel unit. |

### DUAL MONITOR TARGET WIDTH FOR DOWN SCALER HIGH BYTE REGISTER – 0X70F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | N/A | - | **Reserved** |
| 2:0 | RW | 0 | **TAR_W[10:8]**<br><br>Dual monitor down scaler target image width. One pixel unit. |

### DUAL MONITOR TARGET HEIGHT FOR DOWN SCALER LOW BYTE REGISTER – 0X710

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **TAR_H[7:0]**<br><br>Dual monitor down scaler target image height. One line unit. |

### DUAL MONITOR TARGET HEIGHT FOR DOWN SCALER HIGH BYTE REGISTER – 0X711

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | N/A | - | **Reserved** |
| 2:0 | RW | 0 | **TAR_H[10:8]**<br><br>Dual monitor down scaler target image height. One line unit. |

### DUAL MONITOR MODE REGISTER – 0X712

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | - | | Reserved |
| 6 | RW | 0 | **IN16_MODE:** 16 bits input on/off control for down-scaler |
| 5 | RW | 0 | **DM_YC_SWAP:** Control luma and chroma position for BT1120 output |
| 4 | RW | 0 | **DM_MAIN:** Dual monitor video use main display RGB output |
| 3 | RW | 0 | **DM_ENA:** Dual Monitor Enable |
| 2 | RW | 0 | **DM_LCD:** Dual monitor is LCD mode register<br><br>1: When the dual monitor is connected to LCD VGA mode<br>0: When the dual monitor is connected to Analog TV |
| 1 | RW | 0 | **DM_VSPOL:** Dual monitor vertical sync polarity register<br><br>0: negative<br>1: positive |
| 0 | RW | 0 | **DM_HSPOL:** Dual monitor horizontal sync polarity register<br><br>0: negative<br>1: positive |

### DUAL MONITOR HORIZONTAL START LOW BYTE REGISTER – 0X713

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | **DM_HSTART[7:0]**<br><br>Dual monitor DRAM horizontal start register. It is four pixels unit. |

### DUAL MONITOR VERTICAL START LOW BYTE REGISTER – 0X714

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | **DM_VSTART [7:0]**<br><br>Dual monitor DRAM vertical start register. It is one line unit. |

### DUAL MONITOR HSTART AND VSTART HIGH BYTE REGISTER – 0X715

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5:4 | RW | 0 | **DM_HSTART [9:8]**<br><br>Dual monitor DRAM horizontal start register. It is four pixels unit. |
| 3:0 | RW | 0 | **DM_VSTART[11:8]**<br><br>Dual monitor DRAM vertical start register. It is one line unit. |

### DUAL MONITOR DOWN SCALER SOURCE WIDTH LOW BYTE REGISTER – 0X716

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | **SRC_W [7:0]**<br><br>Dual monitor down scaler source image width, 1 pixel unit. |

### DUAL MONITOR DOWN SCALER SOURCE WIDTH HIGH BYTE REGISTER – 0X717

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5:3 | RW | 0 | DM_HSTART [9:8]<br><br>Dual monitor DRAM horizontal start register. It is four pixels unit. |
| 2:0 | RW | 0 | SRC_W[10:8]<br><br>Dual monitor down scaler source image width, 1 pixel unit. |

### DUAL MONITOR DOWN SCALER SOURCE HEIGHT LOW BYTE REGISTER – 0X718

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SRC_H[7:0]<br><br>Dual monitor down scaler source image height, 1 line unit. |

### DUAL MONITOR DOWN SCALER SOURCE HEIGHT HIGH BYTE REGISTER – 0X719

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5:3 | RW | 0 | DM_HSTART [9:8]<br><br>Dual monitor DRAM horizontal start register. It is four pixels unit. |
| 2:0 | RW | 0 | SRC_H[10:8]<br><br>Dual monitor down scaler source image height, 1 line unit. |

### DUAL MONITOR TV ENCODER CONTROL REGISTER 1 – 0X71A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | TVENC_FSC_MODE<br><br>Dual Monitor TV FSCSEL. This register determines the TV modulation frequency standard select.<br>00: NTSC/M<br>01: PAL/B, D, G, H, I<br>10: PAL/M<br>11: PAL/N |
| 5 | RW | 0 | TVENC_SAT_LMT<br><br>Dual Monitor TV video saturation limit. Control the data range of the ITU-R BT 656 output<br>0: Not limit<br>1: Data range is limited to 1 – 254 range |
| 4:1 | - | - | Reserved |
| 0 | RW | 0 | TVENC_PALNT<br><br>Dual Monitor TV PAL or NTSC select. This register determines the CVBS output Standard.<br>0: NSTC mode<br>1: PAL mode. |

### DUAL MONITOR TV ENCODER CONTROL REGISTER 2 – 0X71B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **TVENC_VSPOL:** Dual Monitor TVENC Vertical sync polarity<br><br>0: Active Low<br>1: Active High |
| 6 | RW | 0 | **TVENC_HSPOL:** Dual Monitor TV Encoder Horizontal Sync Polarity<br><br>0: Active Low<br>1: Active High |
| 5 | RW | 0 | **TVENC_VSDEC:** Dual monitor TV Encoder vsync detect signal<br><br>0: vsync signal<br>1: field signal |
| 4 | RW | 0 | **TVENC_PED:** Dual Monitor TV encoder Pedestal Enable. Enable Pedestal for video output.<br><br>0: 0 IRE for pedestal<br>1: 7.5 IRE pedestal level |
| 3 | RW | 0 | **TVENC_PDRST:** Dual Monitor TV Encoder Pedestal Reset. Subcarrier Phase Alternation reset every 8 field in PAL system<br><br>0: Constant relationship, reset every 8 field<br>1: Free running |
| 2 | RW | 0 | **TVENC_PHALT:** Dual Monitor TV Encoder Phase Alternation.<br><br>0: Disable phase alternation for line by line<br>1: Enable phase alternation for line by line |
| 1 | RW | 0 | **TVENC_F_INV:** Dual Monitor TV Encoder Field Output Invert. This register bit inverts the field output level if set. |
| 0 | RW | 0 | **INTERLACE:** Dual Monitor Interlaced mode enable.<br><br>0: progressive<br>1: interlaced |

### DUAL MONITOR TV ENCODER CONTROL REGISTER 3 – 0X71C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | N/A | | **Reserved** |
| 6:5 | RW | 0 | **TVENC_YC_OS[1:0]:** Dual Monitor TV Encoder Active Pixel Offset delay<br><br>00: No Active offset<br>01: Active pixel offset for 1/4 pixel unit<br>10: Active pixel offset for 1/2 pixel unit<br>11: Active pixel offset for 3/4 pixel unit |
| 4 | RW | 1 | **TVENC_OSENC[2]**<br><br>0: one more line delay on odd field<br>1: odd field and even field same delay |
| 3:2 | RW | 00 | **TVENC_OSENC[1:0]:** Dual Monitor TV Encoder Field Offset for the first video line<br><br>00: No delay for odd field and even field<br>01: No delay for even field, delay one line for odd field<br>10: Delay one line for even field, no delay for odd field<br>11: Delay two lines for even field, no delay for odd field |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1 | RW | 0 | **TVENC_FLDENCP:** Dual Monitor TV Encoder Field Polarity. This Register control the field polarity.<br><br>0: Even field is high<br>1: Odd field is high |
| 0 | - | - | **Reserved** |

### DUAL MONITOR TV ENCODER PIXEL DELAY LOW BYTE REGISTER – 0X71D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **TVENC_HSENC[7:0]:** Dual Monitor TV Encoder Active pixel delay. Lower 8 bit of the pixel delay from active video by ½ pixel per step |

### DUAL MONITOR TV ENCODER PIXEL DELAY HIGH BYTE REGISTER – 0X71E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | **TVENC_HSENC[9:8]:** Dual Monitor TV Encoder Active pixel delay. Upper 2 bit of the pixel delay from active video by ½ pixel per step |

### DUAL MONITOR TV ENCODER LINE DELAY REGISTER – 0X71F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5:0 | RW | 0 | **TVENC_VSENC[5:0]:** Dual Monitor TV Encoder Active line delay, 6 bit of the line delay from active video by line per step |

### DUAL MONITOR TV ENCODER LINE OFFSET REGISTER – 0X720

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | RW | 0 | **TVENC_LINE_OS[4:0]:** Dual Monitor TV Encode line Offset, 5 bit of the line offset from active video |

### DUAL MONITOR TV ENCODER PIXEL OFFSET REGISTER – 0X721

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5:0 | RW | 0 | **TVENC_PIXEL_OS[5:0]:** Dual Monitor TV Encode Pixel Offset, 6 bit of the Pixel Offset from active video |

### DUAL MONITOR TV ENCODER CONTROL REGISTER 4 – 0X722

| Bit | R/W | Default | Description |
|---|---|---|---|
| 6 | RW | 0 | **DM _TVENC_C_OFF_D:** Dual Monitor TV Encoder Color Offset |
| 5:4 | RW | 0 | **DM _TVENC_CBW_D:** Dual Monitor TV Encoder Chrominance Bandwidth Control |
| 3:2 | RW | 0 | **DM _TVENC_YBW_D:** Dual Monitor TV Encoder Luminance Bandwidth Control |
| 1:0 | RW | 0 | **DM_TVENC_FLD_OS:** Dual Monitor TV Encoder Active Field offset. |

### DUAL MONITOR TV ENCODER CONTROL REGISTER 5 – 0X723

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **DM_TVENC_TST_FSC_FREE:** Dual monitor TV Encoder Sub-Carrier is set to free run |
| 6 | RW | 0 | **DM_TVENC_T_CCIR_TIM:** Dual monitor TV Encoder CCIR Timing |
| 5 | RW | 0 | **DM_TVENC_T_656_STD:** Dual monitor TV Encoder 656 Standard |
| 4:0 | RW | 0 | **DM_TVENC_VIS_LINE_OS:** Dual monitor TV Encoder line offset |

### DUAL MONITOR CONTROL REGISTER – 0X724

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4 | R/W | 0 | **OSD_CH_EN_SEL:** Simple OSD channel enable select.<br><br>0: use rgb_interface channel enable<br>1: use another register set, [0x728] to [0x72B] |
| 3 | R/W | 0 | **NOVID_SWITCH_EN:** In normal case, this bit set to 0. No video color will show up. If this bit set to 1, SDRAM data will show up. |
| 2 | R/W | 0 | **BND_CH_EN_SEL:** Boundary channel enable select.<br><br>0: use rgb_interface channel enable<br>1: use another register set, [0x728] to [0x72B] |
| 1 | RW | 0 | **NO_DNS:** Turn off down scaler. When this bit set to 1, dm_fclk must be bigger than dm_vclk<br><br>1: bypass down scaler<br>0: use down scaler. |
| 0 | RW | 0 | **VGA_DIS:** Disable dual monitor VGA output. Hsync, Vsync and Data will be low when this bit set to high. |

### DUAL MONITOR DAC POWER DOWN REGISTER – 0X725

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2 | RW | 0 | **DM_PD_R:** Power down R DAC |
| 1 | RW | 0 | **DM_PD_G:** Power down G DAC |
| 0 | RW | 0 | **DM_PD_B:** Power down B DAC |

### DUAL MONITOR MOUSE UPDATE ENABLE REGISTER – 0X726

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | RW | 0 | **MOUSE_UPDATE_EN**<br><br>Mouse data update from SDRAM to local SRAM enable. Hardware will automatically clear this bit when finish operation. |

### DUAL MONITOR DOWN SCALER HORIZONTAL BLANK REGISTER – 0X727

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x20 | **DNS_HBLANK:** Down scaler hblank time. |

### DUAL MONITOR BOUNDARY CHANNEL ENABLE REGISTER 1 – 0X728

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BND_CH_EN[7:0]:** Boundary channel enable. This register is only used when BND_CH_EN_SEL is high. This is only used when user want to show boundary when channel is not enable. |

### DUAL MONITOR BOUNDARY CHANNEL ENABLE REGISTER 2 – 0X729

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BND_CH_EN[15:8]:** Boundary channel enable. This register is only used when BND_CH_EN_SEL is high. This is only used when user want to show boundary when channel is not enable. |

### DUAL MONITOR BOUNDARY CHANNEL ENABLE REGISTER 3 – 0X72A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BND_CH_EN[23:16]:** Boundary channel enable. This register is only used when BND_CH_EN_SEL is high. This is only used when user want to show boundary when channel is not enable. |

### DUAL MONITOR BOUNDARY CHANNEL ENABLE REGISTER 4 – 0X72B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BND_CH_EN[31:24]:** Boundary channel enable. This register is only used when BND_CH_EN_SEL is high. This is only used when user want to show boundary when channel is not enable. |

### DUAL MONITOR RED COLOR FOR NO VIDEO CHANNEL – 0X72C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **NOVID_R:** This color will show up when the channel boundary is enable, but channel has no video. |

### DUAL MONITOR GREEN COLOR FOR NO VIDEO CHANNEL – 0X72D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **NOVID_G:** This color will show up when the channel boundary is enable, but channel has no video. |

### DUAL MONITOR BLUE COLOR FOR NO VIDEO CHANNEL – 0X72E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0xFF | **NOVID_B:** This color will show up when the channel boundary is enable, but channel has no video. |

### DUAL MONITOR MOUSE INDEX REGISTER – 0X72F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4 | RW | 0 | **MOUSE_BUF:** There are two mouse buffer in local SRAM. This bit select which buffer will be updated.<br><br>0: first mouse buffer<br>1: second mouse buffer |
| 3:0 | RW | 0 | **MOUSE_INDEX:** This register set which mouse shape will be loaded from SDRAM. Up to 16 mouse shapes can be saved in SDRAM. If user wants more than 16 mouse shape, user can set different mouse base address. |

### DUAL MONITOR SIMPLE OSD REGISTER – 0X730 TO 0X75E

Please refer to simple OSD chapter.

### DUAL MONITOR HORIZONTAL DISPLAY ENABLE CONTROL REGISTER – 0X75F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1 | RW | 1 | **MOUSE_REG_UPDATE:** When update mouse position register, set this bit to 0. After finish, set this bit to 1. |
| 0 | RW | 1 | **OSG_REG_UPDATE:** When update OSG related register, set this bit to 0. After finish, set this bit to 0 |

### DUAL MONITOR BACKGROUND RED COLOR REGISTER – 0X760

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BAR:** Dual Monitor Background Red Color |

### DUAL MONITOR BACKGROUND GREEN COLOR REGISTER – 0X761

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BAG:** Dual Monitor Background Green Color |

### DUAL MONITOR BACKGROUND BLUE COLOR REGISTER– 0X762

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BAB:** Dual Monitor Background Blue Color |

### DUAL MONITOR CONTROL REGISTER– 0X763

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **SWITCH_EN:** This bit is used for debug mode. When this bit set to 1, background will disabled, data in SDRAM will show up. |
| 6 | RW | 0 | **COLBAR_EN:** Color bar enable in background layer |
| 5 | RW | 0 | **GAIN_EN:** RGB digital gain enable |
| 4 | RW | 0 | **BND_EN:** Dual Monitor Border Width Enable |
| 3:0 | RW | 0 | **BND_WIDTH:** Dual Monitor Border Width |

### DUAL MONITOR BOUNDARY RED COLOR REGISTER– 0X764

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BND_R:** Dual Monitor Border Red Color |

### DUAL MONITOR BOUNDARY GREEN COLOR REGISTER– 0X765

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BND_G:** Dual Monitor Border Green Color |

### DUAL MONITOR BOUNDARY BLUE COLOR REGISTER – 0X766

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **BND_B:** Dual Monitor Border Blue Color |

### DUAL MONITOR VERTICAL TOTAL WINDOW REGISTER – 0X767

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0xFF | **VTT_WIN[7:0]:** Vertical Total can be changed according to input video frame rate. If difference is more than this number, VTT will not be adjusted. Unit is 1 line. |

### DUAL MONITOR VERTICAL TOTAL ADJUST MODE REGISTER – 0X768

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2 | R/W | 0x00 | **VTT_ADJUST:** It is used for manual mode. After set this bit, VTT will be changed according to current input video frame rate. |
| 1 | R/W | 0 | **VTT_ADJUST_MODE**<br><br>1: auto mode, adjust VTT each frame<br>0: manual mode, adjust VTT after set VTT_ADJUST |
| 0 | R/W | 0 | **VTT_ADJUST_EN:** Enable VTT adjustment. If set to 0, use register setting. If set to 1, use input video frame rate. |

### DUAL MONITOR VERTICAL TOTAL FROM INPUT VIDEO LOW BYTE REGISTER – 0X769

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RO | - | **NEW_VTT_RGBW[7:0]:** It is calculated from input video. The number is for two fields. |

### DUAL MONITOR VERTICAL TOTAL FROM INPUT VIDEO HIGH BYTE REGISTER – 0X76A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 3:0 | RO | - | **NEW_VTT_RGBW[11:8]:** It is calculated from input video. The number is for two fields. |

### DUAL MONITOR VERTICAL TOTAL AFTER ADJUSTMENT BYTE REGISTER – 0X76B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RO | - | **NEW_VTT[7:0]:** It is new VTT which is used in output timing. The number is for one field. |

### DUAL MONITOR VERTICAL TOTAL AFTER ADJUSTMENT HIGH REGISTER – 0X76C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RO | - | **NEW_VTT[10:8]:** It is new VTT which is used in output timing. The number is for one field. |

### DUAL MONITOR SINGLE BOX REGISTER – 0X76D TO 0X7AB

**Dual monitor has four single box. Register descriptions are same as main display. Please refer to LCD chapter.**

### DUAL MONITOR MOUSE REGISTER – 0X7AC TO 0X7BD

**Dual monitor has two mouse layer. Register descriptions are same as in main display. Please refer to LCD chapter.**

### HORIZONTAL DOWN SCALER FACTOR LOW BYTE REGISTER – 0X7BE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **POS_HSCALE[7:0]**<br><br>Channel position information down scale factor. 0x1000 means no down scale. Value smaller than 0x1000 means down scale. If image is down scaled, this register must be set correctly. |

### HORIZONTAL DOWN SCALE FACTOR HIGH BYTE REGISTER – 0X7BF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x10 | **POS_HSCALE[15:8]**<br><br>Channel position information down scale factor. 0x1000 means no down scale. Value smaller than 0x1000 means down scale. If image is down scaled, this register must be set correctly. |

### VERTICAL DOWN SCALE FACTOR LOW BYTE REGISTER – 0X7C0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **POS_VSCALE[7:0]**<br><br>Channel position information down scale factor. 0x1000 means no down scale. Value smaller than 0x1000 means down scale. If image is down scaled, this register must be set correctly. |

### VERTICAL DOWN SCALER FACTOR HIGH BYTE REGISTER – 0X7C1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x10 | **POS_VSCALE[15:8]**<br><br>Channel position information down scale factor. 0x1000 means no down scale. Value smaller than 0x1000 means down scale. If image is down scaled, this register must be set correctly. |

### POSITION DOWN SCALER ENABLE CONTROL REGISTER – 0X7C2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | R | 0 | **Reserved** |
| 0 | R/W | 0 | **POS_DNS_EN:** Use down scaled position information if this bit set to high. |

### OSD BLINK RATE REGISTER – 0X7C3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | R/W | 0x00 | **OSD_BLINK_TIME:** OSD blinking frequency control<br><br>00: blinking on each 32 frames<br>01: blinking on each 16 frames<br>10: blinking on each 8 frames<br>11: blinking on each 4 frames |

### DUAL MONITOR OSG REGISTER – 0X7C4 TO 0X7F5

**Dual monitor has two layer OSG. Each layer has four sub-windows. Register descriptions are same as main display. Please refer to OSG chapter.**

### DUAL MONITOR DIGITAL R GAIN REGISTER – 0X7F6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x40 | **R_GAIN:** R digital gain. 0x40 is no gain. More than 0x40 means gain more than 1. |

### DUAL MONITOR DIGITAL G GAIN REGISTER – 0X7F7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x40 | **G_GAIN:** G digital gain. 0x40 is no gain. More than 0x40 means gain more than 1. |

### DUAL MONITOR DIGITAL B GAIN REGISTER – 0X7F8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x40 | **B_GAIN:** B digital gain. 0x40 is no gain. More than 0x40 means gain more than 1. |

### DUAL MONITOR R OFFSET REGISTER – 0X7F9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **R_OFST[7:0]**<br><br>R offset. Rout = Rin * R_GAIN + R_OFST. This value is 2's complement value. R_OFST can be negative data. |

### DUAL MONITOR G OFFSET REGISTER – 0X7FA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | **G_OFST[7:0]**<br><br>G offset. Rout = Rin * R_GAIN + R_OFST. This value is 2's complement value. R_OFST can be negative data. |

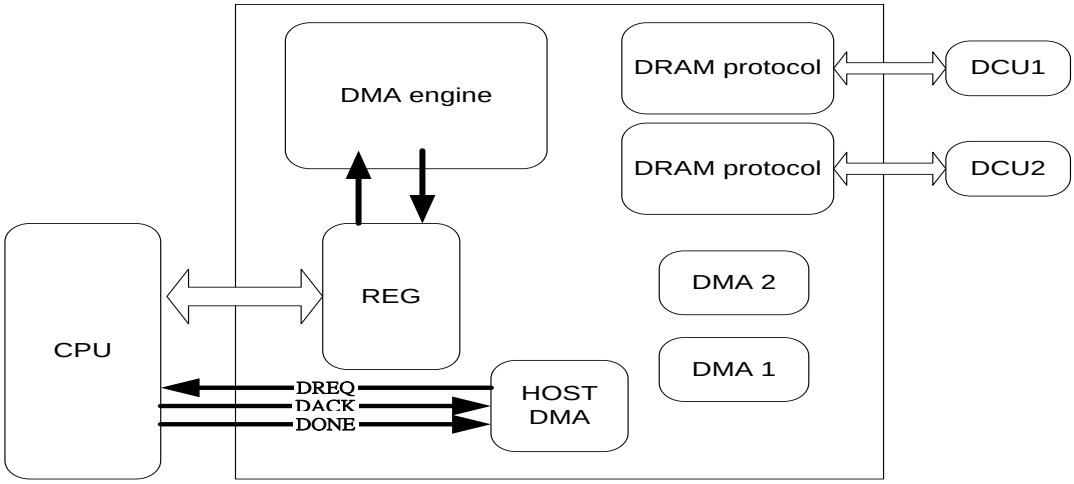
**DUAL MONITOR B OFFSET REGISTER – 0X7FB**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x00 | B_OFST[7:0]<br><br>B offset. Rout = Rin * R_GAIN + R_OFST. This value is 2's complement value. R_OFST can be negative data. |

# DMA Controller (DMAC)

## Overview

**The TW2880's direct memory access controller (DMAC) is a second-generation platform block capable of performing complex data movements through 2 programmable channels, with minimal intervention from the host processor. The hardware micro-architecture includes a DMA engine that performs source and destination address calculations, and the actual data movement operations, along with an DRAM-based memory containing the transfer control descriptors (TCD) for the channels. DMAC can be programmed to move data between the host processor and TW2880's off-chip memory or between two locations in the off-chip DRAM independently with the minimum help from CPU. Two independent DMA channels are supported.**




TW2880 DMAC

## Features

**DMA subsystem features are summarized as follow:**

**DMA Engine**

- 2 independent channels that can move data between
  - Host processor and TW2880's LCD or recording memory
  - Two memory areas in TW2880's LCD or recording memory
- Programmable source / destination starting address

## Theory of Operation

**DMA ENGINE**

**TW2880 DMA operations include:**

- SDRAM to SDRAM block moves
- Host to SDRAM moves

**A DMA operation begins when software enables a DMA channel, after setting the source and destination starting addresses, transfer count, bus transaction size. The DMA Engine moves the data block, and the DMA operation ends when the number of bytes specified by the transfer count has been reached. A DMA operation may also end early by programmer. When a DMA operation ends, an interrupt is sent to the host processor. Status register in each channel can be used to identify the event that caused the interrupt: a normal operation end or any one of several types of error ends.**

The exact transfer count of each DMA operation is controlled by Transfer Size Count Register(0x242 ~0x243, 0x246 ~ 0x247). The data moving sequence is performed as a series of DRAM transactions. The burst memory transaction size is controlled by DCU and is not changeable from DMA.

The source and destination starting addresses are set by firmware programmers. These addresses are incremented by the DMA. All addresses are physical addresses. The DMA control information can be set by direct CPU writes to DMA registers or alternatively, this DMA control information can be read from DMA descriptors stored in the off-screen memory.

**DRAM INTERFACE**

DMA is a low priority DCU client. When transferring data to/from DRAM, the normal handshake is followed. If both channels are activated, a arbiter inside the DMA will do the arbitration to share the DRAM bandwidth. Also programmer needs to make the addresses are correct to prevent overwriting. During the transactions, if time out situation occurs, both DMA channel will report the status and let host CPU decide if a software reset is needed.

**EXTERNAL DMA DREQ/DACK PROTOCOL**

There are two types of external DMA request/acknowledge protocols ( Single service Demand, Single service Handshake ). Each type defines how the signals like DMA request and acknowledge are related to these protocols.

## Basic DMA Timing

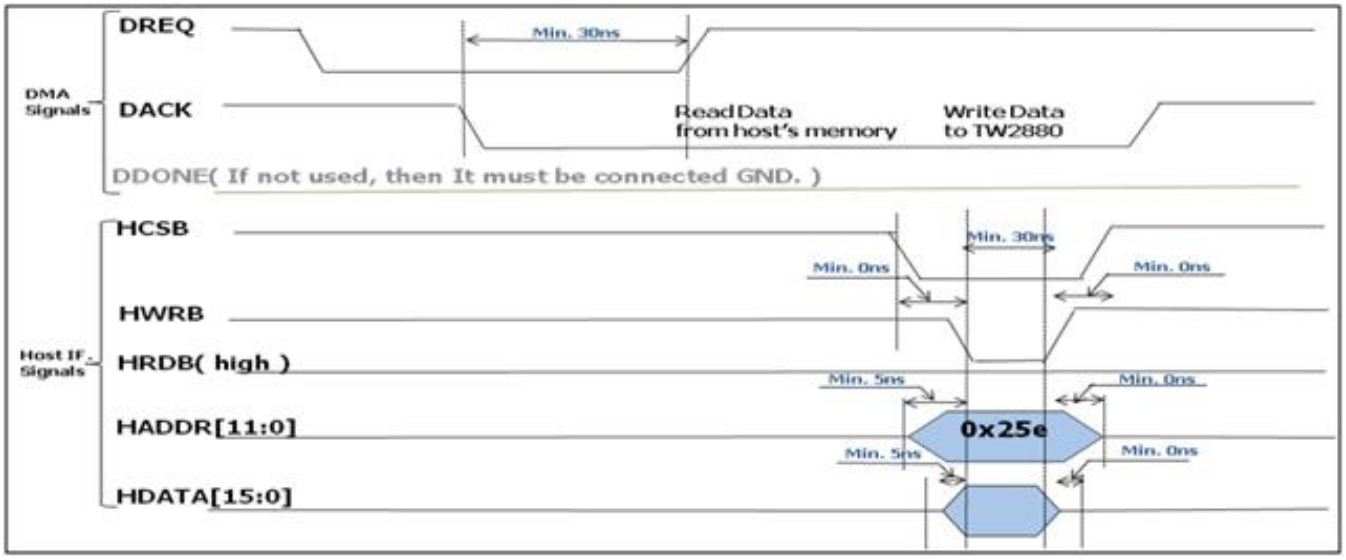
The DMA service means performing paired Reads and Writes cycles during DMA operation, which can make one DMA operation. Under figure shows the basic Timing in the DMA operation.

- The setup time and the delay time of DREQ and DACK are the same in all the modes.

- If the completion of DREQ meets its setup time, it is synchronized twice and then DACK is asserted.

- After assertion of DACK, DMA requests the bus and if it gets the bus it performs its operations.

DACK is negated when DMA operation is completed.




BASIC DMA TIMING DIAGRAM

Demand and Handshake modes are related to the protocol between DREQ and DACK. Figure shows the differences between the two modes.

At the end of one transfer (Single/Burst transfer), DMA checks the state of double-synched DREQ.

**Demand Mode**

- If DREQ remains asserted, the next transfer starts immediately. Otherwise it waits for DREQ to

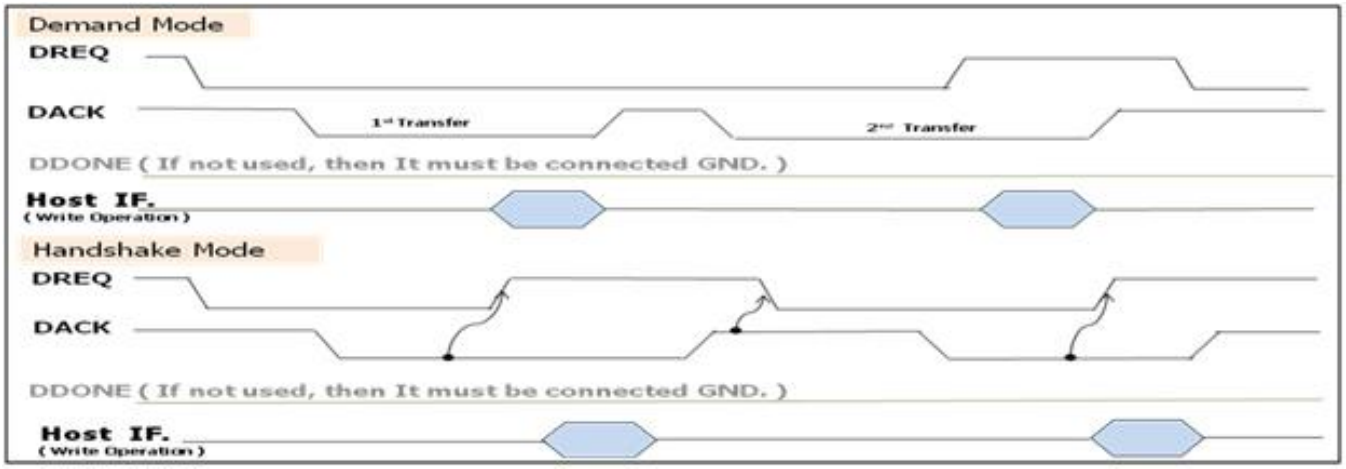
be asserted.

**Handshake Mode**

- If DREQ is negated, DMA negates DACK. Otherwise it waits until DREQ is de-asserted.

Caution: DREQ has to be asserted (low) only after the negation (high) of DACK.

( described in a processor point of view )




DEMAND/HANDSHAKE MODE COMPARISON

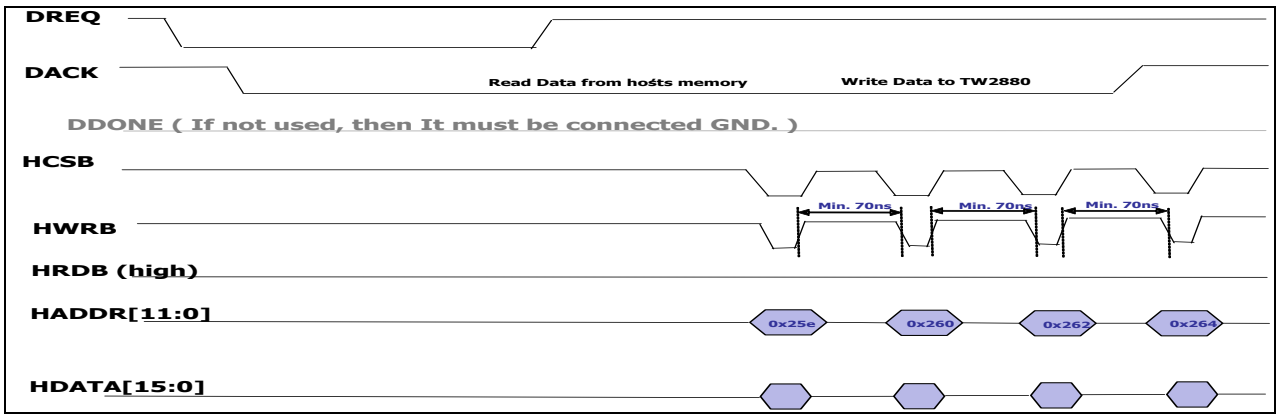
**Transfer Size**

- There are two different transfer sizes; Single and Burst 4.
- DMA holds the bus firmly during the transfer of the chunk of data. Thus, other bus masters cannot get the bus.

**Burst 4 Transfer Size**

Four sequential Reads and Writes respectively are performed in the Burst 4 Transfer.

* Note: Single Transfer size: One read and one write are performed.

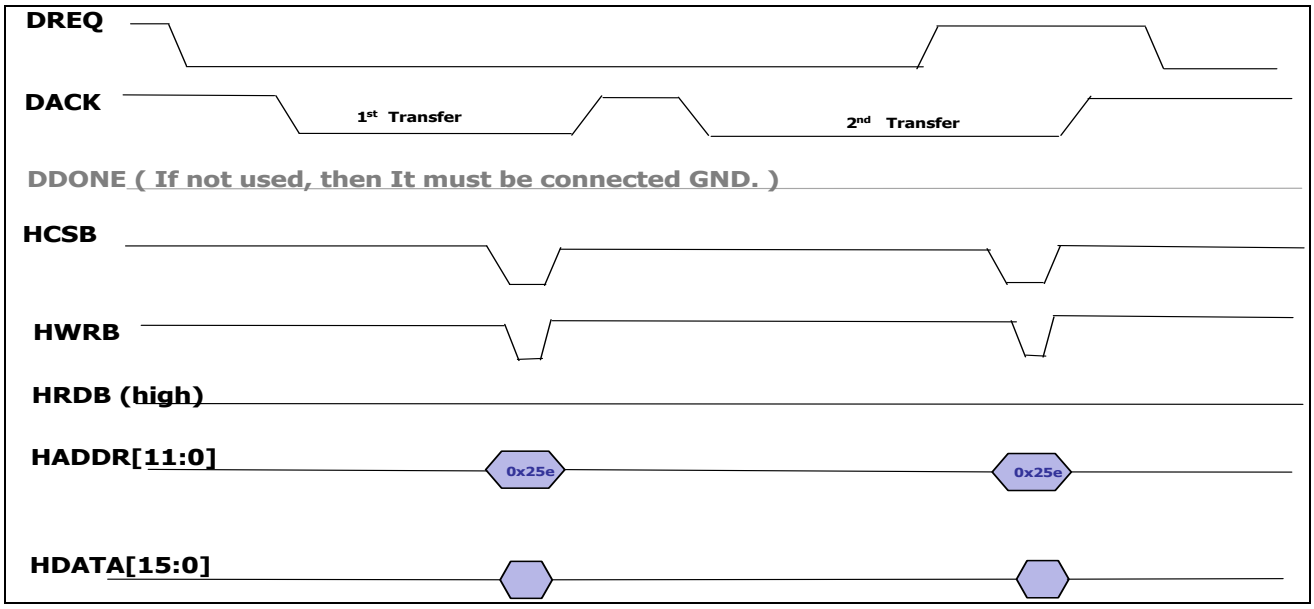



BURST 4 TRANSFER SIZE

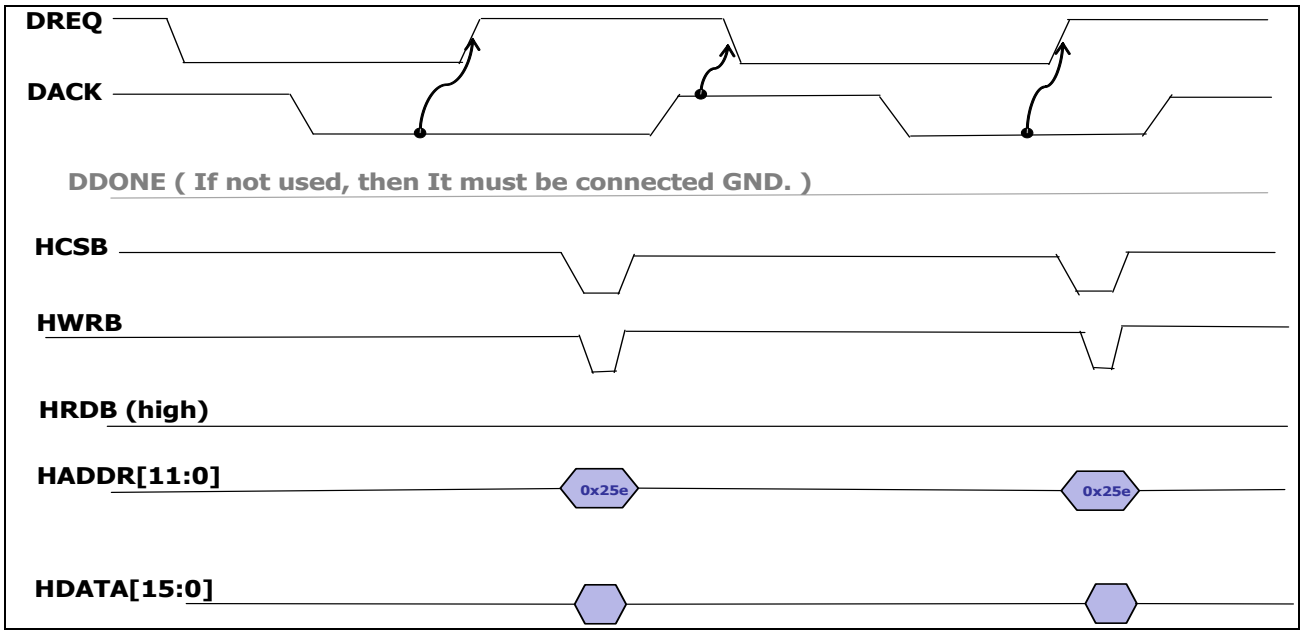
**EXAMPLES**

**Single service in Demand Mode with Single Transfer Size**

**The assertion of DREQ is need for every Single transfer (Single service mode). The operation continues while the DREQ is asserted (Demand mode), and one pair of Read and Write (Single transfer size) is performed.**

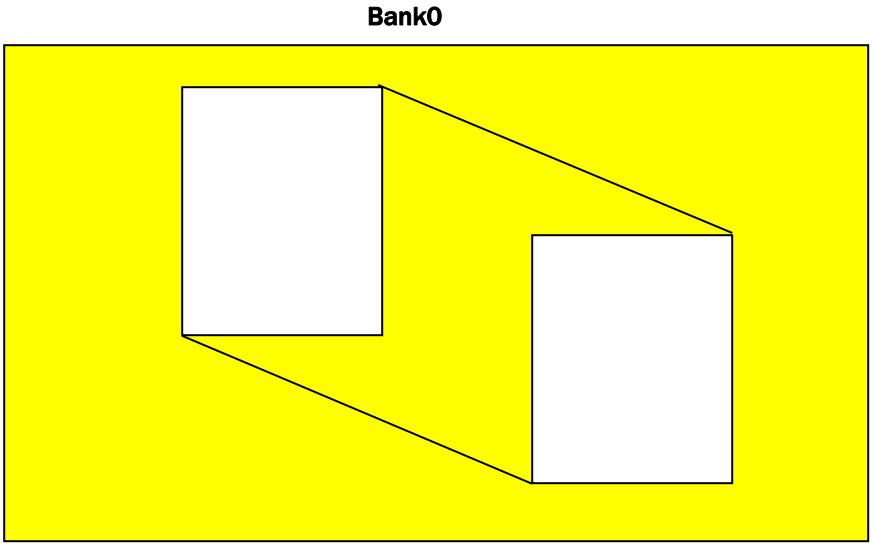



SINGLE SERVICE IN DEMAND MODE WITH SINGLE TRANSFER SIZE

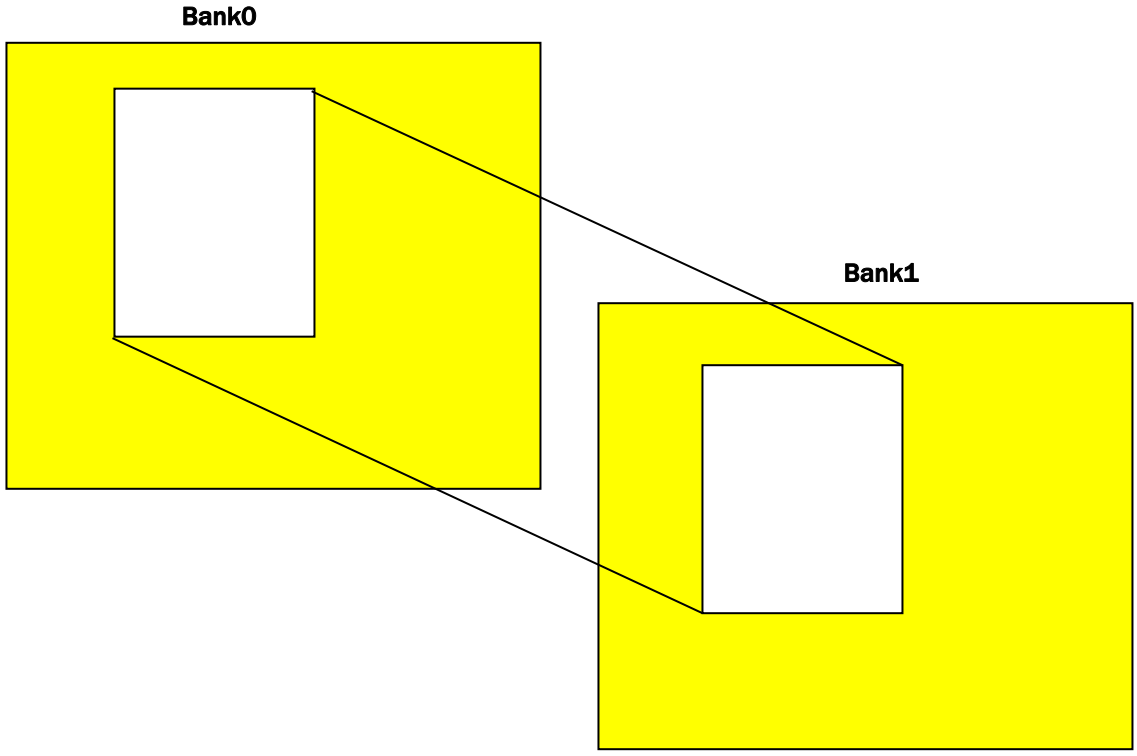



SINGLE SERVICE IN HANDSHAKE MODE WITH SINGLE TRANSFER SIZE

**Example 1 ) DRAM Copy to same bank**




**Example 2 ) DRAM Copy to another bank**



## Register Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x230 | R/W | 0 | Display DRAM Source Vertical Position LSB Register |
| 0x231 | R/W | 0 | Display DRAM Source Vertical Position MSB Register |
| 0x232 | R/W | 0 | Display DRAM Destination Vertical Position LSB Register |
| 0x233 | R/W | 0 | Display DRAM Destination Vertical Position MSB Register |
| 0x234 | R/W | 0 | Display DRAM Vertical Transfer Size LSB Register |
| 0x235 | R/W | 0 | Display DRAM Vertical Transfer Size MSB Register |
| 0x236 | R/W | 0 | Display DRAM Source Horizontal Position LSB Register |
| 0x237 | R/W | 0 | Display DRAM Source Horizontal Position MSB Register |
| 0x238 | R/W | 0 | Display DRAM Destination Hori. Position LSB Register |
| 0x239 | R/W | 0 | Display DRAM Destination Hori. Position MSB Register |
| 0x23A | R/W | 0 | Display DRAM Horizontal Transfer Size LSB Register |
| 0x23B | R/W | 0 | Display DRAM Horizontal Transfer Size MSB Register |
| 0x23C | R/W | 0 | Display DRAM DMA Enable Register |
| 0x240 | R/W | 0 | Host DMA Destination Vertical Position LSB |
| 0x241 | R/W | 0 | Host DMA Destination Vertical Position MSB |
| 0x242 | R/W | 0 | Host DMA Vertical Transfer Size LSB Register |
| 0x243 | R/W | 0 | Host DMA Vertical Transfer Size MSB Register |
| 0x244 | R/W | 0 | Host DMA Destination Horizontal Position LSB Register |
| 0x245 | R/W | 0 | Host DMA Destination Horizontal Position MSB Register |
| 0x246 | R/W | 0 | Host DMA Horizontal Transfer Size Register LSB |
| 0x247 | R/W | 0 | Host DMA Control Register |
| 0x248 | R/W | 0 | Host DMA Enable Register |
| 0x249 | R/W | 0 | Host DMA Transfer Mode Control Register |
| 0x24A | R/W | 0 | Host DMA Status Register |
| 0x24B | R/W | 0 | OSG Total Transfer Size LSB Register 1 |
| 0x24C | R/W | 0 | OSG Total Transfer Size LSB Register 2 |
| 0x24D | R/W | 0 | OSG Total Transfer Size LSB Register 3 |
| 0x24E | R/W | 0 | OSG Total Transfer Size MSB Register |
| 0x25E-0x264 | R/W | 0 | OSG Write Data |
| 0x268 | R/W | 0 | Host DMA Display Pitch LSB Register |
| 0x269 | R/W | 0 | Host DMA Display Pitch MSB Register |
| 0x26A | R/W | 0 | Host DMA Start Address LSB Register 1 |
| 0x26B | R/W | 0 | Host DMA Start Address LSB Register 2 |
| 0x26C | R/W | 0 | Host DMA Start Address MSB Register |
| 0x250 | R/W | 0 | Record DRAM Source Vertical Position LSB Register |
| 0x251 | R/W | 0 | Record DRAM Source Vertical Position MSB Register |
| 0x252 | R/W | 0 | Record DRAM Destination Vertical Position LSB Register |
| 0x253 | R/W | 0 | Record DRAM Destination Vertical Position MSB Register |
| 0x254 | R/W | 0 | Record DRAM Vertical Transfer Size LSB Register |
| 0x255 | R/W | 0 | Record DRAM Vertical Transfer Size MSB Register |
| 0x256 | R/W | 0 | Record DRAM Source Horizontal Position LSB Register |
| 0x257 | R/W | 0 | Record DRAM Source Horizontal Position MSB Register |
| 0x258 | R/W | 0 | Record DRAM Destination Hori. Position LSB Register |
| 0x259 | R/W | 0 | Record DRAM Destination Hori. Position MSB Register |
| 0x25A | R/W | 0 | Record DRAM Horizontal Transfer Size LSB Register |
| 0x25B | R/W | 0 | Record DRAM Horizontal Transfer Size MSB Register |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x25C | R/W | 0 | Record DRAM DMA Enable Register |

## Register Definition

### DISPLAY DRAM SOURCE VERTICAL POSITION LSB REGISTER – 0X230

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Source vertical position[7:0] ( a line) |

### DISPLAY DRAM SOURCE VERTICAL POSITION MSB REGISTER – 0X231

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | W | 0 | Reserved |
| 5:4 | W | 0 | Source bank |
| 3:0 | W | 0 | Source vertical position[11:8] |

### DISPLAY DRAM DESTINATION VERTICAL POSITION LSB REGISTER – 0X232

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Destination vertical position[7:0] (a line) |

### DISPLAY DRAM DESTINATION VERTICAL POSITION MSB REGISTER – 0X233

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | W | 0 | Reserved |
| 5:4 | W | 0 | Destination bank |
| 3:0 | W | 0 | Destination vertical position[11:8] |

### DISPLAY DRAM VERTICAL TRANSFER SIZE LSB REGISTER – 0X234

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Vertical transfer size[7:0] (a line) |

### DISPLAY DRAM VERTICAL TRANSFER SIZE MSB REGISTER – 0X235

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | W | 0 | Reserved |
| 2:0 | W | 0 | Vertical transfer size [10:8] |

### DISPLAY DRAM SOURCE HORIZONTAL POSITION LSB REGISTER – 0X236

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Source horizontal position[7:0] (4 pixels) |

### DISPLAY DRAM SOURCE HORIZONTAL POSITION MSB REGISTER – 0X237

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | W | 0 | Reserved |
| 1:0 | W | 0 | Source vertical position[9:8] |

### DISPLAY DRAM DESTINATION HORIZONTAL POSITION LSB REGISTER – 0X238

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Destination horizontal position[7:0] (4 pixels) |

### DISPLAY DRAM DESTINATION HORIZONTAL POSITION MSB REGISTER – 0X239

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | W | 0 | Reserved |
| 1:0 | W | 0 | Destination horizontal position[9:8] |

### DISPLAY DRAM HORIZONTAL TRANSFER SIZE LSB REGISTER – 0X23A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Horizontal transfer size[7:0] (4 pixels) |

### DISPLAY DRAM HORIZONTAL TRANSFER SIZE MSB REGISTER – 0X23B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | W | 0 | Reserved |
| 0 | W | 0 | Horizontal transfer size [8] |

### DISPLAY DRAM DMA ENABLE REGISTER – 0X23C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | W | 0 | Reserved |
| 0 | W | 0 | Display DRAM DMA enable |

### HOST DMA DESTINATION VERTICAL POSITION LSB REGISTER – 0X240

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Destination vertical position[7:0] (a line) |

### HOST DMA DESTINATION VERTICAL POSITION MSB REGISTER – 0X241

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | W | 0 | Reserved |
| 4:0 | W | 0 | Destination vertical position[12:8] |

### HOST DMA VERTICAL TRANSFER SIZE LSB REGISTER – 0X242

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Vertical transfer size[7:0] (a line) |

### HOST DMA VERTICAL TRANSFER SIZE MSB REGISTER – 0X243

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | W | 0 | Reserved |
| 5:0 | W | 0 | Vertical transfer size [13:8] |

### HOST DMA DESTINATION HORIZONTAL POSITION LSB REGISTER – 0X244

| Bit | R/W | Default | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 7:0 | W | 0 | Destination horizontal position[7:0] (4 pixels) |

### HOST DMA DESTINATION HORIZONTAL POSITION MSB REGISTER – 0X245

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | W | 0 | Reserved |
| 2:0 | W | 0 | Destination horizontal position[10:8] |

### HOST DMA HORIZONTAL TRANSFER SIZE REGISTER LSB – 0X246

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Horizontal transfer size[7:0] (4 pixels) |

### HOST DMA CONTROL REGISTER – 0X247

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | OSG interrupt output enable(Irq PIN) |
| 6 | W | 0 | OSG interrupt enable |
| 5 | W | 0 | hand demand<br>0 = Demand Mode<br>1 = Handshake Mode |
| 4 | W | 0 | Service mode<br>0 = Single Service<br>1 = Not support |
| 3: 0 | W | 0 | Horizontal transfer size [11:8] |

### HOST DMA ENABLE REGISTER – 0X248

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | W | 0 | Reserved |
| 1 | W | 0 | Host DMA enable<br>0 : Disable<br>1 : Enable |
| 0 | W | 0 | OSG DMA enable<br>0 : Disable<br>1 : Enable |

### HOST DMA TRANSFER MODE CONTROL REGISTER – 0X249

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0 | Record DRAM DMA Done status. “1” active |
| 6 | R | 0 | Display DRAM DMA Done status. “1” active |
| 5 | W | 0 | dma_transfer_mode<br><br>0 = Single transfer mode<br>1 = Burst 4 transfer mode |
| 4 | W | | bmp_wait_pin_en - Refer to “Bitmap Wait Enable 1” Register<br>Bitmap wait enable for interrupt Pin<br><br>0 = disable<br>1 = enable |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 3 | W | 0 | big_endian<br><br>0 : little endian<br>1 : big endian |
| 2 | W | 0 | Host DMA Wait enable<br><br>0 : Disable<br>1 : Enable |
| 1 | W | 0 | Bmp_wait_en - Refer to "Bitmap Wait Enable 1" Register<br><br>Bitmap wait enable for interrupt<br><br>0 : Disable<br>1 : Enable |
| 0 | W | 0 | Bmp_Big_Endian – Refer to "Bitmap Wait Enable 1" Register<br><br>"0" : Little Endian<br>"1" : Big Endian |

### HOST DMA STATUS REGISTER – 0X24A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R | | Reserved |
| 2 | R | | Host DMA interrupt signal "1" active |
| 0 | R | | Host DMA wait signal "0" active |

### OSG TOTAL TRANSFER SIZE LSB REGISTER 1 – 0X24B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | OSG total transfer size [7:0] (for Compress Mode) |

### OSG TOTAL TRANSFER SIZE LSB REGISTER 2 – 0X24C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | W | 0 | OSG total transfer size [15:8] |

### OSG TOTAL TRANSFER SIZE LSB REGISTER 3 – 0X24D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | W | 0 | OSG total transfer size [23:16] |

### OSG TOTAL TRANSFER SIZE MSB REGISTER – 0X24E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | W | 0 | Reserved |
| 2:0 | W | 0 | OSG total transfer size [26:24] |

### HOST DMA WRITE DATA – 0X25E, 0X260, 0X262, 0X264

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | W | 0 | Host DMA Write Data |

*** Single transfer mode use only 0x25e address.**

**Burst 4 transfer mode use 0x25e, 0x260, 0x262, 0x264 address.**

### HOST DMA DISPLAY PITCH LSB REGISTER – 0X268

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | W | 0 | Host DMA display Pitch [7:0] |

### HOST DMA DISPLAY PITCH MSB REGISTER – 0X269

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | W | 0 | Reserved |
| 5:0 | W | 0 | Host DMA display Pitch [13:8] |

### HOST DMA START ADDRESS LSB REGISTER 1 – 0X26A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Host DMA Start Address [7:0] |

### HOST DMA START ADDRESS LSB REGISTER 2 – 0X26B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Host DMA Start Address [15:8] |

### HOST DMA START ADDRESS MSB REGISTER – 0X26C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | W | 0 | Reserved |
| 5:0 | W | 0 | Host DMA Start Address [21:16] |

### RECORD DRAM SOURCE VERTICAL POSITION LSB REGISTER – 0X250

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Source vertical position[7:0] (a line) |

### RECORD DRAM SOURCE VERTICAL POSITION MSB REGISTER – 0X251

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | W | 0 | Reserved |
| 5:4 | W | 0 | Source bank |
| 3:0 | W | 0 | Source vertical position[11:8] |

### RECORD DRAM DESTINATION VERTICAL POSITION LSB REGISTER – 0X252

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Destination vertical position[7:0] (a line) |

### RECORD DRAM DESTINATION VERTICAL POSITION MSB REGISTER – 0X253

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | W | 0 | Reserved |
| 5:4 | W | 0 | Destination bank |
| 3:0 | W | 0 | Destination vertical position[11:8] |

### RECORD DRAM VERTICAL TRANSFER SIZE LSB REGISTER – 0X254

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Vertical transfer size[7:0] (a line) |

### RECORD DRAM VERTICAL TRANSFER SIZE MSB REGISTER – 0X255

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | W | 0 | Reserved |
| 2:0 | W | 0 | Vertical transfer size [10:8] |

### RECORD DRAM SOURCE HORIZONTAL POSITION LSB REGISTER – 0X256

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Source horizontal position[7:0] (4 pixels) |

### RECORD DRAM SOURCE HORIZONTAL POSITION MSB REGISTER – 0X257

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | W | 0 | Reserved |
| 1:0 | W | 0 | Source horizontal position[9:8] |

### RECORD DRAM DESTINATION HORIZONTAL POSITION LSB REGISTER – 0X258

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Destination horizontal position[7:0] (4 pixels) |

### RECORD DRAM DESTINATION HORIZONTAL POSITION MSB REGISTER – 0X259

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | W | 0 | Reserved |
| 1:0 | W | 0 | Destination horizontal position[9:8] |

### RECORD DRAM HORIZONTAL TRANSFER SIZE LSB REGISTER – 0X25A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Horizontal transfer size[7:0] (4 pixels) |

### RECORD DRAM HORIZONTAL TRANSFER SIZE MSB REGISTER – 0X25B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | W | 0 | Reserved |
| 0 | W | 0 | Horizontal transfer size [8] |

**RECORD DRAM DMA ENABLE REGISTER – 0X25C**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | W | 0 | Reserved |
| 0 | W | 0 | Copy DMA enable |

# Recording and Output unit

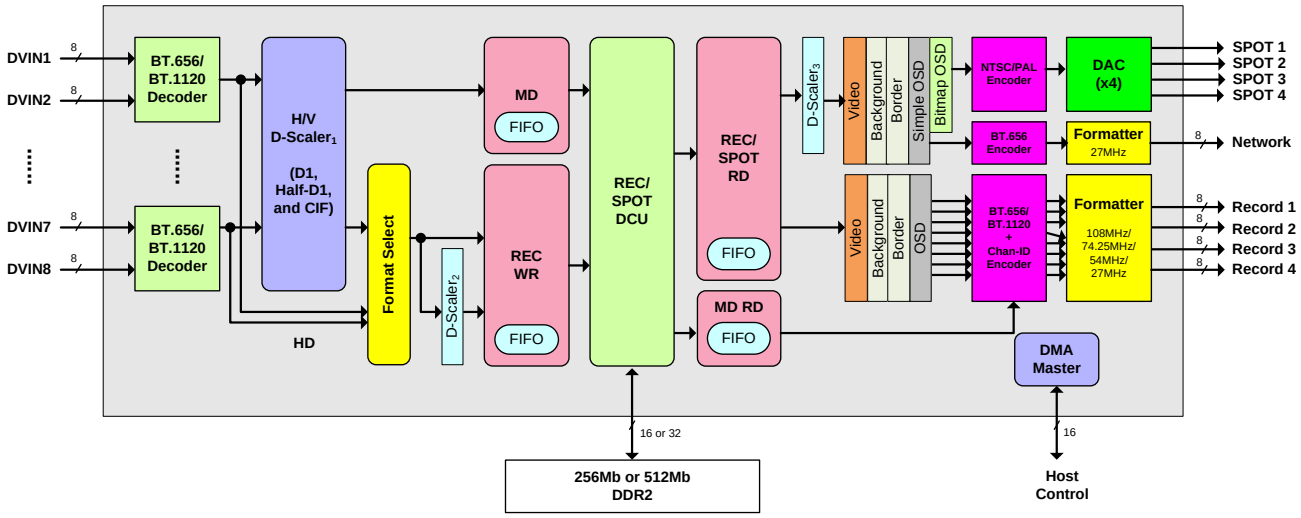
## Introduction

**Recording unit is a very important and complicated part of the TW2880. Due to the DRAM bandwidth limitation, this block uses a separate DRAM controller and a different set of read / writes agents from display controller to complete its functions. To obtain the maximum flexibilities for the user on the output video stream, TW2880 has 2 independent recording units (called write agent) for each input channel. One set of write agents is designed for SPOT output and non-real time output and one set is used for real-time record output. The write agents are grouped into two groups and altogether the number of write agents is 32. Each write agent is capable of choosing camera number (incoming material), recording resolution, recording format and location of the video information inside the DRAM buffer.**

## Functional Description

**The block diagram of recording path is as below:**

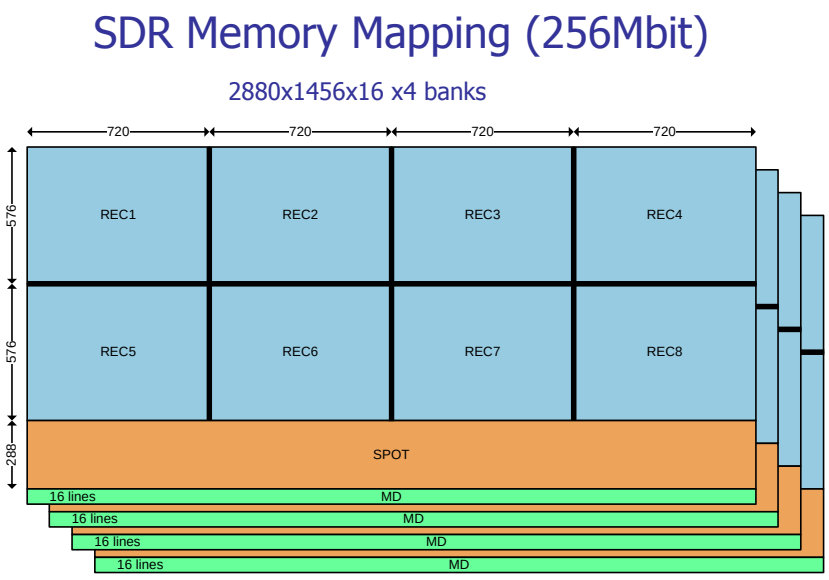
- Down scaler$_1$ unit is shared with display unit but are used with different outputs. For recording path, following fixed resolution outputs are available: D1, Half D1 and CIF.
- Live video stream are stored into proper places in the DRAM buffer according to register setting. Simple Weave de-interlacing technique is used if frame interleaved format is specified.
- A buffer of 4 frames / 4 fields is used for storing video streams. This is needed to prevent video tearing. TW2880 has frame detection / repeat circuitry to ensure the proper read / write sequences. In the Quad and 4D1 mode, read and write bank of each channel can be controlled individually to guarantee read / write sequences.
- The write buffers for SPOT work in a similar way as the live video streams. Record ports and SPOTs can use both recording and SPOT write agents.
- Channels with similar setup should group together to facilitate simple read out functions.
- Motion detection circuit use D1 input as the source for each channel but only use the top field to save cost. The results are stored separately in the specific position of the DRAM buffer.




**BLOCK DIAGRAM FOR RECORD OPERATION**

## Picture Format Control

**TW2880 recording unit supports SDR memory with 256Mb, 128Mb and 64Mb densities. Memory width can be selected as one of 16 or 32 bits. TW2880 needs two x32 SDRAM or four x16 DRAM to form a 64 bit bus in the recording path. Smaller memory will limit the storing capacity of TW2880 and hence is not recommended.**




**256 Mbit Memory will form a 2880x1456 canvas (16 bit per pixel and 4 banks) for user to operate. This memory size can support up to 16 channel D1 output in field interleaved format. There are many ways to store the video channels, but based on the read out unit requirements which will be covered in the subsequent chapters, it is wise to group channels with similar formats and resolutions together to save space. If 16 channel D1 output in Frame Interleaved format is desired, 512 Mb memory are needed.**

## Frame Rate Control Unit

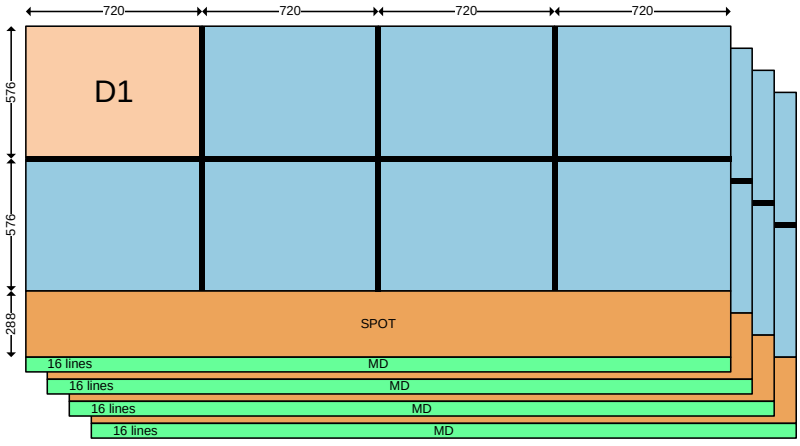
**To prevent video tearing on the record output, 4 frames or 4 fields are used to control frame rate. Each write buffers can select their own read side bank reference from 9 record read ports and 4 SPOTs. In the quad mode and 4D1 mode, each channel has separated frame rate control unit.**

## Record Channels Arrangement Example

**Following are some memory allocation examples for TW2880.**

### 8 D1 CHANNEL FRAME INTERLEAVED REAL TIME OUTPUT

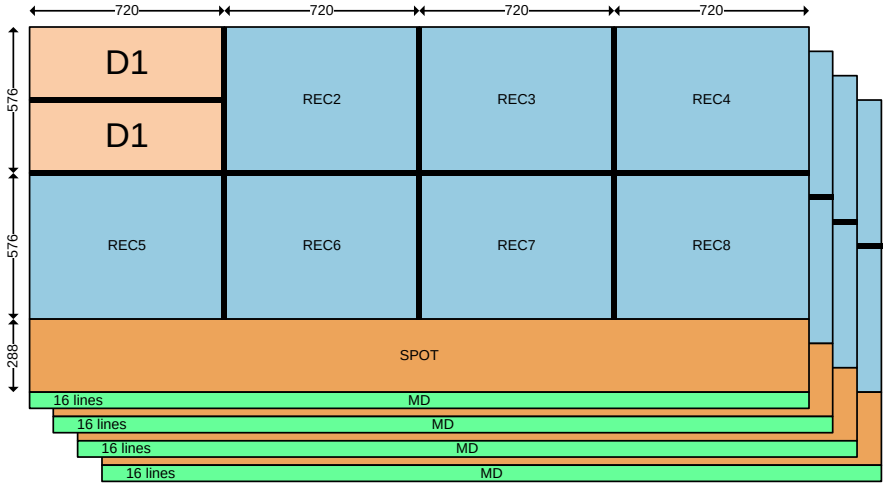
**This mode is used for newer systems when the resolution and frame rate are not to be compromised. To output the video data to the CODEC, the read out unit can be configured at running at 108 MHz (two ports, frame interleaved) or 54 MHz (four ports, frame interleaved) or 27 MHz (8 ports).**



**16 D1 CHANNEL FIELD INTERLEAVED REAL TIME OUTPUT**

**This mode is designed for newer high-end system as all 16 channels are output in D1 resolution. To output the video data to the CODEC, the read out unit can be configured at running at 108 MHz (four ports, field interleaved) or 54 MHz (eight ports, field interleaved).**

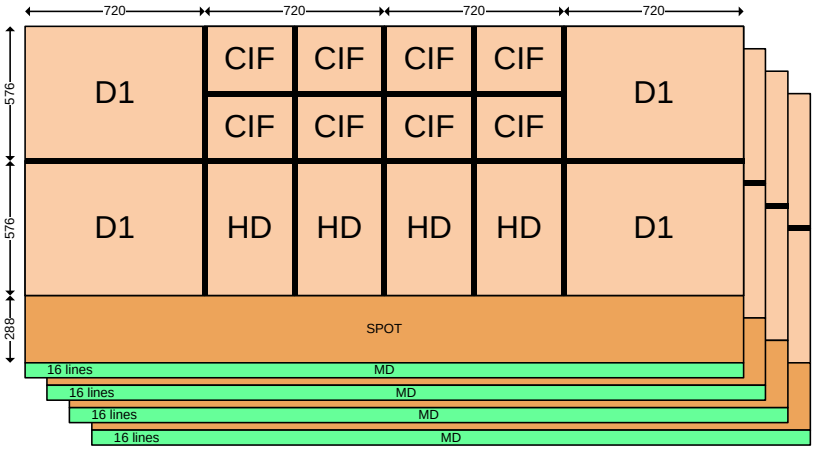
## 16 D1 field interleaved real time record




**16 CHANNEL MIXED FORMAT REAL TIME OUTPUT**

**Here is example showing how several different resolution of video channels are saved into the TW2880 recoding memory. Notice in this setup all 16 write agents are deployed.**

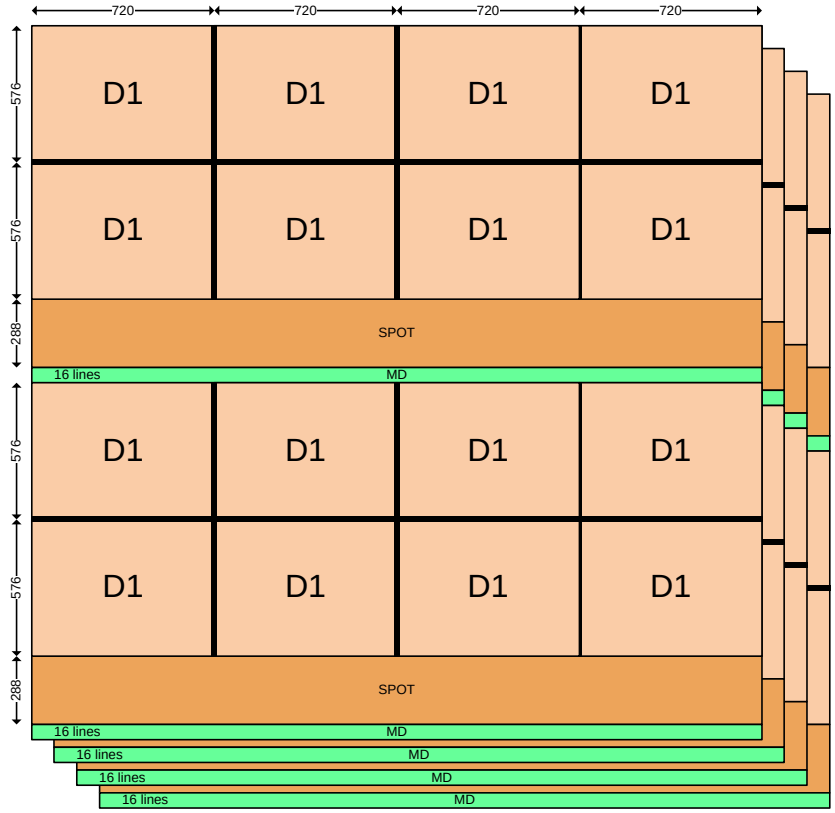
## 16 Mixed frame interleaved real time record




**16 D1 CHANNEL FRAME INTERLEAVED REAL TIME OUTPUT**

**This mode is designed for newer high-end system as all 16 channels are output in D1 resolution. To output the video data to the CODEC, the read out unit can be configured at running at 108 MHz (four ports, frame interleaved) or 54 MHz (eight ports, frame interleaved).**

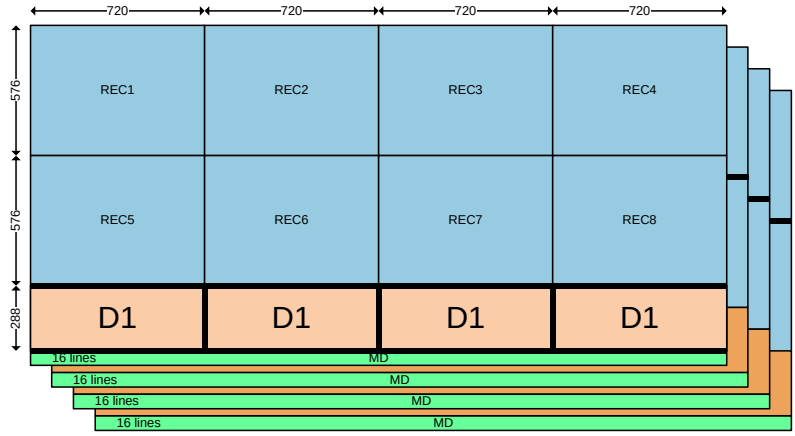
## 16 D1 frame interleaved real time record



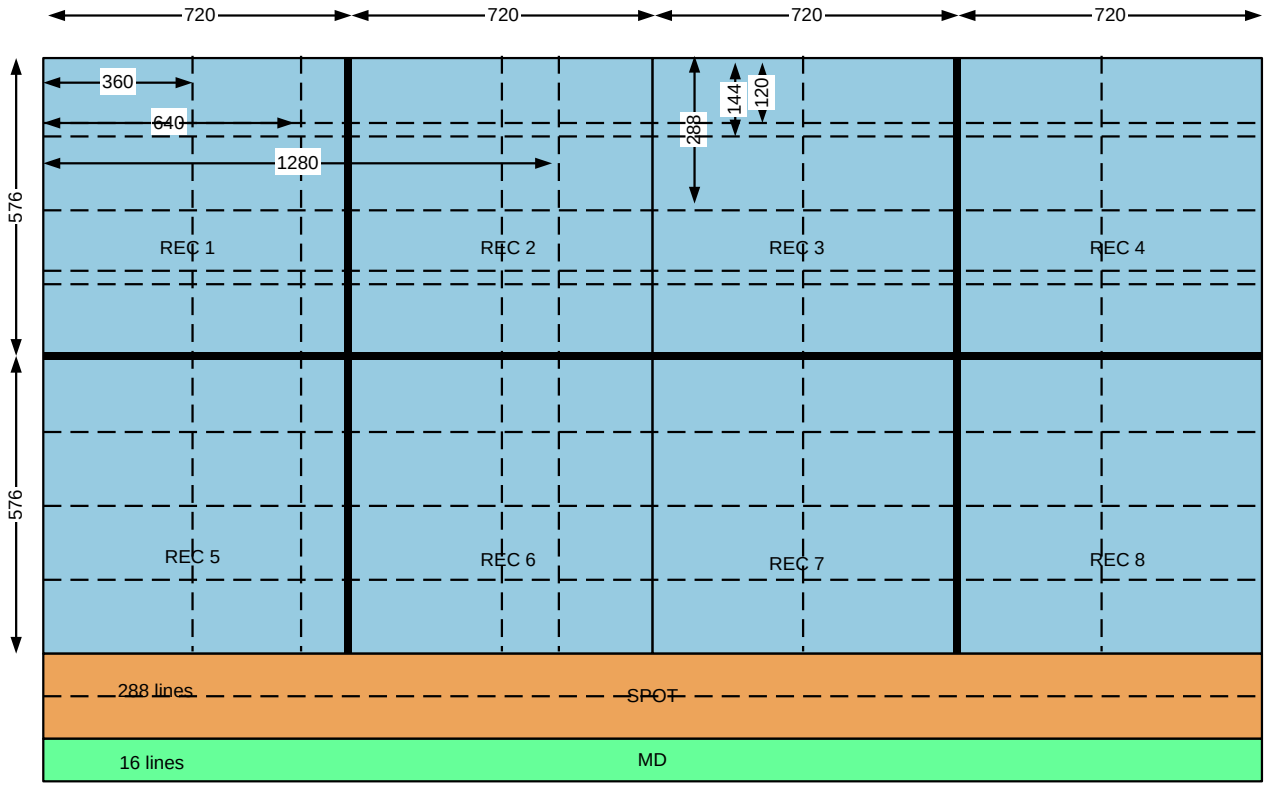

**SPOT MONITOR MEMORY SPACE**

**In the 256 Mbit memory, SPOT can use 2880x288 reserved space. This capacity is big enough to accommodate 4 D1 output for four SPOT. If using quad or half D1 mode to accommodate multi-channels, make sure the recording positions are entered correctly. In the 512 Mbit memory, SPOT can use up to 2x2880x288 memory space.**

## SPOT memory space



# MUX SDRAM Mapping (detailed)



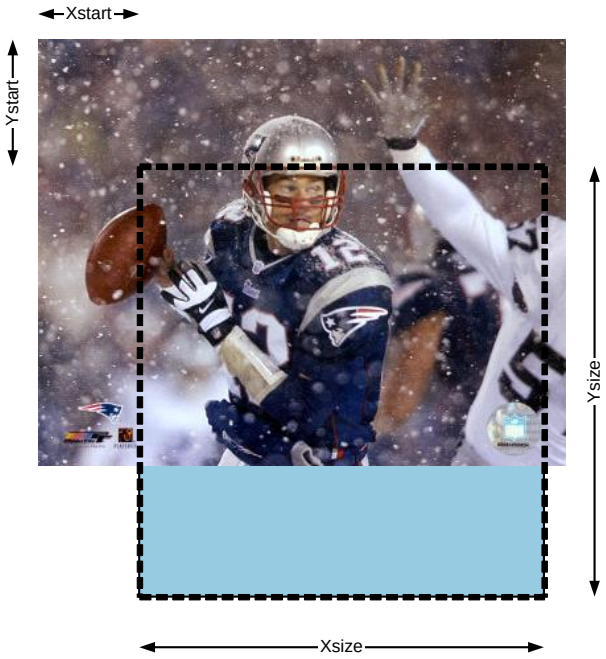

**RECORD POSTIONS IN DETAIL**

**The above diagram shows all kinds of possibilities in choosing channel positions. These positions are selected in recording buffer control register B where the user select a pre-determined horizontal and vertical positions. Because there are too many possible locations to choose from so please refer to manual for actual settings. The offset within a channel is also possible. This is done by changing the starting positions of each channels (Xstart and Ystart). The Xsize and Ysize is determined by the output format. If the size is small than the recording format chosen, cropping will occur. The variable starting position is used to chop off unnecessary artifact in the beginning of the channel.**

**SPOT WRITE BUFFER**

**SPOT write buffers function the same way as recording buffer. The only difference is vertical location. SPOT write buffer can be used by recording unit as a possible inputs.**

## Channel Record Position Control



# Recording Unit

## INTRODUCTION

**TW2880 provides very powerful and versatile recording options and interfaces to the backend CODEC. The functions and output formats for each port are discussed in details in the subsequent sections. There are eight regular output units' shared 4 output pin ports and one network output unit in TW2880. Except the network port, which is usually run at 27 MHz, user can configure each port independently to run at different frequencies and output formats. However, due to the bandwidth limit, not all the output ports can run at the highest frequencies at the same time. The last five port (port 5 to port 9) are equipped with a flexible output option that even one frame per second for a certain channel is supported.**

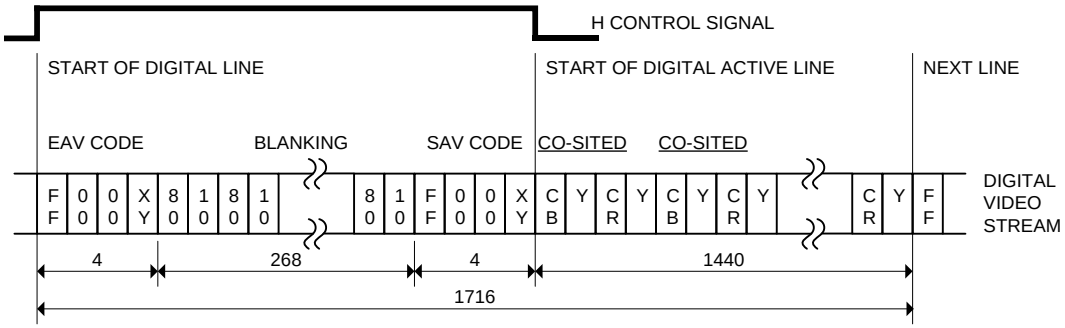
## RECORD OUT FORMAT CONTROL

**The basic output format of TW2880 is followed the BT.656 standard with EAV-HBI-SAV-ACTIVE style of coding. The D1 output sequence is illustrated in the next two figures. The differences between each output format lies in the number of bytes in each format and flags in the HBI area. In the subsequent sections, when we mention a field output or frame output, what we really meant is a series of EAV-HBI-SAV-ACTIVE lines shifting out follow these rules and output continuously to the outside world.**

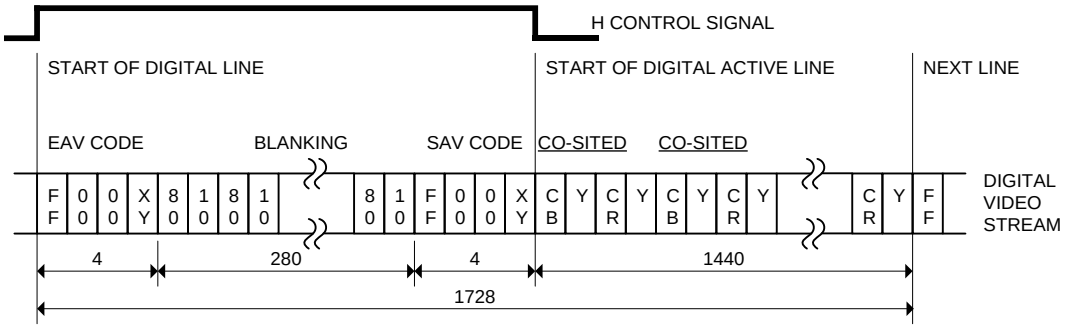
**Several things determined the output video stream:**

- Output frequency:Each port can be configured to run at 108, 54 or 27 MHz. Higher the frequency, higher the data rate and hence more channels supported.
- Output format: Three formats are supported, (1) Field interleaved. (2) Progressive Frame interleaved (3) Interlaced Frame interleaved
- Output resolution: Four resolutions are supported: D1, special CIF, 4D1, 720p and 1080i or 6VGA.
- Output channel selection: This will determine which channel will output from this channel. Because the channel format has been set in the recording side, this setting is actually adjusting the read out DRAM area and order.

**BT.656 8-BIT PARALLEL INTERFACE DATA FORMAT FOR 525/60 VIDEO SYSTEMS**




**BT.656 8-BIT PARALLEL INTERFACE DATA FORMAT FOR 625/50 VIDEO SYSTEMS**



### D1 VIDEO FRAME DATA ARRANGE

Total byte number in a NTSC frame is: 1716 X 525 = 900,900 Bytes where PAL frame has: 1728 x 625 = 1080,000 Bytes. One thing needs to point is the vertical resolution is 525 lines divided as 262 / 263 in the NTSC and 625 lines divided into 312 / 313 in PAL. If frame-interleaved sequence is sent via this interface, the TW2880 will not toggle FLD flag in the VBI status word like when TW2880 send out the interlaced sequence. For a multi-channel sequence, the output clock will get double or quadruple to maintain the output frame rate. The actual frame data will be sent out sequentially follow CH1-CH2-CH3 style and on to the next frame.

### CIF VIDEO FRAME DATA ARRANGE

TW2880 support a special mode where the output resolution is a quarter of the original D1 frame. The mode is called CIF frame. In NTSC case, the number of byte in a CIF frame is: 900,900 byte / 4 = 225,225 bytes. This number cannot be evenly divided by two. The number of bytes in each section is difficult to assign. TW2880 is using 828 x 272 for a CIF frame with the fourth frame's VBI includes a extra 36 clocks.

828 x 272 x 4 + 36 = 900,900

As for PAL, the number is 750 x 360 x 4 = 1,080,000 Bytes.

The CIF is only different in the number of bytes in the output format. The rest is the same as the normal frame output. This format is especially useful when a great number of channels need to be output in a single port. For example, a port running at 108 MHz with a 16 channels in CIF resolution can be accomplished.

### 16-BIT VIDEO FRAME DATA ARRANGE

In addition to the 8-bit BT.656 interface, TW2880 also supports 16-bit video interface with ~~external syncs (BT.601) or~~ embedded sync (BT.1120) format. Data port in both format are divided into upper byte as Y data and lower byte as C data. For the BT.1120 format output TW2880 will output the correct EAV, SAV syntax with the 10-bit extension. ~~For the BT.601 external sync will get sent out.~~ Both formats can be program into different timing and clock rate to send out non-standard video stream.

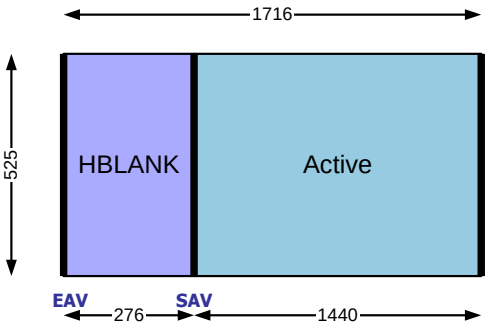
### RECORD OUTPUT METHOD

Two kinds of output methods are provided: real time mode and switch mode. The most important feature provided by the real time mode is synchronization. The different channels in the same video output sequence are guaranteed to start from the beginning. Other benefits include less interrupt to the host and easy data management. Switch mode is provided to user don't care about the frame but care much flexibilities especially in the low frame rate low storage requirement environment.

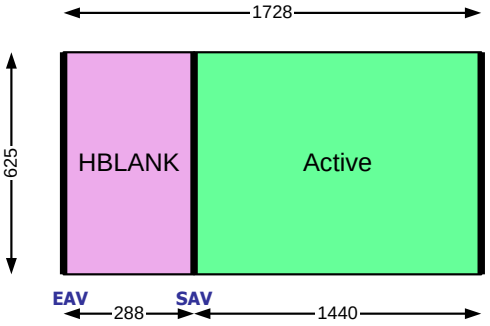
### MULTI-MODE FORMAT CONTROL

This is a special mode to let user achieve the ultimate flexibility in outputting recording channel information. The regular port (port 1 to port4) can select between 1 to 4 source channels. The port 5 to 9 in TW2880 has a larger (128) than normal (4)

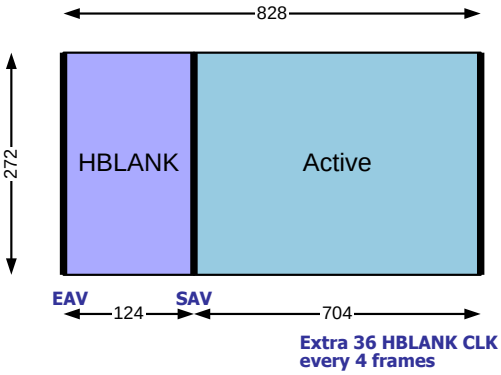
NTSC, D1



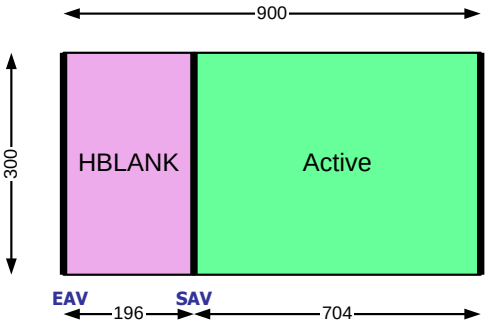

PAL, D1




NTSC, CIF



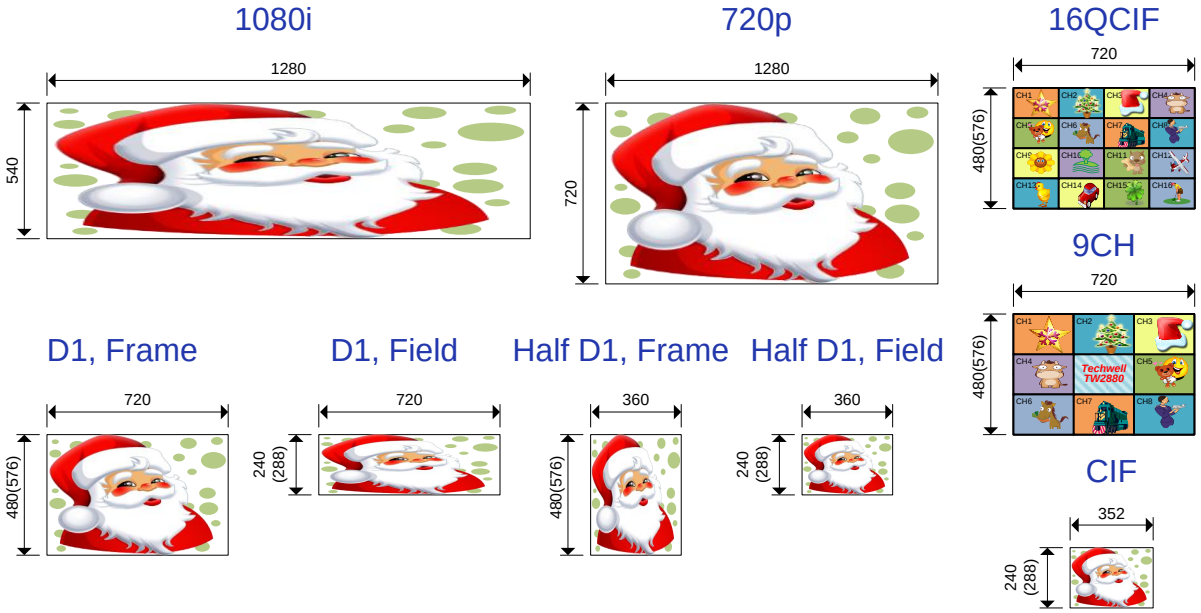

PAL, CIF



**source channel select option. So if configured as running at 27 MHz, user can reduce the recording frame rate to less than one frame per second. Although the channel selection is quite flexible, similar resolution item must be grouped into a single frame except D1 selection. For example, a QUAD frame will consume four CIF selections in the list and if not satisfied garbage data will appear in the sequence. That is to say, if quad frame is output through the sequence, only 32 frame can be entered.**

**RECORDING FORMAT EXAMPLE**

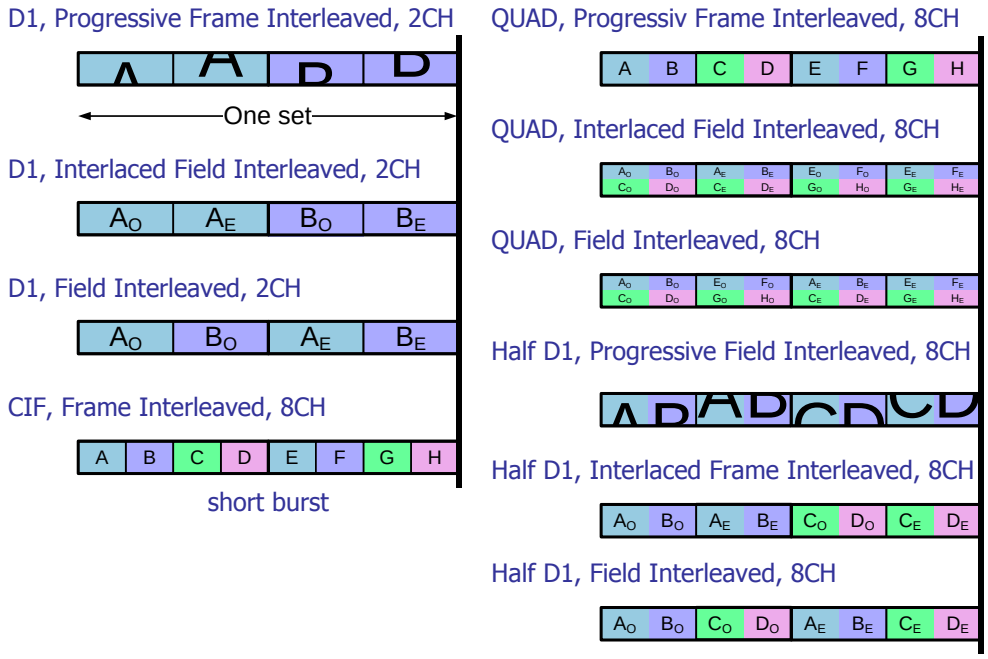
**The following diagrams are the recording resolutions and sizes that TW2880 supports. D1 frame means record in full D1 resolution and do frame interleaved output. This is implied when the vertical sync comes, half of picture is sent followed by the other half. Frame mode has two output methods: progressive and interlaced. D1 field means record in full D1 resolution and do field interleaved output. The odd scan lines will be sent first followed by even scan lines. TW2880 can record the image with high definition size included 1080i and 720p.**



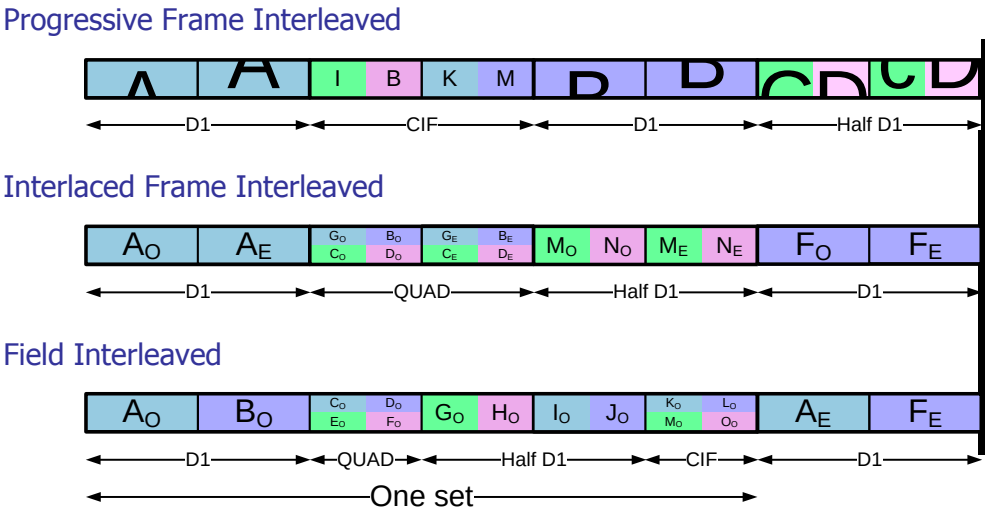
**RECORD OUT CHANNEL OUTPUT EXAMPLE**

**The following is an illustrated example about the possible output sequence of one port running at 54 MHz.**

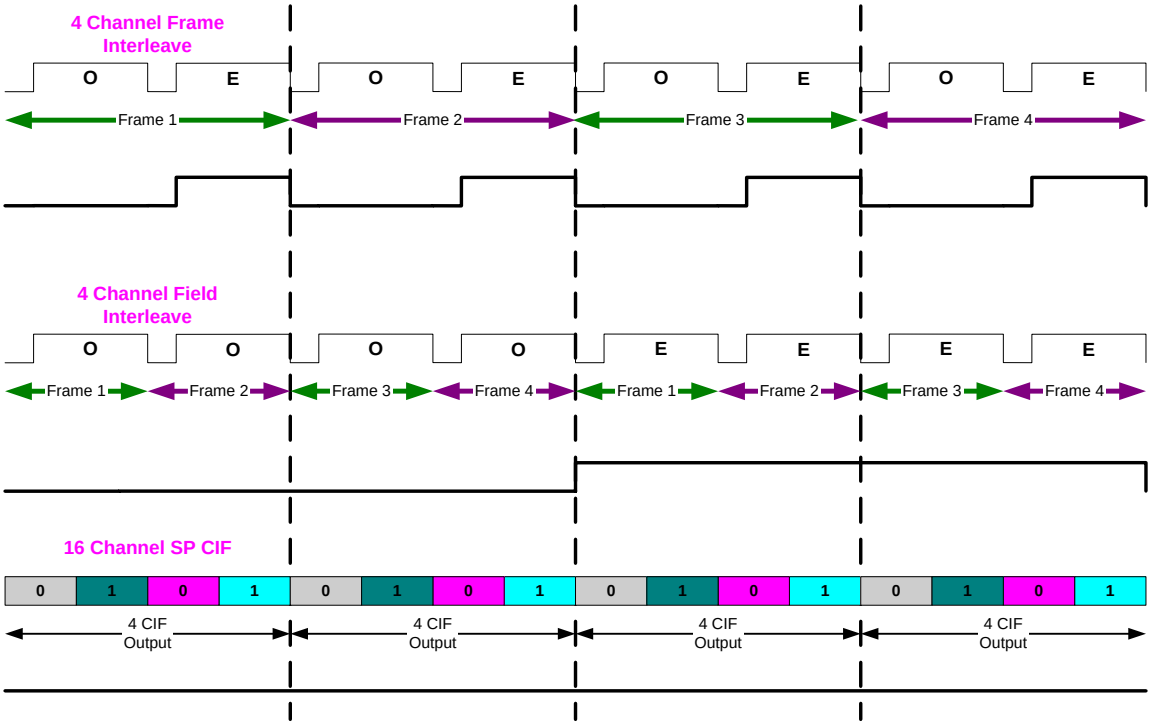
# Record out Option @ 54 MHz




# Multi-mode record out @ 27 MHz



From the above sequence, one can easily tell the difference between frame-interleaved and field-interleaved. This also has great impact on the receiving CODEC because if synchronization between the channels happens, FMI mode will repeat the same frame and FLI mode will repeat the same field and might causing problems for the backend. The scan line output order in the FMI mode has two choices, the default one is the scan lines are output sequentially and followed the picture's nature order. In addition to this progressive FMI mode, TW2880 also support another method which the output scan lines are followed the even and odd field orders into a so called "interlaced FMI mode". This mode is designed to accommodate different CODEC input format.




## RECORD PORT MISC. ITEMS

Port 1-4 has only four sources. User can specify source number from one to four. Output resolution if selected as CIF, this port will output four small CIF frames according to the source registers. If specify as D1, three read out options are available: straight, splitX and splitY. The first method is this port will output a D1 frame based on the source register's origin and does not care about the image size of this source. So using this method it is possible to output all 16 channels in just 4 QUAD. The next mode is used when the two source images want to output in a single D1 frame. The last mode is using together with splitX and output a CIF image as QUAD. The main difference is the location of these sources needs not to be adjacent in the DRAM memory map.

Multi mode port has 128 entries to constitute a customary output sequence. This sequence also has the straight mode and the automatic mode for grabbing the images. The table sequence programming can be done even during the port is enabled. The register write operation will always has the priority over the internal state machine. User can adjust the sequence whenever they seems fit.

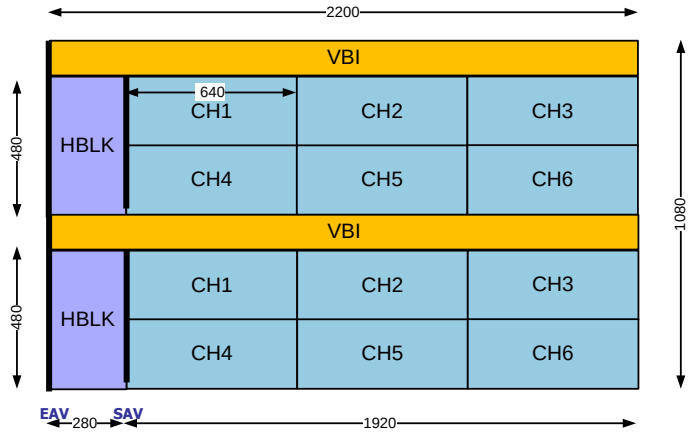
## Special Output Format

**The previous section is mainly discussing output format in D1 or in special CIF resolutions. These two formats can be accepted by most CODEC manufacturers in a multi-channel environment. However, there are situations where CODEC manufacturer does not want to decode EAV-SAV field mark and still want to do multi-channel encoding. For this kind of application, TW2880 supports two more modes to output multi-channel material but with high strobing clock (i.e. 108 MHz), namely, the 4D1 mode and BT.1120 (6VGA) mode. These two modes are the products of utilizing the new transmission standards in the HD era.**

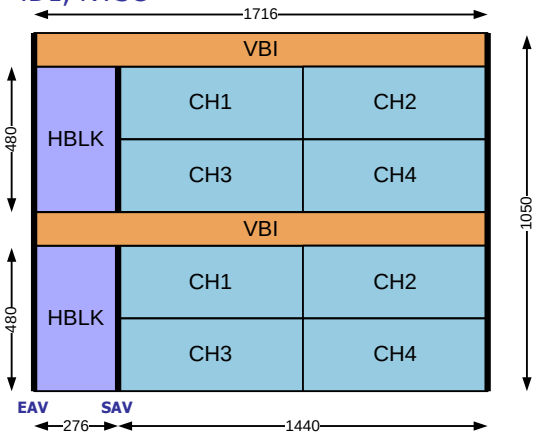
**Basically, 4D1 mode evolves from the old BT.656 mode to accommodate four D1 channel at the same time. It keep the EAV-SAV field mark but enlarge the horizontal pixels to 1440 so two D1 frame can be transmitted. In the vertical area it follows the interlaced format that odd field of the first two D1 streams are transmitted followed by the even fields.**

**The output clock has to set to 108 MHz if real-time recording is desired. Lower output clock will cut down the frame rate.**

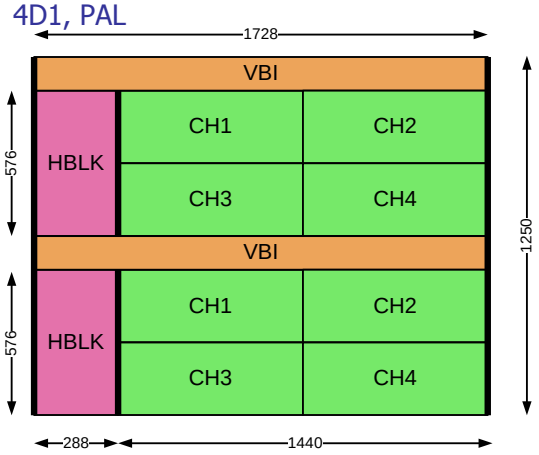
6VGA (BT.1120)




4D1, NTSC




4D1, PAL




**The 6VGA mode is basically the same except the data arrangement and the output clock. The storing selection needs to set as VGA (640) and use Hstart to center the channels. Output clock has to be 148.5 MHz to make it BT.1120 legal.**

## Record Port Capability Chart

| FORMAT | SPECIAL | FIELD INTERLEAVED | INTERLACED FMI | PROGRESSIVE FMI |
|---|---|---|---|---|
| CIF | O | N/A | N/A | N/A |
| D1 | N/A | O | O | O |
| 4D1 | N/A | O | O | N/A |
| BT.1120 | N/A | O | O | N/A |

*** O : Support, N/A : No support**

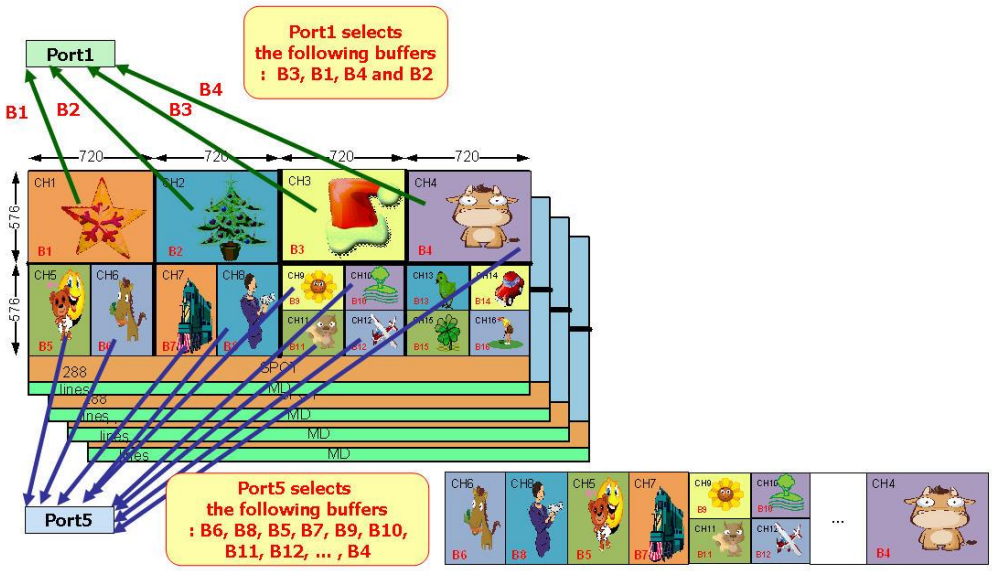
One thing which needs to be pointed out is that the user has the highest flexibility when composing the D1, and 4D1 recording streams. For example, a D1 resolution stream can have be a quad and the 4 channels needed can be reside in any legal locations of the DRAM buffer. Similarly, 4D1 resolution can be used to send 16 CIF through one output pin port. BT.1120 output stream, however, due to the 6 channels nature, does not have this freedom. All channels in the streams needs to be put into adjacent locations.

## Multi-mode Record Port Programming Example

Unlike normal record port 1, 2, 3, 4 can only handle 4 channel images, multi-port 5,6,7 8 can accommodate 128 channel images in a single output stream. By using these options, user can composite the ultimate video stream to the subsequent back end. The source list of the multi-port output is controlled by the three control registers, take port 5 as an example, it is register 0xC35, 0xC36 and 0xC37. 0xC35 specify the location of the table and 0xC36 specify the recording buffer number (not to be confused with the channel number). The reason not to use channel number here is: for one channel user can select different size of image to store. So the important thing to record here is the buffer number. 0xC37 stores the number of valid entry in the table. It doesn't has to be full length.

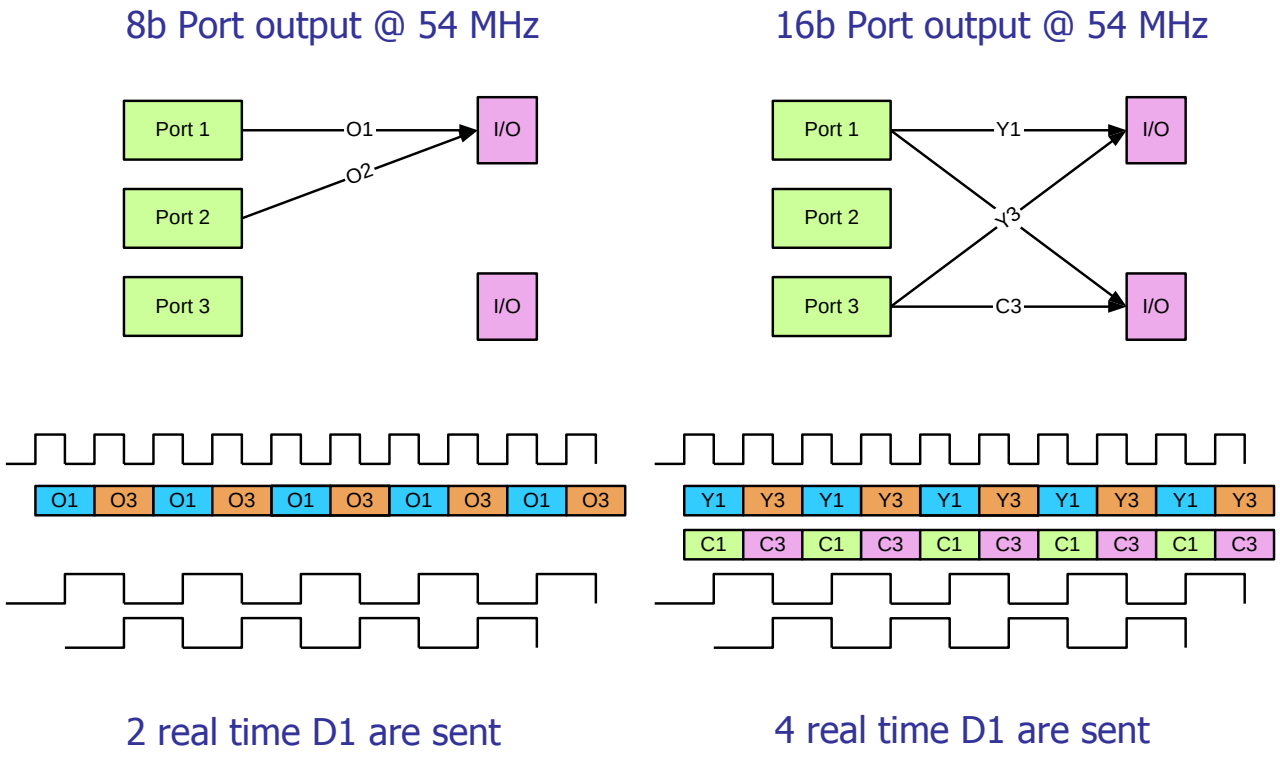
Recording port should be off (not enable) when inputting the source into the table. If users choose the output resolution to be D1, sources smaller than the D1 size will be merged into one D1 frame based on the current size. If users choose the output resolution to be CIF then entry table entry represent a source. This is done for obvious reason. Please pay attention when input the source list into the table otherwise garbage data may appear at the output.

There is no restriction on how to arrange the source list other than the resolution limit described above. Table mode is designed for covering the most of the channel with the lease amount of video data generated in regular non-alarming use. User can adjust or repeat a same buffer several times to increase the total coverage based on importance. Real time coverage is also possible but user needs to remember since output clock has only few selections so it limits the number of sources.




## Output Port

### INTRODUCTION




**TW2880 has 9 record units and 5 set of physical ports, total 54 pins (5 * (8+1) + 3 * 2). The main reason not to let each record unit has its dedicated output pins is to save pins. The output data and clocks have several rather complicated relationships. The following paragraph is used to clarify this:**

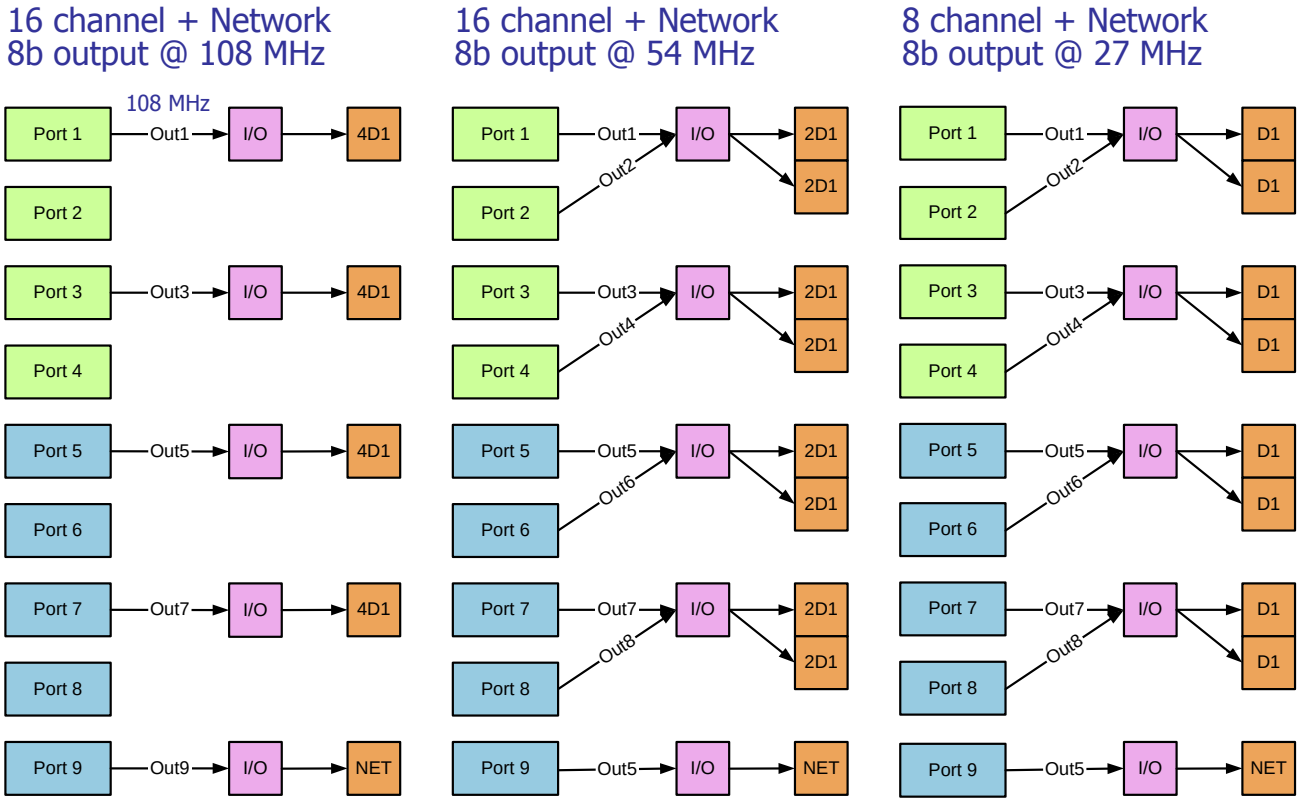
**Suppose we set the two output ports running at 54 MHz which will output 2 real-time D1 stream on each port. The user can set the output port running at 54 MHz. For 8 bit mode, the output data stream will run at 108 MHz with**

**two out-of-phase clocks running at 54 MHz to differentiate the video data. The diagram on the right shows the relationship between the output data and the output clock. One pin set can output two 2-D1 streams to two CODEC chips. In the 16 bit case, the channel number is double so it can support two 4-D1 streams. If the user set the output port running at 27 MHz, for 8 bit mode, the output data stream will run at 54 MHz and one pin set can output two D1 streams to two CODEC chips. So the output channels are cut in half in 27 MHz mode. Other things needed to know for the user:**

- If the associated output ports in a pin set is disabled then number of output channels is cut in half.
- The output rate of the two associated ports must be set to the same value.
- Port 9 is used for network output so its own I/O pins. The output rate can go up to 108 MHz if necessary but since it is sharing bandwidth with other unit so the total performance is subject to active setting.

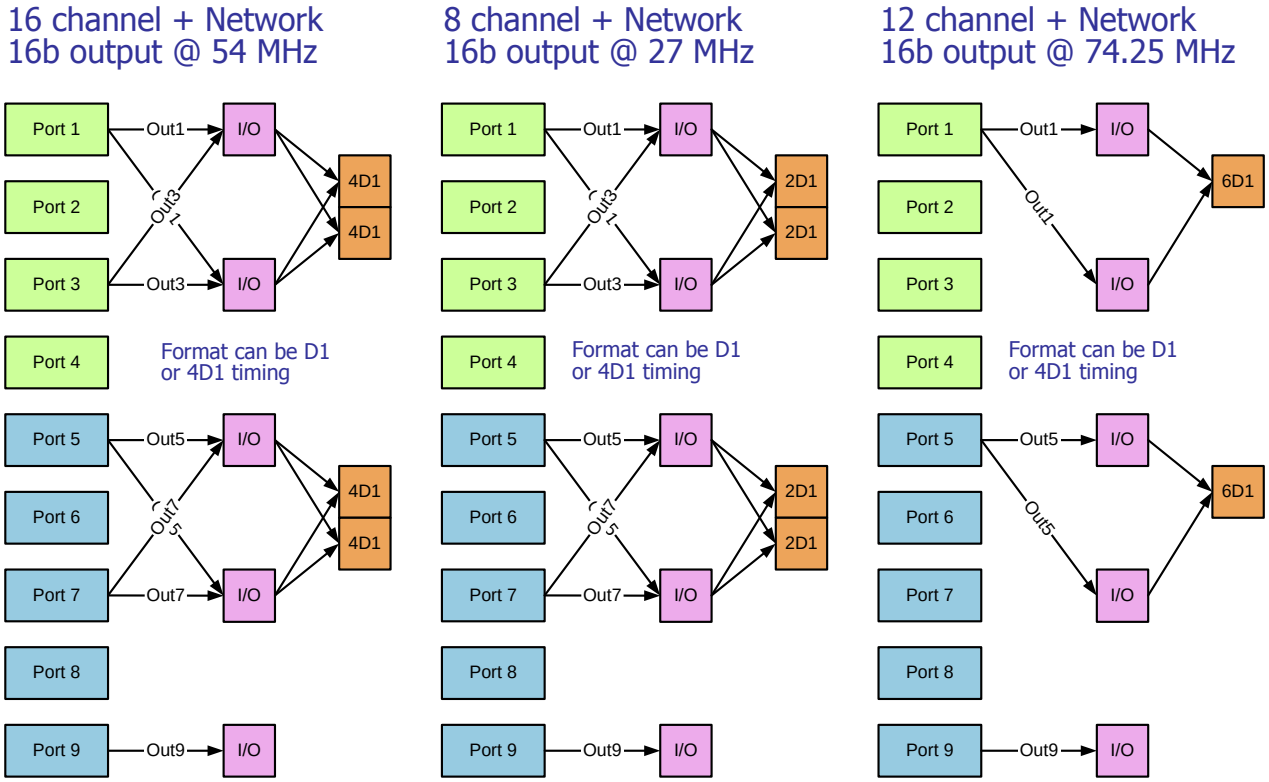
**8-BIT MODE**

**This mode basically deal with BT.656 embedded sync. format. It is possible to support 4 output streams running at 108 MHz and output 16 channels using four recording ports or 8 output streams running at 54 MHz (two clock multiplexed method) and output 16 channels using four recording ports, or 8 output streams running at 27 MHz (two clock multiplexed method) and output 8 channels using four recording ports.**



### 16-BIT MODE

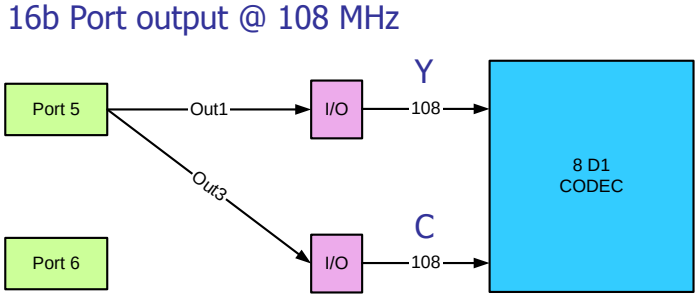
**TW2880 also supports 16-bit output mode where the Y and C data are separated. This setting actually includes support of BT.601 with external sync. and BT.1120 with embedded syncs. For setting at 54 MHz, TW2880 can send out two 4 D1 streams into a single CODEC using two ports. The clock and the data arrangement is mostly the same as in the corresponding 8 bit setup.**




### 16-BIT COMPONENT MODE

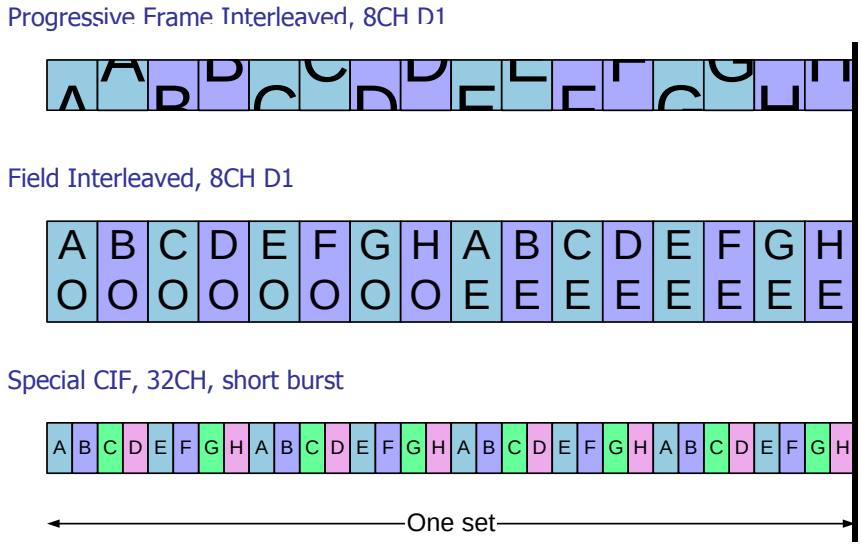
**In a addition to the traditional 16 bit output mode, there is a new 16 bit mode which is designed specifically for new generation high speed CODECs. With this mode, if running at 108 MHz, TW2880 can generate 16 bit 8D1 frame or field interleaved video stream. The idea is simple, setup one recording unit the usual way but instead of output 8 bit Y / C component stream sequentially to the outside, it will output both Y / C components at the same clock to the subsequent modules like OSD and embedded sync module.**

**This means the FIFO output rate doubles as comparison to the normal mode. Because the output speed is clocked at 108 MHz for both component output so 8 D1 channel real time data can be supported. One thing to remember is because the number of output channel is larger than four so the recording unit used is limited to 5, 6, 7, and 8. All four output resolutions are supported but choose BT.1120 will result to a big 8D1 frame (6864x960 for NTSC).**

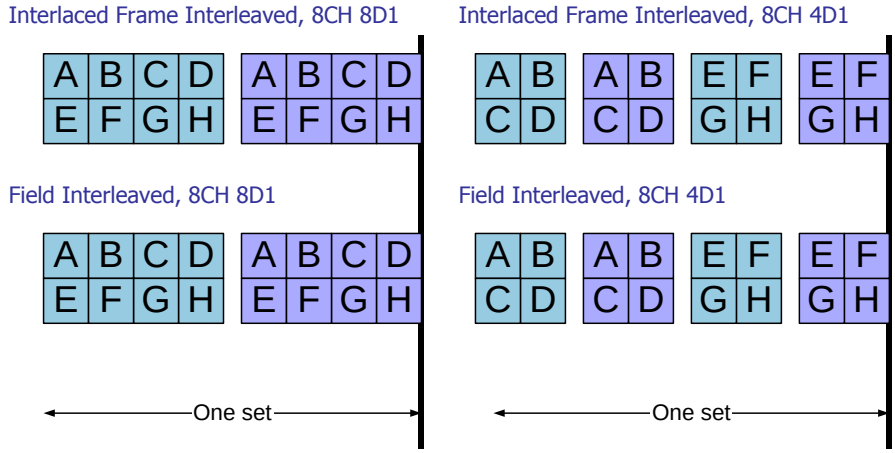


**To use this mode, after setting the recording buffers and the output options, user need to set the responsible bits in register 0xCF0 to enable this special component mode and choose the desired video component. As stated before, only two port 5, 6 are needed for output 16 channel D1 real time to the CODEC. All output resolutions are supported except 6VGA mode, the resulting streams will have different effects based on the resolution chosen. For example, D1 stream will have 8 D1 channel come out before it will repeat itself. NTSC frame interleaved 4D1 mode will have two 1440x960 frame output before repeating itself. Please see the following diagram to for clarification.**

## Record out Option @ 108 MHz, 16 bit

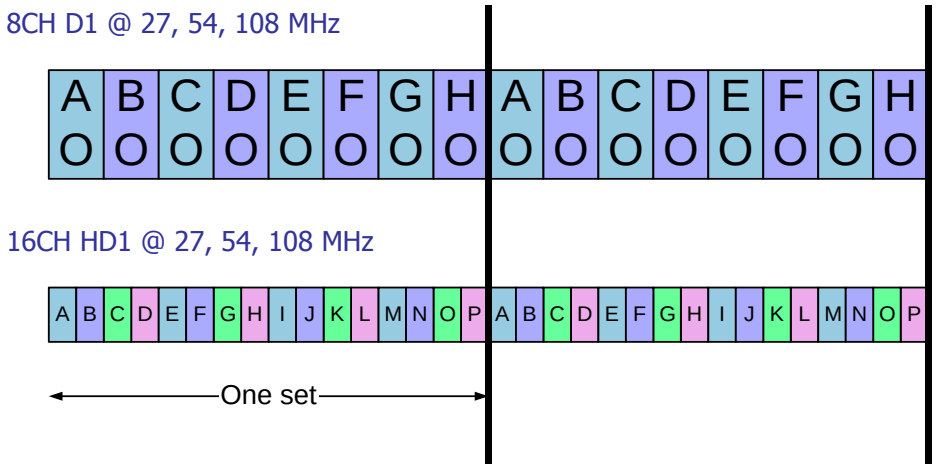



## Record out Option @ 108 MHz, 16 bit



**FIELD SWITCHING MODE**

# Field Switching Record out Option

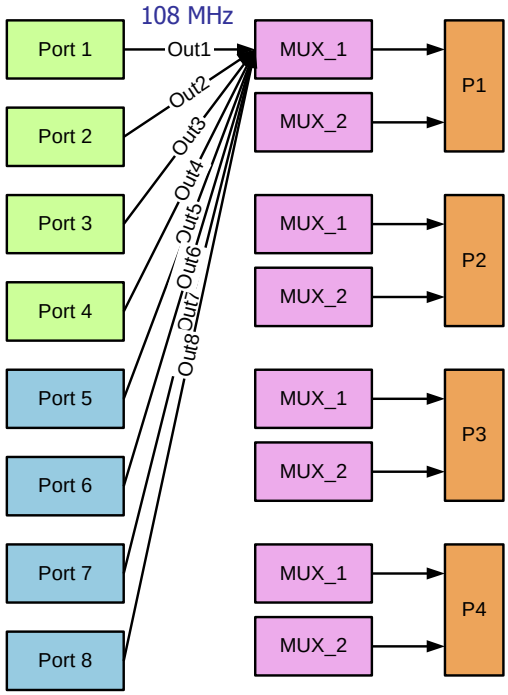



**Field switching mode is designed for generating low bit rate stream to cover more channels at half of the normal data rate. To use it, users need to set registers 0xCF1-0xCF4 to determine which buffer is in field switching mode and which field get recorded, the second thing is set the port run at field interleaved mode then the resulting stream will be in field switching mode. D1, half D1 and quad are all legal output format. Another thing to remember is while user still can select output frequency and channel number in a stream, most of the time the resulting stream are non-real time because the channels covered. This is means each field which representing a channel in the resulting stream may come from different input field number.**

**32 BUFFER MODE**

**The 16 SPOT buffers can be used by the recording port 5-8 to increase the total available buffer to 32. This is done by first setting the corresponding bits in SPOT Buffer Control Register 0xCCE – 0xCF to one. Setting these bits are to change the SPOT buffers feedback input in the frame rate control unit. After setting these bits, user will have 32 buffers at their control to create more options. In the buffer naming, 0x0-0xf is the original recording buffers and the 0x10-0x1f is the spot buffers. For example, user can set buffers 0x0-0xf to record D1 resolution and output 16 channel D1 stream while having a special 16CIF video stream to go out using SPOT buffers at the network port. SPOT function is supported if DRAM bandwidth is enough (182 MHz).**

MUX Selection




**OUTPUT PIN SOURCE CONTROL**

**The source1 and source2 of each output pin set are programmable. The source1 is control by register 0xC4E-0XC51 bit 3 to 0. The source2 is controlled by register 0xCF5-0xCF6. Please see the following diagram, each port can select output from 16 sources. This will make the pair matching in 16 bit mode and byte interleaving in the two CODEC mode easier to program.**

**FIELD SWITCHING QUAD MODE**

Based on the field switching mode, a special field switching quad mode is available for use. This mode is created by setting the write buffer side in half D1 field mode and the recording port is in interlaced frame interleaved mode in D1 resolution. To visual result is similar to the normal QUAD mode but has several advantages over the normal mode:
**The bandwidth saving is very obvious as one field is dropped.**

**No need to worry odd / even field reversed because only one field is used.**

**Image looks sharper because only one low pass filter is used.**

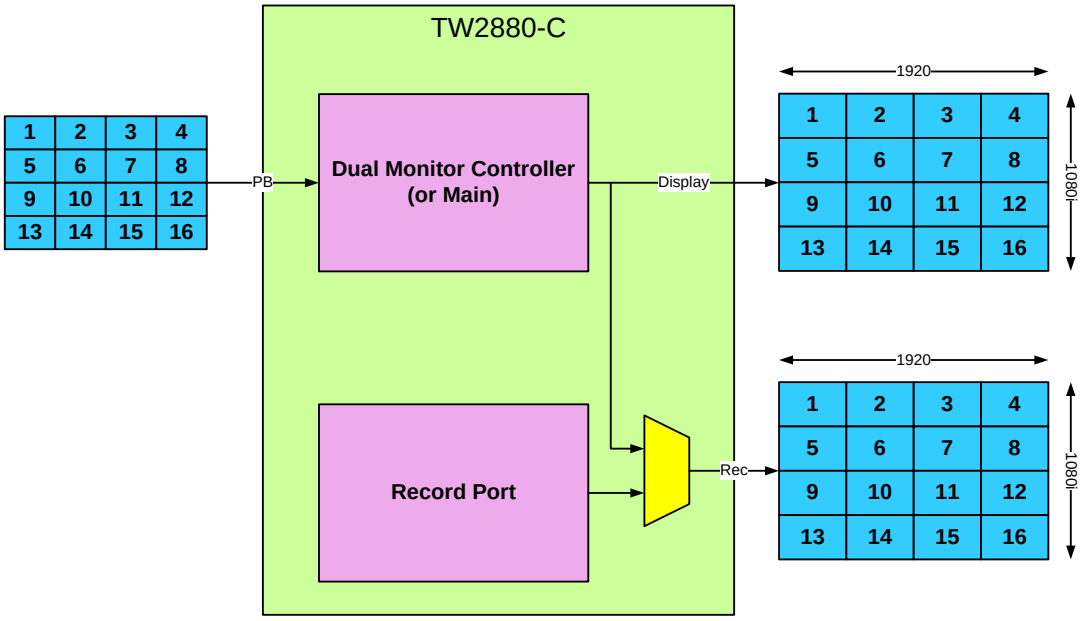
**This combination can be used in recording output and SPOT output. One thing to remember is you can extend the usage to half D1 mode (except the image is horizontally divided) and divide-by-8 mode.**

**SYNCHRONIZATION MODE**

**The table output port 5 to port 9 has a special mode for the TV wall customers. These ports' output can be synchronized together by setting register 0xcf0 bit 0, 2, 4, 6. After setting the corresponding bit the timing of the output port 6, 7, 8, 9 will be controlled by port 5. User can choose any combinations of port to be synchronized together. Another method to synchronize any channels is after setting the output port, do a software reset. After this reset all enabled port will have the same timing if they are configured under the same conditions. By doing this will put a heavy burden on DRAM bandwidth requirement so synchronization is not what you need, after software reset you need to disable and enable the port to scatter the requests.**

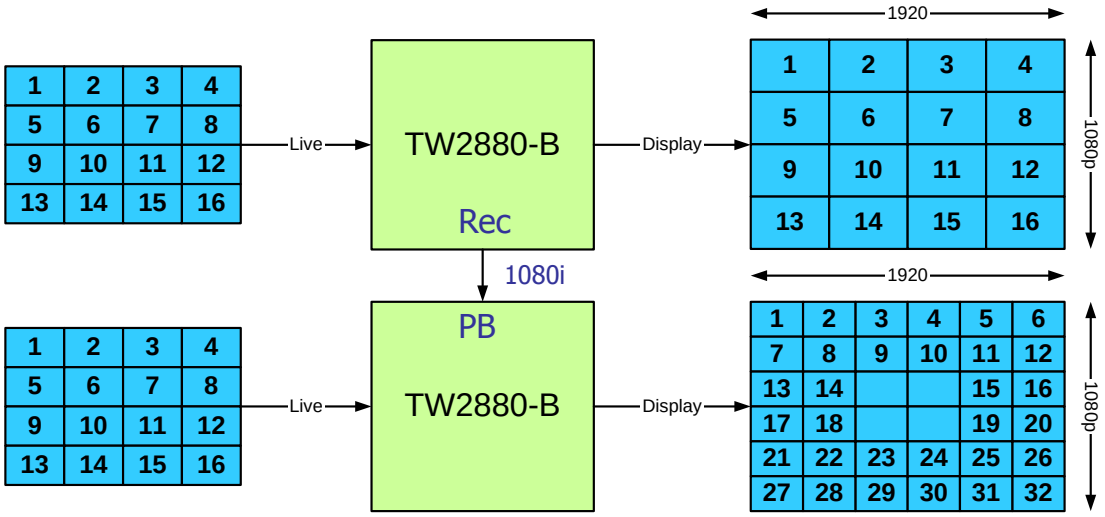
**MULTI-MODE**

**The dual monitor output from the display control unit has an option to route through record port. To use this mode users need to set the dual monitor controller in 1080i timing and set register 0x218 bit 7 to 1. The dual monitor output will come out from pin port 1, 2. This mode enables user to record the image from Live inputs or playback inputs without installing the recording memory.**




Multi-mode Application Example 1

**Another application of this capability is cascading. User now can cascade two TW2880-B chips together to form a 32 channel system. It is illustrated in the following diagram. With the variety options that TW2880-B provide, user can chose to install memories in different places and sizes to tailor to their own needs.**

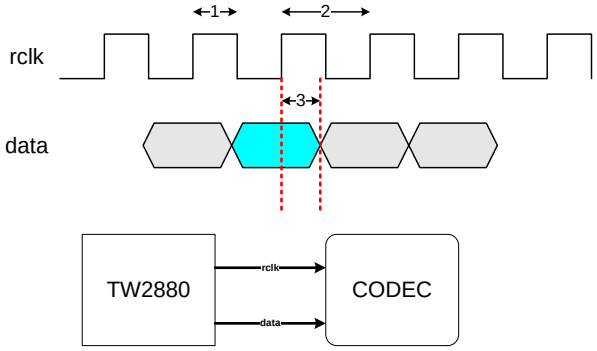



Multi-mode Application Example 2

**CONCLUSION**

**Together with 8 bit mode, TW2880 provides a wide range of connecting options for the backend solutions. One thing to remember is if the new BT.1120 mode is desired, the output will have BT.1120 EAV-SEV syntax output during the sync. period. An external clock (148.5 MHz) is needed when operating in this mode if display output is not in 1080p mode.**

**RECORD OUT AC TIMING**




| PARAMETER | SYMBOL | MIN | TYP | MAX | UNITS |
|---|---|---|---|---|---|
| Output Clock Half Period | 1 | | 4.62(108) 9.26 (54), 18.52 (27) | | ns |
| Output Clock Period | 2 | | 9.25(108) 18.51 (54), 37.03 (27) | | ns |
| Output Data Delay | 3 | 2.6 | | 5.2 | ns |

## Register Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xC00 | R/W | 0 | Recording Buffer 1 Control Register A<br>[7]: Reserved<br>[6]: Recording format select<br>[5:4]: Recording resolution select<br>[3:0]: Channel Select |
| 0xC01 | R/W | 0 | Recording Buffer 2 Control Register A |
| 0xC02 | R/W | 0 | Recording Buffer 3 Control Register A |
| 0xC03 | R/W | 0 | Recording Buffer 4 Control Register A |
| 0xC04 | R/W | 0 | Recording Buffer 5 Control Register A |
| 0xC05 | R/W | 0 | Recording Buffer 6 Control Register A |
| 0xC06 | R/W | 0 | Recording Buffer 7 Control Register A |
| 0xC07 | R/W | 0 | Recording Buffer 8 Control Register A |
| 0xC08 | R/W | 0 | Recording Buffer 9 Control Register A |
| 0xC09 | R/W | 0 | Recording Buffer 10 Control Register A |
| 0xC0A | R/W | 0 | Recording Buffer 11 Control Register A |
| 0xC0B | R/W | 0 | Recording Buffer 12 Control Register A |
| 0xC0C | R/W | 0 | Recording Buffer 13 Control Register A |
| 0xC0D | R/W | 0 | Recording Buffer 14 Control Register A |
| 0xC0E | R/W | 0 | Recording Buffer 15 Control Register A |
| 0xC0F | R/W | 0 | Recording Buffer 16 Control Register A |
| 0xC10 | R/W | 0 | Recording Buffer 1 Control Register B<br>[7]: Recording buffer enable<br>[6:3]: Horizontal position select<br>[2:0]: Vertical position select |
| 0xC11 | R/W | 0 | Recording Buffer 2 Control Register B |
| 0xC12 | R/W | 0 | Recording Buffer 3 Control Register B |
| 0xC13 | R/W | 0 | Recording Buffer 4 Control Register B |
| 0xC14 | R/W | 0 | Recording Buffer 5 Control Register B |
| 0xC15 | R/W | 0 | Recording Buffer 6 Control Register B |
| 0xC16 | R/W | 0 | Recording Buffer 7 Control Register B |
| 0xC17 | R/W | 0 | Recording Buffer 8 Control Register B |
| 0xC18 | R/W | 0 | Recording Buffer 9 Control Register B |
| 0xC19 | R/W | 0 | Recording Buffer 10 Control Register B |
| 0xC1A | R/W | 0 | Recording Buffer 11 Control Register B |
| 0xC1B | R/W | 0 | Recording Buffer 12 Control Register B |
| 0xC1C | R/W | 0 | Recording Buffer 13 Control Register B |
| 0xC1D | R/W | 0 | Recording Buffer 14 Control Register B |
| 0xC1E | R/W | 0 | Recording Buffer 15 Control Register B |
| 0xC1F | R/W | 0 | Recording Buffer 16 Control Register B |
| 0xC20 | R/W | 0 | Record Port 1 Control Register<br>[7:6]: Output resolution select<br>[5]: Output port format select<br>[4]: Horizontal split read select<br>[3]: Vertical split read select<br>[2:1]: Output port clock select<br>[0]: Output port buffer enable |
| 0xC21 | R/W | 0 | Record Port 1 Source Control Register A<br>[7:4]: Source 1 buffer select<br>[3:0]: Source 1 buffer select |
| 0xC22 | R/W | 0 | Record Port 1 Source Control Register B<br>[7:4]: Source 3 buffer select<br>[3:0]: Source 4 buffer select |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xC23 | R/W | 0 | Record Port 1 Custom Horizontal Control Register<br>[7:0]: Value * 32 = HDE |
| 0xC24 | R/W | 0 | Record Port 1 Custom Vertical Control Register<br>[7:0]: Value * 4 - 1 = VDE |
| 0xC25 | R/W | 0 | Record Port 2 Control Register<br>[7:6]: Output resolution select<br>[5]: Output port format select<br>[4]: Horizontal split read select<br>[3]: Vertical split read select<br>[2:1]: Output port clock select<br>[0]: Output port buffer enable |
| 0xC26 | R/W | 0 | Record Port 2 Source Control Register A<br>[7:4]: Source 1 buffer select<br>[3:0]: Source 1 buffer select |
| 0xC27 | R/W | 0 | Record Port 2 Source Control Register B<br>[7:4]: Source 3 buffer select<br>[3:0]: Source 4 buffer select |
| 0xC28 | R/W | 0 | Record Port 2 Custom Horizontal Control Register<br>[7:0]: Value * 32 = HDE |
| 0xC29 | R/W | 0 | Record Port 2 Custom Vertical Control Register<br>[7:0]: Value * 4 - 1 = VDE |
| 0xC2A | R/W | 0 | Record Port 3 Control Register<br>[7:6]: Output resolution select<br>[5]: Output port format select<br>[4]: Horizontal split read select<br>[3]: Vertical split read select<br>[2:1]: Output port clock select<br>[0]: Output port buffer enable |
| 0xC2B | R/W | 0 | Record Port 3 Source Control Register A<br>[7:4]: Source 1 buffer select<br>[3:0]: Source 1 buffer select |
| 0xC2C | R/W | 0 | Record Port 3 Source Control Register B<br>[7:4]: Source 3 buffer select<br>[3:0]: Source 4 buffer select |
| 0xC2D | R/W | 0 | Record Port 3 Custom Horizontal Control Register<br>[7:0]: Value * 32 = HDE |
| 0xC2E | R/W | 0 | Record Port 3 Custom Vertical Control Register<br>[7:0]: Value * 4 - 1 = VDE |
| 0xC2F | R/W | 0 | Record Port 4 Control Register<br>[7:6]: Output resolution select<br>[5]: Output port format select<br>[4]: Horizontal split read select<br>[3]: Vertical split read select<br>[2:1]: Output port clock select<br>[0]: Output port buffer enable |
| 0xC30 | R/W | 0 | Record Port 4 Source Control Register A<br>[7:4]: Source 1 buffer select<br>[3:0]: Source 1 buffer select |
| 0xC31 | R/W | 0 | Record Port 4 Source Control Register B<br>[7:4]: Source 3 buffer select<br>[3:0]: Source 4 buffer select |
| 0xC32 | R/W | 0 | Record Port 4 Custom Horizontal Control Register<br>[7:0]: Value * 32 = HDE |
| 0xC33 | R/W | 0 | Record Port 4 Custom Vertical Control Register<br>[7:0]: Value * 4 - 1 = VDE |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xC34 | R/W | 0 | Record Port 5 Control Register<br>[7:6]: Output resolution select<br>[5]: Output port format select<br>[4:3]: Reserved<br>[2:1]: Output port clock select<br>[0]: Output port buffer enable |
| 0xC35 | R/W | 0 | Record port 5 Source Control Address Register[6:0] |
| 0xC36 | R/W | 0 | Record port 5 Source Control Data Register[4:0] |
| 0xC37 | R/W | 0 | Record port 5 Source Number Register[6:0] |
| 0xC38 | R/W | 0 | Record Port 5 Custom Horizontal Control Register<br>[7:0]: Value * 32 = HDE |
| 0xC39 | R/W | 0 | Record Port 6 Custom Vertical Control Register<br>[7:0]: Value * 4 - 1 = VDE |
| 0xC3A | R/W | 0 | Record Port 6 Control Register<br>[7:6]: Output resolution select<br>[5]: Output port format select<br>[4:3]: Reserved<br>[2:1]: Output port clock select<br>[0]: Output port buffer enable |
| 0xC3B | R/W | 0 | Record port 6 Source Control Address Register[6:0] |
| 0xC3C | R/W | 0 | Record port 6 Source Control Data Register[4:0] |
| 0xC3D | R/W | 0 | Record port 6 Source Number Register[6:0] |
| 0xC3E | R/W | 0 | Record Port 6 Custom Horizontal Control Register<br>[7:0]: Value * 32 = HDE |
| 0xC3F | R/W | 0 | Record Port 6 Custom Vertical Control Register<br>[7:0]: Value * 4 - 1 = VDE |
| 0xC40 | R/W | 0 | Record Port 7 Control Register<br>[7:6]: Output resolution select<br>[5]: Output port format select<br>[4:3]: Reserved<br>[2:1]: Output port clock select<br>[0]: Output port buffer enable |
| 0xC41 | R/W | 0 | Record port 7 Source Control Address Register[6:0] |
| 0xC42 | R/W | 0 | Record port 7 Source Control Data Register[4:0] |
| 0xC43 | R/W | 0 | Record port 7 Source Number Register[6:0] |
| 0xC44 | R/W | 0 | Record Port 7 Custom Horizontal Control Register<br>[7:0]: Value * 32 = HDE |
| 0xC45 | R/W | 0 | Record Port 7 Custom Vertical Control Register<br>[7:0]: Value * 4 - 1 = VDE |
| 0xC46 | R/W | 0 | Record Port 8 Control Register<br>[7:6]: Output resolution select<br>[5]: Output port format select<br>[4:3]: Reserved<br>[2:1]: Output port clock select<br>[0]: Output port buffer enable |
| 0xC47 | R/W | 0 | Record port 8 Source Control Address Register[6:0] |
| 0xC48 | R/W | 0 | Record port 8 Source Control Data Register[4:0] |
| 0xC49 | R/W | 0 | Record port 8 Source Number Register[6:0] |
| 0xC4A | R/W | 0 | Record Port 8 Custom Horizontal Control Register<br>[7:0]: Value * 32 = HDE |
| 0xC4B | R/W | 0 | Record Port 8 Custom Vertical Control Register<br>[7:0]: Value * 4 - 1 = VDE |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xC4C | R/W | 0 | Record Port 1-4 Source Number Register<br>[7:6]: Number of source in port 4<br>[5:4]: Number of source in port 3<br>[3:2]: Number of source in port 2<br>[1:0]: Number of source in port 1 |
| 0xC4D | R/W | 0 | Record Port Operating Mode Select Register<br>[7] ADV[1]<br>[6] ADV[0]<br>[5] Automatic correction<br>[4] BT1120 Select<br>[3] Network Port Interlaced FMI<br>[2] Record test pattern<br>[1] NTSC_PAL<br>[0] WIN_READ |
| 0xC4E | R/W | 0 | Output Pin Set 1 Control Register<br>[7:4] Source 1 select<br>[3] Clock polarity<br>[2] Port width select<br>[1:0] Pin clock select |
| 0xC4F | R/W | 0 | Output Pin Set 2 Control Register<br>[7:4] Source 1 select<br>[3] Clock polarity<br>[2] Port width select<br>[1:0] Pin clock select |
| 0xC50 | R/W | 0 | Output Pin Set 3 Control Register<br>[7:4] Source 1 select<br>[3] Clock polarity<br>[2] Port width select<br>[1:0] Pin clock select |
| 0xC51 | R/W | 0 | Output Pin Set 4 Control Register<br>[7:4] Source 1 select<br>[3] Clock polarity<br>[2] Port width select<br>[1:0] Pin clock select |
| 0xC52 | R/W | 0 | Network Port Control Register<br>[7:6]: Output resolution select<br>[5]: Output port format select<br>[4:3]: Reserved<br>[2:1]: Output port clock select<br>[0]: Output port buffer enable |
| 0xC53 | R/W | 0 | Network Port Source Control Address Register[6:0] |
| 0xC54 | R/W | 0 | Network Port Source Control Data Register[4:0] |
| 0xC55 | R/W | 0 | Network Port Source Number Register[6:0] |
| 0xC56 | R/W | 0 | MISC. Control 1 Register |
| 0xC57 | R/W | 0 | HMARGIN[7:0] Register |
| 0xC58 | R/W | 0 | MISC. Control 2 Register |
| 0xC59 | R/W | 0 | MISC. Control 3 Register |
| 0xC5A | R/W | 0 | HDE selection |
| 0xC5B | R/W | 0 | VDE selection |
| 0xC5C | R/W | 0 | MISC. Control 4 Register |
| 0xC64 | R/W | 0 | Record Buffer Freeze Control Register A |
| 0xC65 | R/W | 0 | Record Buffer Freeze Control Register B |
| 0xC66 | R/W | 0 | SPOT Buffer Freeze Control Register A |
| 0xC67 | R/W | 0 | SPOT Buffer Freeze Control Register B |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xC68 | R/W | 0 | Custom Clock Control Register |
| 0xC69 | R/W | 0 | Interlaced FMI Control Register[7:0] |
| 0xC6A | R/W | 0 | 'wr_page' Reference Selection Control Register A(Write Buffer Index) |
| 0xC6B | R/W | 0 | 'wr_page' Reference Selection Control Register B(Read Port Index) |
| 0xC6C | R/W | 0 | New FRSC control register |
| 0xC90 | R/W | 0 | SPOT Recording Buffer 1 Control Register A<br>[7]: Reserved<br>[6]: Recording format select<br>[5:4]: Recording resolution select<br>[3:0]: Channel Select |
| 0xC91 | R/W | 0 | SPOT Recording Buffer 2 Control Register A |
| 0xC92 | R/W | 0 | SPOT Recording Buffer 3 Control Register A |
| 0xC93 | R/W | 0 | SPOT Recording Buffer 4 Control Register A |
| 0xC94 | R/W | 0 | SPOT Recording Buffer 5 Control Register A |
| 0xC95 | R/W | 0 | SPOT Recording Buffer 6 Control Register A |
| 0xC96 | R/W | 0 | SPOT Recording Buffer 7 Control Register A |
| 0xC97 | R/W | 0 | SPOT Recording Buffer 8 Control Register A |
| 0xC98 | R/W | 0 | SPOT Recording Buffer 9 Control Register A |
| 0xC99 | R/W | 0 | SPOT Recording Buffer 10 Control Register A |
| 0xC9A | R/W | 0 | SPOT Recording Buffer 11 Control Register A |
| 0xC9B | R/W | 0 | SPOT Recording Buffer 12 Control Register A |
| 0xC9C | R/W | 0 | SPOT Recording Buffer 13 Control Register A |
| 0xC9D | R/W | 0 | SPOT Recording Buffer 14 Control Register A |
| 0xC9E | R/W | 0 | SPOT Recording Buffer 15 Control Register A |
| 0xC9F | R/W | 0 | SPOT Recording Buffer 16 Control Register A |
| 0xCA0 | R/W | 0 | SPOT Recording Buffer 1 Control Register B<br>[7]: Recording buffer enable<br>[6:3]: Horizontal position select<br>[2:0]: Vertical position select |
| 0xCA1 | R/W | 0 | SPOT Recording Buffer 2 Control Register B |
| 0xCA2 | R/W | 0 | SPOT Recording Buffer 3 Control Register B |
| 0xCA3 | R/W | 0 | SPOT Recording Buffer 4 Control Register B |
| 0xCA4 | R/W | 0 | SPOT Recording Buffer 5 Control Register B |
| 0xCA5 | R/W | 0 | SPOT Recording Buffer 6 Control Register B |
| 0xCA6 | R/W | 0 | SPOT Recording Buffer 7 Control Register B |
| 0xCA7 | R/W | 0 | SPOT Recording Buffer 8 Control Register B |
| 0xCA8 | R/W | 0 | SPOT Recording Buffer 9 Control Register B |
| 0xCA9 | R/W | 0 | SPOT Recording Buffer 10 Control Register B |
| 0xCAA | R/W | 0 | SPOT Recording Buffer 11 Control Register B |
| 0xCAB | R/W | 0 | SPOT Recording Buffer 12 Control Register B |
| 0xCAC | R/W | 0 | SPOT Recording Buffer 13 Control Register B |
| 0xCAD | R/W | 0 | SPOT Recording Buffer 14 Control Register B |
| 0xCAE | R/W | 0 | SPOT Recording Buffer 15 Control Register B |
| 0xCAF | R/W | 0 | SPOT Recording Buffer 16 Control Register B |
| 0xCC8 | R/W | 0 | Background Buffer Indication Register 1 |
| 0xCC9 | R/W | 0 | Background Buffer Indication Register 2 |
| 0xCCA | R/W | 0 | Second Record DRAM Buffer Register 1 |
| 0xCCB | R/W | 0 | Second Record DRAM Buffer Register 2 |
| 0xCCC | R/W | 0 | Second SPOT DRAM Buffer Register 1 |
| 0xCCD | R/W | 0 | Second SPOT DRAM Buffer Register 2 |
| 0xCCE | R/W | 0 | Record Buffer Alternate Source Register 1 |
| 0xCCF | R/W | 0 | Record Buffer Alternate Source Register 2 |
| 0xCD0 | R/W | 0 | Record Buffer Horizontal Offset Control Register 1 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xCD1 | R/W | 0 | Record Buffer Horizontal Offset Control Register 2 |
| 0xCD2 | R/W | 0 | Record Buffer Horizontal Offset Control Register 3 |
| 0xCD3 | R/W | 0 | Record Buffer Horizontal Offset Control Register 4 |
| 0xCD4 | R/W | 0 | Record Buffer Horizontal Offset Control Register 5 |
| 0xCD5 | R/W | 0 | Record Buffer Horizontal Offset Control Register 6 |
| 0xCD6 | R/W | 0 | Record Buffer Horizontal Offset Control Register 7 |
| 0xCD7 | R/W | 0 | Record Buffer Horizontal Offset Control Register 8 |
| 0xCD8 | R/W | 0 | Record Buffer Horizontal Offset Control Register 9 |
| 0xCD9 | R/W | 0 | Record Buffer Horizontal Offset Control Register 10 |
| 0xCDA | R/W | 0 | Record Buffer Horizontal Offset Control Register 11 |
| 0xCDB | R/W | 0 | Record Buffer Horizontal Offset Control Register 12 |
| 0xCDC | R/W | 0 | Record Buffer Horizontal Offset Control Register 13 |
| 0xCDD | R/W | 0 | Record Buffer Horizontal Offset Control Register 14 |
| 0xCDE | R/W | 0 | Record Buffer Horizontal Offset Control Register 15 |
| 0xCDF | R/W | 0 | Record Buffer Horizontal Offset Control Register 16 |
| 0xCE0 | R/W | 0 | Record Buffer Vertical Offset Control Register 1 |
| 0xCE1 | R/W | 0 | Record Buffer Vertical Offset Control Register 2 |
| 0xCE2 | R/W | 0 | Record Buffer Vertical Offset Control Register 3 |
| 0xCE3 | R/W | 0 | Record Buffer Vertical Offset Control Register 4 |
| 0xCE4 | R/W | 0 | Record Buffer Vertical Offset Control Register 5 |
| 0xCE5 | R/W | 0 | Record Buffer Vertical Offset Control Register 6 |
| 0xCE6 | R/W | 0 | Record Buffer Vertical Offset Control Register 7 |
| 0xCE7 | R/W | 0 | Record Buffer Vertical Offset Control Register 8 |
| 0xCE8 | R/W | 0 | Record Buffer Vertical Offset Control Register 9 |
| 0xCE9 | R/W | 0 | Record Buffer Vertical Offset Control Register 10 |
| 0xCEA | R/W | 0 | Record Buffer Vertical Offset Control Register 11 |
| 0xCEB | R/W | 0 | Record Buffer Vertical Offset Control Register 12 |
| 0xCEC | R/W | 0 | Record Buffer Vertical Offset Control Register 13 |
| 0xCED | R/W | 0 | Record Buffer Vertical Offset Control Register 14 |
| 0xCEE | R/W | 0 | Record Buffer Vertical Offset Control Register 15 |
| 0xCEF | R/W | 0 | Record Buffer Vertical Offset Control Register 16 |
| 0xCF0 | R/W | 0 | Special Component Record mode Register |
| 0xCF1 | R/W | 0 | Record Buffer Field Switching Record Mode Register 1 |
| 0xCF2 | R/W | 0 | Record Buffer Field Switching Record Mode Register 2 |
| 0xCF3 | R/W | 0 | Record Buffer Field Switching Record Mode Field Select Register 1 |
| 0xCF4 | R/W | 0 | Record Buffer Field Switching Record Mode Field Select Register 2 |
| 0xCF5 | R/W | 0 | Output Pin Set Source 2 Control Register A |
| 0xCF6 | R/W | 0 | Output Pin Set Source 2 Control Register B |
| 0xCF7 | R/W | 0 | Frame Rate Controller Update Register |
| 0xCF8 | R/W | 0 | SPOT Buffer Field Switching Record Mode Register 1 |
| 0xCF9 | R/W | 0 | SPOT Buffer Field Switching Record Mode Register 2 |
| 0xCFA | R/W | 0 | SPOT Buffer Field Switching Record Mode Field Select Register 1 |
| 0xCFB | R/W | 0 | SPOT Buffer Field Switching Record Mode Field Select Register 2 |
| 0xCFC | R/W | 0 | Write Buffer Control Register |
| 0xCFD | R/W | 0 | New FRSC Control Register |
| 0xCFE | R/W | 0 | New Write Buffer Mapping Control Register |

# Registers Description

### RECORDING BUFFER 1 CONTROL REGISTER A – 0XC00

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | **Reserved** |
| 6 | R/W | 0x0 | **Recording Format Select**<br><br>0 = Frame Interleaved<br>1 = Field Interleaved |
| 5:4 | R/W | 0x0 | **Recording Resolution Select**<br><br>00 = D1<br>01 = Half D1<br>10 = CIF<br>11 = Reserved |
| 3:0 | R/W | 0x0 | **Incoming Channel Select**<br><br>[3:0] = Channel being recorded |

### RECORDING BUFFER 1 CONTROL REGISTER B – 0XC10

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **Recording Buffer Enable**<br><br>0 = Disable (default)<br>1 = Enable |
| 6:3 | R/W | 0x0 | **Horizontal Position Select**<br><br>0000 = 0<br>0001 = 320 (6VGA), 360<br>0010 = 640 (6VGA), 720<br>0011 = 960 (6VGA), 1080<br>0100 = 1280 (6VGA), 1440<br>0101 = 1600 (6VGA), 1800<br>0110 = 1920 (6VGA), 2160<br>0111 = 2240 (6VGA), 2520<br>Other = reserved |
| 2:0 | R/W | 0x0 | **Vertical Position Select**<br><br>000 = 0<br>001 = 135 (6VGA), 144 (PAL), 120 (NTSC)<br>010 = 270 (6VGA), 288 (PAL), 240 (NTSC)<br>011 = 405 (6VGA), 432 (PAL), 360 (NTSC)<br>100 = 540 (6VGA), 576 (PAL), 480 (NTSC)<br>101 = 675 (6VGA), 720 (PAL), 600 (NTSC)<br>110 = 810 (6VGA), 864 (PAL), 720 (NTSC)<br>111 = 945 (6VGA), 1008 (PAL), 840 (NTSC) |

**Similar assignments to other buffers (2-16) follow: 0xC01 - 0xC0F Recording buffer Control Register A. 0xC11 - 0xC1F Recording buffer Control Register B.**

### RECORD PORT 1 CONTROL REGISTER – 0XC20

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | **Output Resolution Select**<br><br>00 = D1<br>01 = CIF (short burst special format)<br>10 = 4D1 (special format)<br>11 = BT.1120 (1080) |
| 5 | R/W | 0x0 | **Output Port Format Select**<br><br>0 = Frame Interleaved<br>1 = Field Interleaved |
| 4 | R/W | 0x0 | **Horizontal Split Read Select**<br><br>0 = disable<br>1 = enable |
| 3 | R/W | 0x0 | **Vertical Split Read Select**<br><br>0 = disable<br>1 = enable |
| 2:1 | R/W | 0x0 | **Output Port Clock Select**<br><br>00 = ¼ of record clock<br>01 = ½ of record clock<br>10 = full record clock<br>11 = Reserved |
| 0 | R/W | 0x0 | **Output Port Buffer Enable**<br><br>0 = Disable<br>1 = Enable record out function |

### RECORD PORT 1 SOURCE CONTROL REGISTER A – 0XC21

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **Source 1 Buffer Select** |
| 3:0 | R/W | 0x0 | **Source 2 Buffer Select** |

### RECORD PORT 1 SOURCE CONTROL REGISTER B – 0XC22

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **Source 3 Buffer Select** |
| 3:0 | R/W | 0x0 | **Source 4 Buffer Select** |

### RECORD PORT 1 CUSTOM HORIZONTAL CONTROL REGISTER – 0XC23

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **HDE[7:0] Times 16** |

### RECORD PORT 1 CUSTOM VERTICAL CONTROL REGISTER – 0XC24

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **VDE[7:0] Times 4** |

### RECORD PORT 2 CONTROL REGISTER – 0XC25

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | **Output Resolution Select**<br><br>00 = D1<br>01 = CIF (short burst special format)<br>10 = 4D1 (special format)<br>11 = BT.1120 (1080) |
| 5 | R/W | 0x0 | **Output Port Format Select**<br><br>0 = Frame Interleaved<br>1 = Field Interleaved |
| 4 | R/W | 0x0 | **Horizontal Split Read Select**<br><br>0 = disable<br>1 = enable |
| 3 | R/W | 0x0 | **Vertical Split Read Select**<br><br>0 = disable<br>1 = enable |
| 2:1 | R/W | 0x0 | **Output Port Clock Select**<br><br>00 = ¼ of record clock<br>01 = ½ of record clock<br>10 = full record clock<br>11 = Reserved |
| 0 | R/W | 0x0 | **Output Port Buffer Enable**<br><br>0 = Disable<br>1 = Enable record out function |

### RECORD PORT 2 SOURCE CONTROL REGISTER A – 0XC26

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **Source 1 Buffer Select** |
| 3:0 | R/W | 0x0 | **Source 2 Buffer Select** |

### RECORD PORT 2 SOURCE CONTROL REGISTER B – 0XC27

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **Source 3 Buffer Select** |
| 3:0 | R/W | 0x0 | **Source 4 Buffer Select** |

### RECORD PORT 2 CUSTOM HORIZONTAL CONTROL REGISTER – 0XC28

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **HDE[7:0] Times 16** |

### RECORD PORT 2 CUSTOM VERTICAL CONTROL REGISTER – 0XC29

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **VDE[7:0] Times 4** |

### RECORD PORT 3 CONTROL REGISTER – 0XC2A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | **Output Resolution Select**<br><br>00 = D1<br>01 = CIF (short burst special format)<br>10 = 4D1 (special format)<br>11 = BT.1120 (1080) |
| 5 | R/W | 0x0 | **Output Port Format Select**<br><br>0 = Frame Interleaved<br>1 = Field Interleaved |
| 4 | R/W | 0x0 | **Horizontal Split Read Select**<br><br>0 = disable<br>1 = enable |
| 3 | R/W | 0x0 | **Vertical Split Read Select**<br><br>0 = disable<br>1 = enable |
| 2:1 | R/W | 0x0 | **Output Port Clock Select**<br><br>00 = ¼ of record clock<br>01 = ½ of record clock<br>10 = full record clock<br>11 = Reserved |
| 0 | R/W | 0x0 | **Output Port Buffer Enable**<br><br>0 = Disable<br>1 = Enable record out function |

### RECORD PORT 3 SOURCE CONTROL REGISTER A – 0XC2B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **Source 1 Buffer Select** |
| 3:0 | R/W | 0x0 | **Source 2 Buffer Select** |

### RECORD PORT 3 SOURCE CONTROL REGISTER B – 0XC2C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **Source 3 Buffer Select** |
| 3:0 | R/W | 0x0 | **Source 4 Buffer Select** |

### RECORD PORT 3 CUSTOM HORIZONTAL CONTROL REGISTER – 0XC2D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **HDE[7:0] Times 16** |

### RECORD PORT 3 CUSTOM VERTICAL CONTROL REGISTER – 0XC2E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **VDE[7:0] Times 4** |

### RECORD PORT 4 CONTROL REGISTER – 0XC2F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | **Output Resolution Select**<br><br>00 = D1<br>01 = CIF (short burst special format)<br>10 = 4D1 (special format)<br>11 = BT.1120 (1080) |
| 5 | R/W | 0x0 | **Output Port Format Select**<br><br>0 = Frame Interleaved<br>1 = Field Interleaved |
| 4 | R/W | 0x0 | **Horizontal Split Read Select**<br><br>0 = disable<br>1 = enable |
| 3 | R/W | 0x0 | **Vertical Split Read Select**<br><br>0 = disable<br>1 = enable |
| 2:1 | R/W | 0x0 | **Output Port Clock Select**<br><br>00 = ¼ of record clock<br>01 = ½ of record clock<br>10 = full record clock<br>11 = Reserved |
| 0 | R/W | 0x0 | **Output Port Buffer Enable**<br><br>0 = Disable<br>1 = Enable record out function |

### RECORD PORT 4 SOURCE CONTROL REGISTER A – 0XC30

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **Source 1 Buffer Select** |
| 3:0 | R/W | 0x0 | **Source 2 Buffer Select** |

### RECORD PORT 4 SOURCE CONTROL REGISTER B – 0XC31

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **Source 3 Buffer Select** |
| 3:0 | R/W | 0x0 | **Source 4 Buffer Select** |

### RECORD PORT 4 CUSTOM HORIZONTAL CONTROL REGISTER – 0XC32

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **HDE[7:0] Times 16** |

### RECORD PORT 4 CUSTOM VERTICAL CONTROL REGISTER – 0XC33

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **VDE[7:0] Times 4** |

### RECORD PORT 5 CONTROL REGISTER – 0XC34

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | **Output Resolution Select**<br><br>00 = D1<br>01 = CIF (short burst special format)<br>10 = 4D1 (special format)<br>11 = BT.1120 (1080) |
| 5 | R/W | 0x0 | **Output Port Format Select**<br><br>0 = Frame Interleaved<br>1 = Field Interleaved |
| 4:3 | R/W | 0x0 | **Reserved** |
| 2:1 | R/W | 0x0 | **Output Port Clock Select**<br><br>00 = ¼ of record clock<br>01 = ½ of record clock<br>10 = full record clock<br>11 = Reserved |
| 0 | R/W | 0x0 | **Output Port Buffer Enable**<br><br>0 = Disable<br>1 = Enable record out function |

### RECORD PORT 5 SOURCE CONTROL ADDRESS REGISTER – 0XC35

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | **Reserved** |
| 6:0 | R/W | 0x0 | **Address in Source RAM** |

### RECORD PORT 5 SOURCE CONTROL DATA REGISTER – 0XC36

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R | 0x0 | **Reserved** |
| 5 | R/W | 0x0 | **Write 1 to trigger Double Buffer Update** |
| 4:0 | R/W | 0x0 | **Source Buffer Select (32)** |

### RECORD PORT 5 SOURCE NUMBER REGISTER – 0XC37

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **Reserved** |
| 6:0 | R/W | 0x0 | **Number of Source in the list (128 is the largest)** |

### RECORD PORT 5 CUSTOM HORIZONTAL CONTROL REGISTER – 0XC38

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **HDE[7:0] Times 16** |

### RECORD PORT 5 CUSTOM VERTICAL CONTROL REGISTER – 0XC39

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **VDE[7:0] Times 4** |

### RECORD PORT 6 CONTROL REGISTER – 0XC3A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **Output Resolution Select**<br><br>00 = D1<br>01 = CIF (short burst special format)<br>10 = 4D1 (special format)<br>11 = BT.1120 (1080) |
| 5 | R/W | 0x0 | **Output Port Format Select**<br><br>0 = Frame Interleaved<br>1 = Field Interleaved |
| 4:3 | R/W | 0x0 | **Reserved** |
| 2:1 | R/W | 0x0 | **Output Port Clock Select**<br><br>00 = ¼ of record clock<br>01 = ½ of record clock<br>10 = full record clock<br>11 = Reserved |
| 0 | R/W | 0x0 | **Output Port Buffer Enable**<br><br>0 = Disable<br>1 = Enable record out function |

### RECORD PORT 6 SOURCE CONTROL ADDRESS REGISTER – 0XC3B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | **Reserved** |
| 6:0 | R/W | 0x0 | **Address in Source RAM** |

### RECORD PORT 6 SOURCE CONTROL DATA REGISTER – 0XC3C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **Reserved** |
| 5 | R/W | 0x0 | **Write 1 to trigger Double Buffer Update** |
| 4:0 | R/W | 0x0 | **Source Buffer Select (32)** |

### RECORD PORT 6 SOURCE NUMBER REGISTER – 0XC3D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **Reserved** |
| 6:0 | R/W | 0x0 | **Number of Source in the list (128 is the largest)** |

### RECORD PORT 6 CUSTOM HORIZONTAL CONTROL REGISTER – 0XC3E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **HDE[7:0] Times 16** |

### RECORD PORT 6 CUSTOM VERTICAL CONTROL REGISTER – 0XC3F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **VDE[7:0] Times 4** |

### RECORD PORT 7 CONTROL REGISTER – 0XC40

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | **Output Resolution Select**<br><br>00 = D1<br>01 = CIF (short burst special format)<br>10 = 4D1 (special format)<br>11 = BT.1120 (1080) |
| 5 | R/W | 0x0 | **Output Port Format Select**<br><br>0 = Frame Interleaved<br>1 = Field Interleaved |
| 4:3 | R/W | 0x0 | **Reserved** |
| 2:1 | R/W | 0x0 | **Output Port Clock Select**<br><br>00 = ¼ of record clock<br>01 = ½ of record clock<br>10 = full record clock<br>11 = Reserved |
| 0 | R/W | 0x0 | **Output Port Buffer Enable**<br><br>0 = Disable<br>1 = Enable record out function |

### RECORD PORT 7 SOURCE CONTROL ADDRESS REGISTER – 0XC41

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | **Reserved** |
| 6:0 | R/W | 0x0 | **Address in Source RAM** |

### RECORD PORT 7 SOURCE CONTROL DATA REGISTER – 0XC42

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **Reserved** |
| 5 | R/W | 0x0 | **Write 1 to trigger Double Buffer Update** |
| 4:0 | R/W | 0x0 | **Source Buffer Select (32)** |

### RECORD PORT 7 SOURCE NUMBER REGISTER – 0XC43

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **Reserved** |
| 6:0 | R/W | 0x0 | **Number of Source in the list (128 is the largest)** |

### RECORD PORT 7 CUSTOM HORIZONTAL CONTROL REGISTER – 0XC44

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **HDE[7:0] Times 16** |

### RECORD PORT 7 CUSTOM VERTICAL CONTROL REGISTER – 0XC45

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **VDE[7:0] Times 4** |

### RECORD PORT 8 CONTROL REGISTER – 0XC46

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | **Output Resolution Select**<br><br>00 = D1<br>01 = CIF (short burst special format)<br>10 = 4D1 (special format)<br>11 = BT.1120 (1080) |
| 5 | R/W | 0x0 | **Output Port Format Select**<br><br>0 = Frame Interleaved<br>1 = Field Interleaved |
| 4:3 | R/W | 0x0 | **Reserved** |
| 2:1 | R/W | 0x0 | **Output Port Clock Select**<br><br>00 = ¼ of record clock<br>01 = ½ of record clock<br>10 = full record clock<br>11 = Reserved |
| 0 | R/W | 0x0 | **Output Port Buffer Enable**<br><br>0 = Disable<br>1 = Enable record out function |

### RECORD PORT 8 SOURCE CONTROL ADDRESS REGISTER – 0XC47

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | **Reserved** |
| 6:0 | R/W | 0x0 | **Address in Source RAM** |

### RECORD PORT 8 SOURCE CONTROL DATA REGISTER – 0XC48

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | **Reserved** |
| 5 | R/W | 0x0 | **Write 1 to trigger Double Buffer Update** |
| 4:0 | R/W | 0x0 | **Source Buffer Select (32)** |

### RECORD PORT 8 SOURCE NUMBER REGISTER – 0XC49

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **Reserved** |
| 6:0 | R/W | 0x0 | **Number of Source in the list (128 is the largest)** |

### RECORD PORT 8 CUSTOM HORIZONTAL CONTROL REGISTER – 0XC4A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **HDE[7:0] Times 16** |

### RECORD PORT 8 CUSTOM VERTICAL CONTROL REGISTER – 0XC4B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **VDE[7:0] Times 4** |

### RECORD PORT 1-4 SOURCE CONTROL REGISTER – 0XC4C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | Number of Source in the port 4 list (4 is the largest) |
| 5:4 | R/W | 0x0 | Number of Source in the port 3 list (4 is the largest) |
| 3:2 | R/W | 0x0 | Number of Source in the port 2 list (4 is the largest) |
| 1:0 | R/W | 0x0 | Number of Source in the port 1 list (4 is the largest) |

### RECORD PORT OPERATING MODE SELECT REGISTER – 0XC4D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | Bottom field low speed update algorithm select<br>1 = 54 /27 MHz, 0 = normal ADV[1] |
| 6 | R/W | 0x0 | Top field low speed update algorithm select<br>1 = 54 /27 MHz, 0 = normal ADV[0] |
| 5 | R/W | 0x0 | Enable field mode automatic correction<br><br>0 = disable (default)<br>1 = enable |
| 4 | R/W | 0x0 | BT.1120 PAL 50 /60 Hz Selection<br><br>0 = 50 Hz (default)<br>1 = 60 Hz |
| 3 | R/W | 0x0 | Network port Interlaced FMI Mode Control<br><br>0 = Normal (default)<br>1 = Turn on |
| 2 | R/W | 0x0 | Record Source Select<br><br>0 = Live mode (default)<br>1 = Internal test pattern mode |
| 1 | R/W | 0x0 | Record Standard Select<br><br>0 = NTSC mode (default)<br>1 = PAL mode |
| 0 | R/W | 0x0 | Limit Motion Circuitry Read / Write Activity<br><br>0 = Disable (default)<br>1 = Enable |

### OUTPUT PIN SET 1 CONTROL REGISTER – 0XC4E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **Source 1 Select[3:0]**<br><br>1111 = port 8 output high byte<br>1110 = port 8 output low byte<br>1101 = port 7 output high byte<br>1100 = port 7 output low byte<br>1011 = port 6 output high byte<br>1010 = port 6 output low byte<br>1001 = port 5 output high byte<br>1000 = port 5 output low byte<br>0111 = port 4 output high byte<br>0110 = port 4 output low byte<br>0101 = port 3 output high byte<br>0100 = port 3 output low byte<br>0011 = port 2 output high byte<br>0010 = port 2 output low byte<br>0001 = port 1 output high byte<br>0000 = port 1 output low byte (default) |
| 3 | R/W | 0x0 | **Clock Polarity Select**<br><br>0 = normal, 1 = inverted |
| 2 | R/W | 0x0 | **Port width Select**<br><br>0 = 8 bit, 1 = 16 bit |
| 1:0 | R/W | 0x0 | **Pin Clock Select**<br><br>00 = ¼ of record clock<br>01 = ½ of record clock<br>10 = full record clock<br>11 = ½ of the external clock, input from ball AC5 |

### OUTPUT PIN SET 2 CONTROL REGISTER – 0XC4F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **Source 1 Select[3:0]**<br><br>1111 = port 8 output high byte<br>1110 = port 8 output low byte<br>1101 = port 7 output high byte<br>1100 = port 7 output low byte<br>1011 = port 6 output high byte<br>1010 = port 6 output low byte<br>1001 = port 5 output high byte<br>1000 = port 5 output low byte<br>0111 = port 4 output high byte<br>0110 = port 4 output low byte<br>0101 = port 3 output high byte<br>0100 = port 3 output low byte (default)<br>0011 = port 2 output high byte<br>0010 = port 2 output low byte<br>0001 = port 1 output high byte<br>0000 = port 1 output low byte |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 3 | R/W | 0x0 | **Clock Polarity Select**<br><br>0 = normal, 1 = inverted |
| 2 | R/W | 0x0 | **Port width Select**<br><br>0 = 8 bit, 1 = 16 bit |
| 1:0 | R/W | 0x0 | **Pin Clock Select**<br><br>00 = ¼ of record clock<br>01 = ½ of record clock<br>10 = full record clock<br>11 = ½ of the external clock, input from ball AC5 |

**OUTPUT PIN SET 3 CONTROL REGISTER – 0XC50**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **Source 1 Select[3:0]**<br><br>1111 = port 8 output high byte<br>1110 = port 8 output low byte<br>1101 = port 7 output high byte<br>1100 = port 7 output low byte<br>1011 = port 6 output high byte<br>1010 = port 6 output low byte<br>1001 = port 5 output high byte<br>1000 = port 5 output low byte (default)<br>0111 = port 4 output high byte<br>0110 = port 4 output low byte<br>0101 = port 3 output high byte<br>0100 = port 3 output low byte<br>0011 = port 2 output high byte<br>0010 = port 2 output low byte<br>0001 = port 1 output high byte<br>0000 = port 1 output low byte |
| 3 | R/W | 0x0 | **Clock Polarity Select**<br><br>0 = normal, 1 = inverted |
| 2 | R/W | 0x0 | **Port width Select**<br><br>0 = 8 bit, 1 = 16 bit |
| 1:0 | R/W | 0x0 | **Pin Clock Select**<br><br>00 = ¼ of record clock<br>01 = ½ of record clock<br>10 = full record clock<br>11 = ½ of the external clock, input from ball AC5 |

### OUTPUT PIN SET 4 CONTROL REGISTER – 0XC51

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **Source 1 Select[3:0]**<br><br>1111 = port 8 output high byte<br>1110 = port 8 output low byte<br>1101 = port 7 output high byte<br>1100 = port 7 output low byte (default)<br>1011 = port 6 output high byte<br>1010 = port 6 output low byte<br>1001 = port 5 output high byte<br>1000 = port 5 output low byte<br>0111 = port 4 output high byte<br>0110 = port 4 output low byte<br>0101 = port 3 output high byte<br>0100 = port 3 output low byte<br>0011 = port 2 output high byte<br>0010 = port 2 output low byte<br>0001 = port 1 output high byte<br>0000 = port 1 output low byte |
| 3 | R/W | 0x0 | **Clock Polarity Select**<br><br>0 = normal, 1 = inverted |
| 2 | R/W | 0x0 | **Port width Select**<br><br>0 = 8 bit, 1 = 16 bit |
| 1:0 | R/W | 0x0 | **Pin Clock Select**<br><br>00 = ¼ of record clock<br>01 = ½ of record clock<br>10 = full record clock<br>11 = ½ of the external clock, input from ball AC5 |

### NETWORK PORT CONTROL REGISTER – 0XC52

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | **Output Resolution Select**<br><br>00 = D1<br>01 = CIF (short burst special format)<br>10 = 4D1 (special format)<br>11 = BT.1120 (1080) |
| 5 | R/W | 0x0 | **Output Port Format Select**<br><br>0 = Frame Interleaved<br>1 = Field Interleaved |
| 4:3 | R/W | 0x0 | **Reserved** |
| 2:1 | R/W | 0x0 | **Output Port Clock Select**<br><br>00 = ¼ of record clock<br>01 = ½ of record clock<br>10 = full record clock<br>11 = Reserved |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | R/W | 0x0 | Output Port Buffer Enable<br><br>0 = Disable<br>1 = Enable record out function |

### NETWORK PORT CONTROL ADDRESS REGISTER – 0XC53

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | Reserved |
| 6:0 | R/W | 0x0 | Address in Source RAM |

### NETWORK PORT CONTROL DATA REGISTER – 0XC54

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | Reserved |
| 6 | R/W | 0x0 | Write 1 to trigger Double Buffer Update |
| 5:0 | R/W | 0x0 | Source Buffer Select (48) |

### NETWORK PORT SOURCE NUMBER REGISTER – 0XC55

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | Enable Free Running Network Port<br><br>1 = enable, 0 = normal mode |
| 6:0 | R/W | 0x0 | Number of Source in the list (128 is the largest) |

### MISC. CONTROL REGISTER 1 – 0XC56

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | SPREAD<br><br>Spread read sync option<br><br>1 = spread read syn, 0 = no spread |
| 6 | R/W | 0x0 | RST_SEL<br><br>Select different MCLK reset width in FIFO<br><br>1 = Different reset, 0 = Same reset |
| 5 | R/W | 0x0 | Bottom field low speed update algorithm select<br><br>1 = 54 /27 MHz, 0 = normal ADV[1] for SPOT |
| 4 | R/W | 0x0 | Top field low speed update algorithm select<br><br>1 = 54 /27 MHz, 0 = normal ADV[0] for SPOT |
| 3 | R/W | 0x0 | Horizontal Control Override in 6 VGA mode |
| 2:0 | R/W | 0x0 | Select record side FRSC source |

### 4D1 AND QUAD 2ND WRITE HORIZONTAL MARGIN REGISTER – 0XC57

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | HMARGIN[7:0]<br><br>Unit is hcnt. Increase the 2nd write hblank by this number |

### MISC. CONTROL REGISTER 2 – 0XC58

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **1 = Limit Record unit start time if using SPOT buffers**<br><br>1 = disable (default)<br>0 = enable |
| 6 | R/W | 0x0 | **1 = Limit Record unit start time if using record buffers**<br><br>1 = disable (default)<br>0 = enable |
| 5:4 | R/W | 0x0 | **Record Write Buffer Request Level Control (default = 1)** |
| 3:0 | R/W | 0x0 | **Enable no_video reset frame rate control unit**<br><br>Bit 0: ROUT1 – ROUT4<br>Bit 1: ROUT5 – ROUT6<br>Bit 2: ROUT7 – ROUT8<br>Bit 3: ROUT9 |

### MISC. CONTROL REGISTER 3 – 0XC59

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | **SPOT Write Buffer Request Level Control (default = 1)** |
| 5 | R/W | 0x0 | **SPOT Buffer Vertical Invert Line Number Option** |
| 4 | R/W | 0x0 | **Record Buffer Vertical Invert Line Number Option** |
| 3:2 | R/W | 0x0 | **SPOT Buffer Update Option Select** |
| 1:0 | R/W | 0x0 | **Record Buffer Update Option Select** |

### RECORD HDE CONTROL REGISTER – 0XC5A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **HDE selection** |

### RECORD VDE CONTROL REGISTER – 0XC5B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **VDE selection** |

### RECORD BUFFER FREEZE CONTROL REGISTER A – 0XC64

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **Freeze Control [7:0]**<br><br>1 = Channel Freeze<br>0 = Normal |

### RECORD BUFFER FREEZE CONTROL REGISTER B – 0XC65

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **Freeze Control [15:8]**<br><br>1 = Channel Freeze<br>0 = Normal |

### SPOT BUFFER FREEZE CONTROL REGISTER A – 0XC66

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **Freeze Control [7:0]**<br><br>1 = Channel Freeze<br>0 = Normal |

### SPOT BUFFER FREEZE CONTROL REGISTER B – 0XC67

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **Freeze Control [15:8]**<br><br>1 = Channel Freeze<br>0 = Normal |

### CUSTOM CLOCK CONTROL REGISTER – 0XC68

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **Window Position Shadow Register Select**<br><br>0 = recording, 1= SPOT |
| 6:2 | R | 0x0 | **Reserved** |
| 1:0 | R/W | 0x0 | **Custom Clock Source Select**<br><br>00 = Select system clock (108 MHz)<br>01 = Select Internal video clock (vck1), see AP note1 p8<br>10 = Select Internal video clock (vck), see AP note1 p8<br>11 = Select external clock source |

### INTERLACED FRAME OUTPUT MODE CONTROL REGISTER – 0XC69

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **Interlaced Frame Control Port 8 to Port1**<br><br>0 = Progressive Frame (default)<br>1 = Interlaced Frame |

### 'WR_PAGE' REFERENCE SELECTION CONTROL REGISTER A(WRITE BUFFER INDEX) – 0XC6A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R | 0x0 | **Reserved** |
| 5:0 | R/W | 0x0 | **WR_BUF_ADDR**<br>wr_buffer index for selecting 'wr_page' reference from read port<br>0 ~ 15 : recording write buffer 0 ~ 15<br>16 ~ 31 : SPOT write buffer 0 ~ 15 |

### 'WR_PAGE' REFERENCE SELECTION CONTROL REGISTER B(READ PORT INDEX) – 0XC6B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R | 0x0 | Reserved |
| 5 | R/W | 0x0 | BANK_SEL_SEP<br>0 = Use one 'bank_sel' signal for each port<br>1 = Use separated several 'bank_sel' signals for each port |
| 4 | R/W | 0x0 | WR_PAGE_SEP<br>0 = Use one wr_page reference<br>1 = Use separated wr_page reference according to the write buffer |
| 3:0 | R/W | 0x0 | wr_page_sel<br>wr_page reference selection value<br>0 = port1 wr_page reference<br>1 = port2 wr_page reference<br>2 = port3 wr_page reference<br>3 = port4 wr_page reference<br>4 = port5 wr_page reference<br>5 = port6 wr_page reference<br>6 = port7 wr_page reference<br>7 = port8 wr_page reference<br>8 = network port wr_page reference<br>9 = SPOT1 wr_page reference<br>10 = SPOT2 wr_page reference<br>11 = SPOT3 wr_page reference<br>12 = SPOT4 wr_page reference<br>the others = port1 wr_page reference |

### NEW FRSC CONTROL REGISTER B – 0XC6C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | FLD_BANK_INC<br>Bank increase option in the non-real time field mode<br>0 = Bank is increased in every number of channel<br>1 = Bank is increased in every frame rate |
| 6 | R/W | 0x0 | FLD_HVCNT_SEL<br>Field generation option in the hvcnt_rout<br>0 = Old one<br>1 = New one |
| 5 | R/W | 0x0 | FLD_CCIR_SEL<br>Field generation option in the ccir_out_fmt<br>0 = Old one<br>1 = New one |
| 4 | R/W | 0x0 | WR_BUF_FLI_STOP<br>Write buffer stop control selection between frame and field stop<br>0 = Frame stop<br>1 = Field stop |
| 3 | R/W | 0x0 | WR_BUF_STOP<br>Read bank repeat option when write buffer is stop in the 4-frame field mode<br>0 = Repeat one field<br>1 = Repeat one frame |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2 | R/W | 0x0 | FLI_RD_STOP<br>Read bank repeat option when bank stop condition is occurred in the 4-frame field mode<br>0 = Repeat one field<br>1 = Repeat one frame |
| 1 | R/W | 0x0 | FLI_RD<br>Field interleaved read mode with 4 frames<br>0 = 2 frames<br>1 = 4 frames |
| 0 | R/W | 0x0 | RD_FRSC_FLI_EN<br>Read side frame rate control enable for field mode<br>0 = Original frame rate control(bank is increased without checking relation between read and write bank values)<br>1 = New frame rate control in read side(bank is increased if some condition is met between read and write bank values) |

### PB IGNORE BIT CONTROL REGISTER – 0XC6D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | PB_IGNORE<br>PB ignore bit on/off control for each port1 ~ port8<br>0 = Ignore bit disable<br>1 = Ignore bit enable |

### PB IGNORE VALUE REGISTER 1 – 0XC6E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | PB_IG_DATA1<br>2nd PB ignore bit value |
| 3:0 | R/W | 0x0 | PB_IG_DATA0<br>1st PB ignore bit value |

### PB IGNORE VALUE REGISTER 2 – 0XC6F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | PB_IG_DATA3<br>4th PB ignore bit value |
| 3:0 | R/W | 0x0 | PB_IG_DATA2<br>3rd PB ignore bit value |

### BACKGROUND COLOR FOR CB – 0XC70

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | BG_CB<br>Background color of Cb |

### BACKGROUND COLOR FOR CR – 0XC71

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | BG_CR<br>Background color of Cr |

### BACKGROUND COLOR FOR Y – 0XC72

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | BG_Y<br>Background color of Y |

### BACKGROUND ON/OFF CONTROL 1 FOR RECORD WRITE BUFFER – 0XC73

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | BG_ON_REC[7:0]<br>Background on/off control for record write buffer 1 ~ buffer 8 |

### BACKGROUND ON/OFF CONTROL 2 FOR RECORD WRITE BUFFER – 0XC74

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | BG_ON_REC[15:8]<br>Background on/off control for record write buffer 9 ~ buffer 16 |

### BACKGROUND ON/OFF CONTROL 1 FOR SPOT WRITE BUFFER – 0XC75

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | BG_ON_SPOT[7:0]<br>Background on/off control for SPOT write buffer 1 ~ buffer 8 |

### BACKGROUND ON/OFF CONTROL 2 FOR SPOT WRITE BUFFER – 0XC76

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | BG_ON_SPOT[15:8]<br>Background on/off control for SPOT write buffer 9 ~ buffer 16 |

### NEW NON-REAL TIME FIELD SWITCHING MODE FOR PORT 5 ~ PORT 9 – 0XC77

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R/W | 0x0 | Reserved |
| 4:0 | R/W | 0x0 | NEW_FLI_SW<br>New field switching mode for non-real time<br>0 = Original<br>1 = New field switching |

### NEW FIELD SWITCHING MODE FIELD SELECT FOR PORT 5 ~ PORT 9 – 0XC78

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R/W | 0x0 | Reserved |
| 4:0 | R/W | 0x0 | NEW_FLI_SW_SEL<br>Field selection for new field switching mode<br>0 = Even<br>1 = Odd |

### FIELD INTERLEAVED READ MODE WITH 4 FRAMES – 0XC79

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R/W | 0x0 | Reserved |
| 4:0 | R/W | 0x0 | FLI_RD<br>Field interleaved read mode with 4 frames<br>0 = 2 frames<br>1 = 4 frames |

### NEW NON-REAL TIME FIELD SWITCHING MODE FOR PORT 5 ~ PORT 9 – 0XC7A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R/W | 0x0 | Reserved |
| 4:0 | R/W | 0x0 | NEW_FLI_SW<br>New field switching mode for non-real time<br>0 = Original<br>1 = New field switching |

### SPOT RECORDING BUFFER 1 CONTROL REGISTER A – 0XC90

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0x0 | Reserved |
| 6 | R/W | 0x0 | Recording Format Select<br><br>0 = Frame Interleaved<br>1 = Field Interleaved |
| 5:4 | R/W | 0x0 | Recording Resolution Select<br><br>00 = D1<br>01 = Half D1<br>10 = CIF<br>11 = VGA (640) |
| 3:0 | R/W | 0x0 | Incoming Channel Select<br><br>[3:0] = Channel being recorded |

### SPOT RECORDING BUFFER 1 CONTROL REGISTER B – 0XCA0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **Recording Buffer Enable**<br><br>0 = Disable (default)<br>1 = Enable |
| 6:3 | R/W | 0x0 | **Horizontal Position Select**<br><br>0000 = 0<br>0001 = 360<br>0010 = 720<br>0011 = 1080<br>0100 = 1440<br>0101 = 1800<br>0110 = 2160<br>0111 = 2520<br>Other = reserved |
| 2:0 | R/W | 0x0 | **Vertical Position Select**<br><br>000 = 0<br>001 = 144 (PAL), 120 (NTSC)<br>010 = 288 (PAL), 240 (NTSC)<br>011 = 432 (PAL), 360 (NTSC)<br>100 = 576 (PAL), 480 (FMI and NTSC)<br>101 = 720 (PAL), 700 (NTSC)<br>110 = 1152 (PAL), 960 (NTSC)<br>111 = 1296 (PAL), 1080 (NTSC) |

**Similar assignments to other buffers (2-16) follow: 0xC91 - 0xC9F SPOT Recording buffer Control Register A. 0xCA1 - 0xCAF SPOT Recording buffer Control Register B.**

### BACKGROUND BUFFER INDICATION REGISTER 1 – 0XCC8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **Record FIFO [7:0] Enable**<br><br>0 = disable<br>1 = enable |

### BACKGROUND BUFFER INDICATION REGISTER 2 – 0XCC9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **Record FIFO [15:8] Enable**<br><br>0 = disable<br>1 = enable |

### SECOND RECORD BUFFER DRAM CONTROL REGISTER 1 – 0XCCA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **Second Record DRAM Buffer REC_2nd[7:0] Enable**<br><br>0 = disable<br>1 = enable |

### SECOND RECORD BUFFER DRAM CONTROL REGISTER 2 – 0XCCB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Second Record DRAM Buffer REC_2nd[15:8] Enable<br><br>0 = disable<br>1 = enable |

### SECOND SPOT DRAM BUFFER CONTROL REGISTER 1 – 0XCCC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Second Record DRAM Buffer SPOT_2nd[7:0] Enable<br><br>0 = disable<br>1 = enable |

### SECOND SPOT DRAM BUFFER CONTROL REGISTER 2 – 0XCCD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Second Record DRAM Buffer SPOT_2nd[15:8] Enable<br><br>0 = disable<br>1 = enable |

### RECORD BUFFER ALTERNATE SOURCE CONTROL REGISTER 1 – 0XCCE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Use SPOT Buffer as Record Buffer ALTM[7:0] Enable<br><br>0 = disable<br>1 = enable |

### RECORD BUFFER ALTERNATE SOURCE CONTROL REGISTER 2 – 0XCCF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Use SPOT Buffer as Record Buffer ALTM[15:8] Enable<br><br>0 = disable<br>1 = enable |

### RECORD BUFFER HORIZONTAL OFFSET CONTROL REGISTER – 0XCD0-DF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Starting Position to record HSTART[7:0] for CH1-CH16<br><br>Unit is 4 pixel |

### RECORD BUFFER VERTICAL OFFSET CONTROL REGISTER – 0XCE0-EF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Starting Position to record VSTART[7:0] for CH1-CH16<br><br>Unit is line |

### SPECIAL COMPONENT RECORD MODE REGISTER – 0XCF0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | Single element fetching mode for record port 8 (8D1 possible)<br><br>1 = enable, 0 = normal mode |
| 6 | R/W | 0x0 | Synchronized output enable select for record port 9<br><br>1 = synchronize with ROUT5, 0 = normal |
| 5 | R/W | 0x0 | Single element fetching mode for record port 7 (8D1 possible)<br><br>1 = enable, 0 = normal mode |
| 4 | R/W | 0x0 | Synchronized output enable select for record port 8<br><br>1 = synchronize with ROUT5, 0 = normal |
| 3 | R/W | 0x0 | Single element fetching mode for record port 6 (8D1 possible)<br><br>1 = enable, 0 = normal mode |
| 2 | R/W | 0x0 | Synchronized output enable select for record port 7<br><br>1 = synchronize with ROUT5, 0 = normal |
| 1 | R/W | 0x0 | Single element fetching mode for record port 5 (8D1 possible)<br><br>1 = enable, 0 = normal mode |
| 0 | R/W | 0x0 | Synchronized output enable select for record port 6<br><br>1 = synchronize with ROUT5, 0 = normal |

### RECORD BUFFER FIELD SWITCHING RECORD MODE REGISTER I – 0XCF1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Control recording buffer [7:0]<br><br>1 = enable, 0 = normal mode |

### RECORD BUFFER FIELD SWITCHING RECORD MODE REGISTER 2 – 0XCF2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Control recording buffer [15:8]<br><br>1 = enable, 0 = normal mode |

### RECORD BUFFER FIELD SWITCHING RECORD MODE FIELD SELECT REGISTER I – 0XCF3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Control recording buffer [7:0]<br><br>1 = even field, 0 = odd field |

### RECORD BUFFER FIELD SWITCHING RECORD MODE FIELD SELECT REGISTER 2 – 0XCF4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Control recording buffer [15:8]<br><br>1 = even field, 0 = odd field |

### OUTPUT PIN SET SOURCE 2 CONTROL REGISTER A – 0XCF5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R | 0x0 | Pin Set 2 Source 2 Select[3:0]<br><br>1111 = port 8 output high byte<br>1110 = port 8 output low byte<br>1101 = port 7 output high byte<br>1100 = port 7 output low byte (default)<br>1011 = port 6 output high byte<br>1010 = port 6 output low byte<br>1001 = port 5 output high byte<br>1000 = port 5 output low byte<br>0111 = port 4 output high byte<br>0110 = port 4 output low byte<br>0101 = port 3 output high byte<br>0100 = port 3 output low byte<br>0011 = port 2 output high byte<br>0010 = port 2 output low byte<br>0001 = port 1 output high byte<br>0000 = port 1 output low byte |
| 3:0 | R/W | 0x0 | Pin Set 1 Source 2 Select[3:0]<br><br>1111 = port 8 output high byte<br>1110 = port 8 output low byte<br>1101 = port 7 output high byte<br>1100 = port 7 output low byte (default)<br>1011 = port 6 output high byte<br>1010 = port 6 output low byte<br>1001 = port 5 output high byte<br>1000 = port 5 output low byte<br>0111 = port 4 output high byte<br>0110 = port 4 output low byte<br>0101 = port 3 output high byte<br>0100 = port 3 output low byte<br>0011 = port 2 output high byte<br>0010 = port 2 output low byte<br>0001 = port 1 output high byte<br>0000 = port 1 output low byte |

### OUTPUT PIN SET SOURCE 2 CONTROL REGISTER B – 0XCF6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R | 0x0 | Pin Set 4 Source 2 Select[3:0]<br><br>1111 = port 8 output high byte<br>1110 = port 8 output low byte<br>1101 = port 7 output high byte<br>1100 = port 7 output low byte (default)<br>1011 = port 6 output high byte<br>1010 = port 6 output low byte<br>1001 = port 5 output high byte<br>1000 = port 5 output low byte<br>0111 = port 4 output high byte<br>0110 = port 4 output low byte<br>0101 = port 3 output high byte<br>0100 = port 3 output low byte<br>0011 = port 2 output high byte<br>0010 = port 2 output low byte<br>0001 = port 1 output high byte<br>0000 = port 1 output low byte |
| 3:0 | R/W | 0x0 | Pin Set 3 Source 2 Select[3:0]<br><br>1111 = port 8 output high byte<br>1110 = port 8 output low byte<br>1101 = port 7 output high byte<br>1100 = port 7 output low byte (default)<br>1011 = port 6 output high byte<br>1010 = port 6 output low byte<br>1001 = port 5 output high byte<br>1000 = port 5 output low byte<br>0111 = port 4 output high byte<br>0110 = port 4 output low byte<br>0101 = port 3 output high byte<br>0100 = port 3 output low byte<br>0011 = port 2 output high byte<br>0010 = port 2 output low byte<br>0001 = port 1 output high byte<br>0000 = port 1 output low byte |

### FRAME RATE CONTROLLER UPDATE REGISTER – 0XCF7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Enable Free Running Recording Port 7-0<br><br>1 = enable, 0 = normal mode |

### SPOT BUFFER FIELD SWITCHING RECORD MODE REGISTER I – 0XCF8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Control recording buffer [7:0]<br><br>1 = enable, 0 = normal mode |

### SPOT BUFFER FIELD SWITCHING RECORD MODE REGISTER 2 – 0XCF9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Control recording buffer [15:8]<br><br>1 = enable, 0 = normal mode |

### SPOT BUFFER FIELD SWITCHING RECORD MODE FIELD SELECT REGISTER I – 0XCFA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Control recording buffer [7:0]<br><br>1 = even field, 0 = odd field |

### SPOT BUFFER FIELD SWITCHING RECORD MODE FIELD SELECT REGISTER 2 – 0XCFB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | Control recording buffer [15:8]<br><br>1 = even field, 0 = odd field |

### WRITE BUFFER CONTROL REGISTER – 0XCFC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | Recording / SPOT Buffers Write Control Enable<br><br>0 = disable, 1= enable |
| 6 | R | 0x0 | NEW_FLI_SW_SEL<br>Field selection for new field switching mode<br>0 = Even<br>1 = Odd |
| 5 | R | 0x0 | NEW_FLI_SW<br>New field switching mode for non-real time<br>0 = Original<br>1 = New field switching |
| 4 | R/W | 0x0 | SPOT_MEM<br>In the SPOT write buffer, buffer write scheme selection control<br>0 = Use secondary memory area<br>1 = Use left side memory area for 256 Mb memory |
| 3:2 | R/W | 0x0 | SPOT Buffer Source Skip Select<br><br>00 = No skip (default)<br>01 = skip every 5 frames (fields)<br>10 = skip every 6 frames (fields)<br>11 = skip every 7 frames (fields) |
| 1:0 | R/W | 0x0 | Recording Buffer Source Skip Select<br><br>00 = No skip (default)<br>01 = skip every 5 frames (fields)<br>10 = skip every 6 frames (fields)<br>11 = skip every 7 frames (fields) |

### NEW FRSC CONTROL REGISTER A – 0XCFD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | **FLD_FRSC**<br>New frame rate control method by using field signal<br>0x = original frame rate control<br>10 = new frame rate control, wr_page[0] = field<br>11 = new frame rate control, wr_page[0] = ~field |
| 5 | R/W | 0x0 | **NEW_EN**<br>New method for mapping wr_buffers to read ports<br>00 = original mapping<br>01 = new mapping, each read port selects write buffers by setting register 0xCFD[4:0] and 0xCFE[7:0] |
| 4:0 | R/W | 0x0 | **EN_ADDRI**<br>Read port index<br>0 = port5, recording write buffer 0 ~ 7 selection<br>1 = port5, recording write buffer 8 ~ 15 selection<br>2 = port5, SPOT write buffer 0 ~ 7 selection<br>3 = port5, SPOT write buffer 8 ~ 15 selection<br>4 = port6, recording write buffer 0 ~ 7 selection<br>5 = port6, recording write buffer 8 ~ 15 selection<br>6 = port6, SPOT write buffer 0 ~ 7 selection<br>7 = port6, SPOT write buffer 8 ~ 15 selection<br>8 = port7, recording write buffer 0 ~ 7 selection<br>9 = port7, recording write buffer 8 ~ 15 selection<br>10 = port7, SPOT write buffer 0 ~ 7 selection<br>11 = port7, SPOT write buffer 8 ~ 15 selection<br>12 = port8, recording write buffer 0 ~ 7 selection<br>13 = port8, recording write buffer 8 ~ 15 selection<br>14 = port8, SPOT write buffer 0 ~ 7 selection<br>15 = port8, SPOT write buffer 8 ~ 15 selection<br>16 = network, recording write buffer 0 ~ 7 selection<br>17 = network, recording write buffer 8 ~ 15 selection<br>18 = network, SPOT write buffer 0 ~ 7 selection<br>19 = network, SPOT write buffer 8 ~ 15 selection |

### NEW WRITE BUFFER MAPPING CONTROL REGISTER– 0XCFE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | **EN_DATA**<br>New write buffer mapping control data<br>According to the value of EN_ADDRI(0xCFD[4:0]), this value select write buffer for each read port<br>0 = disable, 1= enable |

# Motion Dection Unit

## Introduction

**TW2880 has motion detection circuitry for each incoming video channels (all together the number is 16). The source of MD circuit is the 16 D1 stream coming from the input. To do the job, TW2880 divided the first field of each stream into a 16x12 cell array. From here a unique signature for each cell is extracted and saved into DRAM buffer for later use. The second field of each frame is discarded for simplicity and cost issue. The motion detection algorithm compares the difference of luminance value between current field and reference field to determine whether a motion has occurred.**

**Uses the same detection engine, TW2880 also supports blind and night detection. The motion detector is operated with the second memory controller module.**

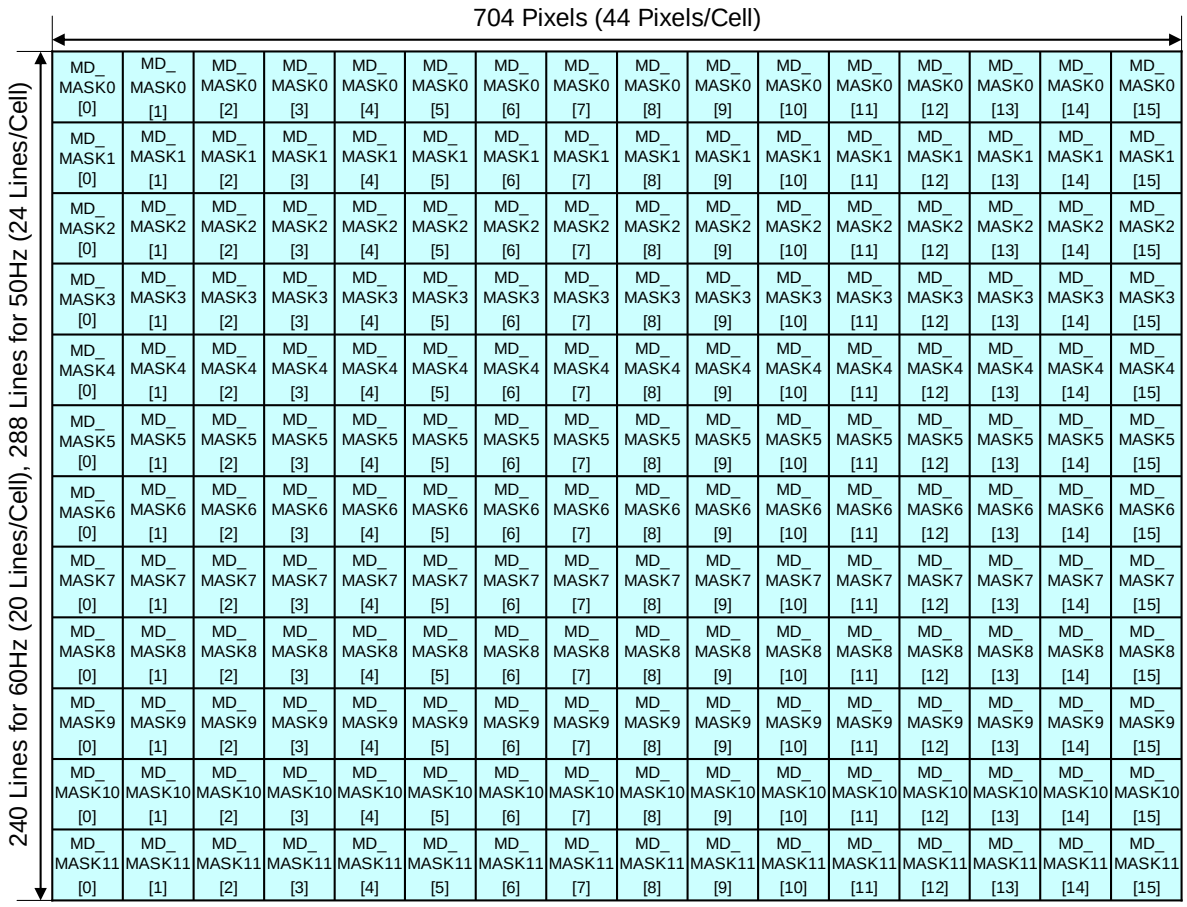


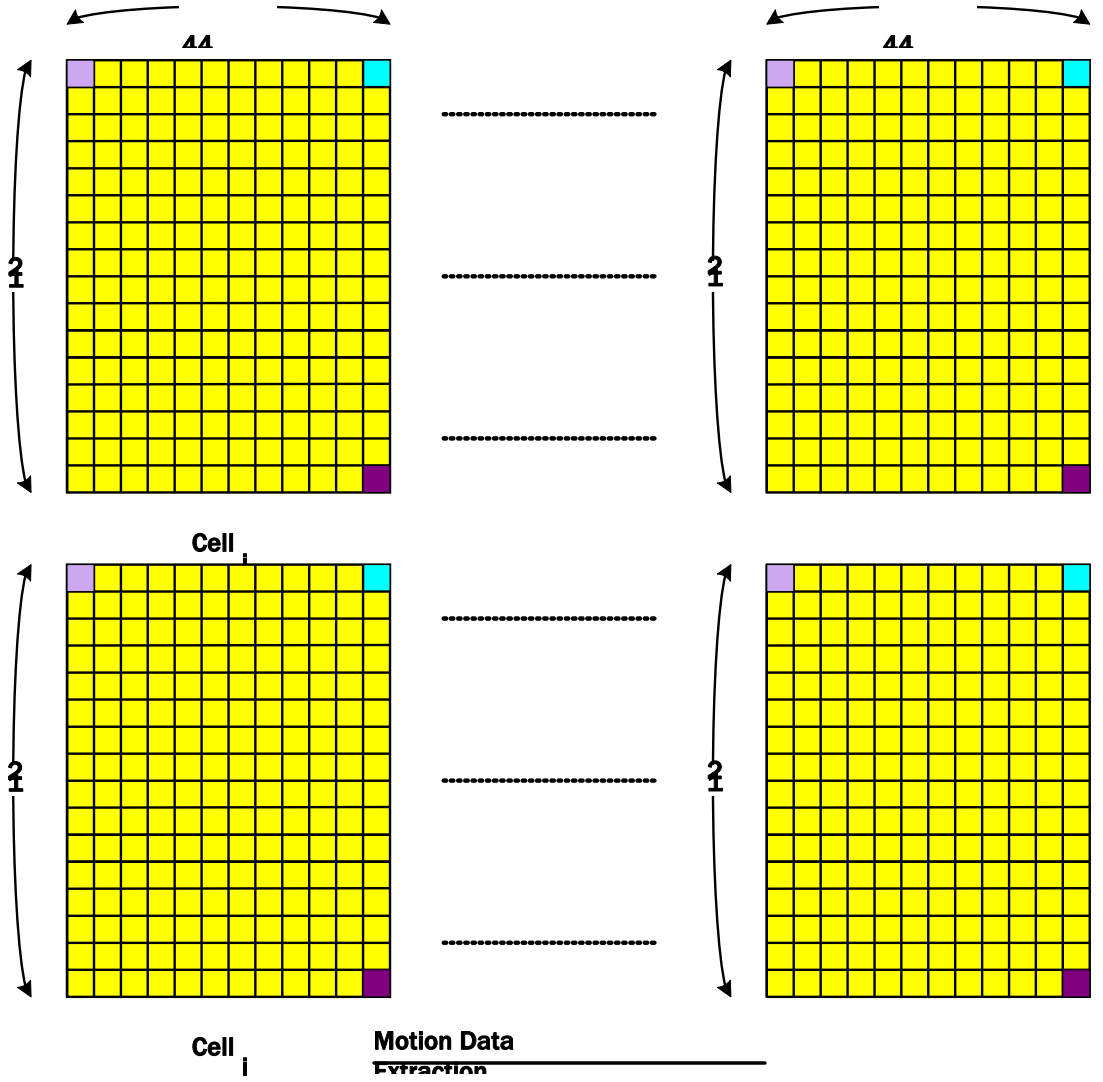

**FIGURE 22. MOTION DETECTION MASK AND CELL DEFINITION**

## Motion Data Extraction

**Motion data extraction process is followed the scan order, from left to right and from top to bottom, performed on a line by line basis. First, the luminance value of each pixel within a cell is added together and divided by a fixed value. The result is stored into a 16x13 FIFO. Please remember the number of horizontal pixels is flexible so the number of pixels in each cell line varies. This adding and dividing process will repeat itself until it reaches the end of the cell where it will repeat 20 times. Each time a new line luminance value is created; it will be added with the previous one and divided with a fixed number. In the end, an 8 bit number is extracted to represent each cell. So for each frame, the data storage needed is:**

**8 x 16 = 128 bit (for a cell row)**

**128 x 12 / 8 = 192 Byte (for a motion frame)**




## Mask and Detection Region Selection

**The motion detection algorithm utilizes the full screen video data and detects individual motion of 32x24 cells. Like the extraction process, this full screen for motion detection consists of 704 pixels and 240 lines for NTSC and 288 lines for PAL.**

**Each cell can be masked via the MD_MASK (0x800/0x900/0xa00/0xb00 ~ 0x817/0x917/0xa17/0xb17, 0x840/0x940/0xa40/0xb40 ~ 0x857/0x957/0xa57/0xb57, 0x880/0x980/0xa80/0xb80 ~ 0x897/0x997/0xa97/0xb97, 0x8c0/0x9c0/0xac0/0xbc0 ~ 0x8d7/0x9d7/0xad7/0xbd7) register as illustrated in Fig 7. If the mask bit in specific cell is programmed to high, the related cell is ignored for motion detection.**

**The motion detection result is stored in registers (0x820/0x920/0xa20/0xb20 ~ 0x837/0x937/0xa37/0xb37, 0x860/0x960/0xa60/0xb60 ~ 0x877/0x977/0xa77/0xb77, 0x8a0/0x9a0/0xaa0/0xba0 ~ 0x8b7/0x9b7/0xab7/0xbb7, 0x8e0/0x9e0/0xae0/0xbe0 ~ 0x8f7/0x9f7/0xaf7/0xbf7) and an "1" indicates detecting motion and "0" denotes no motion detection in the cell.**

To detect motion properly according to situation, TW2880 provides several sensitivity and velocity control parameters for each motion detector. TW2880 supports manual strobe function to update motion detection so that it is more appropriate for user-defined motion sensitivity control.

When no video, motion, blind and night conditions are detected in any video inputs, TW2880 provides the interrupt request to host via the IRQ pin if you enable the request line. The host processor can read the information of motion, blind or night detection by accessing the No Video IRQ Status Register (0xCB9, 0xCB8), Motion IRQ Status Register (0xCBB, 0xCBA), Blind Detection IRQ Status Register (0xCBD, 0xCBC) and the Night Detection IRQ Status Register (0xCBF, 0xCBE). This status information is updated in the vertical blank period of each input.

TW2880 supports an overlay function to display the motion detection result in the picture with 2D arrayed box. Motion detection information can display using the 2-D Box in RGB Path, Record Path can display by Channel ID type. We can get this information from host interface (Status Read) or used by Interrupt / General Purpose IO pin.

## Sensitivity Control

The motion detector has 4 sensitivity parameters to control threshold of motion detection such as the level sensitivity via the MD_LVSENS register, the spatial sensitivity via the MD_SPSENS and MD_CELSENS register, and the temporal sensitivity parameter via the MD_TMPSENS register.

### LEVEL SENSITIVITY

In built-in motion detection algorithm, the motion is detected when luminance level difference between current and reference field is greater than MD_LVSENS value. Motion detector is more sensitive for the smaller MD_LVSENS value and less sensitive for the larger. When the MD_LVSENS is too small, the motion detector may be weak in noise.

### SPATIAL SENSITIVITY

The TW2880 uses 192 (16x12) detection cells in full screen for motion detection. Each detection cell is composed of 44 pixels and 20 lines for NTSC and 24 lines for PAL. Motion detection from only luminance level difference between two fields is very weak in spatial random noise. To remove the fake motion detection from the random noise, the TW2880 supports a spatial filter via the MD_SPSENS register which defines the number of detected cell to decide motion detection in full size image. The large MD_SPSENS value increases the immunity of spatial random noise.

Each detection cell has 4 sub-cells also. Actually motion detection of each cell comes from comparison of sub-cells in it. The MD_CELSENS defines the number of detected sub-cell to decide motion detection in cell. That is, the large MD_CELSENS value increases the immunity of spatial random noise in detection cell.

### TEMPORAL SENSITIVITY

Similarly, temporal filter is used to remove the fake motion detection from the temporal random noise. The MD_TMPSENS regulates the number of taps in the temporal filter to control the temporal sensitivity so that the large MD_TMPSENS value increases the immunity of temporal random noise.

### VELOCITY CONTROL

The motion has various velocities. That is, in a fast motion an object appears and disappears rapidly between the adjacent fields while in a slow motion it is to the contrary. As the built-in motion detection algorithm uses the only luminance level difference between two adjacent fields, a slow motion is inferior in detection rate to a fast motion. To compensate this weakness, MD_SPEED parameter is used which is controllable up to 64 fields(2seconds) for NTSC and 30 fields(2seconds) for PAL 64bit SDRAM and 15 fields(1second) for PAL 32bit SDRAM. MD_SPEED parameter adjusts the field interval in which the luminance level is compared. Thus, for detection of a fast motion a small value is needed and for a slow motion a large value is required. The parameter MD_SPEED value should be greater than MD_TMPSENS value.

Additionally, TW2880 has 2 more parameters to control the selection of reference field. The MD_FIELD[1:0] bit is a field selection parameter such as odd, even, any field or frame.

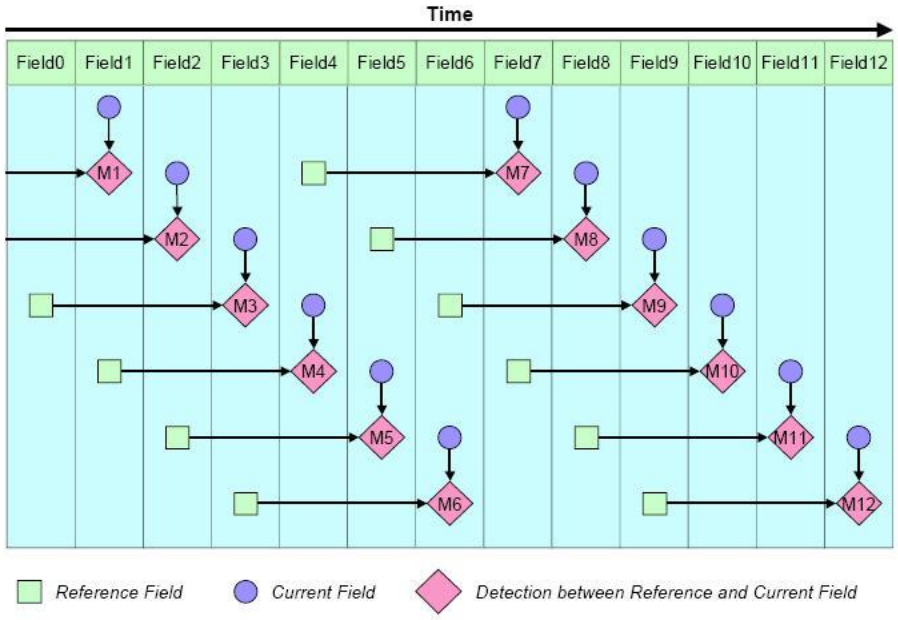



Fig 17 The relat

FIGURE 23. THE RELATIONSHIP BETWEEN CURRENT AND REFERENCE FIELD WHEN MD_REFFLD = "0"

**The MD_REFFLD bit is designed to control the updating period of reference field. If MD_REFFLD = "0", the interval from current field to reference field is always same as the MD_SPEED. It means that the reference field is always updated every field. The Figure 23 shows the relationship between current and reference field for motion detection when MD_REFFLD is set to 0.**

**TW2880 can update the reference field only at the period of MD_SPEED when the MD_REFFLD is high. For this case, the TW2880 can detect a motion with sense of a various velocity. The Fig 18 shows the relationship between current and reference field for motion detection when the MD_REFFLD equals to 1.**

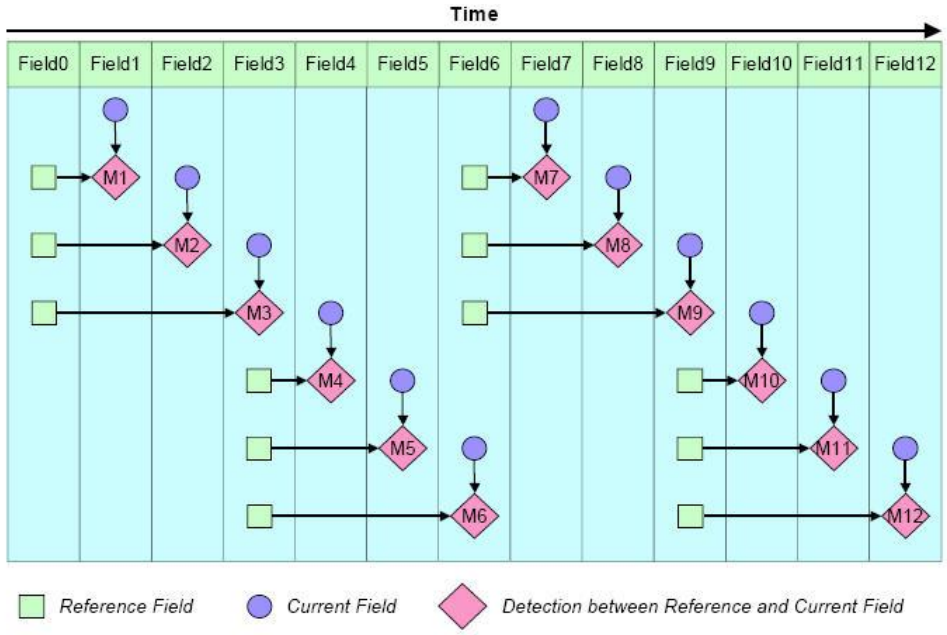



Fig 18 The re

FIGURE 24. THE RELATIONSHIP BETWEEN CURRENT AND REFERENCE FIELD WHEN MD_REFFLD = "1"

**TW2880 also supports the manual detection timing control of the reference field/frame via the MD_STRB_EN and MD_STRB register bits in the MD control registers. If MD_STRB_EN is set to 0, the reference field/frame is automatically updated and reserved on every reference field/frame. If MD_STRB_EN is set to1, the reference field/frame is updated and reserved only when MD_STRB bit is set to 1. If an external strobing signal is used, one can set the mode to 1x and select it. The strobe signal is coming from outside via trigger_in pin. In these two modes, the interval between current and reference field/frame is controlled by user's strobe timing and are very useful for some specific purpose like non-periodical velocity control and very slow motion detection.**

## Blind Detection

**If the luminous level of a video input in every corner area is almost equal to the average luminous level of this frame due to camera got covered by something, this input is defined as blind input. TW2880 supports blind detection individually for all 16 video inputs and generates an interrupt to host CPU.**

**TW2880 uses two sensitivity parameters to detect blind input. One is the level sensitivity via the BD_LVSENS register and the other is spatial sensitivity via the BD_CELSENS register. The TW2880 uses total 192 (16x12) cells in full screen for blind detection. The BD_LVSENS parameter controls the threshold of level between cell and field average. The BD_CELSENS parameter defines the number of cells to detect blind. For BD_CELSENS = "0", the number of cell whose level is same as average of field should be over than 60% to detect blind, 70% for BD_CELSENS = "1", 80% for BD_CELSENS = "2", and 90% for BD_CELSENS = "3". That is, the large value of BD_LVSENS and BD_CELSENS makes blind detector less sensitive.**

**TW2880 also supports dual detection mode for non real time application such as pseudo 8ch application via the MD_DUAL_EN register. The host can read blind detection information for both VIN_A and VIN_B input via the IRQENA_BD (0x17A) register.**

## Night Detection

**TW2880 uses a user defined, fixed value to determine whether a video input is in a broad day light or at night situations. If the average of luminous level is lower than this fixed value, this input is defined as night input. Likewise, the opposite is defined as day input. The TW2880 supports night detection for all 16 video inputs and will generated interrupts to host CPU if triggered.**

**Two parameters are used to detect night input. One is the level sensitivity via the ND_LVSENS register and the other is temporal sensitivity via the ND_TMPSENS register. The ND_LVSENS parameter controls threshold level of day and night. The ND_TMPSENS parameter regulates the number of taps in the temporal low pass filter to control the temporal sensitivity. The large value of ND_LVSENS and ND_TMPSENS makes night detector less sensitive.**

**The TW2880 also supports dual detection mode for non-real-time application such as pseudo 8ch application via the MD_DUAL_EN register. The host can read night detection information for both VIN_A and VIN_B input via the IRQENA_ND (0x17B) register.**

## Register Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x800 | R/W | 0x00 | Motion Detection Cell Line 0 Mask Register 1 |
| 0x801 | R/W | 0x00 | Motion Detection Cell Line 0 Mask Register 2 |
| 0x802 | R/W | 0x00 | Motion Detection Cell Line 1 Mask Register 1 |
| 0x803 | R/W | 0x00 | Motion Detection Cell Line 1 Mask Register 2 |
| 0x804 | R/W | 0x00 | Motion Detection Cell Line 2 Mask Register 1 |
| 0x805 | R/W | 0x00 | Motion Detection Cell Line 2 Mask Register 1 |
| 0x806 | R/W | 0x00 | Motion Detection Cell Line 3 Mask Register 2 |
| 0x807 | R/W | 0x00 | Motion Detection Cell Line 3 Mask Register 1 |
| 0x808 | R/W | 0x00 | Motion Detection Cell Line 4 Mask Register 2 |
| 0x809 | R/W | 0x00 | Motion Detection Cell Line 4 Mask Register 1 |
| 0x80A | R/W | 0x00 | Motion Detection Cell Line 5 Mask Register 1 |
| 0x80B | R/W | 0x00 | Motion Detection Cell Line 5 Mask Register 2 |
| 0x80C | R/W | 0x00 | Motion Detection Cell Line 6 Mask Register 1 |
| 0x80D | R/W | 0x00 | Motion Detection Cell Line 6 Mask Register 2 |
| 0x80E | R/W | 0x00 | Motion Detection Cell Line 7 Mask Register 1 |
| 0x80F | R/W | 0x00 | Motion Detection Cell Line 7 Mask Register 2 |
| 0x810 | R/W | 0x00 | Motion Detection Cell Line 8 Mask Register 1 |
| 0x811 | R/W | 0x00 | Motion Detection Cell Line 8 Mask Register 2 |
| 0x812 | R/W | 0x00 | Motion Detection Cell Line 9 Mask Register 1 |
| 0x813 | R/W | 0x00 | Motion Detection Cell Line 9 Mask Register 2 |
| 0x814 | R/W | 0x00 | Motion Detection Cell Line 10 Mask Register 1 |
| 0x815 | R/W | 0x00 | Motion Detection Cell Line 10 Mask Register 2 |
| 0x816 | R/W | 0x00 | Motion Detection Cell Line 11 Mask Register 1 |
| 0x817 | R/W | 0x00 | Motion Detection Cell Line 11 Mask Register 2 |
| 0x818 | R/W | 0x00 | Motion Detection Control Register 1 |
| 0x819 | R/W | 0x00 | Motion Detection Control Register 2 |
| 0x81A | R/W | 0x00 | Motion Detection Control Register 3 |
| 0x81B | R/W | 0x00 | Motion Detection Control Register 4 |
| 0x81E | R/W | 0x00 | Motion Detection Control Register 5 |
| 0x81F | R/W | 0x00 | Motion Detection Control Register 6 |
| 0x820 | R/W | 0x00 | Motion Detection Cell Line 0 Result Register 1 |
| 0x821 | R/W | 0x00 | Motion Detection Cell Line 0 Result Register 2 |
| 0x822 | R/W | 0x00 | Motion Detection Cell Line 1 Result Register 1 |
| 0x823 | R/W | 0x00 | Motion Detection Cell Line 1 Result Register 2 |
| 0x824 | R/W | 0x00 | Motion Detection Cell Line 2 Result Register 1 |
| 0x825 | R/W | 0x00 | Motion Detection Cell Line 2 Result Register 2 |
| 0x826 | R/W | 0x00 | Motion Detection Cell Line 3 Result Register 1 |
| 0x827 | R/W | 0x00 | Motion Detection Cell Line 3 Result Register 2 |
| 0x828 | R/W | 0x00 | Motion Detection Cell Line 4 Result Register 1 |
| 0x829 | R/W | 0x00 | Motion Detection Cell Line 4 Result Register 2 |
| 0x82A | R/W | 0x00 | Motion Detection Cell Line 5 Result Register 1 |
| 0x82B | R/W | 0x00 | Motion Detection Cell Line 5 Result Register 2 |
| 0x82C | R/W | 0x00 | Motion Detection Cell Line 6 Result Register 1 |
| 0x82D | R/W | 0x00 | Motion Detection Cell Line 6 Result Register 2 |
| 0x82E | R/W | 0x00 | Motion Detection Cell Line 7 Result Register 1 |
| 0x82F | R/W | 0x00 | Motion Detection Cell Line 7 Result Register 2 |
| 0x830 | R/W | 0x00 | Motion Detection Cell Line 8 Result Register 1 |
| 0x831 | R/W | 0x00 | Motion Detection Cell Line 8 Result Register 2 |
| 0x832 | R/W | 0x00 | Motion Detection Cell Line 9 Result Register 1 |
| 0x833 | R/W | 0x00 | Motion Detection Cell Line 9 Result Register 2 |
| 0x834 | R/W | 0x00 | Motion Detection Cell Line 10 Result Register 1 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x835 | R/W | 0x00 | Motion Detection Cell Line 10 Result Register 2 |
| 0x836 | R/W | 0x00 | Motion Detection Cell Line 11 Result Register 1 |
| 0x837 | R/W | 0x00 | Motion Detection Cell Line 11 Result Register 2 |

## Registers Description

### MOTION DETECTION CELL LINE 0 MASK REGISTER 1 – 0X800

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK0[7:0] |

This is for channel 0. For other channel the assignments are: 0x840, 0x880, 0x8C0, 0x900, 0x940, 0x980, 0x9C0, 0xA00, 0xA40, 0xA80, 0xAC0, 0xB00, 0xB40, 0xB80, 0xBC0 respectively

### MOTION DETECTION CELL LINE 0 MASK REGISTER 2 – 0X801

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK0[15:8] |

### MOTION DETECTION CELL LINE 1 MASK REGISTER 1 – 0X802

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK1[7:0] |

### MOTION DETECTION CELL LINE 1 MASK REGISTER 2 – 0X803

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK1[15:8] |

### MOTION DETECTION CELL LINE 2 MASK REGISTER 1 – 0X804

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK2[7:0] |

### MOTION DETECTION CELL LINE 2 MASK REGISTER 2 – 0X805

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK2[15:8] |

### MOTION DETECTION CELL LINE 3 MASK REGISTER 1 – 0X806

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK3[7:0] |

### MOTION DETECTION CELL LINE 3 MASK REGISTER 2 – 0X807

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK3[15:8] |

### MOTION DETECTION CELL LINE 4 MASK REGISTER 1 – 0X808

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK4[7:0] |

### MOTION DETECTION CELL LINE 4 MASK REGISTER 2 – 0X809

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK4[15:8] |

### MOTION DETECTION CELL LINE 5 MASK REGISTER 1 – 0X80A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK5[7:0] |

### MOTION DETECTION CELL LINE 5 MASK REGISTER 2 – 0X80B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK5[15:8] |

### MOTION DETECTION CELL LINE 6 MASK REGISTER 1 – 0X80C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK6[7:0] |

### MOTION DETECTION CELL LINE 6 MASK REGISTER 2 – 0X80D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK6[15:8] |

### MOTION DETECTION CELL LINE 7 MASK REGISTER 1 – 0X80E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK7[7:0] |

### MOTION DETECTION CELL LINE 7 MASK REGISTER 2 – 0X80F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK7[15:8] |

### MOTION DETECTION CELL LINE 8 MASK REGISTER 1 – 0X810

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK8[7:0] |

### MOTION DETECTION CELL LINE 8 MASK REGISTER 2 – 0X811

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK8[15:8] |

### MOTION DETECTION CELL LINE 9 MASK REGISTER 1 – 0X812

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK9[7:0] |

### MOTION DETECTION CELL LINE 9 MASK REGISTER 2 – 0X813

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK9[15:8] |

### MOTION DETECTION CELL LINE 10 MASK REGISTER 1 – 0X814

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK10[7:0] |

### MOTION DETECTION CELL LINE 10 MASK REGISTER 2 – 0X815

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK10[15:8] |

### MOTION DETECTION CELL LINE 11 MASK REGISTER 1 – 0X816

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK11[7:0] |

### MOTION DETECTION CELL LINE 11 MASK REGISTER 2 – 0X817

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | MD_MASK11[15:8] |

### MOTION DETECTION CONTROL REGISTER 1 – 0X818

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **MD_DIS,** channel disable, 1 = disable |
| 6 | R/W | 0x0 | **Reserved** |
| 5:4 | R/W | 0x0 | **BD_CELLSENS,** blind cell sensitivity adjust<br><br>00 = 60%<br>01 = 70%<br>10 = 80%<br>11 = 90% |
| 3:0 | R/W | 0x0 | **BD_LVSENS,** Blind level sensitivity select |

### MOTION DETECTION CONTROL REGISTER 2 – 0X819

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **ND_LVSENS,** night detection level sensitivity adjust |
| 3:0 | R/W | 0x0 | **ND_TMPSENS,** night detection temporal sensitivity adjust |

### MOTION DETECTION CONTROL REGISTER 3 – 0X81A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0x0 | **MD_FIELD,** motion detection field select |
| 5 | R/W | 0x0 | **Reserved** |
| 4:0 | R/W | 0x0 | **MD_LVSENS,** motion detection level sensitivity adjust |

### MOTION DETECTION CONTROL REGISTER 4 – 0X81B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **Reserved** |
| 6 | R/W | 0x0 | **MD_REFFLD,** Reference Field Update Mode<br><br>0 : Update Every Field, 1 : Update Every MD_SPEED Period |
| 5:0 | R/W | 0x0 | **MD_SPEED,** temporal sensitivity adjust |

### MOTION DETECTION CONTROL REGISTER 5 – 0X81E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | **MD_TMPSENS,** motion detection temporal sensitivity adjust |
| 3:0 | R/W | 0x0 | **MD_SPSENS,** motion spatial sensitivity adjust |

### MOTION DETECTION CONTROL REGISTER 6 – 0X81F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0x0 | **MD_VINV,** motion detection display mode, 1 = vertical invert |
| 6 | R/W | 0x0 | **MD_HINV,** motion detection display mode, 1 = horizontal invert |
| 5:4 | R/W | 0x0 | **MD_CELLSENS,**<br>Define the threshold of sub-cell number for motion detection<br><br>0 = Motion is detected if 1 sub-cell has motion(default)<br>1 = Motion is detected if 2 sub-cell has motion<br>2 = Motion is detected if 3 sub-cell has motion<br>3 = Motion is detected if 4 sub-cell has motion(less sensitive) |
| 3 | R/W | 0x0 | **Reserved** |
| 2 | R/W | 0x0 | **MD_STRB,** internal user strobe signal |
| 1:0 | R/W | 0x0 | **Strobe mode**<br><br>00 = automatic mode<br>01 = internal strobe mode, **MD_STRB_EN = 1**<br>1x = external strobe mode |

### MOTION DETECTION CELL LINE 0 RESULT REGISTER 1 – 0X820

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | **md_det0[7:0]** |

**This is for channel 0. For other channel the assignments are: 0x860, 0x8A0, 0x8E0, 0x920, 0x960, 0x9A0, 0x9E0, 0xA20, 0xA60, 0xAA0, 0xAE0, 0xB20, 0xB60, 0xBA0, 0xBE0 respectively**

### MOTION DETECTION CELL LINE 0 RESULT REGISTER 2 – 0X821

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | **md_det0[15:8]** |

### MOTION DETECTION CELL LINE 1 RESULT REGISTER 1 – 0X822

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det1[7:0] |

### MOTION DETECTION CELL LINE 1 RESULT REGISTER 2 – 0X823

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det1[15:8] |

### MOTION DETECTION CELL LINE 2 RESULT REGISTER 1 – 0X824

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det2[7:0] |

### MOTION DETECTION CELL LINE 2 RESULT REGISTER 2 – 0X825

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det2[15:8] |

### MOTION DETECTION CELL LINE 3 RESULT REGISTER 1 – 0X826

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det3[7:0] |

### MOTION DETECTION CELL LINE 3 RESULT REGISTER 2 – 0X827

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det3[15:8] |

### MOTION DETECTION CELL LINE 4 RESULT REGISTER 1 – 0X828

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det4[7:0] |

### MOTION DETECTION CELL LINE 4 RESULT REGISTER 2 – 0X829

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det4[15:8] |

### MOTION DETECTION CELL LINE 5 RESULT REGISTER 1 – 0X82A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det5[7:0] |

### MOTION DETECTION CELL LINE 5 RESULT REGISTER 2 – 0X82B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det5[15:8] |

### MOTION DETECTION CELL LINE 6 RESULT REGISTER 1 – 0X82C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det6[7:0] |

### MOTION DETECTION CELL LINE 6 RESULT REGISTER 2 – 0X82D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det6[15:8] |

### MOTION DETECTION CELL LINE 7 RESULT REGISTER 1 – 0X82E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det7[7:0] |

### MOTION DETECTION CELL LINE 7 RESULT REGISTER 2 – 0X82F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det7[15:8] |

### MOTION DETECTION CELL LINE 8 RESULT REGISTER 1 – 0X830

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det8[7:0] |

### MOTION DETECTION CELL LINE 8 RESULT REGISTER 2 – 0X831

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det8[15:8] |

### MOTION DETECTION CELL LINE 9 RESULT REGISTER 1 – 0X832

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det9[7:0] |

### MOTION DETECTION CELL LINE 9 RESULT REGISTER 2 – 0X833

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det9[15:8] |

### MOTION DETECTION CELL LINE 10 RESULT REGISTER 1 – 0X834

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det10[7:0] |

### MOTION DETECTION CELL LINE 10 RESULT REGISTER 2 – 0X835

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det10[15:8] |

### MOTION DETECTION CELL LINE 11 RESULT REGISTER 1 – 0X836

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det11[7:0] |

### MOTION DETECTION CELL LINE 11 RESULT REGISTER 2 – 0X837

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0x0 | md_det11[15:8] |

# Channel ID Encoder Unit

## Channel ID Encoder

The video output sequences from each of the TW2880 output port have critical system information embedded in the VBI area called channel ID. This information is very useful for the people who try to capture and utilize the video sequence later on. TW2880 supports three kinds of channel ID: User channel ID, Detection channel ID and Auto channel ID. The output order of these IDs in a TW2880 generated video sequence is fixed as Auto -> Repeat Auto(only in analog format) -> Detection -> User Channel ID. Definitions of each channel ID are described in detailed in the following sections. Each channel ID unit for each output port can be enabled and disabled independently. The Channel ID Format of TW2880 is very similar to TW2835, but total amount of data increase a little bit.

## Channel ID Definition

### AUTO CHANNEL ID

TW2880's Auto channel ID is 5 bytes long and is used for automatic identification of picture configuration such as video input path number, analog switch, event, region enable, field/frame mode information and CIF channel info. The ID information will repeat once.

| TYPE | BIT | NAME | FUNCTION |
|---|---|---|---|
| Auto Channel ID | [39:38] | ID_TYPE | 2'b00 = Auto Channel ID |
| | [37:36] | LINE_NUM | Line Number : 0 : 1st Line, 1 : 2nd Line (auto repeat) |
| | [35] | F_OUT | Encoder Field Polarity. 0 : Odd, 1 : Even |
| | [34:32] | Record port | 0 – 7 = Rec0 – Rec7 |
| | [31:28] | CH_NUM1 | 1/4 Quadrant Channel Number |
| | [27:24] | CH_NUM2 | 2/4 Quadrant Channel Number |
| | [23:20] | CH_NUM3 | 3/4 Quadrant Channel Number |
| | [19:16] | CH_NUM4 | 4/4 Quadrant Channel Number |
| | [15] | Interleave | 0: Frame, 1: Field |
| | [14] | Interlace | In frame interleave, 0: Progressive, 1: Interlaced.<br>In field interleave, 0: non field switch, 1: field switch |
| | [13:12] | Resolution | 0 : Full D1, 1 : CIF, 2 : 4D1, 3 : BT1120 |
| | [11] | H_SPLIT | 0: no split, 1: split |
| | [10] | V_SPLIT | 0: no split, 1: split |
| | [9] | FLD_MODE | 0: top field is 0, 1: top field is 1 |
| | [8] | - | 0: [7:4] use for no video, 1: [7:4] is used for ignored bit |

| TYPE | BIT | NAME | FUNCTION |
|---|---|---|---|
| | [7:0] | DET_INFO | [7]: No video or ignored function for 1/4 quadrant channel<br>[6]: No video or ignored function for 2/4 quadrant channel<br>[5]: No video or ignored function for 3/4 quadrant channel<br>[4]: No video or ignored function for 4/4 quadrant channel<br>[3]: Blind for 1/4 quadrant channel<br>[2]: Blind for 2/4 quadrant channel<br>[1]: Blind for 3/4 quadrant channel<br>[0]: Blind for 4/4 quadrant channel |

**DETECTION CHANNEL ID**

Detection channel ID is composed of 40 byte data (8 lines) and is used for real time detected information such as motion, video loss, blind and night detection.

| TYPE | BIT | NAME | FUNCTION |
|---|---|---|---|
| Detection Channel ID | [39:38] | ID_TYPE | Detection Channel ID = 2'b10 |
| | [37:36] | LINE_NUM | Line Number : 0 : 1st Line, 1 : 2nd Line |
| | [35] | F_OUT | Encoder Field Polarity. 0 : Odd, 1 : Even |
| | [34:32] | Record port | 0 – 7 = Rec0 – Rec7 |
| | [31:0] | DET_INFO | [31:28] : 0/8 Ch DET_INFO (two lines)<br>~~<br>[03:00] : 7/15 Ch DET_INFO<br>DET_INFO[3] : Video Loss Information<br>DET_INFO[2] : Motion Information<br>DET_INFO[1] : Blind Information<br>DET_INFO[0] : Night Information |

196-bit motion detection information is also included in Detection channel ID, total is 24 bytes (6 lines). Two lines use ID_TYPE 2'b10, the rest four lines use ID_TYPE 2'b01;

| TYPE | BIT | NAME | FUNCTION |
|---|---|---|---|
| Detection Channel ID | [39:38] | ID_TYPE | Motion Detection Channel ID = 2'b10 (shared) |
| | [37:36] | LINE_NUM | Line Number : 2 : 3rd Line, 3 : 4th Line |
| | [35:32] | MD_SEL | Which channel is shown on 192 bits. |
| | [31:0] | DET_INFO | Line2: MD_DET[191:160] : Motion Detection Info<br>Line3: MD_DET[159:128] : Motion Detection Info |

| TYPE | BIT | NAME | FUNCTION |
|---|---|---|---|
| Detection Channel ID | [39:38] | ID_TYPE | Motion Detection Channel ID = 2'b01 |
| | [37:36] | LINE_NUM | Line Number : 0 : 5 th Line, 1 : 6th Line, 2 : 7th line, 3 : 8th line |
| | [35:32] | MD_SEL | Which channel is shown on 192 bits. |
| | [31:0] | DET_INFO | Line 0: MD_DET[127:96] : Motion Detection Info<br>Line 1: MD_DET[95:64] : Motion Detection Info<br>Line 2: MD_DET[63: 32] : Motion Detection Info<br>Line 3: MD_DET[31:0] : Motion Detection Info |

192 bit motion information is only for one channel. Each channel is divided by 16x12=192 windows. Each bit indicates 1 window motion status. To select different channel motion data, configure register MOTION_SEL.

**USER CHANNEL ID**

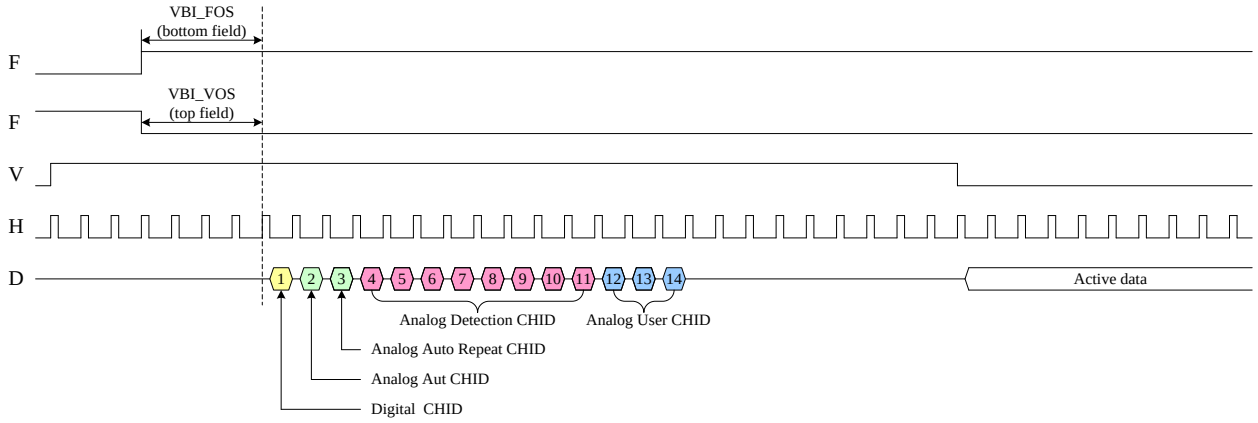
User Channel ID is composed of 15 bytes data (three lines) and is used for indicating customized information such as system information and date.

| TYPE | BIT | NAME | FUNCTION |
|---|---|---|---|
| User Channel ID | [39:38] | ID_TYPE | 2'b11 = User Channel ID |
| | [37:36] | LINE_NUM | Line Number : 0 : 1st Line, 1 : 2nd Line, 2 : 3rd Line |
| | [35] | F_OUT | Encoder Field Polarity. 0 : Odd, 1 : Even |
| | [34:32] | Record port | 0 – 7 = Rec0 – Rec7 |
| | [31:0] | MAN_ID | [31:0] : Manual User ID |

## Channel ID Output Method

TW2880 has two ways of transmitting channel ID: analog type channel ID and digital type channel ID. There are totally 14 lines channel ID data, which include 1 line of digital channel ID and 13 lines of analog channel ID. In analog channel ID, there are 1 line of auto channel ID, 1 line repeat auto channel ID, 8 lines of detection channel ID and 3 lines of user channel ID.

| | CHANNEL ID LINE NUMBER |
|---|---|
| 1 | Digital channel ID |
| 2 | Analog auto channel ID |
| 3 | Analog auto repeat channel ID |
| 4 | Analog detection channel ID 1 |
| 5 | Analog detection channel ID 2 |
| 6 | Analog detection channel ID 3 |
| 7 | Analog detection channel ID 4 |
| 8 | Analog detection channel ID 5 |
| 9 | Analog detection channel ID 6 |
| 10 | Analog detection channel ID 7 |
| 11 | Analog detection channel ID 8 |
| 12 | Analog user channel ID 1 |
| 13 | Analog user channel ID 2 |
| 14 | Analog user channel ID 3 |



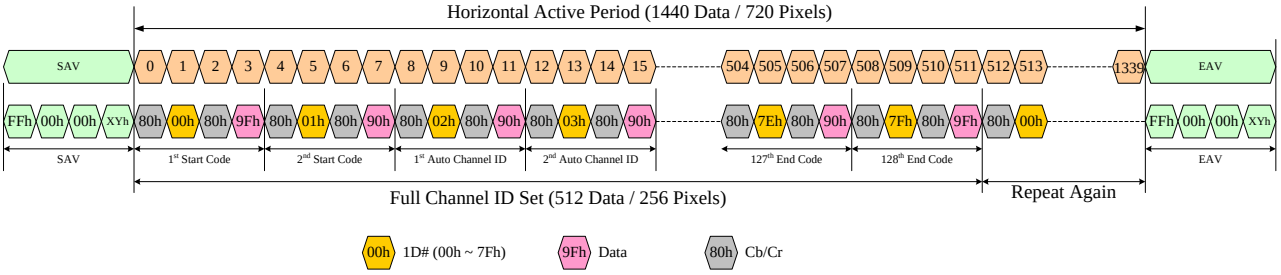

**TIMING DIAGRAM OF CHANNEL ID**

## DIGITAL CHANNEL ID

**TW2880 provides the digital type channel ID during VBI period. It's useful for DSP application because the channel ID can be inserted in just one line with special format. The digital channel ID is located before analog channel ID line. The digital channel ID can be enabled via the VIS_CODE_EN register.**

**The digital channel ID is inserted in Y data in BT656 or BT1120 stream and composed of ID # and channel information. The ID # indicates the index of digital type channel ID including the start code, auto/detection/user channel ID and end code. The ID number has 0 ~ 0x7F index and each channel information of one byte is divided into 2 bytes of 4 LSB that takes "0x90" offset against ID # for discrimination. The start code is located in ID# 0 ~1 and auto channel ID is situated in ID # 2 ~ 11. The detection channel ID is located in ID # 12 ~ 91 and the user channel ID is situated in ID # 92 ~ 121. The end codes occupied the others. The digital channel ID will be repeated during horizontal active period. There is no repeat auto channel ID in digital format. The following figure shows the illustration of digital channel ID.**

**In the TW2880 output sequence, the vertical offset of odd field is controlled by the VIS_LINE_VOS register with 1 line unit and even field is control by the VIS_LINE_FOS register with 1 line unit. Channel ID can be flexibly enabled by register VIS_ID_OEN and VIS_ID_EEN for odd field and even field respectively. These registers make it possible to insert channel ID in vertical active region if user required.**




**DIGITAL CHANNEL ID TIMING DIAGRAM**

| 1D# | DATA | DESCRIPTION |
|---|---|---|
| 0 (00h) | 9Fh | Start Code |
| 1 (01h) | 90h | |
| 2 (02h) | {9, A0_MSB} | Auto Channel ID (5x2=10 bytes) |
| 3 (03h) | {9, A0_LSB} | |
| ... | ... | |
| 10 (0Ah) | {9, A4_MSB} | |
| 11 (0Bh) | {9, A4_LSB} | |
| 12 (0Ch) | **{9, D0_MSB}** | **Detection Channel ID (40x2=80) bytes** |
| **13 (0Dh)** | **{9, D0_LSB}** | |
| **...** | **...** | |
| **90 (5Ah)** | **{9, D39_MSB}** | |
| **91 (5Bh)** | **{9, D39_LSB}** | |
| 92 (5Ch) | {9, U0_MSB} | User Channel ID (15x2=30 bytes) |
| 93 (5Dh) | {9, U0_LSB} | |
| ... | ... | |
| 120 (78h) | {9, D14_MSB} | |
| 121 (79h) | {9, D14_LSB} | |
| **122 (7Ah)** | **90h** | **End Code** |
| **123 (7Bh)** | **9Fh** | |
| **124 (7Ch)** | **90h** | **End Code** |
| **125 (7Dh)** | **9Fh** | |
| **126 (7Eh)** | **90h** | **End Code** |
| **127 (7Fh)** | **9Fh** | |

DIGITAL CHANNEL ID DATA FORMAT

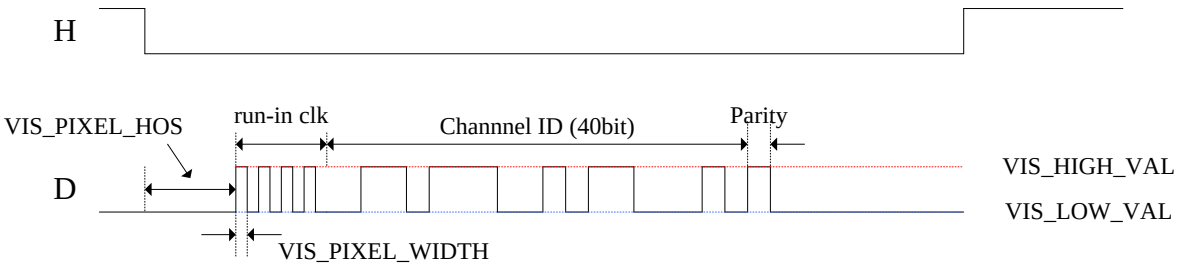
**Digital channel ID data sequence is: Cb, ID#, Cr, 9Xh, Cb ID#, Cr, 9Xh, .... After one full set, same data must be repeated in active area.**

**ANALOG CHANNEL ID**

**TW2880 supports analog type channel ID during VBI period. The analog channel ID can include an Auto channel ID, Repeat auto channel ID, Detection channel ID and User channel ID. Each channel ID can be enabled via the VIS_AUTO_EN, AUTO_RPT_EN, VIS_DET_EN, VIS_USER_EN registers. Auto channel ID requires one line basically, but can need one more line for repetition. Detection channel ID requires 8 lines and User channel ID require three lines so that total thirteen lines are used for analog type channel ID.**

**Analog channel ID is located right after digital channel ID line. Default starting position is right after SAV. The horizontal starting position offset is defined via VIS_PIXEL_HOS register with 1 pixel unit for data insertion and run in clock. The vertical start point is defined via VIS_VOS for odd field and VIS_FOS is for even field in one line offset. The pixel width of each bit is controlled by the VIS_PIXEL_WIDTH register and the magnitude of each bit is defined by the VIS_HIGH_VAL and VIS_LOW_VAL register.**

**The analog channel ID consists of run-in clock, channel ID data, type and parity bit. The run-in clock insertion is enabled via the VIS_RIC_EN register. There are four run in clock cycles. The channel ID data can include 4 byte information and the channel ID type contains 2 bits that "0" is meant for Auto channel ID and repeated channel ID, "2" and "4" for Detection channel ID, "3" for User channel ID of VIS_MAN0~5. The parity is 1 bit width and used for even parity. This definition is used for both digital and analog channel ID. There are five bytes plus 1 parity data in one line.**




**ANALOG CHANNEL ID FORMAT**

**In each line of analog channel ID, there are 4 run in clock cycle, 40 channel ID data cycle and 1 parity cycle. Each cycle has "VIS_PIXEL_WIDTH x 2" pixels. Analog channel ID is only located in Y position.**

## Channel ID in Active Region

**For CODECs that do not have the capability or bandwidth to process channel ID in the vertical blanking period, TW2880 also provides a simple way to let the channels to identify themselves. The kind of channel ID is stored in the active video area in a way that is robust against horizontal and vertical scaling. To insert the channel ID, each channel will have a 2x2 luma pixel slot in the first two lines in the active region for each field. In this slot the luma value is substituted to 0x01 or the value stored in the register 0xDAF. Take next diagram for example, channel 0 will have luma pixels Y0 and Y1, channel 1 will have luma pixels Y2 and Y3, channel 2 will have luma pixels Y4 and Y5 change to 0x01 for the first two lines.**

Line 0

| FF | 00 | 00 | SAV | CB0 | Y0 | CR0 | Y1 | CB1 | Y2 | CR1 | Y3 | CB2 | Y4 | CR2 | Y5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Line 1

| FF | 00 | 00 | SAV | CB0 | Y0 | CR0 | Y1 | CB1 | Y2 | CR1 | Y3 | CB2 | Y4 | CR2 | Y5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

**The CODEC's firmware should recover the channel ID and replaced with a interpolated value from the adjacent pixels to minimize the impact. Next diagram illustrate the output for channel 0.**

Line 0

| FF | 00 | 00 | SAV | CB0 | 01 | CR0 | 01 | CB1 | Y2 | CR1 | Y3 | CB2 | Y4 | CR2 | Y5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Line 1

| FF | 00 | 00 | SAV | CB0 | 01 | CR0 | 01 | CB1 | Y2 | CR1 | Y3 | CB2 | Y4 | CR2 | Y5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

**The substitution rule basically stay the same for 16 bit output. In this configuration, two adjacent luma pixels will get replaced while the chroma section is untouched. Registers 0xDAD and 0xDAE is used for limit the active channel ID only appears in the first frame of a output sequence. This is useful for the customers that already knows the output sequence of the incoming video and only want TW2880 to help identifying the first channel.**

## Registers Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xD00 | R/W | 0x04 | [7]: FLDMODE_PB1<br>[6]: INVALID_MD_PB1<br>[5:3]: VIN_SEL_PB1<br>[2:0]: VIS_PIXEL_WIDTH_PB1 |
| 0xD01 | R/W | 0xFF | [7]: VIS_ID_OEN_PB1<br>[6]: VIS_AUTO_EN_PB1<br>[5]: VIS_RPT_EN_PB1<br>[4]: VIS_DET_EN_PB1<br>[3]: VIS_USER_EN_PB1<br>[2]: VIS_CODE_EN_PB1<br>[1]: VIS_RIC_PB1<br>[0]: VIS_ID_EEN_PB1 |
| 0xD02 | R/W | 0x00 | VIS_HOS_PB1[7:0] |
| 0xD03 | R/W | 0x3F | VIS_HIGH_VAL_PB1[7:0] |
| 0xD04 | R/W | 0x05 | VIS_LOW_VAL_PB1[7:0] |
| 0xD05 | R/W | 0x00 | [4:0]: VIS_VOS_PB1 |
| 0xD06 | R/W | 0x00 | [7:0]: VIS_MAN0_PB1_L |
| 0xD07 | R/W | 0x00 | [7:0]: VIS_MAN0_PB1_H |
| 0xD08 | R/W | 0x00 | [7:0]: VIS_MAN1_PB1_L |
| 0xD09 | R/W | 0x00 | [7:0]: VIS_MAN1_PB1_H |
| 0xD0A | R/W | 0x00 | [7:0]: VIS_MAN2_PB1_L |
| 0xD0B | R/W | 0x00 | [7:0]: VIS_MAN2_PB1_H |
| 0xD0C | R/W | 0x00 | [7:0]: VIS_MAN3_PB1_L |
| 0xD0D | R/W | 0x00 | [7:0]: VIS_MAN3_PB1_H |
| 0xD0E | R/W | 0x00 | [7:0]: VIS_MAN4_PB1_L |
| 0xD0F | R/W | 0x00 | [7:0]: VIS_MAN4_PB1_H |
| 0xD10 | R/W | 0x00 | [7:0]: VIS_MAN5_PB1_L |
| 0xD11 | R/W | 0x00 | [7:0]: VIS_MAN5_PB1_H |
| 0xD12 | R/W | 0x04 | [7]: FLDMODE_PB2<br>[6]: INVALID_MD_PB2<br>[5:3]: VIN_SEL_PB2<br>[2:0]: VIS_PIXEL_WIDTH_PB2 |
| 0xD13 | R/W | 0xFF | [7]: VIS_ID_OEN_PB2<br>[6]: VIS_AUTO_EN_PB2<br>[5]: VIS_RPT_EN_PB2<br>[4]: VIS_DET_EN_PB2<br>[3]: VIS_USER_EN_PB2<br>[2]: VIS_CODE_EN_PB2<br>[1]: VIS_RIC_PB2<br>[0]: VIS_ID_EEN_PB2 |
| 0xD14 | R/W | 0x00 | VIS_HOS_PB2[7:0] |
| 0xD15 | R/W | 0x3F | VIS_HIGH_VAL_PB2[7:0] |
| 0xD16 | R/W | 0x05 | VIS_LOW_VAL_PB2[7:0] |
| 0xD17 | R/W | 0x00 | [4:0]: VIS_VOS_PB2 |
| 0xD18 | R/W | 0x00 | [7:0]: VIS_MAN0_PB2_L |
| 0xD19 | R/W | 0x00 | [7:0]: VIS_MAN0_PB2_H |
| 0xD1A | R/W | 0x00 | [7:0]: VIS_MAN1_PB2_L |
| 0xD1B | R/W | 0x00 | [7:0]: VIS_MAN1_PB2_H |
| 0xD1C | R/W | 0x00 | [7:0]: VIS_MAN2_PB2_L |
| 0xD1D | R/W | 0x00 | [7:0]: VIS_MAN2_PB2_H |
| 0xD1E | R/W | 0x00 | [7:0]: VIS_MAN3_PB2_L |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xD1F | R/W | 0x00 | [7:0]: VIS_MAN3_PB2_H |
| 0xD20 | R/W | 0x00 | [7:0]: VIS_MAN4_PB2_L |
| 0xD21 | R/W | 0x00 | [7:0]: VIS_MAN4_PB2_H |
| 0xD22 | R/W | 0x00 | [7:0]: VIS_MAN5_PB2_L |
| 0xD23 | R/W | 0x00 | [7:0]: VIS_MAN5_PB2_H |
| 0xD24 | R/W | 0x04 | [7]: FLDMODE_PB3<br>[6]: INVALID_MD_PB3<br>[5:3]: VIN_SEL_PB3<br>[2:0]: VIS_PIXEL_WIDTH_PB3 |
| 0xD25 | R/W | 0xFF | [7]: VIS_ID_OEN_PB3<br>[6]: VIS_AUTO_EN_PB3<br>[5]: VIS_RPT_EN_PB3<br>[4]: VIS_DET_EN_PB3<br>[3]: VIS_USER_EN_PB3<br>[2]: VIS_CODE_EN_PB3<br>[1]: VIS_RIC_PB3<br>[0]: VIS_ID_EEN_PB3 |
| 0xD26 | R/W | 0x00 | VIS_HOS_PB3[7:0] |
| 0xD27 | R/W | 0x3F | VIS_HIGH_VAL_PB3[7:0] |
| 0xD28 | R/W | 0x05 | VIS_LOW_VAL_PB3[7:0] |
| 0xD29 | R/W | 0x00 | [4:0]: VIS_VOS_PB3 |
| 0xD2A | R/W | 0x00 | [7:0]: VIS_MAN0_PB3_L |
| 0xD2B | R/W | 0x00 | [7:0]: VIS_MAN0_PB3_H |
| 0xD2C | R/W | 0x00 | [7:0]: VIS_MAN1_PB3_L |
| 0xD2D | R/W | 0x00 | [7:0]: VIS_MAN1_PB3_H |
| 0xD2E | R/W | 0x00 | [7:0]: VIS_MAN2_PB3_L |
| 0xD2F | R/W | 0x00 | [7:0]: VIS_MAN2_PB3_H |
| 0xD30 | R/W | 0x00 | [7:0]: VIS_MAN3_PB3_L |
| 0xD31 | R/W | 0x00 | [7:0]: VIS_MAN3_PB3_H |
| 0xD32 | R/W | 0x00 | [7:0]: VIS_MAN4_PB3_L |
| 0xD33 | R/W | 0x00 | [7:0]: VIS_MAN4_PB3_H |
| 0xD34 | R/W | 0x00 | [7:0]: VIS_MAN5_PB3_L |
| 0xD35 | R/W | 0x00 | [7:0]: VIS_MAN5_PB3_H |
| 0xD36 | R/W | 0x04 | [7]: FLDMODE_PB4<br>[6]: INVALID_MD_PB4<br>[5:3]: VIN_SEL_PB4<br>[2:0]: VIS_PIXEL_WIDTH_PB4 |
| 0xD37 | R/W | 0xFF | [7]: VIS_ID_OEN_PB4<br>[6]: VIS_AUTO_EN_PB4<br>[5]: VIS_RPT_EN_PB4<br>[4]: VIS_DET_EN_PB4<br>[3]: VIS_USER_EN_PB4<br>[2]: VIS_CODE_EN_PB4<br>[1]: VIS_RIC_PB4<br>[0]: VIS_ID_EEN_PB4 |
| 0xD38 | R/W | 0x00 | VIS_HOS_PB4[7:0] |
| 0xD39 | R/W | 0x3F | VIS_HIGH_VAL_PB4[7:0] |
| 0xD3A | R/W | 0x05 | VIS_LOW_VAL_PB4[7:0] |
| 0xD3B | R/W | 0x00 | [4:0]: VIS_VOS_PB4 |
| 0xD3C | R/W | 0x00 | [7:0]: VIS_MAN0_PB4_L |
| 0xD3D | R/W | 0x00 | [7:0]: VIS_MAN0_PB4_H |
| 0xD3E | R/W | 0x00 | [7:0]: VIS_MAN1_PB4_L |
| 0xD3F | R/W | 0x00 | [7:0]: VIS_MAN1_PB4_H |
| 0xD40 | R/W | 0x00 | [7:0]: VIS_MAN2_PB4_L |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xD41 | R/W | 0x00 | [7:0]: VIS_MAN2_PB4_H |
| 0xD42 | R/W | 0x00 | [7:0]: VIS_MAN3_PB4_L |
| 0xD43 | R/W | 0x00 | [7:0]: VIS_MAN3_PB4_H |
| 0xD44 | R/W | 0x00 | [7:0]: VIS_MAN4_PB4_L |
| 0xD45 | R/W | 0x00 | [7:0]: VIS_MAN4_PB4_H |
| 0xD46 | R/W | 0x00 | [7:0]: VIS_MAN5_PB4_L |
| 0xD47 | R/W | 0x00 | [7:0]: VIS_MAN5_PB4_H |
| 0xD48 | R/W | 0x07 | [7]: FLDMODE_PB5<br>[6]: INVALID_MD_PB5<br>[5:3]: VIN_SEL_PB5<br>[2:0]: VIS_PIXEL_WIDTH_PB5 |
| 0xD49 | R/W | 0xFF | [7]: VIS_ID_OEN_PB5<br>[6]: VIS_AUTO_EN_PB5<br>[5]: VIS_RPT_EN_PB5<br>[4]: VIS_DET_EN_PB5<br>[3]: VIS_USER_EN_PB5<br>[2]: VIS_CODE_EN_PB5<br>[1]: VIS_RIC_PB5<br>[0]: VIS_ID_EEN_PB5 |
| 0xD4A | R/W | 0x00 | VIS_HOS_PB5[7:0] |
| 0xD4B | R/W | 0x3F | VIS_HIGH_VAL_PB5[7:0] |
| 0xD4C | R/W | 0x05 | VIS_LOW_VAL_PB5[7:0] |
| 0xD4D | R/W | 0x00 | [4:0]: VIS_VOS_PB5 |
| 0xD4E | R/W | 0x00 | [7:0]: VIS_MAN0_PB5_L |
| 0xD4F | R/W | 0x00 | [7:0]: VIS_MAN0_PB5_H |
| 0xD50 | R/W | 0x00 | [7:0]: VIS_MAN1_PB5_L |
| 0xD51 | R/W | 0x00 | [7:0]: VIS_MAN1_PB5_H |
| 0xD52 | R/W | 0x00 | [7:0]: VIS_MAN2_PB5_L |
| 0xD53 | R/W | 0x00 | [7:0]: VIS_MAN2_PB5_H |
| 0xD54 | R/W | 0x00 | [7:0]: VIS_MAN3_PB5_L |
| 0xD55 | R/W | 0x00 | [7:0]: VIS_MAN3_PB5_H |
| 0xD56 | R/W | 0x00 | [7:0]: VIS_MAN4_PB5_L |
| 0xD57 | R/W | 0x00 | [7:0]: VIS_MAN4_PB5_H |
| 0xD58 | R/W | 0x00 | [7:0]: VIS_MAN5_PB5_L |
| 0xD59 | R/W | 0x00 | [7:0]: VIS_MAN5_PB5_H |
| 0xD5A | R/W | 0x07 | [7]: FLDMODE_PB6<br>[6]: INVALID_MD_PB6<br>[5:3]: VIN_SEL_PB6<br>[2:0]: VIS_PIXEL_WIDTH_PB6 |
| 0xD5B | R/W | 0xFF | [7]: VIS_ID_OEN_PB6<br>[6]: VIS_AUTO_EN_PB6<br>[5]: VIS_RPT_EN_PB6<br>[4]: VIS_DET_EN_PB6<br>[3]: VIS_USER_EN_PB6<br>[2]: VIS_CODE_EN_PB6<br>[1]: VIS_RIC_PB6<br>[0]: VIS_ID_EEN_PB6 |
| 0xD5C | R/W | 0x00 | VIS_HOS_PB6[7:0] |
| 0xD5D | R/W | 0x3F | VIS_HIGH_VAL_PB6[7:0] |
| 0xD5E | R/W | 0x05 | VIS_LOW_VAL_PB6[7:0] |
| 0xD5F | R/W | 0x00 | [4:0]: VIS_VOS_PB6 |
| 0xD60 | R/W | 0x00 | [7:0]: VIS_MAN0_PB6_L |
| 0xD61 | R/W | 0x00 | [7:0]: VIS_MAN0_PB6_H |
| 0xD62 | R/W | 0x00 | [7:0]: VIS_MAN1_PB6_L |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xD63 | R/W | 0x00 | [7:0]: VIS_MAN1_PB6_H |
| 0xD64 | R/W | 0x00 | [7:0]: VIS_MAN2_PB6_L |
| 0xD65 | R/W | 0x00 | [7:0]: VIS_MAN2_PB6_H |
| 0xD66 | R/W | 0x00 | [7:0]: VIS_MAN3_PB6_L |
| 0xD67 | R/W | 0x00 | [7:0]: VIS_MAN3_PB6_H |
| 0xD68 | R/W | 0x00 | [7:0]: VIS_MAN4_PB6_L |
| 0xD69 | R/W | 0x00 | [7:0]: VIS_MAN4_PB6_H |
| 0xD6A | R/W | 0x00 | [7:0]: VIS_MAN5_PB6_L |
| 0xD6B | R/W | 0x00 | [7:0]: VIS_MAN5_PB6_H |
| 0xD6C | R/W | 0x07 | [7]: FLDMODE_PB7<br>[6]: INVALID_MD_PB7<br>[5:3]: VIN_SEL_PB7<br>[2:0]: VIS_PIXEL_WIDTH_PB7 |
| 0xD6D | R/W | 0xFF | [7]: VIS_ID_OEN_PB7<br>[6]: VIS_AUTO_EN_PB7<br>[5]: VIS_RPT_EN_PB7<br>[4]: VIS_DET_EN_PB7<br>[3]: VIS_USER_EN_PB7<br>[2]: VIS_CODE_EN_PB7<br>[1]: VIS_RIC_PB7<br>[0]: VIS_ID_EEN_PB7 |
| 0xD6E | R/W | 0x00 | VIS_HOS_PB7[7:0] |
| 0xD6F | R/W | 0x3F | VIS_HIGH_VAL_PB7[7:0] |
| 0xD70 | R/W | 0x05 | VIS_LOW_VAL_PB7[7:0] |
| 0xD71 | R/W | 0x00 | [4:0]: VIS_VOS_PB7 |
| 0xD72 | R/W | 0x00 | [7:0]: VIS_MAN0_PB7_L |
| 0xD73 | R/W | 0x00 | [7:0]: VIS_MAN0_PB7_H |
| 0xD74 | R/W | 0x00 | [7:0]: VIS_MAN1_PB7_L |
| 0xD75 | R/W | 0x00 | [7:0]: VIS_MAN1_PB7_H |
| 0xD76 | R/W | 0x00 | [7:0]: VIS_MAN2_PB7_L |
| 0xD77 | R/W | 0x00 | [7:0]: VIS_MAN2_PB7_H |
| 0xD78 | R/W | 0x00 | [7:0]: VIS_MAN3_PB7_L |
| 0xD79 | R/W | 0x00 | [7:0]: VIS_MAN3_PB7_H |
| 0xD7A | R/W | 0x00 | [7:0]: VIS_MAN4_PB7_L |
| 0xD7B | R/W | 0x00 | [7:0]: VIS_MAN4_PB7_H |
| 0xD7C | R/W | 0x00 | [7:0]: VIS_MAN5_PB7_L |
| 0xD7D | R/W | 0x00 | [7:0]: VIS_MAN5_PB7_H |
| 0xD7E | R/W | 0x07 | [7]: FLDMODE_PB8<br>[6]: INVALID_MD_PB8<br>[5:3]: VIN_SEL_PB8<br>[2:0]: VIS_PIXEL_WIDTH_PB8 |
| 0xD7F | R/W | 0xFF | [7]: VIS_ID_OEN_PB8<br>[6]: VIS_AUTO_EN_PB8<br>[5]: VIS_RPT_EN_PB8<br>[4]: VIS_DET_EN_PB8<br>[3]: VIS_USER_EN_PB8<br>[2]: VIS_CODE_EN_PB8<br>[1]: VIS_RIC_PB8<br>[0]: VIS_ID_EEN_PB8 |
| 0xD80 | R/W | 0x00 | VIS_HOS_PB8[7:0] |
| 0xD81 | R/W | 0x3F | VIS_HIGH_VAL_PB8[7:0] |
| 0xD82 | R/W | 0x05 | VIS_LOW_VAL_PB8[7:0] |
| 0xD83 | R/W | 0x00 | [4:0]: VIS_VOS_PB8 |
| 0xD84 | R/W | 0x00 | [7:0]: VIS_MAN0_PB8_L |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xD85 | R/W | 0x00 | [7:0]: VIS_MAN0_PB8_H |
| 0xD86 | R/W | 0x00 | [7:0]: VIS_MAN1_PB8_L |
| 0xD87 | R/W | 0x00 | [7:0]: VIS_MAN1_PB8_H |
| 0xD88 | R/W | 0x00 | [7:0]: VIS_MAN2_PB8_L |
| 0xD89 | R/W | 0x00 | [7:0]: VIS_MAN2_PB8_H |
| 0xD8A | R/W | 0x00 | [7:0]: VIS_MAN3_PB8_L |
| 0xD8B | R/W | 0x00 | [7:0]: VIS_MAN3_PB8_H |
| 0xD8C | R/W | 0x00 | [7:0]: VIS_MAN4_PB8_L |
| 0xD8D | R/W | 0x00 | [7:0]: VIS_MAN4_PB8_H |
| 0xD8E | R/W | 0x00 | [7:0]: VIS_MAN5_PB8_L |
| 0xD8F | R/W | 0x00 | [7:0]: VIS_MAN5_PB8_H |
| 0xD90 | R/W | 0x00 | [4:0]: VIS_FOS_PB1 |
| 0xD91 | R/W | 0x00 | [4:0]: VIS_FOS_PB2 |
| 0xD92 | R/W | 0x00 | [4:0]: VIS_FOS_PB3 |
| 0xD93 | R/W | 0x00 | [4:0]: VIS_FOS_PB4 |
| 0xD94 | R/W | 0x00 | [4:0]: VIS_FOS_PB5 |
| 0xD95 | R/W | 0x00 | [4:0]: VIS_FOS_PB6 |
| 0xD96 | R/W | 0x00 | [4:0]: VIS_FOS_PB7 |
| 0xD97 | R/W | 0x00 | [4:0]: VIS_FOS_PB8 |
| 0xD98 | R/W | 0x07 | [7]: FLDMODE_PB9<br>[6]: INVALID_MD_PB9<br>[5:3]: VIN_SEL_PB9<br>[2:0]: VIS_PIXEL_WIDTH_PB9 |
| 0xD99 | R/W | 0xFF | [7]: VIS_ID_OEN_PB9<br>[6]: VIS_AUTO_EN_PB9<br>[5]: VIS_RPT_EN_PB9<br>[4]: VIS_DET_EN_PB9<br>[3]: VIS_USER_EN_PB9<br>[2]: VIS_CODE_EN_PB9<br>[1]: VIS_RIC_PB9<br>[0]: VIS_ID_EEN_PB9 |
| 0xD9A | R/W | 0x00 | VIS_HOS_PB9[7:0] |
| 0xD9B | R/W | 0x3F | VIS_HIGH_VAL_PB9[7:0] |
| 0xD9C | R/W | 0x05 | VIS_LOW_VAL_PB9[7:0] |
| 0xD9D | R/W | 0x00 | [4:0]: VIS_VOS_PB9 |
| 0xD9E | R/W | 0x00 | [7:0]: VIS_MAN0_PB9_L |
| 0xD9F | R/W | 0x00 | [7:0]: VIS_MAN0_PB9_H |
| 0xDA0 | R/W | 0x00 | [7:0]: VIS_MAN1_PB9_L |
| 0xDA1 | R/W | 0x00 | [7:0]: VIS_MAN1_PB9_H |
| 0xDA2 | R/W | 0x00 | [7:0]: VIS_MAN2_PB9_L |
| 0xDA3 | R/W | 0x00 | [7:0]: VIS_MAN2_PB9_H |
| 0xDA4 | R/W | 0x00 | [7:0]: VIS_MAN3_PB9_L |
| 0xDA5 | R/W | 0x00 | [7:0]: VIS_MAN3_PB9_H |
| 0xDA6 | R/W | 0x00 | [7:0]: VIS_MAN4_PB9_L |
| 0xDA7 | R/W | 0x00 | [7:0]: VIS_MAN4_PB9_H |
| 0xDA8 | R/W | 0x00 | [7:0]: VIS_MAN5_PB9_L |
| 0xDA9 | R/W | 0x00 | [7:0]: VIS_MAN5_PB9_H |
| 0xDAA | R/W | 0x00 | [4:0]: VIS_FOS_PB9 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xDAB | R/W | 0x00 | [7]: ACTIVE_chid_EN_PB8<br>[6]: ACTIVE_chid_EN_PB7<br>[5]: ACTIVE_chid_EN_PB6<br>[4]: ACTIVE_chid_EN_PB5<br>[3]: ACTIVE_chid_EN_PB4<br>[2]: ACTIVE_chid_EN_PB3<br>[1]: ACTIVE_chid_EN_PB2<br>[0]: ACTIVE_chid_EN_PB1 |
| 0xDAC | R/W | 0x00 | [0]: ACTIVE_chid_EN_PB9 |
| 0xDAD | R/W | 0x00 | [7]: FIRST_FRAME_EN_P8<br>[6]: FIRST_FRAME_EN_P7<br>[5]: FIRST_FRAME_EN_P6<br>[4]: FIRST_FRAME_EN_P5<br>[3]: FIRST_FRAME_EN_P4<br>[2]: FIRST_FRAME_EN_P3<br>[1]: FIRST_FRAME_EN_P2<br>[0]: FIRST_FRAME_EN_P1 |
| 0xDAE | R/W | 0x00 | [0]: FIRST_FRAME_EN_P9 |
| 0xDAF | R/W | 0x01 | [0]: ACTIVE_CHID |
| 0xDB0 | R/W | 0x00 | [7:4]: MOTION_SEL_2<br>[3:0]: MOTION_SEL_1 |
| 0xDB1 | R/W | 0x00 | [7:4]: MOTION_SEL_4<br>[3:0]: MOTION_SEL_3 |
| 0xDB2 | R/W | 0x00 | [7:4]: MOTION_SEL_6<br>[3:0]: MOTION_SEL_5 |
| 0xDB3 | R/W | 0x00 | [7:4]: MOTION_SEL_8<br>[3:0]: MOTION_SEL_7 |
| 0xDB4 | R/W | 0x00 | [3:0]: MOTION_SEL_9 |
| 0xDB5 | R/W | 0x00 | FLD_SEL[7:0] |
| 0xDB6 | R/W | 0x00 | FLD_SEL[8] |

# Registers Description

### PB1 CHANNEL ID ENC MISC REGISTER – 0XD00

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **FLDMODE_PB1:** Encoder field parity<br><br>1: 1 for top field<br>0: 0 for top field |
| 6 | RW | 0 | **INVALID_MD_PB1:** Invalid motion detection<br><br>1: enable. CHID enc generates 0 in V and H blanking.<br>0: disable. CHID enc generates 0x80/0x10 in V and H blanking |
| 5:3 | RW | 0 | **VIN_SEL_PB1:** Record port selection.<br><br>0: port1<br>1: port2<br>2: port3<br>3: port4<br>4: port5<br>5: port6<br>6: port7<br>7: port8 |
| 2:0 | RW | 4 | **VIS_PIXEL_WIDTH_PB1**<br><br>Define analog CHID data sample rate. The value is half period of data period. Unit is 1 pixel. |

### PB1 CHANNEL ID ENC ENABLE REGISTER – 0XD01

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 1 | **VIS_ID_OEN_PB1:** Enable output of D CHID and A CHID in odd field<br><br>1: enable<br>0: disable |
| 6 | RW | 1 | **VIS_AUTO_EN_PB1:** Enable analog auto CHID encoding.<br><br>1: enable<br>0: disable |
| 5 | RW | 1 | **VIS_RPT_EN_PB1:** Enable analog auto repeat CHID encoding.<br><br>1: enable<br>0: disable |
| 4 | RW | 1 | **VIS_DET_EN_PB1:** Enable analog detection CHID encoding.<br><br>1: enable<br>0: disable |
| 3 | RW | 1 | **VIS_USER_EN_PB1:** Enable analog user CHID encoding.<br><br>1: enable<br>0: disable |
| 2 | RW | 1 | **VIS_CODE_EN_PB1:** Enable digital CHID encoding.<br><br>1: enable<br>0: disable |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1 | RW | 1 | **VIS_RIC_PB1:** Enable run in clock for analog CHID<br><br>1: enable<br>0: disable |
| 0 | RW | 1 | **VIS_ID_EEN_PB1:** Enable output of D CHID and A CHID in even field<br><br>1: enable<br>0: disable |

### PB1 CHANNEL ID ENC HORIZONTAL OFFSET REGISTER – 0XD02

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **VIS_HOS_PB1**<br><br>Horizontal offset for analog CHID in unit of pixel. |

### PB1 CHANNEL ID ENC HIGH VALUE REGISTER – 0XD03

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 3f | **VIS_HIGH_VAL_PB1**<br><br>Define high value of analog CHID data bit if it is 1. |

### PB1 CHANNEL ID ENC LOW VALUE REGISTER – 0XD04

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 5 | **VIS_LOW_VAL_PB1**<br><br>Define low value of analog CHID data bit if it is 0. |

### PB1 ODD FIELD VERTICAL LINE OFFSET REGISTER – 0XD05

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | RW | 0 | **VIS_VOS_PB1**<br><br>Vertical offset of channel ID in odd field. |

### PB1 CHANNEL ID ENC MAN0 LSB REGISTER – 0XD06

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **VIS_MAN0_PB1_L**<br><br>User CHID. |

### PB1 CHANNEL ID ENC MAN0 MSB REGISTER – 0XD07

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | VIS_MAN0_PB1_H<br><br>User CHID. |

### PB1 CHANNEL ID ENC MAN1 LSB REGISTER – 0XD08

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | VIS_MAN1_PB1_L<br><br>User CHID. |

### PB1 CHANNEL ID ENC MAN1 SB REGISTER – 0XD09

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | VIS_MAN1_PB1_H<br><br>User CHID. |

### PB1 CHANNEL ID ENC MAN2 LSB REGISTER – 0XD0A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | VIS_MAN2_PB1_L<br><br>User CHID. |

### PB1 CHANNEL ID ENC MAN2 MSB REGISTER – 0XD0B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | VIS_MAN2_PB1_H<br><br>User CHID. |

### PB1 CHANNEL ID ENC MAN3 LSB REGISTER – 0XD0C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | VIS_MAN3_PB1_L<br><br>User CHID. |

### PB1 CHANNEL ID ENC MAN3 MSB REGISTER – 0XD0D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | VIS_MAN3_PB1_H<br><br>User CHID. |

**PB1 CHANNEL ID ENC MAN4 LSB REGISTER – 0XD0E**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | VIS_MAN4_PB1_L<br><br>User CHID. |

**PB1 CHANNEL ID ENC MAN4 MSB REGISTER – 0XD0F**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | VIS_MAN4_PB1_H<br><br>User CHID. |

**PB1 CHANNEL ID ENC MAN5 LSB REGISTER – 0XD10**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | VIS_MAN5_PB1_L<br><br>User CHID. |

**PB1 CHANNEL ID ENC MAN5 MSB REGISTER – 0XD11**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | VIS_MAN5_PB1_H<br><br>User CHID. |

**This is for record unit 1. For record unit 2 – 8, the assignments are: 0xD12 – 0xD8F, for unit 9, 0xD98 – 0xDAA**

**PB1 EVEN FIELD VERTICAL OFFSET REGISTER – 0XD90**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | RW | 0 | VIS_FOS_PB1<br><br>Vertical offset of channel ID in even field. |

**This is for record unit 1. For record unit 2 – 8, the assignments are: 0xD91 – 0xD97**

### ACTIVE AREA CHANNEL ID ENABLE REGISTER 1 – 0XDAB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ACTIVE_chid_EN<br><br>Simple channel ID in active area using “0x01” in first two lines<br>[7]: ACTIVE_chid_EN_PB8<br>[6]: ACTIVE_chid_EN_PB7<br>[5]: ACTIVE_chid_EN_PB6<br>[4]: ACTIVE_chid_EN_PB5<br>[3]: ACTIVE_chid_EN_PB4<br>[2]: ACTIVE_chid_EN_PB3<br>[1]: ACTIVE_chid_EN_PB2<br>[0]: ACTIVE_chid_EN_PB1 |

### ACTIVE AREA CHANNEL ID ENABLE REGISTER 2 – 0XDAC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | RW | 0 | ACTIVE_chid_EN<br><br>Simple channel ID in active area using “0x01” in first two lines<br>[0]: ACTIVE_chid_EN_PB9 |

### ACTIVE AREA CHANNEL ID IN FIRST FRAME REGISTER 1 – 0XDAD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | FIRST_FRAME_EN<br><br>Active Channel ID only in first frame.<br>[7]: FIRST_FRAME_EN_P8<br>[6]: FIRST_FRAME_EN_P7<br>[5]: FIRST_FRAME_EN_P6<br>[4]: FIRST_FRAME_EN_P5<br>[3]: FIRST_FRAME_EN_P4<br>[2]: FIRST_FRAME_EN_P3<br>[1]: FIRST_FRAME_EN_P2<br>[0]: FIRST_FRAME_EN_P1 |

### ACTIVE AREA CHANNEL ID IN FIRST FRAME REGISTER 2 – 0XDAE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | FIRST_FRAME_EN<br><br>Active Channel ID only in first frame.<br>[0]: FIRST_FRAME_EN_P9 |

### ACTIVE AREA CHANNEL ID DATA REGISTER – 0XDAF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x01 | ACTIVE_CHID<br><br>Active Channel ID data. If the video data is same as active channel ID data, the LSB of video data will be inverted. |

**MOTION CHANNEL SELECTION REGISTER – 0XDB0**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0x00 | MOTION_SEL_2<br><br>This register is used for which channel motion information will be shown in 192 bit motion channel ID. It's for read out port 2. |
| 3:0 | RW | 0x00 | MOTION_SEL_1<br><br>This register is used for which channel motion information will be shown in 192 bit motion channel ID. It's for read out port 1. |

**Register 0xDB1 to 0xDB4 are for port 2 to 9**

**FIELD SELECTION REGISTER – 0XDB5**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x00 | FLD_SEL[7:0]<br><br>This register is used for field switch mode. They are for record port 0 to 7.<br><br>0: odd field<br>1: even field |

**FIELD SELECTION REGISTER – 0XDB6**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | RW | 0x00 | FLD_SEL[8]<br><br>This register is used for field switch mode. It is for network port.<br><br>0: odd field<br>1: even field |

# OSD for Recording Path

## Introduction

**TW2880 OSD controller for recording path displays channel information for each channel. There are 9 recording ports supported. It is a font based design. All font tables are stored in local SRAM. Channel information includes 32 characters date/time, 16 characters title and 16 characters status. There are totally 64 fonts saved in SRAM. Font width can be selected from 6, 8, 12 or 16. Font height can be selected from 8, 10, 16 or 20. Channel information is also saved in local SRAM. Besides channel information layer, there are background layer and channel boundary layer.**

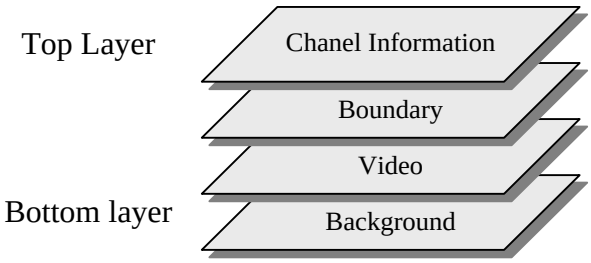
## Features

- 9 OSD engines for 9 recording ports
- 64 fonts table saved in SRAM
- Channel information table saved in SRAM
- Three lines channel information
- 32 characters date/time
- 16 characters channel title for each channel
- 16 characters channel status for each channel
- Font size can be changed
- Channel Boundary
- Background
- Color Bar

## Description

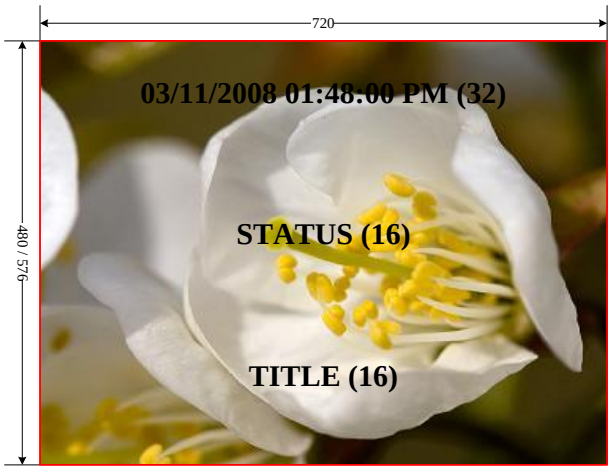
### LAYERS

**There are four layers including video. The bottom layer is background. Background is a pure color under video layer. Background color can be programmed. It is shown up when this channel of video is not enabled. Boundary layer is a boundary for each channel. Boundary width and color can be programmed. The top layer is channel information.**

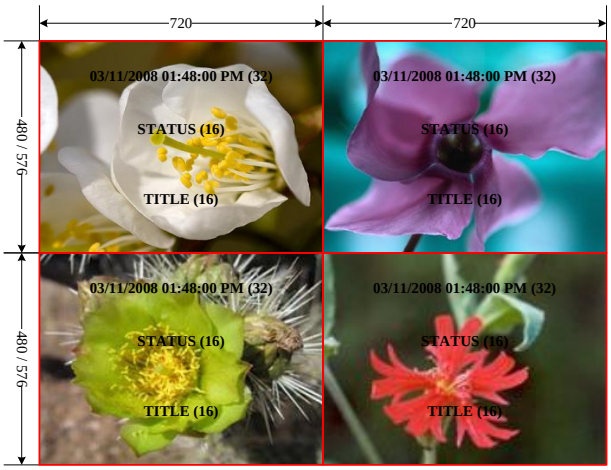



### RESOLUTIONS

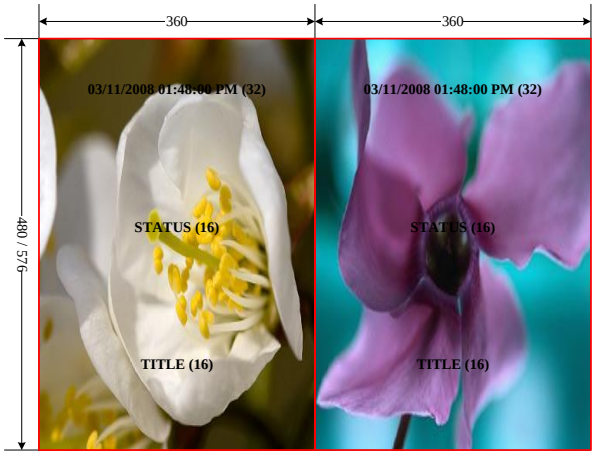
**There are four resolutions supported: D1 (720x480 or 720x576),CIF (352x240 or 352x288), 4D1 (1440x960 or 1440x1152) and BT1120 (1920x1080). In D1 case, it can be full D1 picture, or two half D1 pictures or four quad pictures. In 4D1 case, it has 4 D1 pictures in one frame. In BT1120, it has 6 VGA pictures in one frame. Channel information picture is shown below. For each resolution, font size can be programmed.**



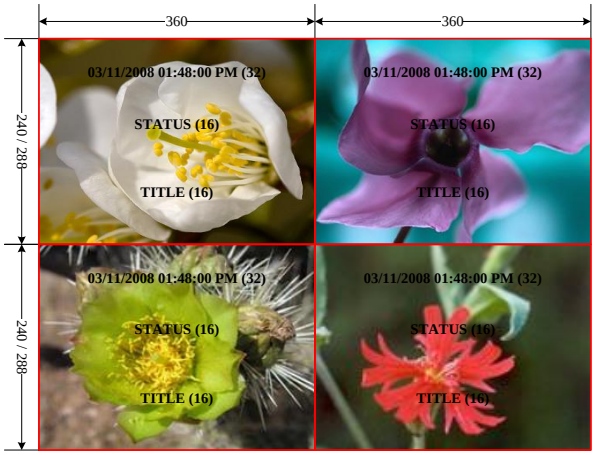

D1



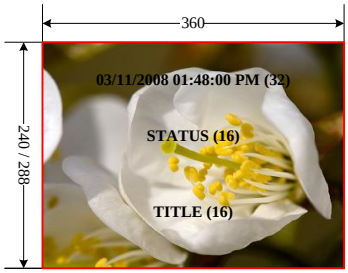

4D1



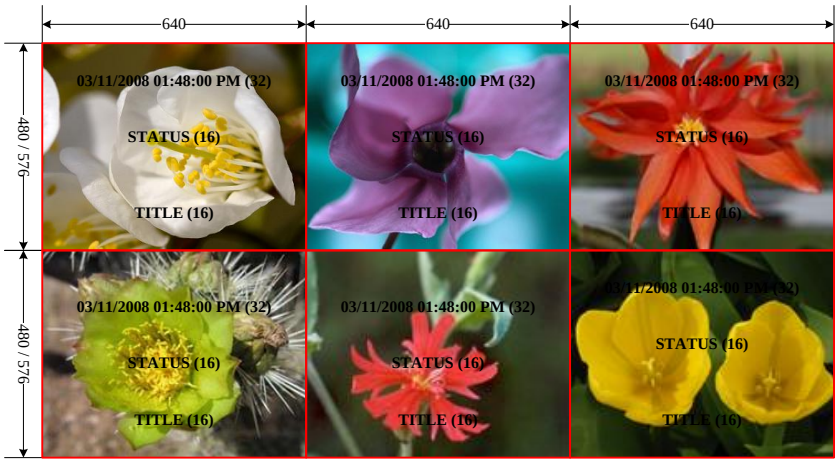

Half D1




Quad



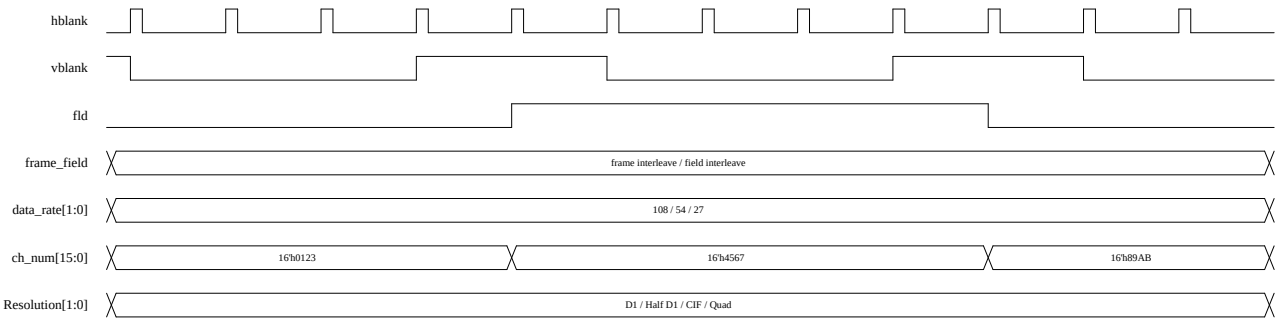

CIF




BT1120

# Timing Diagram

## VERTICAL TIMING



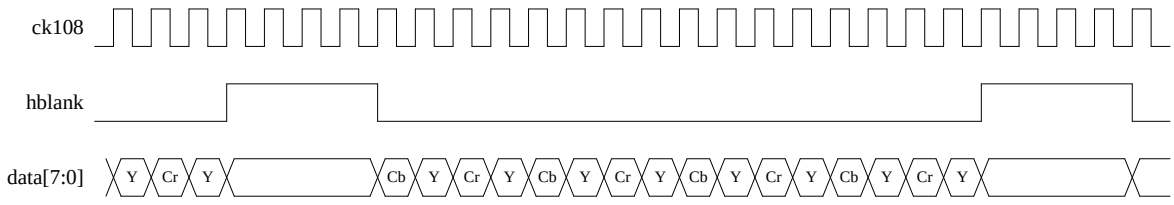

Vertical Timing

**"fld" signal has different means between frame interleave and field interleave. In frame interleave, "fld" is selection for top half frame and bottom half frame. In field interleave, "fld" is selection for odd field and even field.**

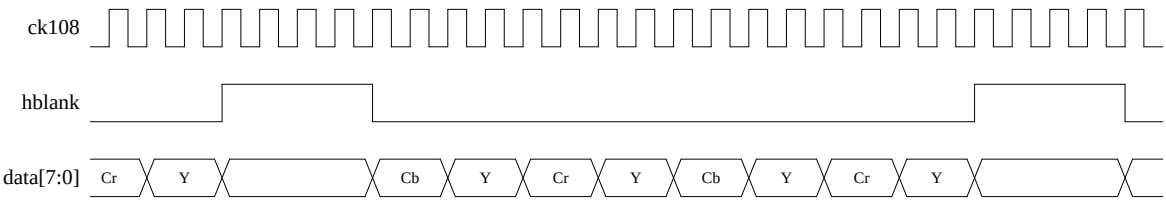
**Channel number is 16 bits. If Quad mode is used, total 16 bits are used. [15:12] is for top left channel, [11:8] is for top right channel, [7:4] is for bottom left channel, and [3:0] is for bottom right channel. If D1 or CIF mode is selected, only bit [15:12] is used. If Half D1 mode is selected, bit [15:8] is used. [15:12] is for left channel and [11:8] is for right channel.**

**Resolution select different size of each channel. D1, Half D1 and Quad have same timing, but the contents in active area are different. In D1 mode, there is only one channel data. In Half D1 mode, there is two channels data. In Quad mode, there is four channels data. In CIF mode, there is only one channel but size is 1/4 of D1.**

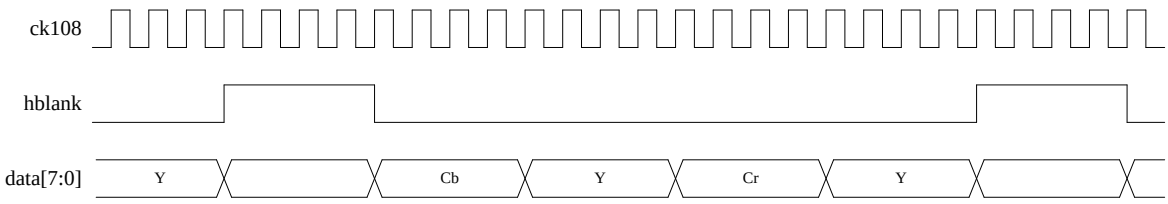
## HORIZONTAL TIMING




Horizontal Timing (108M)




Horizontal Timing (54M)

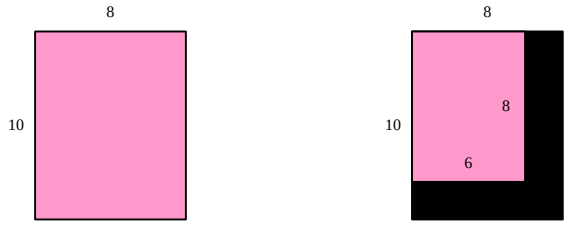



Horizontal Timing (27M)

**Input clock is always 108MHz. But data rate can be select from 108MHz, 54MHz and 27MHz. Input and output data are 8 bits.**

## Font Composition

**There are totally 64 fonts which can be saved in SRAM. The font size saved in SRAM is fixed to 8x10. But displayed font size can be changed. Horizontal can display four sizes: 6, 8, 12 or 16. Size 12 or 16 are doubled from size 6 or 8. If 6 or 12 are selected, fonts saved in SRAM must have small size. But additional two pixels must be saved in SRAM. Vertical can display four sizes: 8, 10, 16 or 20. Size 16 or 20 are doubled from size 8 or 10. If 8 or 16 are selected, fonts saved in SRAM must have small size. But additional two lines must be saved in SRAM. The following picture shows SRAM data. For 8x10 size font, all SRAM data are valid. For 6x8 size font, only 6x8 area are valid. Black area is for dummy data.**




**There are 2 bits for each pixel color. 00 means transparent, color 01, 10 and 11 can be set by registers: ROSD_FONT_R1 (G1, B1), ROSD_FONT_R2 (G2, B2) and ROSD_FONT_R3 (G3, B3).**

**Data saved in SRAM is shown below. There are totally 64x8x10*2=640x16 bits in SRAM. Data are saved font by font. For each font, data is saved line by line. Pixel data in each line is in big endian.**

**Different size font is used for different resolution. For example, 16x20 can be used for D1, 8x10 can be use for CIF and Quad, and 8x20 can be use for Half D1.**

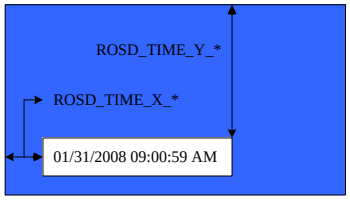
**The font size can be changed according to register ROSD_FONT_HSIZE_* and ROSD_FONT_VSIZE_*. But remember the fonts saved in memory are always 8x10. If double size is selected, just repeat every pixel twice.**

**ROSD_FONT_HSIZE_FULL is for D1 mode. ROSD_FONT_HSIZE_HALF is for Half D1 and Quad and CIF mode. ROSD_FONT_VSIZE_FULL is for D1 and Half D1 mode. ROSD_FONT_VSIZE_HALF is for Quad and CIF mode.**

# Display Channel Information

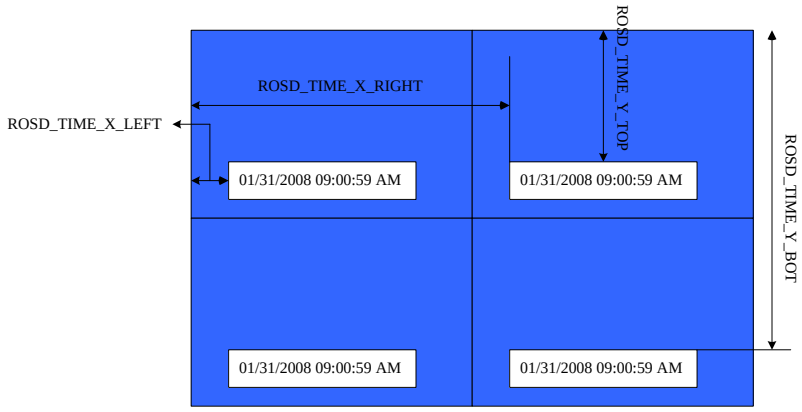
### DATE AND TIME

**There are totally 32 fonts can be displayed including space. Date and time are same for each channel. The position for date and time can be programmed for different resolutions, not for channels. Each font is selected from 64-font table. So index for each font is 6 bit. Time and date can be enabled by setting register ROSD_TIME_EN and ROSD_EN to high. It can be mixed with video data by setting register ROSD_TIME_MIX to high. Mix percentage is 50% video plus 50% channel number. The font index is saved in display SRAM. It needs 32x6 bits.**



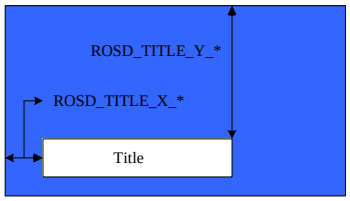

**ROSD_TIME_Y_* defines vertical start position. There are three different vertical positions. ROSD_TIME_Y_FULL is for D1 and Half D1, ROSD_TIME_Y_TOP is for Quad top channels or CIF, and ROSD_TIME_Y_BOTTOM is for Quad bottom channels. All the number is based on (0,0) of whole image.**

**ROSD_TIME_X_* defines horizontal start position. There are three different horizontal positions. ROSD_TIME_X_FULL is for D1, ROSD_TIME_X_LEFT is for Quad left channels or CIF, and ROSD_TIME_X_RIGHT is for Quad right channels. All the number is based on (0,0) of whole image.**




### TITLE

**For each channel, there is 16-font title information. Each font is selected from 64-font table. So index for each font is 6 bit. Title can be enabled by setting register ROSD_TITLE_EN and ROSD_EN to high. Title can be mixed with video data by setting register ROSD_TITLE_MIX to high. Mix percentage is 50% video plus 50% channel number. The position for date and time can be programmed for different resolutions, not for channels.**




**ROSD_TITLE_Y_* defines vertical start position. There are three different vertical positions. ROSD_TITLE_Y_FULL is for D1 and Half D1, ROSD_TITLE_Y_TOP is for Quad top channels or CIF, and ROSD_TITLE_Y_BOTTOM is for Quad bottom channels.**

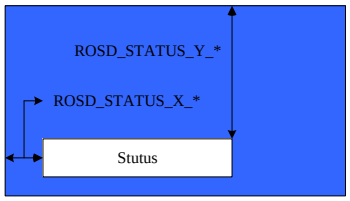
**ROSD_TITLE_X_* defines horizontal start position. There are three different horizontal positions. ROSD_TITLE_X_FULL is for D1, ROSD_TITLE_X_LEFT is for Quad left channels or CIF, and ROSD_TITLE_X_RIGHT is for Quad right channels.**

**If channel number information has less than 16 fonts, you can set the remaining font to space. So you need to put space font in the 64-font table.**

**Channel information for display is saved in display SRAM. It contains 16x6x16=256x6. The sequence is channel by channel. In each channel, the sequence is font by font.**

### STATUS

**Status is same as title. It has 16 fonts. It needs 16x6x16 bit SRAM size.**

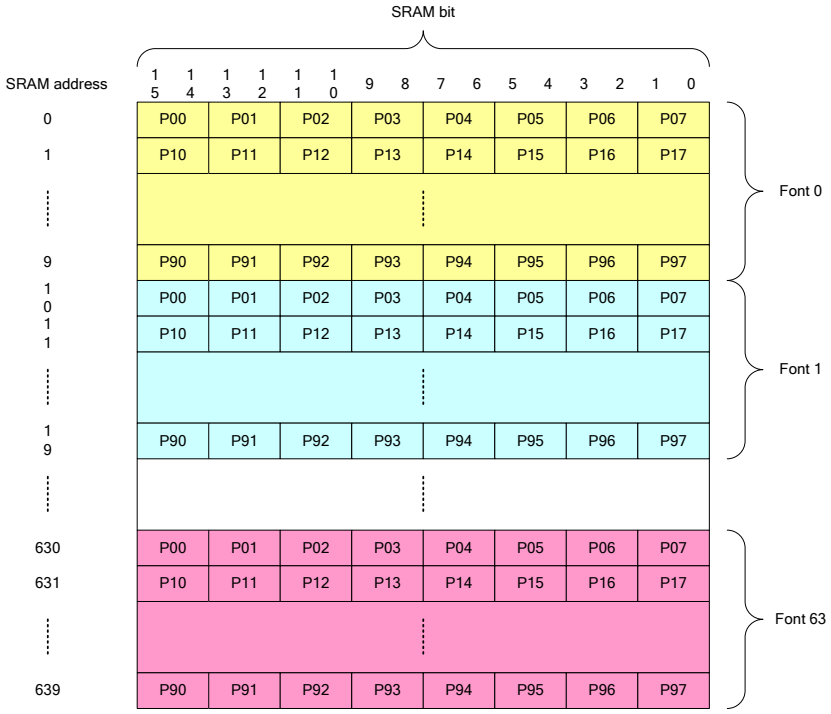



## SRAM Allocation

**There are two SRAM blocks in this design. One is for font. The other is for display information.**

### FONT SRAM

**Fonts are saved in one SRAM block. Font SRAM size is 640x16 bits.**




**During system initialization, host need to write font data in this SRAM. The write sequence is:**

**1. ROSD_FRAM_ADDR[7:0]**

**2. ROSD_FRAM_ADDR[10:0]**

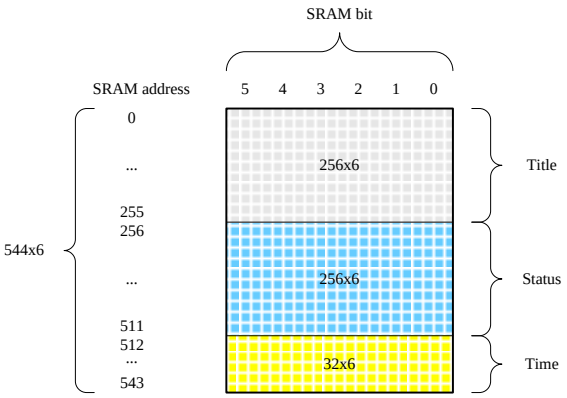
**3. ROSD_FRAM_DATA[7:0]**

**4. ROSD_FRAM_DATA[15:8]**

**ROSD_FRAM_DATA[15:8] must be the last one.**

**DISPLAY SRAM**

**Display SRAM includes time and date, channel title and status. Time and date needs 32x6 bits, channel title needs 256x6 bits, and channel status needs 256x16 bits also. So the total SRAM size is 544x6 bits.**




**During display, host need to write index data in this SRAM. The write sequence is:**

**Set ROSD_DRAM_ADDR, and then set ROSD_DRAM_DATA.**

## Privacy Windows

**This feature is used to provide privacies for the monitored objects. In many occasions we need to provide options to block certain private information from the monitoring people. Take monitoring banking ATMs in the shopping mall for example, although we want to monitor the overall operations and the activities for this object to prevent any malpractice, we want to protect the customers PIDs to be recorded and made public later on. To achieve this, system providers can use the privacy windows offered by TW2880 to block these kind of hot spots and make sure no one else get access to customers' code.**

**There are actually three sets of independent privacy window in the TW2880. One for live video (dual monitor is controlled by the same set), one for SPOT and one for record. To simplify the control of these three sets, we have shadowed the three sets under the control of register 0xE4F bit 1 and 0. 0x00 will control the reading and writing of the recording privacy windows, 0x01 will control the SPOT window set and 0x10 will control live window set. To read / write the respective register sets you need to put proper values to these two bits first.**

**INTRODUCTION**

**There are four windows in each channel for each set so altogether there are 192 windows to play with. To use these windows, user needs to program the horizontal start, vertical start, horizontal size, vertical size and effect that you desired and enable the window, you will see portion of the image is affected by the window.**

**The unit for horizontal start is double pixel, with the 8 bit provided you can cover the whole horizontal span. Same thing for the vertical except the unit is double line and the control bit is 7 bit. For the window size TW2880 provides 8 horizontal selections and four vertical selections for the users.**

**MASKING EFFECT**

**8 masking effects are provided including mosaic. Except the two solid color masks which users can not see anything under the mask, the rest effects are provided to let monitoring people see the some of the actions but cannot see the details of the operation hence the privacy protection. The "over exposed" and "under exposed"**

masks are generated by playing the luminance values of the pixels that are under the window boundaries. The “over exposed” mask is generated by adding a pre-defined value to every pixel’s luminance value within the window and go through a limiter. The “under exposed” mask is generated by dividing a fixed value to every pixels under the window. The values and the factors are selectable.

### MOSAIC EFFECT

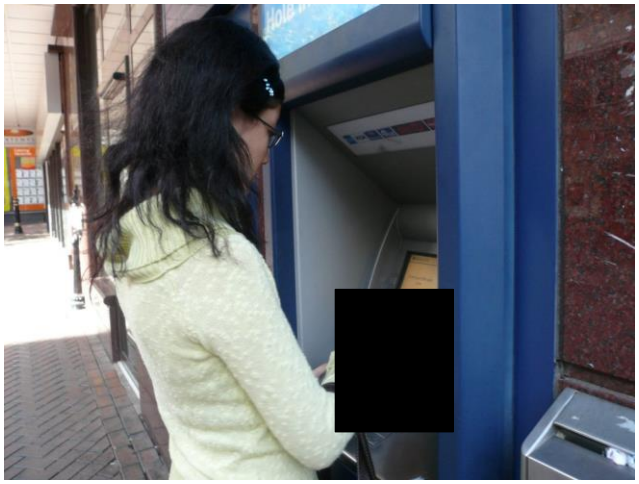
4 Mosaic sizes are provided, they are 8x8, 16x16, 32x32 and 64x64. Take 16x16 size for example, this will divide a NTSC field into a 45x15 cell. (45x18 for PAL). If mosaic effect is turn on, every pixel inside the cell will have the same luminance value. Visually, this will create a tiled cell array such that any detailed information underneath the window is impossible to see through. Smaller sizes of mosaic tile is used to block detailed movement but let the observer see the general motion.



Original Video Image



White Mask Image



Black Mask Image



Mosaic Mask Image

## Registers Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xE00 | R/W | 0 | [7]: ROSD_EN8<br>[6]: ROSD_EN7<br>[5]: ROSD_EN6<br>[4]: ROSD_EN5<br>[3]: ROSD_EN4<br>[2]: ROSD_EN3<br>[1]: ROSD_EN2<br>[0]: ROSD_EN1 |
| 0xE01 | R/W | 0 | [7]: ROSD_TIME_EN8<br>[6]: ROSD_TIME_EN7<br>[5]: ROSD_TIME_EN6<br>[4]: ROSD_TIME_EN5<br>[3]: ROSD_TIME_EN4<br>[2]: ROSD_TIME_EN3<br>[1]: ROSD_TIME_EN2<br>[0]: ROSD_TIME_EN1 |
| 0xE02 | R/W | 0 | [7]: ROSD_TITLE_EN8<br>[6]: ROSD_TITLE_EN7<br>[5]: ROSD_TITLE_EN6<br>[4]: ROSD_TITLE_EN5<br>[3]: ROSD_TITLE_EN4<br>[2]: ROSD_TITLE_EN3<br>[1]: ROSD_TITLE_EN2<br>[0]: ROSD_TITLE_EN1 |
| 0xE03 | R/W | 0 | [7]: ROSD_STATUS_EN8<br>[6]: ROSD_STATUS_EN7<br>[5]: ROSD_STATUS_EN6<br>[4]: ROSD_STATUS_EN5<br>[3]: ROSD_STATUS_EN4<br>[2]: ROSD_STATUS_EN3<br>[1]: ROSD_STATUS_EN2<br>[0]: ROSD_STATUS_EN1 |
| 0xE04 | R/W | 0 | [7]: ROSD_BNDRY_EN8<br>[6]: ROSD_BNDRY_EN7<br>[5]: ROSD_BNDRY_EN6<br>[4]: ROSD_BNDRY_EN5<br>[3]: ROSD_BNDRY_EN4<br>[2]: ROSD_BNDRY_EN3<br>[1]: ROSD_BNDRY_EN2<br>[0]: ROSD_BNDRY_EN1 |
| 0xE05 | R/W | 0 | [7]: ROSD_BGND_EN8<br>[6]: ROSD_BGND_EN7<br>[5]: ROSD_BGND_EN6<br>[4]: ROSD_BGND_EN5<br>[3]: ROSD_BGND_EN4<br>[2]: ROSD_BGND_EN3<br>[1]: ROSD_BGND_EN2<br>[0]: ROSD_BGND_EN1 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xE06 | R/W | 0 | [7]: ROSD_MIX_8<br>[6]: ROSD_MIX_7<br>[5]: ROSD_MIX_6<br>[4]: ROSD_MIX_5<br>[3]: ROSD_MIX_4<br>[2]: ROSD_MIX_3<br>[1]: ROSD_MIX_2<br>[0]: ROSD_MIX_1 |
| 0xE07 | R/W | 0x2 | [3:0]: ROSD_BNDRY_WIDTH[3:0] |
| 0xE08 | R/W | 0x77 | [7:6]: ROSD_FONT_VSIZE_HALF<br>[5:4]: ROSD_FONT_VSIZE_FULL<br>[3:2]: ROSD_FONT_HSIZE_HALF<br>[1:0]: ROSD_FONT_HSIZE_FULL |
| 0xE09 | R/W | 0 | ROSD_TIME_X_FULL[7:0] |
| 0xE0A | R/W | 0 | ROSD_TIME_X_FULL[9:8] |
| 0xE0B | R/W | 0 | ROSD_TIME_X_LEFT[7:0] |
| 0xE0C | R/W | 0 | ROSD_TIME_X_LEFT[9:8] |
| 0xE0D | R/W | 0 | ROSD_TIME_X_RIGHT[7:0] |
| 0xE0E | R/W | 0 | ROSD_TIME_X_RIGHT[9:8] |
| 0xE0F | R/W | 0 | ROSD_TIME_Y_FULL[7:0] |
| 0xE10 | R/W | 0 | ROSD_TIME_Y_FULL[9:8] |
| 0xE11 | R/W | 0 | ROSD_TIME_Y_TOP[7:0] |
| 0xE12 | R/W | 0 | ROSD_TIME_Y_TOP[9:8] |
| 0xE13 | R/W | 0 | ROSD_TIME_Y_BOTTOM[7:0] |
| 0xE14 | R/W | 0 | ROSD_TIME_Y_BOTTOM[9:8] |
| 0xE15 | R/W | 0 | ROSD_TITLE_X_FULL[7:0] |
| 0xE16 | R/W | 0 | ROSD_TITLE_X_FULL[9:8] |
| 0xE17 | R/W | 0 | ROSD_TITLE_X_LEFT[7:0] |
| 0xE18 | R/W | 0 | ROSD_TITLE_X_LEFT[9:8] |
| 0xE19 | R/W | 0 | ROSD_TITLE_X_RIGHT[7:0] |
| 0xE1A | R/W | 0 | ROSD_TITLE_X_RIGHT[9:8] |
| 0xE1B | R/W | 0 | ROSD_TITLE_Y_FULL[7:0] |
| 0xE1C | R/W | 0 | ROSD_TITLE_Y_FULL[9:8] |
| 0xE1D | R/W | 0 | ROSD_TITLE_Y_TOP[7:0] |
| 0xE1E | R/W | 0 | ROSD_TITLE_Y_TOP[9:8] |
| 0xE1F | R/W | 0 | ROSD_TITLE_Y_BOTTOM[7:0] |
| 0xE20 | R/W | 0 | ROSD_TITLE_Y_BOTTOM[9:8] |
| 0xE21 | R/W | 0 | ROSD_STATUS_X_FULL[7:0] |
| 0xE22 | R/W | 0 | ROSD_STATUS_X_FULL[9:8] |
| 0xE23 | R/W | 0 | ROSD_STATUS_X_LEFT[7:0] |
| 0xE24 | R/W | 0 | ROSD_STATUS_X_LEFT[9:8] |
| 0xE25 | R/W | 0 | ROSD_STATUS_X_RIGHT[7:0] |
| 0xE26 | R/W | 0 | ROSD_STATUS_X_RIGHT[9:8] |
| 0xE27 | R/W | 0 | ROSD_STATUS_Y_FULL[7:0] |
| 0xE28 | R/W | 0 | ROSD_STATUS_Y_FULL[9:8] |
| 0xE29 | R/W | 0 | ROSD_STATUS_Y_TOP[7:0] |
| 0xE2A | R/W | 0 | ROSD_STATUS_Y_TOP[9:8] |
| 0xE2B | R/W | 0 | ROSD_STATUS_Y_BOTTOM[7:0] |
| 0xE2C | R/W | 0 | ROSD_STATUS_Y_BOTTOM[9:8] |
| 0xE2D | R/W | 0x10 | ROSD_FONT_Y1[7:0] |
| 0xE2E | R/W | 0x80 | ROSD_FONT_CB1[7:0] |
| 0xE2F | R/W | 0x80 | ROSD_FONT_CR1[7:0] |
| 0xE30 | R/W | 0x10 | ROSD_FONT_Y2[7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xE31 | R/W | 0x80 | ROSD_FONT_CB2[7:0] |
| 0xE32 | R/W | 0x80 | ROSD_FONT_CR2[7:0] |
| 0xE33 | R/W | 0x10 | ROSD_FONT_Y3[7:0] |
| 0xE34 | R/W | 0x80 | ROSD_FONT_CB3[7:0] |
| 0xE35 | R/W | 0x80 | ROSD_FONT_CR3[7:0] |
| 0xE36 | R/W | 0x10 | ROSD_BGND_Y[7:0] |
| 0xE37 | R/W | 0x80 | ROSD_BGND_CB[7:0] |
| 0xE38 | R/W | 0x80 | ROSD_BGND_CR[7:0] |
| 0xE39 | R/W | 0x10 | ROSD_BNDRY_Y[7:0] |
| 0xE3A | R/W | 0x80 | ROSD_BNDRY_CB[7:0] |
| 0xE3B | R/W | 0x80 | ROSD_BNDRY_CR[7:0] |
| 0xE3C | W | 0 | ROSD_FRAM_ADDR[7:0] |
| 0xE3D | W | 0 | ROSD_FRAM_ADDR[9:8] |
| 0xE3E | W | 0 | ROSD_FRAM_DATA[7:0] |
| 0xE3F | W | 0 | ROSD_FRAM_DATA[15:8] |
| 0xE40 | W | 0 | ROSD_DRAM_ADDR[7:0] |
| 0xE41 | W | 0 | ROSD_DRAM_ADDR[9:8] |
| 0xE42 | W | 0 | ROSD_DRAM_DATA[5:0] |
| 0xE43 | R/W | 0 | [7]: ROSD_COLBAR_EN8<br>[6]: ROSD_COLBAR_EN7<br>[5]: ROSD_COLBAR_EN6<br>[4]: ROSD_COLBAR_EN 5<br>[3]: ROSD_COLBAR_EN4<br>[2]: ROSD_COLBAR_EN3<br>[1]: ROSD_COLBAR_EN2<br>[0]: ROSD_COLBAR_EN1 |
| 0xE44 | R/W | 0 | [7]: ROSD_COLBAR_EN9<br>[6]: ROSD_MIX_9<br>[5]: ROSD_BGND_EN9<br>[4]: ROSD_BNDRY_EN9<br>[3]: ROSD_STATUS_EN9<br>[2]: ROSD_TITLE_EN9<br>[1]: ROSD_TIME_EN9<br>[0]: ROSD_EN9 |
| 0xE45 | R/W | 0 | ROSD_COLBAR_VPOS[7:0] |
| 0xE46 | R/W | 0 | ROSD_FSWITCH[7:0] |
| 0xE47 | R/W | 0 | Privacy Window Overflow Register 1 for window 1 |
| 0xE48 | R/W | 0 | Privacy Window Overflow Register 2 for window 1 |
| 0xE49 | R/W | 0 | Privacy Window Overflow Register 1 for window 2 |
| 0xE4A | R/W | 0 | Privacy Window Overflow Register 2 for window 2 |
| 0xE4B | R/W | 0 | Privacy Window Overflow Register 1 for window 3 |
| 0xE4C | R/W | 0 | Privacy Window Overflow Register 2 for window 3 |
| 0xE4D | R/W | 0 | Privacy Window Overflow Register 1 for window 4 |
| 0xE4E | R/W | 0 | Privacy Window Overflow Register 2 for window 4 |
| 0xE4F | R/W | 0 | Privacy Window Shadow Set Control Register |
| 0xE50-0xE5F | R/W | 0 | Privacy Window 1 Horizontal Start Register |
| 0xE60-0xE6F | R/W | 0 | Privacy Window 1 Vertical Start Register |
| 0xE70-0xE7F | R/W | 0 | Privacy Window 1 Control Register |
| 0xE80-0xE8F | R/W | 0 | Privacy Window 2 Horizontal Start Register |
| 0xE90-0xE9F | R/W | 0 | Privacy Window 2 Vertical Start Register |
| 0xEA0-0xEAF | R/W | 0 | Privacy Window 2 Control Register |
| 0xEB0-0xEBF | R/W | 0 | Privacy Window 3 Horizontal Start Register |
| 0xEC0-0xECF | R/W | 0 | Privacy Window 3 Vertical Start Register |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xED0-0xEDF | R/W | 0 | Privacy Window 3 Control Register |
| 0xDD0-0xDDF | R/W | 0 | Privacy Window 4 Horizontal Start Register |
| 0xDE0-0xDEF | R/W | 0 | Privacy Window 4 Vertical Start Register |
| 0xDF0-0xDFF | R/W | 0 | Privacy Window 4 Control Register |

# Registers Description

### ROSD ENABLE REGISTER – 0XE00

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_EN<br><br>[7]: ROSD_EN8<br>[6]: ROSD_EN7<br>[5]: ROSD_EN6<br>[4]: ROSD_EN5<br>[3]: ROSD_EN4<br>[2]: ROSD_EN3<br>[1]: ROSD_EN2<br>[0]: ROSD_EN1<br><br>Recording port n (1 to 8) OSD enable or disable. If this register is set to 1, OSD is enabled. But you need to set ROSD_TIME_EN or ROSD_TITLE_EN or ROSD_STATUS_EN if needed. If ROSD_EN is set to 0, all ROSD display is disabled. |

### ROSD TIME AND DATE ENABLE REGISTER – 0XE01

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TIME_EN<br><br>[7]: ROSD_TIME_EN8<br>[6]: ROSD_TIME_EN7<br>[5]: ROSD_TIME_EN6<br>[4]: ROSD_TIME_EN5<br>[3]: ROSD_TIME_EN4<br>[2]: ROSD_TIME_EN3<br>[1]: ROSD_TIME_EN2<br>[0]: ROSD_TIME_EN1<br><br>Recording port n (1 to 8) time and date enable or disable. To enable time and date, you need to set ROSD_EN and ROSD_TIME_EN. |

### ROSD TITLE ENABLE REGISTER – 0XE02

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TITLE_EN<br><br>[7]: ROSD_TITLE_EN8<br>[6]: ROSD_TITLE_EN7<br>[5]: ROSD_TITLE_EN6<br>[4]: ROSD_TITLE_EN5<br>[3]: ROSD_TITLE_EN4<br>[2]: ROSD_TITLE_EN3<br>[1]: ROSD_TITLE_EN2<br>[0]: ROSD_TITLE_EN1<br><br>Recording port n (1 to 8) title enable or disable. To enable title, you need to set ROSD_EN and ROSD_TITLE_EN. |

### ROSD STATUS ENABLE REGISTER – 0XE03

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_STATUS_EN<br><br>[7]: ROSD_STATUS_EN8<br>[6]: ROSD_STATUS_EN7<br>[5]: ROSD_STATUS_EN6<br>[4]: ROSD_STATUS_EN5<br>[3]: ROSD_STATUS_EN4<br>[2]: ROSD_STATUS_EN3<br>[1]: ROSD_STATUS_EN2<br>[0]: ROSD_STATUS_EN1<br><br>Recording port n (1 to 8) status enable or disable. To enable status, you need to set ROSD_EN and ROSD_STATUS_EN. |

### ROSD BOUNDARY ENABLE REGISTER – 0XE04

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_BNDRY_EN<br><br>[7]: ROSD_BNDRY_EN8<br>[6]: ROSD_BNDRY_EN7<br>[5]: ROSD_BNDRY_EN6<br>[4]: ROSD_BNDRY_EN5<br>[3]: ROSD_BNDRY_EN4<br>[2]: ROSD_BNDRY_EN3<br>[1]: ROSD_BNDRY_EN2<br>[0]: ROSD_BNDRY_EN1<br><br>Recording port n (1 to 8) boundary enable or disable. To enable boundary, you need to set ROSD_EN and ROSD_BNDRY_EN. |

### ROSD BACKGROUND ENABLE REGISTER – 0XE05

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_BGND_EN<br><br>[7]: ROSD_BGND_EN8<br>[6]: ROSD_BGND_EN7<br>[5]: ROSD_BGND_EN6<br>[4]: ROSD_BGND_EN5<br>[3]: ROSD_BGND_EN4<br>[2]: ROSD_BGND_EN3<br>[1]: ROSD_BGND_EN2<br>[0]: ROSD_BGND_EN1<br><br>Recording port n (1 to 8) back ground enable or disable. To enable back ground, you need to set ROSD_EN and ROSD_BGND_EN. |

### ROSD MIX CONTROL ENABLE REGISTER – 0XE06

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_MIX<br><br>[7]: ROSD_MIX_8<br>[6]: ROSD_MIX_7<br>[5]: ROSD_MIX_6<br>[4]: ROSD_MIX_5<br>[3]: ROSD_MIX_4<br>[2]: ROSD_MIX_3<br>[1]: ROSD_MIX_2<br>[0]: ROSD_MIX_1<br><br>Display font mix enable bit. If set to 1, font data will be 50% blending with video<br>1: enable<br>0: disable |

### ROSD BOUNDARY WIDTH REGISTER – 0XE07

| Bit | R/W | Default | Description |
|---|---|---|---|
| 3:0 | RW | 2 | ROSD_BNDRY_WIDTH<br><br>Boundary width from 0 to 15. Unit is 2 pixels. Only Quad has boundary. |

### ROSD FONT SIZE REGISTER – 0XE08

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 1 | ROSD_FONT_VSIZE_HALF<br><br>Font vertical size for Quad and CIF mode. There are four choices.<br>00: 8<br>01: 10<br>10: 16, scale up from 8<br>11: 20, scale up from 10 |
| 5:4 | RW | 3 | ROSD_FONT_VSIZE_FULL<br><br>Font vertical size for D1 and half D1 mode. There are four choices.<br>00: 8<br>01: 10<br>10: 16, scale up from 8<br>11: 20, scale up from 10 |
| 3:2 | RW | 1 | ROSD_FONT_HSIZE_HALF<br><br>Font horizontal size for Quad, half D1 and CIF mode. There are four choices.<br>00: 6<br>01: 8<br>10: 12, scale up from 6<br>11: 16, scale up from 8 |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 1 | ROSD_FONT_HSIZE_FULL<br><br>Font horizontal size for D1 mode. There are four choices.<br>00: 6<br>01: 8<br>10: 12, scale up from 6<br>11: 16, scale up from 8 |

### ROSD TIME HORIZONTAL POSTION FOR FULL MODE LOW BYTE REGISTER – 0XE09

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TIME_X_FULL[7:0]<br><br>Time and Date horizontal position for D1 mode. Unit is 1 pixel |

### ROSD TIME HORIZONTAL POSTION FOR FULL MODE HIGH BYTE REGISTER – 0XE0A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_TIME_X_FULL[9:8]<br><br>Time and Date horizontal position for D1 mode. Unit is 1 pixel |

### ROSD TIME HORIZONTAL LEFT POSTION FOR HALF MODE LOW BYTE REGISTER – 0XE0B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TIME_X_LEFT[7:0]<br><br>Time and Date horizontal left position for half D1, Quad and CIF mode. Unit is 1 pixel |

### ROSD TIME HORIZONTAL LEFT POSTION FOR HALF MODE HIGH BYTE REGISTER – 0XE0C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_TIME_X_LEFT[9:8]<br><br>Time and Date horizontal left position for half D1, Quad and CIF mode. Unit is 1 pixel |

### ROSD TIME HORIZONTAL RIGHT POSTION FOR HALF MODE LOW BYTE REGISTER – 0XE0D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TIME_X_RIGHT[7:0]<br><br>Time and Date horizontal right position for half D1, Quad and CIF mode. Unit is 1 pixel |

### ROSD TIME HORIZONTAL RIGHT POSTION FOR HALF MODE HIGH BYTE REGISTER – 0XE0E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_TIME_X_RIGHT[9:8]<br><br>Time and Date horizontal right position for half D1, Quad and CIF mode. Unit is 1 pixel |

### ROSD TIME VERTICAL POSTION FOR FULL MODE LOW BYTE REGISTER – 0XE0F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TIME_Y_FULL[7:0]<br><br>Time and Date vertical position for D1 and half D1 mode. Unit is 1 pixel |

### ROSD TIME VERTICAL POSTION FOR FULL MODE HIGH BYTE REGISTER – 0XE10

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_TIME_Y_FULL[9:8]<br><br>Time and Date vertical position for D1 and half D1 mode. Unit is 1 pixel |

### ROSD TIME VERTICAL TOP POSTION FOR HALF MODE LOW BYTE REGISTER – 0XE11

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TIME_Y_TOP[7:0]<br><br>Time and Date vertical top position for Quad and CIF mode. Unit is 1 pixel |

### ROSD TIME VERTICAL TOP POSTION FOR HALF MODE HIGH BYTE REGISTER – 0XE12

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_TIME_Y_TOP[9:8]<br><br>Time and Date vertical top position for Quad and CIF mode. Unit is 1 pixel |

### ROSD TIME VERTICAL BOTTOM POSTION FOR HALF MODE LOW BYTE REGISTER – 0XE13

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TIME_Y_BOTTOM[7:0]<br><br>Time and Date vertical bottom position for Quad and CIF mode. Unit is 1 pixel |

### ROSD TIME VERTICAL BOTTOM POSTION FOR HALF MODE HIGH BYTE REGISTER – 0XE14

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_TIME_Y_BOTTOM[9:8]<br><br>Time and Date vertical bottom position for Quad and CIF mode. Unit is 1 pixel |

### ROSD TITLE HORIZONTAL POSTION FOR FULL MODE LOW BYTE REGISTER – 0XE15

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TITLE_X_FULL[7:0]<br><br>Title horizontal position for D1 mode. Unit is 1 pixel |

### ROSD TITLE HORIZONTAL POSTION FOR FULL MODE HIGH BYTE REGISTER – 0XE16

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_TITLE_X_FULL[9:8]<br><br>Title horizontal position for D1 mode. Unit is 1 pixel |

### ROSD TITLE HORIZONTAL LEFT POSTION FOR HALF MODE LOW BYTE REGISTER – 0XE17

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TITLE_X_LEFT[7:0]<br><br>Title horizontal left position for half D1, Quad and CIF mode. Unit is 1 pixel |

### ROSD TITLE HORIZONTAL LEFT POSTION FOR HALF MODE HIGH BYTE REGISTER – 0XE18

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_TITLE_X_LEFT[9:8]<br><br>Title horizontal left position for half D1, Quad and CIF mode. Unit is 1 pixel |

### ROSD TITLE HORIZONTAL RIGHT POSTION FOR HALF MODE LOW BYTE REGISTER – 0XE19

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TITLE_X_RIGHT[7:0]<br><br>Title horizontal right position for half D1, Quad and CIF mode. Unit is 1 pixel |

### ROSD TITLE HORIZONTAL RIGHT POSTION FOR HALF MODE HIGH BYTE REGISTER – 0XE1A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_TITLE_X_RIGHT[9:8]<br><br>Title horizontal right position for half D1, Quad and CIF mode. Unit is 1 pixel |

### ROSD TITLE VERTICAL POSTION FOR FULL MODE LOW BYTE REGISTER – 0XE1B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TITLE_Y_FULL[7:0]<br><br>Title vertical position for D1 and half D1 mode. Unit is 1 pixel |

### ROSD TITLE VERTICAL POSTION FOR FULL MODE HIGH BYTE REGISTER – 0XE1C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_TITLE_Y_FULL[9:8]<br><br>Title vertical position for D1 and half D1 mode. Unit is 1 pixel |

### ROSD TITLE VERTICAL TOP POSTION FOR HALF MODE LOW BYTE REGISTER – 0XE1D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TITLE_Y_TOP[7:0]<br><br>Title vertical top position for Quad and CIF mode. Unit is 1 pixel |

### ROSD TITLE VERTICAL TOP POSTION FOR HALF MODE HIGH BYTE REGISTER – 0XE1E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_TITLE_Y_TOP[9:8]<br><br>Title vertical top position for Quad and CIF mode. Unit is 1 pixel |

### ROSD TITLE VERTICAL BOTTOM POSTION FOR HALF MODE LOW BYTE REGISTER – 0XE1F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_TITLE_Y_BOTTOM[7:0]<br><br>Title vertical bottom position for Quad and CIF mode. Unit is 1 pixel |

### ROSD TITLE VERTICAL BOTTOM POSTION FOR HALF MODE HIGH BYTE REGISTER – 0XE20

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_TITLE_Y_BOTTOM[9:8]<br><br>Title vertical bottom position for Quad and CIF mode. Unit is 1 pixel |

### ROSD STATUS HORIZONTAL POSTION FOR FULL MODE LOW BYTE REGISTER – 0XE21

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_STATUS_X_FULL[7:0]<br><br>Status horizontal position for D1 mode. Unit is 1 pixel |

### ROSD STATUS HORIZONTAL POSTION FOR FULL MODE HIGH BYTE REGISTER – 0XE22

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_STATUS_X_FULL[9:8]<br><br>Status horizontal position for D1 mode. Unit is 1 pixel |

### ROSD STATUS HORIZONTAL LEFT POSTION FOR HALF MODE LOW BYTE REGISTER – 0XE23

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_STATUS_X_LEFT[7:0]<br><br>Status horizontal left position for half D1, Quad and CIF mode. Unit is 1 pixel |

### ROSD STATUS HORIZONTAL LEFT POSTION FOR HALF MODE HIGH BYTE REGISTER – 0XE24

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_STATUS_X_LEFT[9:8]<br><br>Status horizontal left position for half D1, Quad and CIF mode. Unit is 1 pixel |

### ROSD STATUS HORIZONTAL RIGHT POSTION FOR HALF MODE LOW BYTE REGISTER – 0XE25

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_STATUS_X_RIGHT[7:0]<br><br>Status horizontal right position for half D1, Quad and CIF mode. Unit is 1 pixel |

### ROSD STATUS HORIZONTAL RIGHT POSTION FOR HALF MODE HIGH BYTE REGISTER – 0XE26

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_STATUS_X_RIGHT[9:8]<br><br>Status horizontal right position for half D1, Quad and CIF mode. Unit is 1 pixel |

### ROSD STATUS VERTICAL POSTION FOR FULL MODE LOW BYTE REGISTER – 0XE27

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_STATUS_Y_FULL[7:0]<br><br>Status vertical position for D1 and half D1 mode. Unit is 1 pixel |

### ROSD STATUS VERTICAL POSTION FOR FULL MODE HIGH BYTE REGISTER – 0XE28

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_STATUS_Y_FULL[9:8]<br><br>Status vertical position for D1 and half D1 mode. Unit is 1 pixel |

### ROSD STATUS VERTICAL TOP POSTION FOR HALF MODE LOW BYTE REGISTER – 0XE29

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_STATUS_Y_TOP[7:0]<br><br>Status vertical top position for Quad and CIF mode. Unit is 1 pixel |

### ROSD STATUS VERTICAL TOP POSTION FOR HALF MODE HIGH BYTE REGISTER – 0XE2A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_STATUS_Y_TOP[9:8]<br><br>Status vertical top position for Quad and CIF mode. Unit is 1 pixel |

### ROSD STATUS VERTICAL BOTTOM POSTION FOR HALF MODE LOW BYTE REGISTER – 0XE2B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_STATUS_Y_BOTTOM[7:0]<br><br>Status vertical bottom position for Quad and CIF mode. Unit is 1 pixel |

### ROSD STATUS VERTICAL BOTTOM POSTION FOR HALF MODE HIGH BYTE REGISTER – 0XE2C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_STATUS_Y_BOTTOM[9:8]<br><br>Status vertical bottom position for Quad and CIF mode. Unit is 1 pixel |

### ROSD FONT Y COLOR 1 REGISTER – 0XE2D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x10 | ROSD_FONT_Y1[7:0]<br><br>Font color Y for index 1 |

### ROSD FONT CB COLOR 1 REGISTER – 0XE2E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x80 | ROSD_FONT_CB1[7:0]<br><br>Font color Cb for index 1 |

### ROSD FONT CR COLOR 1 REGISTER – 0XE2F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x80 | ROSD_FONT_CR1[7:0]<br><br>Font color Cr for index 1 |

### ROSD FONT Y COLOR 2 REGISTER – 0XE30

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x10 | ROSD_FONT_Y2[7:0]<br><br>Font color Y for index 2 |

### ROSD FONT CB COLOR 2 REGISTER – 0XE31

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x80 | ROSD_FONT_CB2[7:0]<br><br>Font color Cb for index 2 |

### ROSD FONT CR COLOR 2 REGISTER – 0XE32

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x80 | ROSD_FONT_CR2[7:0]<br><br>Font color Cr for index 2 |

### ROSD FONT Y COLOR 3 REGISTER – 0XE33

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x10 | ROSD_FONT_Y3[7:0]<br><br>Font color Y for index 3 |

### ROSD FONT CB COLOR 3 REGISTER – 0XE34

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x80 | ROSD_FONT_CB3[7:0]<br><br>Font color Cb for index 3 |

### ROSD FONT CR COLOR 3 REGISTER – 0XE35

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x80 | ROSD_FONT_CR3[7:0]<br><br>Font color Cr for index 3 |

### ROSD BACKGROUND Y COLOR REGISTER – 0XE36

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x10 | ROSD_BGND_Y [7:0]<br><br>Background color Y |

### ROSD BACKGROUND CB COLOR REGISTER – 0XE37

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x80 | ROSD_BGND_CB[7:0]<br><br>Background color Cb |

### ROSD BACKGROUND CR COLOR REGISTER – 0XE38

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x80 | ROSD_BGND_CR[7:0]<br><br>Background color Cr |

### ROSD BOUNDARY Y COLOR REGISTER – 0XE39

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x10 | ROSD_BNDRY_Y[7:0]<br><br>Boundary color Y |

### ROSD BOUNDARY CB COLOR REGISTER – 0XE3A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x80 | ROSD_BNDRY_CB[7:0]<br><br>Boundary color Cb |

### ROSD BOUNDARY CR COLOR REGISTER – 0XE3B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0x80 | ROSD_BNDRY_CR[7:0]<br><br>Boundary color Cr |

### ROSD FONT RAM ADDRESS LOW BYTE REGISTER – 0XE3C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_FRAM_ADDR[7:0]<br><br>Font SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 640x16. |

### ROSD FONT RAM ADDRESS HIGH BYTE REGISTER – 0XE3D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_FRAM_ADDR[9:8]<br><br>Font SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 640x16. |

### ROSD FONT RAM DATA LOW BYTE REGISTER – 0XE3E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_FRAM_DATA[7:0]<br><br>Font SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 640x16. |

### ROSD FONT RAM DATA HIGH BYTE REGISTER – 0XE3F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_FRAM_DATA[15:8]<br><br>Font SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 640x16. |

### ROSD DISPLAY RAM ADDRESS REGISTER – 0XE40

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_DRAM_ADDR[7:0]<br><br>Display SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 544x6. |

### ROSD DISPLAY RAM ADDRESS REGISTER – 0XE41

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | ROSD_DRAM_ADDR[9:8]<br><br>Display SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 544x6. |

### ROSD DISPLAY RAM DATA REGISTER – 0XE42

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_DRAM_Data[5:0]<br><br>Display SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 544x6. |

### ROSD COLOR BAR ENABLE REGISTER – 0XE43

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_COLBAR_EN<br><br>[7]: ROSD_COLBAR_EN8<br>[6]: ROSD_COLBAR _EN7<br>[5]: ROSD_COLBAR _EN6<br>[4]: ROSD_COLBAR _EN5<br>[3]: ROSD_COLBAR _EN4<br>[2]: ROSD_COLBAR _EN3<br>[1]: ROSD_COLBAR _EN2<br>[0]: ROSD_COLBAR _EN1<br><br>Recording port n (1 to 8) color bar enable or disable. |

### ROSD ENABLE FOR PORT 9 REGISTER – 0XE44

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | [7]: ROSD_COLBAR_EN9<br>[6]: ROSD_MIX_9<br>[5]: ROSD_BGND_EN9<br>[4]: ROSD_BNDRY_EN9<br>[3]: ROSD_STATUS_EN9<br>[2]: ROSD_TITLE_EN9<br>[1]: ROSD_TIME_EN9<br>[0]: ROSD_EN9 |

### ROSD COLBAR VERITICAL BLACK BAR POSITION REGISTER – 0XE45

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_COLBAR_VPOS[7:0]<br><br>There is a black bar (32 lines) in color bar. This register defines start position for this bar. It is for test purpose. |

### ROSD FIELD SWITCHING CONTROL REGISTER – 0XE46

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROSD_FSWITCH[7:0]: Field switch mode for record port 0 to 7<br><br>0: not field switch<br>1: field switch |

### PRIVACY WINDOW OVERFLOW REGISTER 1 – 0XE47

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | Channel 7-0 Overflow bit for Window 1 |

### PRIVACY WINDOW OVERFLOW REGISTER 2 – 0XE48

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | Channel 15-8 Overflow bit for Window 1 |

**Similar functions are distributed in 0xE49 – 0xE4E for Window 2 – Window 4**

### PRIVACY WINDOW SHADOW SET CONTROL REGISTER – 0XE4F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | ROSD_FSWITCH[8]: Field switch mode for network port<br><br>0: not field switch<br>1: field switch |
| 6:3 | R | 0 | Reserved |
| 2 | RW | 0 | Select Vertical<br><br>This bit control the overflow bit. 1 = Select vertical |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1:0 | RW | 0 | **Shadow set control for each live, record, SPOT privacy channels**<br>**00: Record**<br>**01: SPOT**<br>**10: Live**<br>11: Reserved |

**PRIVACY WINDOW 1 HORIZONTAL START REGISTER – 0XE50 – 0XE5F**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | RW | 0 | **HSTART[7:0] for each channel**<br>**This values is the horizontal starting point for privacy window 1. Unit is double pixel.** |

**PRIVACY WINDOW 1 VERTICAL START REGISTER – 0XE60 – 0XE6F**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | **1 = Enable Window** |
| 6:0 | RW | 0 | **VSTART[6:0] for each channel**<br>**This values is the vertical starting point for privacy window 1. Unit is quadruple line.** |

**PRIVACY WINDOW 1 CONTROL REGISTER – 0XE70 – 0XE7F**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R/W | 0 | **Content operation**<br>**000: solid white color**<br>**001: solid black color**<br>**010: over exposed**<br>**011: under exposed**<br>**100: Mosaic block size is 8x8**<br>**101: Mosaic block size is 16x16**<br>**110: Mosaic block size is 32x32**<br>111: Mosaic block size is 64x64 |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:2 | R/W | 0 | Horizontal size select<br>000: 32<br>001: 64<br>010: 128<br>011: 256<br>100: 320<br>101: 640<br>110: 960<br>111:1440 |
| 1:0 | R/W | 0 | Vertical size select<br>00: 32<br>01: 64<br>10: 128<br>11: 288 |

**PRIVACY WINDOW 2 HORIZONTAL START REGISTER – 0XE80 – 0XE8F**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | RW | 0 | HSTART[7:0] for each channel<br>This values is the horizontal starting point for privacy window 2. Unit is double pixel. If WIN4_CON[6] is one, this becomes user color Y register. |

**PRIVACY WINDOW 2 VERTICAL START REGISTER – 0XE90 – 0XE9F**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | 1 = Enable Window |
| 6:0 | RW | 0 | VSTART[6:0] for each channel<br>This values is the vertical starting point for privacy window 2. Unit is quadruple line. |

**PRIVACY WINDOW 2 CONTROL REGISTER – 0XEA0 – 0XEAF**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R/W | 0 | Control over exposed value<br>000: light, 111: is the over |
| 4:2 | R/W | 0 | Horizontal size select<br>000: 32<br>001: 64<br>010: 128<br>011: 256<br>100: 320<br>101: 640<br>110: 960<br>111:1440 |
| 1:0 | R/W | 0 | Vertical size select<br>00: 32<br>01: 64<br>10: 128<br>11: 288 |

**PRIVACY WINDOW 3 HORIZONTAL START REGISTER – 0XEB0 – 0XEBF**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | RW | 0 | HSTART[7:0] for each channel<br>This values is the horizontal starting point for privacy window 3. Unit is double pixel. If WIN4_CON[6] is one, this becomes user color Cb register. |

**PRIVACY WINDOW 3 VERTICAL START REGISTER – 0XEC0 – 0XECF**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | 1 = Enable Window |
| 6:0 | RW | 0 | VSTART[6:0] for each channel<br>This values is the vertical starting point for privacy window 3. Unit is quadruple line. |

**PRIVACY WINDOW 3 CONTROL REGISTER – 0XED0 – 0XEDF**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | Reserved |
| 6:5 | R/W | 0 | Under exposed value select<br>00: 1/8 of the original<br>01: 1/16 of the original<br>10: 1/32 of the original<br>11: 1/64 of the original |
| 4:2 | R/W | 0 | Horizontal size select<br>000: 32<br>001: 64<br>010: 128<br>011: 256<br>100: 320<br>101: 640<br>110: 960<br>111:1440 |
| 1:0 | R/W | 0 | Vertical size select<br>00: 32<br>01: 64<br>10: 128<br>11: 288 |

**PRIVACY WINDOW 4 HORIZONTAL START REGISTER – 0XDD0 – 0XDDF**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | RW | 0 | HSTART[7:0] for each channel<br>This values is the horizontal starting point for privacy window 4. Unit is double pixel. If WIN4_CON[6] is one, this becomes user color Cr register. |

**PRIVACY WINDOW 4 VERTICAL START REGISTER – 0XDE0 – 0XDEF**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | 1 = Enable Window |
| 6:0 | RW | 0 | VSTART[6:0] for each channel<br>This values is the vertical starting point for privacy window 4. Unit is quadruple line. |

**PRIVACY WINDOW 4 CONTROL REGISTER – 0XDF0 – 0XDFF**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | Horizontal Resolution Select<br>1 = use HSTART[7;0] as HSTART[8:1]<br>0 = use overflow H bit as HSTART[8] |
| 6 | R/W | 0 | User color update enable<br>1 = Enable user to update desired color into HPOS2, HPOS3, HPOS4 as Y. Cb, Cr<br>0 = Normal operation |
| 5 | R/W | 0 | User color substitute enable<br>1 = Enable the user color switch<br>0 = Normal operation |
| 4:2 | R/W | 0 | Horizontal size select<br>000: 32<br>001: 64<br>010: 128<br>011: 256<br>100: 320<br>101: 640<br>110: 960<br>111:1440 |
| 1:0 | R/W | 0 | Vertical size select<br>00: 32<br>01: 64<br>10: 128<br>11: 288 |

# SPOT Controller Unit

## Introduction

**TW2880 has a very powerful SPOT display controller which can support four independent TV output (NTSC or PAL). By integrating the TV encoder and the DAC, this controller provide glues less interface to the traditional TV to show the recording content which include the current field being recorded, background color, camera number, border divider and OSD for the date and time being recorded. With properly software registers setup the SPOT display controller can show all sixteen scaled down D1 windows on one display or shows four D1 recording video stream in a serial rotating fashion. The rotating rate, sources and sizes are programmable through registers.**

## Features

- Supports NTSC / PAL standard TV monitor with integrated 10 bit DAC
- Four SPOT controllers to display four D1 or Quad channel simultaneously
- Down scaling support up to 1440x960 resolution
- 27 MHZ output frequency
- Programmable sync timing
- Programmable Channel rotation
- Programmable Back ground color
- Programmable Border control
- Programmable Camera on-off display and location
- Programmable OSD display font and location for the time, date
- Programmable to display REC ports or Network Port to any SPOT

## SPOT Controller Architecture

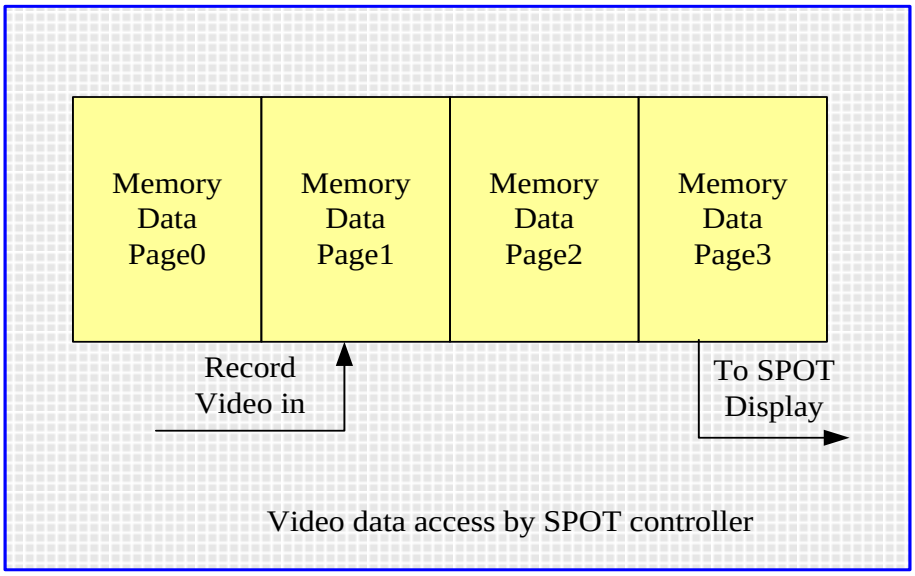
**For the SPOT controller to display the REC video channels on the TV screen, the memory location need to be programmed for the controller to fetch data at the right area, Firmware needs to set the each spot starting address, size, down scale factors, background color, border width and all TV parameters control registers for the CRT monitor to display properly.**

### MEMORY ACCESS CONTROL

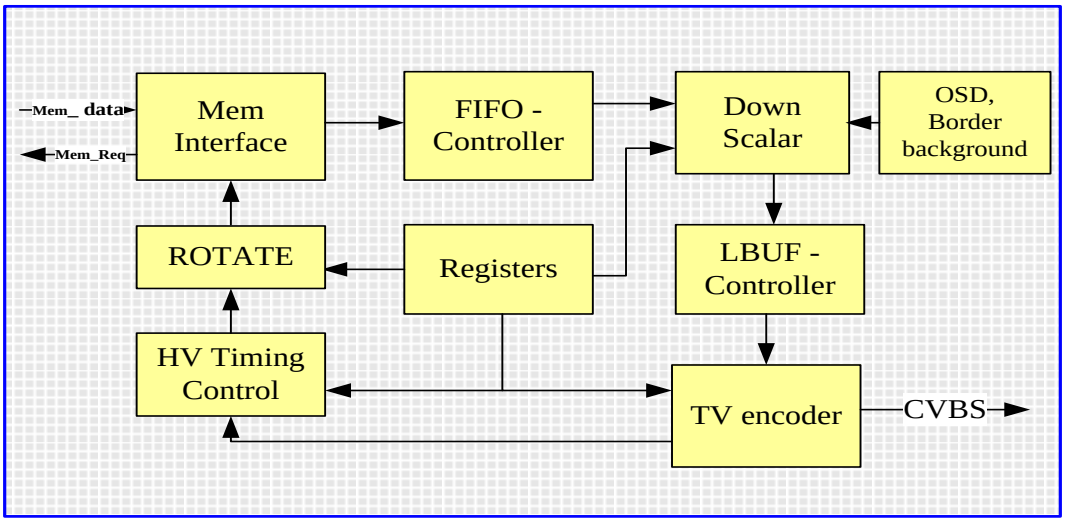
**In the memory mapping, the SPOT controller needs to access the location where the recorded data were written by the encoder interface module. By programming the SPOT HSTART, SPOT VSTART registers, the SPOT controller will display the channel video as the record path.**

### FRAME DATA SYNCHONIZATION

**To prevent the video tearing, the SPOT controller takes use the frame rate synchronization circuit which will guarantee two frames difference between active display frame and active storing frame. If the frame rate of the SPOT controller is higher than the incoming video pipe, frame repeating process is employed to make sure the two frame boundary is maintained.**




**SPOT CONTROLLER BLOCK DIAGRAM**



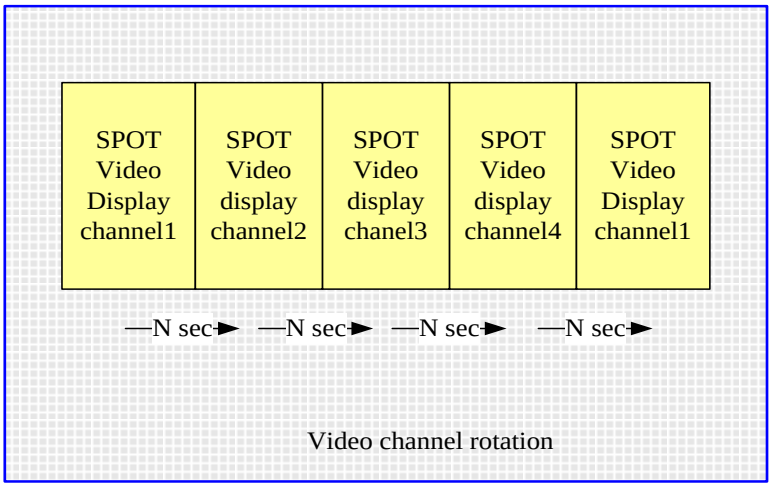

SPOT controller Block Diagram

**MEMORY INTERFACE**

**Base on the information of Horizontal and vertical timing, and under registers control, Memory Interface is a module uses to generate request to the DRAM controller and ask for the DRAM access to get the display data from the DRAM and send it the FIFO.**

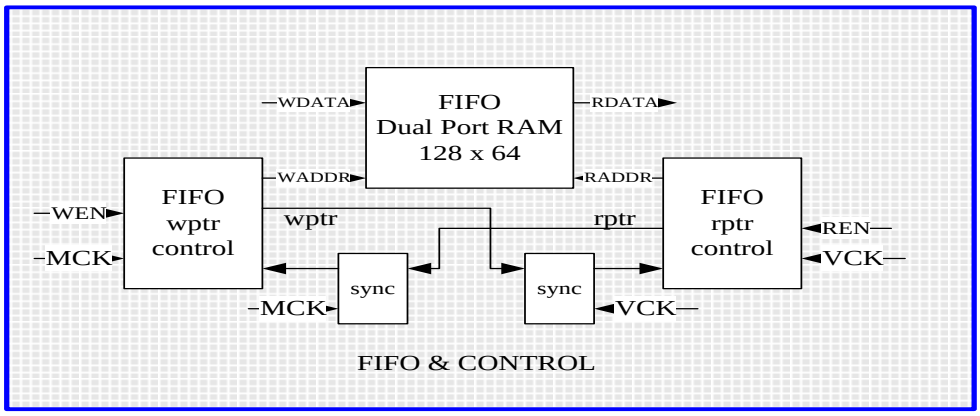
### ROTATION CONTROL

**Base on the information of Horizontal and vertical timing, and under registers control, Rotation control will jump to other location to fetch the new data to display on the screen with the programmed update rate for each video channel.**




### FIFO INTERFACE

**Due to the memory clock and video clock running in different frequency, SPOT controller includes a 128 deep by 64 bit wide FIFO to synchronize the data for cross clock domain operation.**




### HV TIMING CONTROL

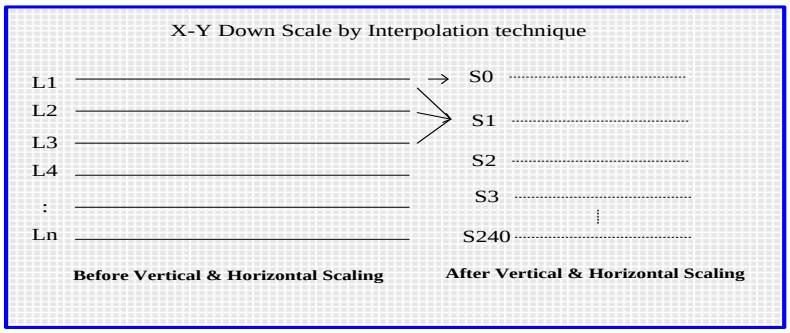
**Base on the setting of horizontal and vertical registers and the inputs information of the TV Encoder, The HV timing control module generates all pixels, lines, and blanking signals and feed them to the Memory interface module to request data from DRAM.**

**Depends on the Odd or Even field indicator output from the TV, The Dram Interface module will generate the memory address in that field and fetches data from the line memory, and send to the Down Scale controller and processes the image according the scale factor and send it to the Line Buffers.**

### DOWN SCALAR

**Down scalar is a module uses to scale down more channels on the display to fit the TV standard timing. So on the second monitor, we can pack as many channels as seen on the LCD and display them on a regular TV.**

**By using the linear interpolation technique, with independent setting of horizontal and vertical scale factors and others registers. We can scale the number of channels, the size and the location of the channels to the display's native resolution.**




### LBUF CONTROL

**Due to the clock rate difference of the Down Scalar and the TV Encoder, there are two line buffers size of 720x16 to accommodate the timing of the TV Encoder.**

### OSD, BACKGROUND AND BORDER CONTROL

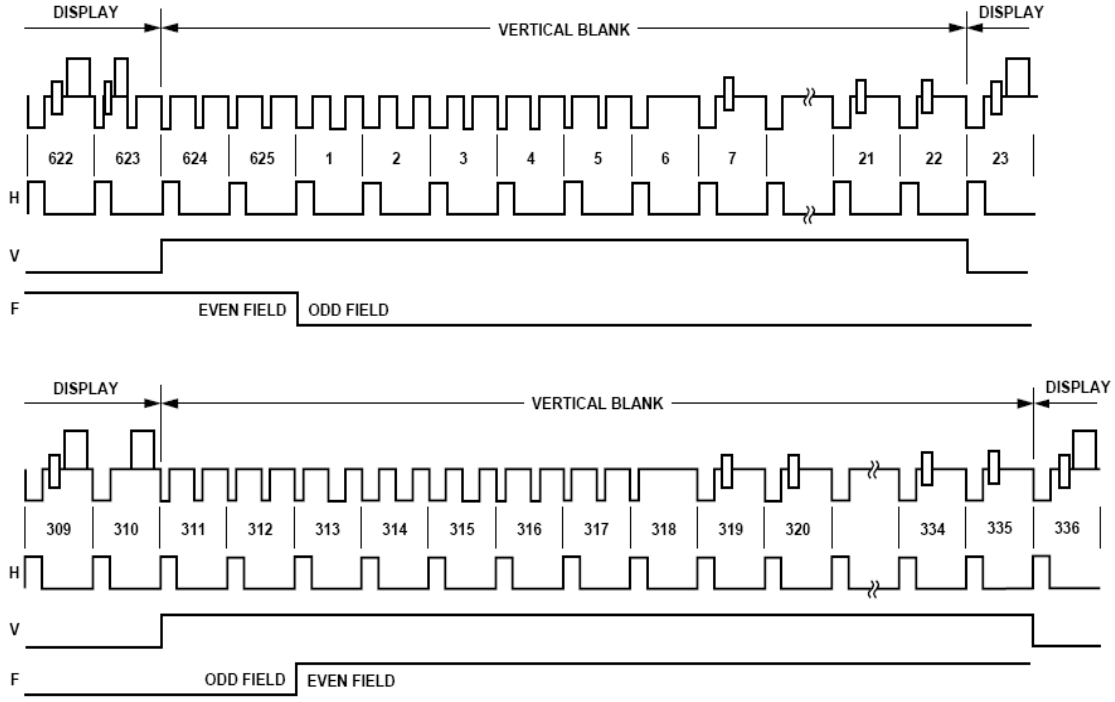
**SPOT controller has an OSD controller for recording path displays channel information for each channel. The OSD controller is a font based design. All font tables are stored in local SRAM. Channel information includes 32 characters date/time, 16 characters title and 16 characters status. There are totally 64 fonts saved in SRAM. Font width can be selected from 6, 8, 12 or 16. Font height can be selected from 8, 10, 16 or 20. Channel information is also saved in local SRAM. Besides channel information layer, there are background layer and channel boundary layer. Following are the features supported in the SPOT display operation.**

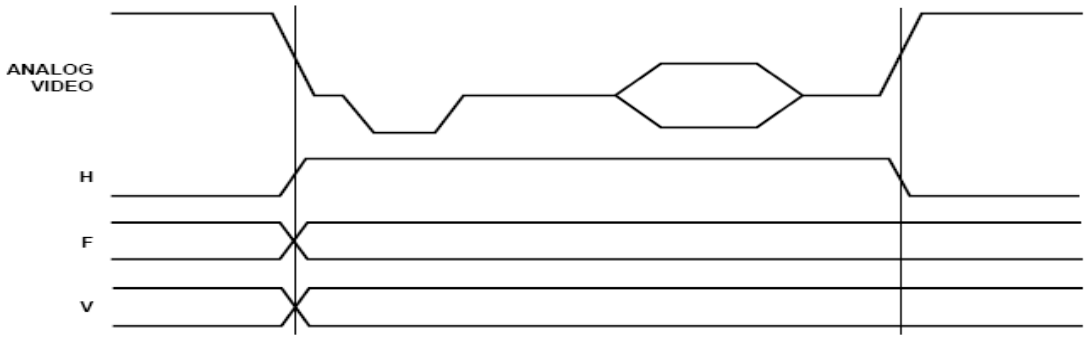
- 4 OSD engines for 4 SPOT ports
- 64 fonts table saved in SRAM
- Channel information table saved in SRAM
- Three lines channel information
- 32 characters date/time
- 8 characters channel title for each channel
- 1 picture for each channel
- 32 characters status for each channel
- font size can be changed
- Channel Boundary
- Background
- Color Bar

**For detailed operation, please refer to the "OSD for Recording Path" section beginning on page 452.**

## TV ENCODER

**TV Encoder is the module converts all component data from scalar into a standard analog baseband television signal (CVBS) which is compatible with worldwide standards. Follow is the PAL timing.**




## THEORY OF OPERATION

**When the SPOT controller is enabling, the controller can access the whole memory map and display the content with the start address set by the HSTART and the VSTART registers. Of course, if the setting is out of bound, the display will show something on the screen with no meaning at all. So for the controller to operate properly, some registers need to be set.**

**While the controller has quit a few registers, there are a few registers need to be adjust, others just are default values. Here are the registers needed to change if the resolutions, scaling, and different memory location wanted to be displayed.**

**In the calculation, the vertical period time is based on the TV standard vertical period.**

**In NTSC, this is 16.667 ms in each field, in the 1440x960 mode. Base on the 960 line per field in interlace mode, there are 480 lines in either odd or even field and scaled down to 240 lines plus some blank lines for NTSC standard.**

**In NTSC mode:**

**Each field total pixel = 525 * (1440 + undisplay_pixel)**

**Each field total time = 16.678 ms**

**Total pixels per field (calculated from field period and 108 MHz) = 16.678 ms / (1/108 MHZ) = 1,801224 pixels**

**Undisplay_pixel = (1801224 – 525 * 1440 ) / 525 = 1045224 / 525 = 1991**

**SP_HTT = 1991 +1440 = 3431 = 'hD67**

**SP_HDE = 1439 = 'h59F**

**SP_VTT = 523 = 'h20B**

**SP_VDE = 479 = 'h1DF**

**SP_SRC_WID = 1440 = 'h5A0**

**SP_SRC_HI = 480 = 'h1E0**

**SP_DST_WID = 720 = 'h2D0**

**SP_DST_HI = 240 = 'hF0**

**If SPOT is set to display in D1 or quad mode.**

**SP_HTT = 6863 = 'h1ACF**

**SP_HDE = 719 = 'h2CF**

**SP_VTT = 261 = 'h105**

**SP_VDE = 239 = 'hEF**

**SP_SRC_WID = 720 = 'h2D0**

**SP_SRC_HI = 240 = 'hF0**

**SP_DST_WID = 720 = 'h2D0**

**SP_DST_HI = 240 = 'hF0**

**In Pal mode:**

**The SPOT is set to D1 or quad mode**

**SP_HTT = 6863 = 'h1ACF**

**SP_HDE = 719 = 'h2CF**

**SP_VTT = 311 = 'h137**

**SP_VDE = 287 = 'h11F**

**SP_SRC_WID = 720 = 'h2D0**

**SP_SRC_HI = 240 = 'hF0**

**SP_DST_WID = 720 = 'h2D0**

**SP_DST_HI = 240 = 'hF0**

**SP_TV_CTRL(reg 'hF17) = 'h41**

**SP_TV_CTRL(reg 'hF18) = 'h05**

## Registers Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xF00 | R/W | 0x00 | SP1_HTT[7:0], SPOT1 Horizontal Total Register |
| 0xF01 | R/W | 0x00 | SP1_HTT[12:8], SPOT1 Horizontal Total Register |
| 0xF02 | R/W | 0x00 | SP1_HDE[7:0], SPOT1 Horizontal Display Enable Register |
| 0xF03 | R/W | 0x00 | SP1_HDE[11:8], SPOT1 Horizontal Display Enable Register |
| 0xF04 | R/W | 0x00 | SP1_VDE[7:0], SPOT1 Vertical Display Enable Register |
| 0xF05 | R/W | 0x00 | [7:4] BND_CH_EN[3:0] SPOT Border Channel Enable<br>[2:0] SP1_VDE[10:8], SPOT1 Vertical Display Enable Register |
| 0xF06 | R/W | 0x00 | SP1_HSYNC[7:0], SPOT1 Horizontal Sync register |
| 0xF07 | R/W | 0x00 | SP_BND_WIDTH_V[3:0], Border Width in Vertical direction in Field Switching Mode<br>[7:4] SP_BND_WIDTH_V[3:0]<br>[3:2] SP1_FRSC<br>[1] SPOT Reset FRSC bank sel when no video detection active<br>[0] SP1_HSYNC[8] |
| 0xF08 | R/W | 0x00 | SP1_VSYNC[7:0], SPOT1 Vertical Sync Register |
| 0xF09 | R/W | 0x00 | SP1_VTT[7:0], SPOT1 Vertical Total Register low byte, Hi Byte is in F0A registers |
| 0xF0A | R/W | 0x06 | SP1_VTT[10:8], SPOT1 VTT Total Register<br><br>[7] SP1_Half_D1 Select<br>[6:3] RESERVED<br>[2:0] SP1_VTT[10:8] |
| 0xF0B | R/W | 0x00 | SP Frame Rate Control and Vblank Select<br><br>[7:6] Vblank Select from one of the four SPOT for FRSC<br>[5] SP1 Colorbar Enable<br>[4] RESERVED<br>[3:2] SP1_FLD select<br>11: use bank0 from the FRSC as the field<br>10: use bank0 from the FRSC inverse as the filed<br>0x: use HVCNT generated field<br>[1:0] reserved |
| 0xF0C | R/W | 0x00 | SP_ROT_INT_EN<br><br>[7] SP4 Rot Int status<br>[6] SP3 Rot Int status<br>[5] SP2 Rot Int status<br>[4] SP1 Rot Int status<br><br>[3] SP4 Rot Int Enable<br>[2] SP3 Rot Int Enable<br>[1] SP2 Rot Int Enable<br>[0] SP1 Rot Int Enable |
| 0xF0D | R/W | 0x00 | SP1_DNS Upper Bits, lower byte in reg F13, F14, F15, F16<br><br>[7:6] SP1 Down Scaler Destination Width upper bit [9:8]<br>[5:3] SP1 Down Scaler Source Height upper bit [10:8]<br>[2:0] SP1 Down Scaler Source Width upper bit [10:8] |
| 0xF0E | R/W | 0x00 | [7] SP1 REC or Network Port input enable<br>[6:4] SP1 Internal Color Bar test Select<br>[3:1] SP1 Line Buffer First line Control<br>[0] SP1 Down scaler Destination Height upper bit [8] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xF0F | R/W | 0x00 | SPOT1 Enable Register<br>[7] SP1_FLD_MODE<br>[6] SP1_2ND_MEM<br>[5] SP1_FRMCNT_EN, SPOT1 Frame Count Enable<br>[4] SP1_ENA, SPOT1 Enable<br>[3:2] SP1_ROT_EN, SPOT1 Rotate Enable<br>[1] SP1_VS_POL, SPOT1 Vsync Polarity Select<br>[0] SP1_HS_POL, SPOT1 Hsync Polarity Select |
| 0xF10 | R/W | 0x00 | SP1_HSTART[7:0], SPOT1 Horizontal Start Address Register |
| 0xF11 | R/W | 0x00 | SP1_VSTART[7:0], SPOT1 Vertical Start Address Register |
| 0xF12 | R/W | 0x00 | [7] SP1 Mode Enable<br>[5:4] SP1_HSTART[9:8], SPOT1 Horizontal Start Address Register<br>[3:0] SP1_VSTART[11:8], SPOT1 Vertical Start Address Register |
| 0xF13 | R/W | 0x00 | SP1_SRC_Width[7:0],<br>SPOT1 DNS Source width Register, upper 3 bit locates in 0xF0D[2:0] |
| 0xF14 | R/W | 0x00 | SP1_SRC_Height[7:0],<br>SPOT1 DNS Source height Register, upper 3 bit locates in 0xF0D[5:3] |
| 0xF15 | R/W | 0x00 | SP1_DST_Width[7:0],<br>SPOT1 DNS Destination width Register, upper 3 bit locates in 0xF0D[7:6] |
| 0xF16 | R/W | 0x00 | SP1_DST_Height[7:0],<br>SPOT1 DNS Destination height Register, upper 1 bit locates in 0xF0E[0] |
| 0xF17 | R/W | 0x00 | SP1_TVE_STD[7:0], SPOT1 TV Standard Register |
| 0xF18 | R/W | 0x00 | SP1_TVE_POL[7:0], SPOT1 TV Sync Polarity Register |
| 0xF19 | R/W | 0x00 | SP1_TVE_FDET[6:0], SPOT1 TV Field Control Register |
| 0xF1A | R/W | 0x00 | SP1_TVE_HSENC[7:0], SPOT1 TV Encoder Active Pixel delay |
| 0xF1B | R/W | 0x00 | SP1_TVE_HSENC[9:8],<br>[7:2] SPOT1 Rotate Time Offset<br>[1:0] SPOT1 TV Encoder Active Pixel delay |
| 0xF1C | R/W | 0x00 | [7:6] SP1_CVBS_SRC_SEL<br>[5:0] SP1_TVE_VSENC[5:0], SPOT1 TV Encoder Active Line delay |
| 0xF1D | R/W | 0x00 | SP1_TVE_LINE_OS[4:0], SPOT1 TV Encoder Active Line offset |
| 0xF1E | R/W | 0x00 | [7;6] SP1_FLD_SW_MODE<br>[5:0] SP1_TVE_PIXEL_OS[5:0], SPOT1 TV Encoder Active Pixel Offset |
| 0xF1F | R/W | 0x00 | SP1_TVE_FLD_OS[1:0], SPOT1 TV Encoder Active Field offset |
| 0xF20 | R/W | 0x00 | SP1_TVE_FSC_FREE[7], SPOT1 TV Encoder Subcarrier run free<br>[6:0] RESERVED |
| 0xF21 | R/W | 0x00 | [7:0] Vertical Start Address offset [7:0] in 256 or 512 Mb mem config. |
| 0xF22 | R/W | 0x00 | SPOT_BACKGND_R[7:0], SPOT Background Red Color |
| 0xF23 | R/W | 0x00 | SPOT_BACKGND_G[7:0], SPOT Background Green Color |
| 0xF24 | R/W | 0x00 | SPOT_BACKGND_B[7:0], SPOT Background Blue Color |
| 0xF25 | R/W | 0x00 | [7] SPOT Switch Enable<br>[6] SPOT Internal Color Bar Enable<br>[4] SP1_BRDR_EN, SPOT1 Border Enable<br>[3:0] SPOT Border Width |
| 0xF26 | R/W | 0x00 | SPOT_BORDER COLOR R[7:0], SPOT BORDER Red Color |
| 0xF27 | R/W | 0x00 | SPOT_BORDER COLOR G[7:0], SPOT BORDER Green Color |
| 0xF28 | R/W | 0x00 | SPOT_BORDER COLOR B[7:0], SPOT BORDER Blue Color |
| 0xF29 | R/W | 0x00 | SPOT_ROT_FREQ[7:0], SPOT Channel Rotate Frequency |
| 0xF2A | R/W | 0x00 | SP2_OSD_DRAM_ADDR[7:0] |
| 0xF2B | R/W | 0x00 | SP2_OSD_DRAM_DATA[5:0] |
| 0xF2C | R/W | 0x00 | SP3_OSD_DRAM_ADDR[7:0] |
| 0xF2D | R/W | 0x00 | SP3_OSD_DRAM_DATA[5:0] |
| 0xF2E | R/W | 0x00 | SP4_OSD_DRAM_ADDR[7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xF2F | R/W | 0x00 | SP4_OSD_DRAM_DATA[5:0] |
| 0xF30 | R/W | 0x00 | SPOT_BORDER_ENABLE<br>———————<br>[7] SPOT4_BORDER_EN<br>[6] SPOT3_BORDER_EN<br>[5] SPOT2_BORDER_EN<br>———————<br>[4] SP1_OSD_TITLE_EN<br>[3] SP1_OSD_TIME_EN<br>[2] SP1_OSD_CHPIC_EN<br>[1] SP1_OSD_CHNUM_EN<br>[0] SP1_OSD_EN |
| 0xF31 | R/W | 0x00 | SPOT1 OSD Control Register<br>[5] SP1_OSD_CHPIC_BLINK<br>[4] SP1_OSD_CHPIC_TRANS<br>[3] SP1_OSD_TITLE_MIX<br>[2] SP1_OSD_TIME_MIX<br>[1] SP1_OSD_CHPIC_MIX<br>[0] SP1_OSD_CHNUM_MIX |
| 0xF32 | R/W | 0x00 | [7:6] SP1_OSD_CHPIC_POS<br>[5:4] SP1_OSD_CHNUM_POS<br>[3:2] SP1_OSD_FONT_VSIZE<br>[1:0] SP1_OSD_FONT_HSIZE |
| 0xF33 | R/W | 0x00 | SPOT_OSD_CHNUM_HPOS[7:0] |
| 0xF34 | R/W | 0x00 | SPOT_OSD_CHNUM_HPOS[10:8] |
| 0xF35 | R/W | 0x00 | SPOT_OSD_CHNUM_VPOS[7:0] |
| 0xF36 | R/W | 0x00 | [7:3] SP1_FRC_CTL<br>[2:0] SPOT_OSD_CHNUM_VPOS[10:8] |
| 0xF37 | R/W | 0x80 | SPOT_OSD_CHPIC_HPOS[7:0] |
| 0xF38 | R/W | 0x80 | [7:3] SP2_FRC_CTL<br>SPOT_OSD_CHPIC_HPOS[10:8] |
| 0xF39 | R/W | 0x10 | SPOT_OSD_CHPIC_VPOS[7:0] |
| 0xF3A | R/W | 0x00 | [7:3] SP3_FRC_CTL<br>SPOT_OSD_CHPIC_VPOS[10:8] |
| 0xF3B | R/W | 0x00 | SPOT_OSD_TIME_HPOS[7:0] |
| 0xF3C | R/W | 0x00 | [7:3] SP4_FRC_CTL<br>SPOT_OSD_TIME_HPOS[10:8] |
| 0xF3D | R/W | 0x00 | SPOT_OSD_TIME_VPOS[7:0] |
| 0xF3E | R/W | 0x00 | SPOT_OSD_TIME_VPOS[10:8] |
| 0xF3F | R/W | 0x00 | SPOT_OSD_TITLE_HPOS[7:0] |
| 0xF40 | R/W | 0x00 | SPOT_OSD_TITLE_HPOS[10:8] |
| 0xF41 | R/W | 0x00 | SPOT_OSD_TITLE_VPOS[7:0] |
| 0xF42 | R/W | 0x00 | SPOT_OSD_TITLE_VPOS[10:8] |
| 0xF43 | R/W | 0x00 | SPOT_OSD_FONT_PR1[7:0] |
| 0xF44 | R/W | 0x00 | SPOT_OSD_FONT_Y1[7:0] |
| 0xF45 | R/W | 0x00 | SPOT_OSD_FONT_PB1[7:0] |
| 0xF46 | R/W | 0x00 | SPOT_OSD_FONT_PR2[7:0] |
| 0xF47 | R/W | 0x00 | SPOT_OSD_FONT_Y2[7:0] |
| 0xF48 | R/W | 0x00 | SPOT_OSD_FONT_PB2[7:0] |
| 0xF49 | R/W | 0x00 | SPOT_OSD_FONT_PR3[7:0] |
| 0xF4A | R/W | 0x00 | SPOT_OSD_FONT_Y3[7:0] |
| 0xF4B | R/W | 0x00 | SPOT_OSD_FONT_PB3[7:0] |
| 0xF4C | R/W | 0x00 | SPOT_OSD_PIC_PR0[7:0] |
| 0xF4D | R/W | 0x00 | SPOT_OSD_PIC_Y0[7:0] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xF4E | R/W | 0x00 | SPOT_OSD_PIC_PB0[7:0] |
| 0xF4F | R/W | 0x00 | SPOT_OSD_PIC_PR1[7:0] |
| 0xF50 | R/W | 0x00 | SPOT_OSD_PIC_Y1[7:0] |
| 0xF51 | R/W | 0x00 | SPOT_OSD_PIC_PB1[7:0] |
| 0xF52 | R/W | 0x00 | SPOT_OSD_PIC_PR2[7:0] |
| 0xF53 | R/W | 0x00 | SPOT_OSD_PIC_Y2[7:0] |
| 0xF54 | R/W | 0x00 | SPOT_OSD_PIC_PB2[7:0] |
| 0xF55 | R/W | 0x00 | SPOT_OSD_PIC_PR3[7:0] |
| 0xF56 | R/W | 0x00 | SPOT_OSD_PIC_Y3[7:0] |
| 0xF57 | R/W | 0x00 | SPOT_OSD_PIC_PB3[7:0] |
| 0xF58 | R/W | 0x00 | SP1_OSD_FRAM_ADDR[7:0] |
| 0xF59 | R/W | 0x00 | SP1_OSD_FRAM_ADDR[10:8] |
| 0xF5A | R/W | 0x00 | SP1_OSD_FRAM_DATA[7:0] |
| 0xF5B | R/W | 0x00 | SP1_OSD_FRAM_DATA[15:8] |
| 0xF5C | R/W | 0x00 | SP1_OSD_DRAM_ADDR[7:0] |
| 0xF5D | R/W | 0x00 | SP1_OSD_DRAM_DATA[5:0] |
| 0xF5E | R/W | 0x00 | SPOT_DISP_PITCH[7:0] |
| 0xF5F | R/W | 0x00 | SPOT_ADJUST_ENA[7:0] |
| 0xF60 | R/W | 0x00 | SP2_HTT[7:0], SPOT2 Horizontal Total Register |
| 0xF61 | R/W | 0x00 | SP2_HTT[12:8], SPOT2 Horizontal Total Register |
| 0xF62 | R/W | 0x00 | SP2_HDE[7:0], SPOT2 Horizontal Display Enable Register |
| 0xF63 | R/W | 0x00 | SP2_HDE[11:8], SPOT2 Horizontal Display Enable Register |
| 0xF64 | R/W | 0x00 | SP2_VDE[7:0], SPOT2 Vertical Display Enable Register |
| 0xF65 | R/W | 0x00 | [7:4] BND_CH_EN SPOT Border Channel Enable<br>[3] RESERVED<br>[2:0] SP2_VDE[10:8], SPOT2 Vertical Display Enable Register |
| 0xF66 | R/W | 0x00 | SP2_HSYNC[7:0], SPOT2 Horizontal Position Sync Register |
| 0xF67 | R/W | 0x00 | SP_BND_WIDTH_SEL[3:0], Border Width in Vertical direction in Field Switching Mode<br>[7:4] SP_BND_WIDTH_V[3:0]<br>[3:2] SP2_FRSC<br>[1] SP2_PHB[8]<br>[0] SP2_PHSYNC[8] |
| 0xF68 | R/W | 0x00 | SP2_VSYNC[7:0], SPOT2 Vertical Position Sync Register |
| 0xF69 | R/W | 0x00 | SP2_ VTT[7:0] SPOT2 Vertical Total Register |
| 0xF6A | R/W | 0x00 | SP2 _VTT[10:8] SPOT2 Vertical Total Upper Bits Register<br><br>[7] SP2_Half_D1 Select<br>[6] RESERVED<br>[5:4] SEL656P Select 1 of 4 SPOT output to send to Network port<br>[3] RESERVED<br>[2:0] SP2_VTT[10:8] |
| 0xF6B | R/W | 0x00 | [7:6] RESERVED<br>[5] SP2_COLORBAR, SPOT2 Color Bar Enable Register<br>[4] SP2_BG_SW, SPOT2 Switch Background Enable Register<br>[3:2] SP2_FLD select<br>11: use bank0 from the FRSC as the field<br>10: use bank0 from the FRSC inverse as the filed<br>0x: use HVCNT generated field<br>[1:0] reserved |
| 0xF6C | R/W | 0x00 | [7:0] RESERVED |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xF6D | R/W | 0x00 | SP2_DNS SPOT2 DOWN SCALE Upper Bits Register<br>[7:6] SP2 Destination Width upper bit [9:8]<br>[5:3] SP2 Source Height upper bit [10:8]<br>[2:0] SP2 Source Width upper bit [10:8] |
| 0xF6E | R/W | 0x00 | [7] SP2 Network Port input enable<br>[6:5] SP2 Internal Color Bar test Select<br>[3:1] SP2 Line Buffer First line Control<br>[0] SP2 Destination Height upper bit [8] |
| 0xF6D | R/W | 0x00 | SP2_HBDEL[7:0], SPOT2 Horizontal Blank Delay Register |
| 0xF6E | R/W | 0x00 | [7] SP2_BW_CTRL,<br>[6:4] SP2_CBAR,<br>[0] SP2_DHB[8] |
| 0xF6F | R/W | 0x00 | SPOT2 Enable Register<br>[7] SP2_FLD_MODE<br>[6] SP2_2ND_MEM<br>[5] SP2_FRMCNT_EN, SPOT2 Frame Count Enable<br>[4] SP2_ENA, SPOT2 Enable<br>[3:2] SP2_ROT_EN, SPOT2 Rotate Enable<br>[1] SP2_VS_POL, SPOT2 Vsync Polarity Select<br>[0] SP2_HS_POL, SPOT2 Hsync Polarity Select |
| 0xF70 | R/W | 0x00 | SP2_HSTART [7:0], SPOT2 Horizontal Start Address Register |
| 0xF71 | R/W | 0x00 | SP2_VSTART [7:0], SPOT2 Vertical Start Address Register |
| 0xF72 | R/W | 0x00 | [7] SP2 Quad Mode Enable<br>[5:4] SP2_HSTART [9:8], SPOT2 Horizontal Start Address Register<br>[3:0] SP2_VSTART [11:8], SPOT2 Vertical Start Address Register |
| 0xF73 | R/W | 0x00 | SP2_SRC_WID[7:0], SPOT2 Source width Register |
| 0xF74 | R/W | 0x00 | SP2_SRC_HI[7:0], SPOT2 Source height Register |
| 0xF75 | R/W | 0x00 | SP2_DST_WID[7:0], SPOT2 Destination width Register |
| 0xF76 | R/W | 0x00 | SP2_DST_HI[7:0], SPOT2 Destination height Register |
| 0xF77 | R/W | 0x00 | SPOT2 TV Encoder Standard Register<br>[7:6] SP2_TVE_FSC_SEL, SPOT2 TV Encoder Fsc Select<br>[5] SP2_TVE_SAT_LMT, SPOT2 TV Saturation Limit<br>[0] SP2_TVE_STD, SPOT2 TV Standard Select |
| 0xF78 | R/W | 0x00 | SPOT2 TV Encoder Standard Register<br>[7] SP2_TVE_VSPOL, SPOT2 TV Encoder Vertical Sync Polarity<br>[6] SP2_TVE_HSPOL, SPOT2 TV Encoder Horiz Sync Polarity<br>[5] SP2_TVE_VSDET, SPOT2 TV Encoder Vsync detect<br>[4] SP2_TVE_PED, SPOT2 TV Encoder Pedestal Enable<br>[3] SP2_TVE_PDRST, SPOT2 TV Encoder Pedestal Reset<br>[2] SP2_TVE_PHALT, SPOT2 TV Encoder Phase Alternated<br>[1] SP2_TVE_F_INV, SPOT2 TV Encoder Field Inverted<br>[0] SP2_TVE_INTL, SPOT2 TV Encoder Interlace Select |
| 0xF79 | R/W | 0x00 | SPOT2 TV Encoder Offset Register<br>[5:4] SP2_TVE_PIX_OS, SPOT2 TV Encoder Pixel Offset<br>[3:2] SP2_TVE_FLD_OS, SPOT2 TV Encoder field Offset<br>[1] SP2_TVE_FLD_POL, SPOT2 TV Encoder Field Polarity<br>[0] SP2_TVE_ FLDDET, SPOT2 Encoder Field Detect |
| 0xF7A | R/W | 0x00 | SP2_TVE_HSENC[7:0], SPOT2 TV Encoder Active Pixel delay |
| 0xF7B | R/W | 0x00 | SP2_TVE_HSENC[9:8],<br>[7:2] SPOT2 Rotate Time Offset<br>[1:0] SPOT2 TV Encoder Active Pixel delay |
| 0xF7C | R/W | 0x40 | [7:6] SP2_CVBS_SRC_SEL<br>[5:0] SP2_TVE_VSENC[5:0], SPOT2 TV Encoder Active Line delay |
| 0xF7D | R/W | 0x00 | SP2_TVE_LINE_OS[4:0], SPOT2 TV Encoder Active Line offset |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xF7E | R/W | 0x00 | [7;6] SP2_FLD_SW_MODE<br>[5:0] SP2_TVE_PIXEL_OS[5:0], SPOT2 TV Encoder Active Pixel offset |
| 0xF7F | R/W | 0x00 | SP2_TVE_FLD_OS[1:0], SPOT2 TV Encoder Active Field offset |
| 0xF80 | R/W | 0x00 | SP2_TVE_FSC_FREE[7], SPOT2 TV Encoder Subcarrier run free<br>[6:0] RESERVED |
| 0xF81 | R/W | 0x00 | SPOT2 OSD Enable Register<br>[4] SP2_OSD_TITLE_EN<br>[3] SP2_OSD_TIME_EN<br>[2] SP2_OSD_CHPIC_EN<br>[1] SP2_OSD_CHNUM_EN<br>[0] SP2_OSD_OSD_EN |
| 0xF82 | R/W | 0x00 | SPOT2 OSD Control Register<br>[5] SP2_OSD_CHPIC_BLINK<br>[4] SP2_OSD_CHPIC_TRANS<br>[3] SP2_OSD_TITLE_MIX<br>[2] SP2_OSD_TIME_MIX<br>[1] SP2_OSD_CHPIC_MIX<br>[0] SP2_OSD_CHNUM_MIX |
| 0xF83 | R/W | 0x00 | [7:6] SP2_OSD_CHPIC_POS<br>[5:4] SP2_OSD_CHNUM_POS<br>[3:2] SP2_OSD_FONT_VSIZE<br>[1:0] SP2_OSD_FONT_HSIZE |
| 0xF84 | R/W | 0x00 | SP2_OSD_TIME_HPOS[7:0] |
| 0xF85 | R/W | 0x00 | SP2_OSD_TIME_HPOS[10:8] |
| 0xF86 | R/W | 0x00 | SP2_OSD_TIME_VPOS[7:0] |
| 0xF87 | R/W | 0x00 | SP2_OSD_TIME_VPOS[10:8] |
| 0xF88 | R/W | 0x00 | SP2_OSD_TITLE_HPOS[7:0] |
| 0xF89 | R/W | 0x00 | [7:3] Vertical Start Address Offset [12:8] in 256 or 512 Mb mem config<br>SP2_OSD_TITLE_HPOS[10:8] |
| 0xF8A | R/W | 0x00 | SP2_OSD_TITLE_VPOS[7:0] |
| 0xF8B | R/W | 0x00 | SP2_OSD_TITLE_VPOS[10:8] |
| 0xF8C | R/W | 0x00 | SP2_OSD_FRAME_ADDR[7:0] |
| 0xF8D | R/W | 0x00 | SP2_OSD_FRAME_ADDR[10:8] |
| 0xF8E | R/W | 0x00 | SP2_OSD_FRAME_DATA[7:0] |
| 0xF8F | R/W | 0x00 | SP2_OSD_FRAME_DATA [15:8] |
| 0xF90 | R/W | 0x00 | SP3_HTT[7:0], SPOT3 Horizontal Total Register |
| 0xF91 | R/W | 0x00 | SP3_HTT[12:8], SPOT3 Horizontal Total Register |
| 0xF92 | R/W | 0x00 | SP3_HDE[7:0], SPOT3 Horizontal Display Enable Register |
| 0xF93 | R/W | 0x00 | SP3_HDE[11:8], SPOT3 Horizontal Display Enable Register |
| 0xF94 | R/W | 0x00 | SP3_VDE[7:0], SPOT3 Vertical Display Enable Register |
| 0xF95 | R/W | 0x00 | [7:4] BND_CH_EN [11:8] SPOT Border Channel Enable<br>[3] RESERVED<br>[2:0] SP3_VDE[10:8], SPOT3 Vertical Display Enable Register |
| 0xF96 | R/W | 0x00 | SP3_HSYNC[7:0], SPOT3 Horizontal Position Sync Register |
| 0xF97 | R/W | 0x00 | [7:4] RESERVED<br>[3:2] SP3_FRSC<br>[1] SP3_HBLNK[8], SPOT3 Horizontal Blank Position Register<br>[0] SP3_HSYNC[8], SPOT3 Horizontal Sync Position Register |
| 0xF98 | R/W | 0x00 | SP3_VSYNC[7:0], SPOT3 Vertical Position Sync Register |
| 0xF99 | R/W | 0x00 | SP3_VTT[7:0], SPOT3 VTT Register |
| 0xF9A | R/W | 0x00 | SP3_VTT[10:8], SPOT3 VTT Register<br>[7] SP3_Half_D1 Select<br>[6:3] RESERVED<br>[2:0] SP3_VTT[10:8] |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xF9B | R/W | 0x00 | [7:6] RESERVED<br>[5] SP3_COLORBAR, SPOT3 Color Bar Enable Register<br>[4] SP3_BG_SW, SPOT3 Switch Background Enable Register<br>[3:2] SP3_FLD select<br>11: use bank0 from the FRSC as the field<br>10: use bank0 from the FRSC inverse as the filed<br>0x: use HVCNT generated field<br>[1] SP2_PD_EN<br>[0] SP1_PD_EN |
| 0xF9C | R/W | 0x00 | SP_USE_BY_REC_NETWK_PORT<br>[7:4] Select REC ports 9-0 or network to replace SP2 as display port<br>[3:0] Select REC ports 9-0 or network to replace SP1 as display port |
| 0xF9D | R/W | 0x00 | SP3_DNS Down Scale register Upper Bits<br>[7:6] SP3 Destination Width upper bit [9:8]<br>[5:3] SP3 Source Height upper bit [10:8]<br>[2:0] SP3 Source Width upper bit [10:8] |
| 0xF9E | R/W | 0x00 | [7] SP3 Network Port input enable<br>[6:5] SP3 Internal Color Bar test Select<br>[3:1] SP3 Line Buffer First line Control<br>[0] SP3 Destination Height upper bit [8] |
| 0xF9F | R/W | 0x00 | SPOT3 Enable Register<br>[7] SP3_FLD_MODE<br>[6] SP3_2ND_MEM<br>[5] SP3_FRMCNT_EN, SPOT3 Frame Count Enable<br>[4] SP3_ENA, SPOT3 Enable<br>[3:2] SP3_ROT_EN, SPOT3 Rotate Enable<br>[1] SP3_VS_POL, SPOT3 Vsync Polarity Select<br>[0] SP3_HS_POL, SPOT3 Hsync Polarity Select |
| 0xFA0 | R/W | 0x00 | SP3_HSTART [7:0], SPOT3 Horizontal Start Address Register |
| 0xFA1 | R/W | 0x00 | SP3_VSTART [7:0], SPOT3 Vertical Start Address Register |
| 0xFA2 | R/W | 0x00 | [7] SP3 Quad Mode Enable<br>[5:4] SP3_HSTART [9:8], SPOT3 Horizontal Start Address Register<br>[3:0] SP3_VSTART [11:8], SPOT3 Vertical Start Address Register |
| 0xFA3 | R/W | 0x00 | SP3_SRC_WID[7:0], SPOT3 Source Width Register |
| 0xFA4 | R/W | 0x00 | SP3_SRC_HI[7:0], SPOT3 Source Height Register |
| 0xFA5 | R/W | 0x00 | SP3_DST_WID[7:0], SPOT3 Destination Width Register |
| 0xFA6 | R/W | 0x00 | SP3_DST_HI[7:0], SPOT3 Destination Height Register |
| 0xFA7 | R/W | 0x00 | SPOT3 TV Encoder Standard Register<br>[7:6] SP3_TVE_FSC_SEL, SPOT3 TV Encoder Fsc Select<br>[5] SP3_TVE_SAT_LMT, SPOT3 TV Saturation Limit<br>[0] SP3_TVE_STD, SPOT3 TV Standard Select |
| 0xFA8 | R/W | 0x00 | SPOT3 TV Encoder Standard Register<br>[7] SP3_TVE_VSPOL, SPOT3 TV Encoder Vertical Sync Polarity<br>[6] SP3_TVE_HSPOL, SPOT3 TV Encoder Horiz Sync Polarity<br>[5] SP3_TVE_VSDET, SPOT3 TV Encoder Vsync detect<br>[4] SP3_TVE_PED, SPOT3 TV Encoder Pedestal Enable<br>[3] SP3_TVE_PDRST, SPOT3 TV Encoder Pedestal Reset<br>[2] SP3_TVE_PHALT, SPOT3 TV Encoder Phase Alternated<br>[1] SP3_TVE_F_INV, SPOT3 TV Encoder Field Inverted<br>[0] SP3_TVE_INTL, SPOT3 TV Encoder Interlace Select |
| 0xFA9 | R/W | 0x00 | SPOT3 TV Encoder Offset Register<br>[5:4] SP3_TVE_PIX_OS, SPOT3 TV Encoder Pixel Offset<br>[3:2] SP3_TVE_FLD_OS, SPOT3 TV Encoder field Offset<br>[1] SP3_TVE_FLD_POL, SPOT3 TV Encoder Field Polarity<br>[0] SP3_TVE_ FLDDET, SPOT3 Encoder Field Detect |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xFAA | R/W | 0x00 | SP3_TVE_HSENC[7:0], SPOT3 TV Encoder Active Pixel delay |
| 0xFAB | R/W | 0x00 | SP3_TVE_HSENC[9:8],<br>[7:2] SP3 Rotate Time Offset<br>[1:0] SP3 TV Encoder Active Pixel delay |
| 0xFAC | R/W | 0x80 | [7:6] SP3_CVBS_SRC_SEL<br>[5:0] SP3_TVE_VSENC[5:0], SPOT3 TV Encoder Active Line delay |
| 0xFAD | R/W | 0x00 | SP3_TVE_LINE_OS[4:0], SPOT3 TV Encoder Active Line offset |
| 0xFAE | R/W | 0x00 | [7;6] SP3_FLD_SW_MODE<br>[5:0] SP3_TVE_PIXEL_OS[5:0], SPOT3 TV Encoder Active Pixel offset |
| 0xFAF | R/W | 0x00 | SP3_TVE_FLD_OS[1:0], SPOT3 TV Encoder Active Field offset |
| 0xFB0 | R/W | 0x00 | SP3_TVE_FSC_FREE[7], SPOT3 TV Encoder Subcarrier run free<br>[6:0] RESERVED |
| 0xFB1 | R/W | 0x00 | SPOT3 OSD Enable Register<br>[4] SP3_OSD_TITLE_EN<br>[3] SP3_OSD_TIME_EN<br>[2] SP3_OSD_CHPIC_EN<br>[1] SP3_OSD_CHNUM_EN<br>[0] SP3_OSD_OSD_EN |
| 0xFB2 | R/W | 0x00 | SPOT3 OSD Control Register<br>[5] SP3_OSD_CHPIC_BLINK<br>[4] SP3_OSD_CHPIC_TRANS<br>[3] SP3_OSD_TITLE_MIX<br>[2] SP3_OSD_TIME_MIX<br>[1] SP3_OSD_CHPIC_MIX<br>[0] SP3_OSD_CHNUM_MIX |
| 0xFB3 | R/W | 0x00 | [7:6] SP3_OSD_CHPIC_POS<br>[5:4] SP3_OSD_CHNUM_POS<br>[3:2] SP3_OSD_FONT_VSIZE<br>[1:0] SP3_OSD_FONT_HSIZE |
| 0xFB4 | R/W | 0x00 | SP3_OSD_TIME_HPOS[7:0] |
| 0xFB5 | R/W | 0x00 | SP3_OSD_TIME_HPOS[10:8] |
| 0xFB6 | R/W | 0x00 | SP3_OSD_TIME_VPOS[7:0] |
| 0xFB7 | R/W | 0x00 | SP3_OSD_TIME_VPOS[10:8] |
| 0xFB8 | R/W | 0x00 | SP3_OSD_TITLE_HPOS[7:0] |
| 0xFB9 | R/W | 0x00 | SP3_OSD_TITLE_HPOS[10:8] |
| 0xFBA | R/W | 0x00 | SP3_OSD_TITLE_VPOS[7:0] |
| 0xFBB | R/W | 0x00 | SP3_OSD_TITLE_VPOS[10:8] |
| 0xFBC | R/W | 0x00 | SP3_OSD_FRAME_ADDR[7:0] |
| 0xFBD | R/W | 0x00 | SP3_OSD_FRAME_ADDR[10:8] |
| 0xFBE | R/W | 0x00 | SP3_OSD_FRAME_DATA[7:0] |
| 0xFBF | R/W | 0x00 | SP3_OSD_FRAME_DATA [15:8] |
| 0xFC0 | R/W | 0x00 | SP4_HTT[7:0], SPOT4 Horizontal Total Register |
| 0xFC1 | R/W | 0x00 | SP4_HTT[12:8], SPOT4 Horizontal Total Register |
| 0xFC2 | R/W | 0x00 | SP4_HDE[7:0], SPOT4 Horizontal Display Enable Register |
| 0xFC3 | R/W | 0x00 | SP4_HDE[11:8], SPOT4 Horizontal Display Enable Register |
| 0xFC4 | R/W | 0x00 | SP4_VDE[7:0], SPOT4 Vertical Display Enable Register |
| 0xFC5 | R/W | 0x00 | [7:4] BND_CH_EN[15:12] SPOT Border Channel Enable<br>[2:0] SP4_VDE[10:8], SPOT4 Vertical Display Enable Register |
| 0xFC6 | R/W | 0x00 | SP4_HSYNC[7:0], SPOT4 Horizontal Position Sync Register |
| 0xFC7 | R/W | 0x00 | [3:2] SP4_FRSC<br>[1] SP4_HBLNK[8], SPOT4 Horizontal Blank Position Register<br>[0] SP4_HSYNC[8], SPOT4 Horizontal Sync Position Register |
| 0xFC8 | R/W | 0x00 | SP4_VSYNC[7:0], SPOT4 Vertical Position Sync Register |
| 0xFC9 | R/W | 0x00 | SP4_VTT[7:0], SPOT4 VTT register |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xFCA | R/W | 0x00 | SP4_VTT[10:8], SPOT4 VTT Register<br>[7] SP4_Half_D1 Select<br>[2:0] SP4_VTT[10:8] |
| 0xFCB | R/W | 0x00 | [5] SP4_COLORBAR, SPOT4 Color Bar Enable Register<br>[4] SP4_BG_SW, SPOT4 Switch Background Enable Register<br>[3:2] SP4_FLD select<br>11: use bank0 from the FRSC as the field<br>10: use bank0 from the FRSC inverse as the filed<br>0x: use HVCNT generated field<br>[1] SP4_PD_EN<br>[0] SP3_PD_EN |
| 0xFCC | R/W | 0x00 | SP_USE_BY_REC_NETWORK_PORT<br>[7:4] Select REC ports 9-0 or network to replace SP4 as display port<br>[3:0] Select REC ports 9-0 or network to replace SP3 as display port |
| 0xFCD | R/W | 0x00 | SP4_DNS DOWN SCALE Register Upper Bits<br>[7:6] SP4 Destination Width upper bit [9:8]<br>[5:3] SP4 Source Height upper bit [10:8]<br>[2:0] SP4 Source Width upper bit [10:8] |
| 0xFCE | R/W | 0x00 | [7] SP4 REC or Network Port input enable<br>[6:5] SP4 Internal Color Bar test Select<br>[3:1] SP4 Line Buffer First line Control<br>[0] SP4 Destination Height upper bit [8] |
| 0xFCF | R/W | 0x00 | SPOT4 Enable Register<br>[7] SP4_FLD_MODE<br>[6] SP4_2ND_MEM<br>[5] SP4_FRMCNT_EN, SPOT4 Frame Count Enable<br>[4] SP4_ENA, SPOT4 Enable<br>[3:2] SP4_ROT_EN, SPOT4 Rotate Enable<br>[1] SP4_VS_POL, SPOT4 Vsync Polarity Select<br>[0] SP4_HS_POL, SPOT4 Hsync Polarity Select |
| 0xFD0 | R/W | 0x00 | SP4_HSTART [7:0], SPOT4 Horizontal Start Address Register |
| 0xFD1 | R/W | 0x00 | SP4_VSTART[7:0], SPOT4 Vertical Start Address Register |
| 0xFD2 | R/W | 0x00 | [7] SP4 Quad Mode Enable<br>[5:4] SP4_HSTART[9:8], SPOT4 Horizontal Start Address Register<br>[3:0] SP4_VSTART[11:8], SPOT4 Vertical Start Address Register |
| 0xFD3 | R/W | 0x00 | SP4_SRC_WID[7:0], SPOT4 Source width Register |
| 0xFD4 | R/W | 0x00 | SP4_SRC_Hi[7:0], SPOT4 Source height Register |
| 0xFD5 | R/W | 0x00 | SP4_DST_WID[7:0], SPOT4 Destination width Register |
| 0xFD6 | R/W | 0x00 | SP4_DST_Hi[7:0], SPOT4 Destination height Register |
| 0xFD7 | R/W | 0x00 | SPOT4 TV Encoder Standard Register<br>[7:6] SP4_TVE_FSC_SEL, SPOT4 TV Encoder Fsc Select<br>[5] SP4_TVE_SAT_LMT, SPOT4 TV Saturation Limit<br>[0] SP4_TVE_STD, SPOT4 TV Standard Select |
| 0xFD8 | R/W | 0x00 | SPOT4 TV Encoder Standard Register<br>[7] SP4_TVE_VSPOL, SPOT4 TV Encoder Vertical Sync Polarity<br>[6] SP4_TVE_HSPOL, SPOT4 TV Encoder Horiz Sync Polarity<br>[5] SP4_TVE_VSDET, SPOT4 TV Encoder Vsync detect<br>[4] SP4_TVE_PED, SPOT4 TV Encoder Pedestal Enable<br>[3] SP4_TVE_PDRST, SPOT4 TV Encoder Pedestal Reset<br>[2] SP4_TVE_PHALT, SPOT4 TV Encoder Phase Alternated<br>[1] SP4_TVE_F_INV, SPOT4 TV Encoder Field Inverted<br>[0] SP4_TVE_INTL, SPOT4 TV Encoder Interlace Select |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xFD9 | R/W | 0x00 | SPOT4 TV Encoder Offset Register<br>[5:4] SP4_TVE_PIX_OS, SPOT4 TV Encoder Pixel Offset<br>[3:2] SP4_TVE_FLD_OS, SPOT4 TV Encoder field Offset<br>[1] SP4_TVE_FLD_POL, SPOT4 TV Encoder Field Polarity<br>[0] SP4_TVE_ FLDDET, SPOT4 Encoder Field Detect |
| 0xFDA | R/W | 0x00 | SP4_TVE_HSENC[7:0], SPOT4 TV Encoder Active Pixel delay |
| 0xFDB | R/W | 0x00 | SP4_TVE_HSENC[9:8],<br>[7:2] SP4 Rotate Time Offset<br>[1:0] SP4 TV Encoder Active Pixel delay |
| 0xFDC | R/W | 0xC0 | [7:6] SP4_CVBS_SRC_SEL<br>[5:0] SP4_TVE_VSENC[5:0], SPOT4 TV Encoder Active Line delay |
| 0xFDD | R/W | 0x00 | SP4_TVE_LINE_OS[4:0], SPOT4 TV Encoder Active Line offset |
| 0xFDE | R/W | 0x00 | [7;6] SP4_FLD_SW_MODE<br>[5:0] SP4_TVE_PIXEL_OS[5:0], SPOT4 TV Encoder Active Pixel Offset |
| 0xFDF | R/W | 0x00 | SP4_TVE_FLD_OS[1:0], SPOT4 TV Encoder Active Field offset |
| 0xFE0 | R/W | 0x00 | SP4_TVE_FSC_FREE[7], SPOT4 TV Encoder Subcarrier run free |
| 0xFE1 | R/W | 0x00 | SPOT4 OSD Enable Register<br>[4] SP4_OSD_TITLE_EN<br>[3] SP4_OSD_TIME_EN<br>[2] SP4_OSD_CHPIC_EN<br>[1] SP4_OSD_CHNUM_EN<br>[0] SP4_OSD_OSD_EN |
| 0xFE2 | R/W | 0x00 | SPOT4 OSD Control Register<br>[5] SP4_OSD_CHPIC_BLINK<br>[4] SP4_OSD_CHPIC_TRANS<br>[3] SP4_OSD_TITLE_MIX<br>[2] SP4_OSD_TIME_MIX<br>[1] SP4_OSD_CHPIC_MIX<br>[0] SP4_OSD_CHNUM_MIX |
| 0xFE3 | R/W | 0x00 | [7:6] SP4_OSD_CHPIC_POS<br>[5:4] SP4_OSD_CHNUM_POS<br>[3:2] SP4_OSD_FONT_VSIZE<br>[1:0] SP4_OSD_FONT_HSIZE |
| 0xFE4 | R/W | 0x00 | SP4_OSD_TIME_HPOS[7:0] |
| 0xFE5 | R/W | 0x00 | SP4_OSD_TIME_HPOS[10:8] |
| 0xFE6 | R/W | 0x00 | SP4_OSD_TIME_VPOS[7:0] |
| 0xFE7 | R/W | 0x00 | SP4_OSD_TIME_VPOS[10:8] |
| 0xFE8 | R/W | 0x00 | SP4_OSD_TITLE_HPOS[7:0] |
| 0xFE9 | R/W | 0x00 | SP4_OSD_TITLE_HPOS[10:8] |
| 0xFEA | R/W | 0x00 | SP4_OSD_TITLE_VPOS[7:0] |
| 0xFEB | R/W | 0x00 | SP4_OSD_TITLE_VPOS[10:8] |
| 0xFEC | R/W | 0x00 | SP4_OSD_FRAME_ADDR[7:0] |
| 0xFED | R/W | 0x00 | SP4_OSD_FRAME_ADDR[10:8] |
| 0xFEE | R/W | 0x00 | SP4_OSD_FRAME_DATA[7:0] |
| 0xFEF | R/W | 0x00 | SP4_OSD_FRAME_DATA [15:8] |
| 0XFF0 | R/W | 0x00 | SPOT Channel0 Mapping Register<br>[7] CH0_WR_EN<br>[6] CH0_VSIZE_SEL<br>[5:4] CH0_VPOS_SEL<br>[3] CH0_HSIZE_SEL<br>[2:0] CH0_HPOS_SEL |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0XFF1 | R/W | 0x00 | SPOT Channel1 Mapping Register<br>[7] CH1_WR_EN<br>[6] CH1_VSIZE_SEL<br>[5:4] CH1_VPOS_SEL<br>[3] CH1_HSIZE_SEL<br>[2:0] CH1_HPOS_SEL |
| 0XFF2 | R/W | 0x00 | SPOT Channel2 Mapping Register<br>[7] CH2_WR_EN<br>[6] CH2_VSIZE_SEL<br>[5:4] CH2_VPOS_SEL<br>[3] CH2_HSIZE_SEL<br>[2:0] CH2_HPOS_SEL |
| 0XFF3 | R/W | 0x00 | SPOT Channel3 Mapping Register<br>[7] CH3_WR_EN<br>[6] CH3_VSIZE_SEL<br>[5:4] CH3_VPOS_SEL<br>[3] CH3_HSIZE_SEL<br>[2:0] CH3_HPOS_SEL |
| 0XFF4 | R/W | 0x00 | SPOT Channel4 Mapping Register<br>[7] CH4_WR_EN<br>[6] CH4_VSIZE_SEL<br>[5:4] CH4_VPOS_SEL<br>[3] CH4_HSIZE_SEL<br>[2:0] CH4_HPOS_SEL |
| 0XFF5 | R/W | 0x00 | SPOT Channel5 Mapping Register<br>[7] CH5_WR_EN<br>[6] CH5_VSIZE_SEL<br>[5:4] CH5_VPOS_SEL<br>[3] CH5_HSIZE_SEL<br>[2:0] CH5_HPOS_SEL |
| 0XFF6 | R/W | 0x00 | SPOT Channel6 Mapping Register<br>[7] CH6_WR_EN<br>[6] CH6_VSIZE_SEL<br>[5:4] CH6_VPOS_SEL<br>[3] CH6_HSIZE_SEL<br>[2:0] CH6_HPOS_SEL |
| 0XFF7 | R/W | 0x00 | SPOT Channel7 Mapping Register<br>[7] CH7_WR_EN<br>[6] CH7_VSIZE_SEL<br>[5:4] CH7_VPOS_SEL<br>[3] CH7_HSIZE_SEL<br>[2:0] CH7_HPOS_SEL |
| 0XFF8 | R/W | 0x00 | SPOT Channel8 Mapping Register<br>[7] CH8_WR_EN<br>[6] CH8_VSIZE_SEL<br>[5:4] CH8_VPOS_SEL<br>[3] CH8_HSIZE_SEL<br>[2:0] CH8_HPOS_SEL |
| 0XFF9 | R/W | 0x00 | SPOT Channel9 Mapping Register<br>[7] CH9_WR_EN<br>[6] CH9_VSIZE_SEL<br>[5:4] CH9_VPOS_SEL<br>[3] CH9_HSIZE_SEL<br>[2:0] CH9_HPOS_SEL |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0XFFA | R/W | 0x00 | SPOT Channel10 Mapping Register<br>[7] CH10_WR_EN<br>[6] CH10_VSIZE_SEL<br>[5:4] CH10_VPOS_SEL<br>[3] CH10_HSIZE_SEL<br>[2:0] CH10_HPOS_SEL |
| 0XFFB | R/W | 0x00 | SPOT Channel11 Mapping Register<br>[7] CH11_WR_EN<br>[6] CH11_VSIZE_SEL<br>[5:4] CH11_VPOS_SEL<br>[3] CH11_HSIZE_SEL<br>[2:0] CH11_HPOS_SEL |
| 0XFFC | R/W | 0x00 | SPOT Channel12 Mapping Register<br>[7] CH12_WR_EN<br>[6] CH12_VSIZE_SEL<br>[5:4] CH12_VPOS_SEL<br>[3] CH12_HSIZE_SEL<br>[2:0] CH12_HPOS_SEL |
| 0XFFD | R/W | 0x00 | SPOT Channel13 Mapping Register<br>[7] CH13_WR_EN<br>[6] CH13_VSIZE_SEL<br>[5:4] CH13_VPOS_SEL<br>[3] CH13_HSIZE_SEL<br>[2:0] CH13_HPOS_SEL |
| 0XFFE | R/W | 0x00 | SPOT Channel14 Mapping Register<br>[7] CH14_WR_EN<br>[6] CH14_VSIZE_SEL<br>[5:4] CH14_VPOS_SEL<br>[3] CH14_HSIZE_SEL<br>[2:0] CH14_HPOS_SEL |
| 0XFFF | R/W | 0x00 | SPOT Channel15 Mapping Register<br>[7] CH15_WR_EN<br>[6] CH15_VSIZE_SEL<br>[5:4] CH15_VPOS_SEL<br>[3] CH15_HSIZE_SEL<br>[2:0] CH15_HPOS_SEL |

# Registers

### SPOT1 CONTROL REGISTER– 0XF00

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **SP1_HTT [7:0], SPOT1 Horizontal Total Register**<br><br>Lower 8 bits of the SP1_HTT |

### SPOT1 CONTROL REGISTER– 0XF01

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4:0 | RW | 0 | **SP1_HTT [12:8], SPOT1 Horizontal Total Register**<br><br>Upper 5 bits of the SP1_HTT |
| 7:5 | N/A | | Reserved |

### SPOT1 CONTROL REGISTER– 0XF02

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **SP1_HDE [7:0], SPOT1 Horizontal Display Enable Register**<br><br>Lower 8 bits of the HDE register |

### SPOT1 CONTROL REGISTER– 0XF03

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | N/A | | Reserved |
| 3:0 | RW | 0 | **SP1_HDE [11:8], SPOT1 Horizontal Display Enable Register**<br><br>Upper 4 bits of the SP1_HDE |

### SPOT1 CONTROL REGISTER– 0XF04

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **SP1_VDE [7:0], SPOT1 Vertical Display Enable Register**<br><br>Lower 8 bits of the VDE register |

### SPOT1 CONTROL REGISTER– 0XF05

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | N/A | 0 | **BND_CH_EN[3:0] Border Channel Enable** |
| 3 | N/A | 0 | Reserved |
| 2:0 | RW | 0 | **SP1_VDE [10:8], SPOT1 Vertical Display Enable Register**<br><br>Upper 3 bits of the Vertical Display Enable Register |

### SPOT1 CONTROL REGISTER– 0XF06

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **SP1_HSYNC[7:0], SPOT1 Horizontal Sync register** |

### SPOT1 CONTROL REGISTER– 0XF07

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0 | **BND_WIDTH_V Vertical Border Width in Field Switching Mode**<br><br>[7:4] SP Vertical Border Width Select |
| 3:2 | RW | 0 | **SP1 FRSC Control**<br>11: bank_sel equals bank_selB<br>10: bank_sel equals {vblank, fld}<br>01: bank_sel equals {bank_selA[1], fld}<br>00: bank_sel equals bank_selA<br><br>11: wr_page equals wr_pageB<br>10: wr_page equals {~vblank, 0}<br>01: wr_page equals {!bank_selA[1], fld}<br>00: wr_page equals wr_pageA |
| 1 | RW | 0 | **SPOT FRSC No Video Control**<br><br>Enable to detect no in coming video to reset FRSC bank select |
| 0 | RW | 0 | **SP1 HSYNC[8], SPOT1Horizontal Position Sync Register**<br><br>Upper 1 bit of the SP1 HSYNC |

### SPOT1 CONTROL REGISTER– 0XF08

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | x44 | **SP1_PVSYNC [7:0], SPOT1 Vertical Position Sync Register**<br><br>Lower 8 bits of the SP1_PVSYNC<br>The register value determine when the horizontal sync start when the horizontal display counter reach to the sum of SP1_VDE and SP1_PVSYNC |

### SPOT1 CONTROL REGISTER– 0XF09

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **SP1_VTT [7:0], SPOT1 VTT Register**<br><br>Lower 8 bits of the SP1_ VTT register |

### SPOT1 CONTROL REGISTER– 0XF0A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | [7] **SP1_Half_D1 Select** |
| 6:3 | RW | 0 | Reserve |
| 2:0 | RW | 0 | [2:0] **SP1_VTT**[10:8] |

### SPOT1 CONTROL REGISTER– 0XF0B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | SP1 VBLANK_SEL<br><br>[7:6] = 11, Select sp4 vlank as the source to the FRSC<br>= 10, Select sp3 vlank as the source to the FRSC<br>= 01, Select sp2 vlank as the source to the FRSC<br>= 00, Select sp1 vlank as the source to the FRSC |
| 5 | RW | 0 | SP1 COLOR BAR ENA |
| 4 | RW | 0 | SP1_SWITCH_EN<br><br>0: Background enable<br>1: Background display disable |
| 3:2 | RW | 0 | SP1_FLD select<br><br>11: use bank0 from the FRSC as the field<br>10: use bank0 from the FRSC inverse as the filed<br>0x: use HVCNT generated field |
| 1:0 | RW | 0 | Reserved |

### SPOT1 CONTROL REGISTER– 0XF0C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_ROT_INT_EN<br><br>[7] SP4 Rot Int status<br>[6] SP3 Rot Int status<br>[5] SP2 Rot Int status<br>[4] SP1 Rot Int status<br><br>[3] SP4 Rot Int Enable<br>[2] SP3 Rot Int Enable<br>[1] SP2 Rot Int Enable<br>[0] SP1 Rot Int Enable |

### SPOT1 CONTROL REGISTER – 0XF0D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_DNS Upper Bits<br><br>[7:6] SP1 Destination Width upper bit [9:8]<br>[5:3] SP1 Source Height upper bit [10:8]<br>[2:0] SP1 Source Width upper bit [10:8] |

### SPOT1 CONTROL REGISTER – 0XF0E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | SP1 REC or Network port Enable<br>Enable the REC or Network port to use SP1 as display |
| 6:4 | RW | 0 | SP1 Color Bar Select |
| 3:1 | RW | 0 | SP1_Firsrt_Line_Control |
| 0 | RW | 0 | SP1 Destination Height Upper Bit [8] |

**SPOT1 CONTROL REGISTER – 0XF0F**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | N/A | 1 | SP1_Field Mode Select<br>1: Field mode<br>0: Frame mode |
| 6 | RW | 0 | SP1_2ND_MEM Enable |
| 5 | RW | 0 | SP1_FRMCNT_EN SPOT1 Frame Count Enable |
| 4 | RW | 0 | SP1_ENA SPOT1 Enable |
| 3:2 | RW | 0 | SP1_ROT Rotation Enable |
| 1 | RW | 0 | SP1_VSPOL SPOT1 Vertical Sync Polarity Control |
| 0 | RW | 0 | SP1_HSPOL SPOT1 Horizontal Sync Polarity Control |

**SPOT1 CONTROL REGISTER – 0XF10**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP1_HSTART [7:0], SPOT1 Horizontal Start Register<br><br>This register determines the lower 8 bits of SPOT1 Horizontal start address |

**SPOT1 CONTROL REGISTER – 0XF11**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 01 | SP1_VSTART [7:0], SPOT1 Vertical Start Register<br><br>This register determines the lower 8 bits of SPOT1 Vertical start address |

**SPOT1 CONTROL REGISTER– 0XF12**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | N/A | 0 | SP1 Quad Mode Enable<br><br>When Enable, the SP1 memory controller will read half of each line data and start a new a new address calculation from second half line as well as half field for the vertical address to prevent whole screen jumping when weak video is presence.<br><br>This bit should set to one when screen image is set to quad mode. |
| 6 | N/A | | Reserved |
| 5:4 | RW | 0 | SP1_HSTART [9:8], SPOT1 Horizontal Start Register<br><br>Upper 2 bits of the SPOT1 Horizontal start register in 4 pixels / unit |
| 3:0 | RW | 0 | SP1_VSTART [11:8], SPOT1 Vertical Start Register<br><br>Upper 4 bits of the SPOT1 Vertical start register in 1 line / unit |

### SPOT1 CONTROL REGISTER – 0XF13

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP1_SRC_WID [7:0], SPOT1 Down Scale Source Width Register<br><br>This register determines the source width of the image, The upper 3 bits are in register 0xF0D. |

### SPOT1 CONTROL REGISTER– 0XF14

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP1_SRC_HI [7:0], SPOT1 Down Scale Source Height Register<br><br>This register determines the source height of the image. The upper 3 bits are in register 0xF0D. |

### SPOT1 CONTROL REGISTER – 0XF15

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP1_DST_WID [7:0], SPOT1 Down Scale Destination Width Register<br><br>This register determines the source width of the image. The upper 2 bits are in register 0xF0D. |

### SPOT1 CONTROL REGISTER– 0XF16

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP1_DST_HI [7:0], SPOT1 Down Scale Destination Height Register<br><br>This register determines the source height of the image. The upper 1 bit is in register 0xF0E. |

### SPOT1 CONTROL REGISTER– 0XF17

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | TVENC_FSC_MODE, SPOT1 TV FSCSEL<br><br>This register determines the TV modulation frequency standard select.<br><br>00 - NTSC M<br>01 - PAL B, D, G, H, I<br>10 - PAL M<br>11 - PAL N |
| 5 | RW | 0 | TVENC_SAT_LMT, SPOT1 TV video saturation limit<br><br>Control the data range of the ITU-R BT 656 output<br><br>0 - Not limit<br>1 – Data range is limited to 1 – 254 range |
| 4:1 | RW | 0 | Reserved |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | RW | 0 | **TVENC_PALNT, SPOT1 TV PAL or NTSC select**<br><br>This register determines the CVBS output Standard.<br><br>0: NSTC mode<br>1: PAL mode. |

**SPOT1 CONTROL REGISTER– 0XF18**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **SP1_TVENC_VSPOL, SPOT1 TVENC Vertical sync polarity**<br><br>0: Active Low<br>1: Active High |
| 6 | RW | 0 | **SP1_TVENC_HSPOL, SPOT1 TV Encoder Horizontal Sync Polarity**<br><br>0: Active Low<br>1: Active High |
| 5 | RW | 0 | **SP1_TVENC_VSDET, SPOT1 TV Encoder Vsync detain select**<br><br>**This bit is used by TV encoder as an option to generate an reset by using vsync or field signals.**<br><br>**If this bit is 1, the reset sync generated from field signal.**<br>**If this bit is 0, the reset sync generated from the vsync signal.** |
| 4 | RW | 0 | **SP1_TVENC_PED, SPOT1 TV Encoder Pedestal Enable**<br><br>Enable Pedestal for video output<br><br>0: 0 IRE for pedestal<br>1: 7.5 IRE pedestal level |
| 3 | RW | 0 | **SP1_TVENC_PDRST, SPOT1 TV Encoder Pedestal Reset**<br><br>Subcarrier Phase Alternation reset every 8 field in PAL system<br><br>0: Constant relationship, reset every 8 field<br>1: Free running |
| 2 | RW | 0 | **SP1_TVENC_PHALT, SPOT1 TV Encoder Phase Alternation**<br><br>0: Disable phase alternation for line by line. Use in NTSC<br>1: Enable phase alternation for line by line. Use in PAL |
| 1 | RW | 0 | **SP1_TVENC_F_INV, SPOT1 TV Encoder Field Output Invert**<br><br>This register bit inverts the field output level if set. |
| 0 | RW | 0 | **SP1_TVENC_INTERLACE, SPOT1 TVENC Interlace Select**<br><br>This register bit determines the TV Standard in Interlace timing mode.<br><br>0 - Progressive<br>1 - Interlace |

**SPOT1 CONTROL REGISTER– 0XF19**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | N/A | | Reserved |
| 5:4 | RW | 0 | SP1_TVENC_YC_OS[1:0], SPOT1 TV Encoder Active Pixel Offset delay<br><br>00: No Active offset<br>01: Active pixel offset for 1/4 pixel unit<br>10: Active pixel offset for 1/2 pixel unit<br>11: Active pixel offset for 3/4 pixel unit |
| 3:2 | RW | 0 | SP1_TVENC_OSENC[1:0], SPOT1 TV Encoder Field Offset for the first video line<br><br>00: Apply same ENC_VDSEL for odd and even field<br>01: Apply same ENC_VDSEL for odd and odd field<br>10: Apply TVENC_VSDEL for odd field and TVENC_VSDEL+1 for even field<br>11: Apply TVENC_VSDEL for odd field and TVENC_VSDEL+2 for even field |
| 1 | RW | 0 | SP1_TVENC_FLDENCP, SPOT1 TV Encoder Field Polarity<br><br>This Register control the field polarity<br>0: Even field is high<br>1: Odd field is high |
| 0 | RW | 0 | SP1_TVENC_FLDDET, SPOT1 TV Encoder Field Detect |

**SPOT1 CONTROL REGISTER– 0XF1A**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_TVENC_HSENC[7:0], SPOT1 TV Encoder Active pixel delay<br><br>Lower 8 bits of the pixel delay from active video by ½ pixel per step |

**SPOT1 CONTROL REGISTER– 0XF1B**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | N/A | 0 | SP1_Rotate_Time_Offset |
| 1:0 | RW | 0 | SP1_TVENC_HSENC[9:8], SPOT1 TV Encoder Active pixel delay<br><br>Upper 2 bits of the pixel delay from active video by ½ pixel per step |

**SPOT1 CONTROL REGISTER– 0XF1C**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | N/A | 0 | SP1_CVBS_SEL<br><br>11 : SP1 CVBS output is coming from SP4<br>10 : SP1 CVBS output is coming from SP3<br>01 : SP1 CVBS output is coming from SP2<br>00 : SP1 CVBS output is coming from SP1 |
| 5:0 | RW | 0 | SP1_TVENC_VSENC, SPOT1 TV Encoder Active line delay<br><br>6 bits of the line delay from active video by line per step |

**SPOT1 CONTROL REGISTER– 0XF1D**

| Bit | R/W | Default | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 7:6 | N/A | | Reserved |
| 4:0 | RW | 0 | SP1_TVENC_LINE_OS, SPOT1 TV Encoder Active line offset<br><br>5 bits of the line offset |

**SPOT1 CONTROL REGISTER– 0XF1E**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | N/A | 0 | [7;6] SP1_FLD_SW_MODE<br><br>[7] Field Switching Mode enable<br>[6] Field Switch Mode image height is 240 lines Enable |
| 5:0 | RW | 0 | SP1_TVENC_PIXEL_OS, SPOT1 TV Encoder Active Pixel offset<br><br>6 bits of the pixel offset |

**SPOT1 CONTROL REGISTER– 0XF1F**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | N/A | | Reserved |
| 6 | RW | 0 | SP1_TVENC_C_OFF_D, SPOT1 TV Encoder C Off<br><br>Set the Color Off if set |
| 5:4 | RW | 0 | SP1_TVENC_CBW_D, SPOT1 TV Encoder CBCR delay<br><br>2 bits of the CbCr delay |
| 3:2 | RW | 0 | SP1_TVENC_YBW_D, SPOT1 TV Encoder Luma delay<br><br>2 bits of the Luminous delay |
| 1:0 | RW | 0 | SP1_TVENC_FLD_OS, SPOT1 TV Encoder Active Field offset<br><br>2 bits of the field offset |

**SPOT1 CONTROL REGISTER– 0XF20**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | SP1_TVENC_TST_FSC_FREE, SPOT1 TV Encoder Sub-Carrier is<br><br>set to free run |
| 6 | RW | 0 | SP1_TVENC_T_CCIR_TIM, SPOT1 TV Encoder CCOR Timing<br><br>2 bits of the CbCr delay |
| 5 | RW | 0 | SP1_TVENC_T_656_STD, SPOT1 TV Encoder 656 Standard<br><br>2 bits of the Luminous delay |
| 4:0 | RW | 0 | SP1_TVENC_VIS_LINE_OS, SPOT1 TV Encoder line offset<br><br>5 bits Line offset |

**SPOT CONTROL REGISTER– 0XF21**

| Bit | R/W | Default | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 7:0 | N/A | 0 | Vertical Start Address offset [7:0] in 256 or 512 Mb memory Config.<br><br>In 128Mb config, the SPOT can only read 4 D1 images written in the memory, when the Record memory size increased to 256 Mb, SPOT write can store 4 more D1 images in SDRAM, to read these images, we need to set the $SP_{1,2,3,4}$ VSTART address together with this address offset registers to match the SPOT write address to get the correct images.<br><br>To use this features, $SP_x$ 2ND_MEM bit need to set to 1. |

**SPOT CONTROL REGISTER– 0XF22**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | BACKGROUNG Pr, SPOT Background Color Red Color<br><br>8 bits Red color value |

**SPOT CONTROL REGISTER– 0XF23**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | BACKGROUNG Y, SPOT Background Color Green Color<br><br>8 bits Green color value |

**SPOT CONTROL REGISTER– 0XF24**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | BACKGROUNG Pb, SPOT Background Color Blue Color<br><br>8 bits Blue color value |

**SPOT CONTROL REGISTER– 0XF25**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | SWITCH_EN, For test purpose |
| 6 | RW | 0 | COLBAR_EN, For test purpose |
| 5 | RW | 0 | Reserve |
| 4 | RW | 0 | BND_EN[0], SPOT Border Width Enable bit 0 |
| 3:0 | RW | 0 | BORDER_WIDTH, SPOT Border Width Select<br><br>4 bits Border Width Select |

**SPOT CONTROL REGISTER– 0XF26**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | BND_Pr, SPOT Border Red Color<br><br>8 bits Red color value |

**SPOT CONTROL REGISTER– 0XF27**

| Bit | R/W | Default | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 7:0 | RW | 0 | BND_Y, SPOT Border Green Color<br><br>8 bits Green color value |

**SPOT CONTROL REGISTER – 0XF28**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | BND_Pb, SPOT Border Blue Color<br><br>8 bits Blue color value |

**SPOT CONTROL REGISTER – 0XF29**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | ROT_FREQ, SPOT Channel Rotate Frequency<br><br>8 bits Rotate Frequency Control in field / unit |

**SPOT2 CONTROL REGISTER – 0XF2A**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP2_OSD_DRAM_ADDR |

**SPOT2 CONTROL REGISTER – 0XF2B**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | | Reserve |
| 5:0 | RW | 0 | SP2_OSD_DRAM_DATA |

**SPOT3 CONTROL REGISTER – 0XF2C**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP3_OSD_DRAM_ADDR |

**SPOT3 CONTROL REGISTER – 0XF2D**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | | Reserve |
| 5:0 | RW | 0 | SP3_OSD_DRAM_DATA |

**SPOT4 CONTROL REGISTER – 0XF2E**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_OSD_DRAM_ADDR |

### SPOT4 CONTROL REGISTER – 0XF2F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | | Reserve |
| 5:0 | RW | 0 | SP4_OSD_DRAM_DATA |

### SPOT1 CONTROL REGISTER – 0XF30

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | RW | 0 | BND_EN[3:1] SPOT Border Enable for each channel<br><br>Display Border enable for channel 4, channel 3, and channel 2<br>[7] : channel 4<br>[6] : channel 3<br>[5] : channel 2<br><br>1: enable<br>0: disable |
| 4 | RW | 0 | SP1_OSD_TITLE_EN<br><br>Display title enable<br>1: enable<br>0: disable |
| 3 | RW | 0 | SP1_OSD_TIME_EN<br><br>Display time and date enable<br>1: enable<br>0: disable |
| 2 | RW | 0 | SP1_OSD_CHPIC_EN<br><br>Display picture for each channel<br>1: enable<br>0: disable |
| 1 | RW | 0 | SP1_OSD_CHNUM_EN<br><br>Display channel number for each channel<br>1: enable<br>0: disable |
| 0 | RW | 0 | SP1_OSD_EN<br><br>All OSD information enable or disable. If set to 1, each information is enabled by bit [4:1]. If set to 0, all OSD is disabled.<br>1: enable<br>0: disable |

**SPOT1 CONTROL REGISTER – 0XF31**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | Reserved |
| 5 | RW | 0 | SP1_OSD_CHPIC_BLINK<br><br>If set to 1, picture for each channel blink in picture index 2 and 3. Picture 0 and 1 will not blink<br>1: enable<br>0: disable |
| 4 | RW | 0 | SP1_OSD_CHPIC_TRANS<br><br>If set to 1, picture in color index “00” will be transparent<br>1: enable<br>0: disable |
| 3 | RW | 0 | SP1_OSD_TITLE_MIX<br><br>Display title mix enable bit. If set to 1, title will be 50% blending with video<br>1: enable<br>0: disable |
| 2 | RW | 0 | SP1_OSD_TIME_MIX<br><br>Display time and date mix enable bit. If set to 1, time/date will be 50% blending with video<br>1: enable<br>0: disable |
| 1 | RW | 0 | SP1_OSD_CHPIC_MIX<br><br>Display picture mix enable bit. If set to 1, picture will be 50% blending with video<br>1: enable<br>0: disable |
| 0 | RW | 0 | SP1_OSD_CHNUM_MIX<br><br>Display channel number information mix enable bit. If set to 1, channel information will be 50% blending with video<br>1: enable<br>0: disable |

### SPOT1 OSD FONT SIZE REGISTER – 0XF32

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0x1 | SP1_OSD_CHPIC_POS<br><br>Channel picture corner position.<br>00: left top<br>01: right top<br>10: left bottom<br>11: right bottom |
| 5:4 | RW | 0 | SP1_OSD_CHNUM_POS<br><br>Channel number corner position.<br>00: left top<br>01: right top<br>10: left bottom<br>11: right bottom |
| 3:2 | RW | 0x3 | SP1_OSD_FONT_VSIZE<br><br>Font vertical size for display. There are four choices.<br>00: 8<br>01: 10<br>10: 16, scale up from 8<br>11: 20, scale up from 10 |
| 1:0 | RW | 0x3 | SP1_OSD_FONT_HSIZE<br><br>Font horizontal size for display. There are four choices.<br>00: 6<br>01: 8<br>10: 12, scale up from 6<br>11: 16, scale up from 8 |

### SPOT1 OSD CHANNEL NUMBER HORIZONTAL POSITION LOW BYTE REGISTER – 0XF33

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_OSD_CHNUM_HPOS[7:0]<br><br>Channel number information horizontal position offset to each channel horizontal start position. It is one pixel unit. |

### SPOT1 OSD CHANNEL NUMBER HORIZONTAL POSITION HIGH BYTE REGISTER – 0XF34

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP1_OSD_CHNUM_HPOS[10:8]<br><br>Channel number information horizontal position offset to each channel horizontal start position. It is one pixel unit. |

### SPOT1 OSD CHANNEL NUMBER VERTICAL POSITION LOW BYTE REGISTER – 0XF35

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_OSD_CHNUM_VPOS[7:0]<br><br>Channel number information vertical position offset to each channel vertical start position. It is one pixel unit. |

### SPOT1 OSD CHANNEL NUMBER HORIZONTAL POSITION HIGH BYTE REGISTER – 0XF36

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | RW | 0 | SP1_FRC_CTL[4:0]<br><br>[4] Auto Correction Jumps two field when set to 1.<br>One field if this bit set to 0.<br>[3] Enable Correction<br>[2:0] Select one of quad spot wr pages as reference<br>0: Select spot wr page 0,1,2,3<br>1: Select spot wr page 4,5,6,7<br>2: Select spot wr page 7,8,9,10<br>3: Select spot wr page 11,12,13,14<br>4-7: use quad position to select as reference |
| 2:0 | RW | 0 | SP1_OSD_CHNUM_VPOS[10:8]<br><br>Channel number information vertical position offset to each channel vertical start position. It is one pixel unit. |

### SPOT1 OSD CHANNEL PICTURE HORIZONTAL POSITION LOW BYTE REGISTER – 0XF37

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_OSD_CHPIC_HPOS[7:0]<br><br>Channel picture information horizontal position offset to each channel horizontal start position. It is one pixel unit. |

### SPOT1 OSD CHANNEL PICTURE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XF38

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | RW | 0 | SP2_FRC_CTL[4:0]<br><br>[4] Auto Correction Jumps two field when set to 1.<br>One field if this bit set to 0.<br>[3] Enable Correction<br>[2:0] Select one of quad spot wr pages as reference<br>0: Select spot wr page 0,1,2,3<br>1: Select spot wr page 4,5,6,7<br>2: Select spot wr page 7,8,9,10<br>3: Select spot wr page 11,12,13,14<br>4-7: use quad position to select as reference |
| 2:0 | RW | 0 | SP1_OSD_CHPIC_HPOS[10:8]<br><br>Channel picture information horizontal position offset to each channel horizontal start position. It is one pixel unit. |

### SPOT1 OSD CHANNEL PICTURE VERTICAL POSITION LOW BYTE REGISTER – 0XF39

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_OSD_CHPIC_VPOS[7:0]<br><br>Channel picture information vertical position offset to each channel vertical start position. It is one pixel unit. |

### SPOT1 OSD CHANNEL PICTURE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XF3A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | RW | 0 | SP3_FRC_CTL[4:0]<br><br>[4] Auto Correction Jumps two field when set to 1.<br>One field if this bit set to 0.<br>[3] Enable Correction<br>[2:0] Select one of quad spot wr pages as reference<br>0: Select spot wr page 0,1,2,3<br>1: Select spot wr page 4,5,6,7<br>2: Select spot wr page 7,8,9,10<br>3: Select spot wr page 11,12,13,14<br>4-7: use quad position to select as reference |
| 2:0 | RW | 0 | SP1_OSD_CHPIC_VPOS[10:8]<br><br>Channel number information vertical position offset to each channel vertical start position. It is one pixel unit. |

### SPOT1 OSD TIME AND DATE HORIZONTAL POSITION LOW BYTE REGISTER – 0XF3B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_OSD_TIME_HPOS[7:0]<br><br>Time and Date information horizontal position. It is one pixel unit. |

### SPOT1 OSD TIME AND DATE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XF3C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | RW | 0 | SP4_FRC_CTL[4:0]<br><br>[4] Auto Correction Jumps two field when set to 1.<br>One field if this bit set to 0.<br>[3] Enable Correction<br>[2:0] Select one of quad spot wr pages as reference<br>0: Select spot wr page 0,1,2,3<br>1: Select spot wr page 4,5,6,7<br>2: Select spot wr page 7,8,9,10<br>3: Select spot wr page 11,12,13,14<br>4-7: use quad position to select as reference |
| 2:0 | RW | 0 | SP1_OSD_TIME_HPOS[10:8]<br><br>Time and Date information horizontal position. It is one pixel unit. |

### SPOT1 OSD TIME AND DATE VERTICAL POSITION LOW BYTE REGISTER – 0XF3D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_OSD_TIME_VPOS[7:0]<br><br>Time and Date information vertical position. It is one pixel unit. |

### SPOT1 OSD TIME AND DATE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XF3E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP1_OSD_TIME_VPOS[10:8]<br><br>Time and Date information vertical position. It is one pixel unit. |

### SPOT1 OSD TITLE HORIZONTAL POSITION LOW BYTE REGISTER – 0XF3F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_OSD_TITLE_HPOS[7:0]<br><br>Title information horizontal position. It is one pixel unit. |

### SPOT1 OSD TITLE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XF40

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP1_OSD_TITLE_HPOS[10:8]<br><br>Title information horizontal position. It is one pixel unit. |

### SPOT1 OSD TITLE VERTICAL POSITION LOW BYTE REGISTER – 0XF41

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_OSD_TITLE_VPOS[7:0]<br><br>Title information vertical position. It is one pixel unit. |

### SPOT1 OSD TITLE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XF42

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP1_OSD_TITLE_VPOS[10:8]<br><br>Title information vertical position. It is one pixel unit. |

### SPOT OSD FONT RED COLOR 1 REGISTER – 0XF43

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_FONT_PR1[7:0]<br><br>Font red color for color index 1. |

### SPOT OSD FONT GREEN COLOR 1 REGISTER – 0XF44

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_FONT_Y1[7:0]<br><br>Font green color for color index 1. |

### SPOT OSD FONT BLUE COLOR 1 REGISTER – 0XF45

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_FONT_PB1[7:0]<br><br>Font blue color for color index 1. |

### SPOT OSD FONT RED COLOR 2 REGISTER – 0XF46

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_FONT_PR2[7:0]<br><br>Font red color for color index 2. |

### SPOT OSD FONT GREEN COLOR 2 REGISTER – 0XF47

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_FONT_Y2[7:0]<br><br>Font green color for color index 2. |

### SPOT OSD FONT BLUE COLOR 2 REGISTER – 0XF48

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_FONT_PB2[7:0]<br><br>Font blue color for color index 2. |

### SPOT OSD FONT RED COLOR 3 REGISTER – 0XF49

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_FONT_PR3[7:0]<br><br>Font red color for color index 3. |

### SPOT OSD FONT GREEN COLOR 3 REGISTER – 0XF4A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_FONT_Y3[7:0]<br><br>Font green color for color index 3. |

### SPOT OSD FONT BLUE COLOR 3 REGISTER – 0XF4B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_FONT_PB3[7:0]<br><br>Font blue color for color index 3. |

### SPOT OSD PICTURE RED COLOR 0 REGISTER – 0XF4C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_PIC_PR0[7:0]<br><br>Picture red color for color index 0. |

### SPOT OSD PICTURE GREEN COLOR 0 REGISTER – 0XF4D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_PIC_Y0[7:0]<br><br>Picture green color for color index 0. |

### SPOT OSD PICTURE BLUE COLOR 0 REGISTER – 0XF4E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_PIC_PB0[7:0]<br><br>Picture blue color for color index 0. |

### SPOT OSD PICTURE RED COLOR 1 REGISTER – 0XF4F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_PIC_PR1[7:0]<br><br>Picture red color for color index 1. |

### SPOT OSD PICTURE GREEN COLOR 1 REGISTER – 0XF50

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_PIC_Y1[7:0]<br><br>Picture green color for color index 1. |

### SPOT OSD PICTURE BLUE COLOR 1 REGISTER – 0XF51

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_PIC_PB1[7:0]<br><br>Picture blue color for color index 1. |

### SPOT OSD PICTURE RED COLOR 2 REGISTER – 0XF52

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_PIC_PR2[7:0]<br><br>Picture red color for color index 2. |

### SPOT OSD PICTURE GREEN COLOR 2 REGISTER – 0XF53

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_PIC_Y2[7:0]<br><br>Picture green color for color index 2. |

### SPOT OSD PICTURE BLUE COLOR 2 REGISTER – 0XF54

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_PIC_PB2[7:0]<br><br>Picture blue color for color index 2. |

### SPOT OSD PICTURE RED COLOR 3 REGISTER – 0XF55

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_PIC_PR3[7:0]<br><br>Picture red color for color index 3. |

### SPOT OSD PICTURE GREEN COLOR 3 REGISTER – 0XF56

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP_OSD_PIC_Y3[7:0]<br><br>Picture green color for color index 3. |

### SPOT OSD PICTURE BLUE COLOR 3 REGISTER – 0XF57

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_OSD_PIC_PB3[7:0]<br><br>Picture blue color for color index 3. |

### SPOT1 OSD FONT AND PICTURE RAM ADDRESS LOW BYTE REGISTER – 0XF58

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_OSD_FRAM_ADDR[7:0]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT1 OSD FONT AND PICTURE RAM ADDRESS HIGH BYTE REGISTER – 0XF59

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP1_OSD_FRAM_ADDR[10:8]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT1 OSD FONT AND PICTURE RAM DATA LOW BYTE REGISTER – 0XF5A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_OSD_FRAM_DATA[7:0]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT1 OSD FONT AND PICTURE RAM DATA HIGH BYTE REGISTER – 0XF5B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_OSD_FRAM_DATA[15:8]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT1 OSD DISPLAY RAM ADDRESS REGISTER – 0XF5C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP1_OSD_DRAM_ADDR[7:0]<br><br>Display SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 244x6. Channel number size is 160x6, time/date size is 32x6, title size is 32x6 and picture index size is 20x6. |

### SPOT1 OSD DISPLAY RAM DATA REGISTER – 0XF5D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | | Reserve |
| 5:0 | RW | 0 | SP1_OSD_DRAM_Data[5:0]<br><br>Display SRAM Data. When host write data to this SRAM, address is written first and then data. SRAM size is 244x6. Channel number size is 160x6, time/date size is 32x6, title size is 32x6 and picture index size is 20x6. |

### SPOT DISP_PITCH REGISTER – 0XF5E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SPOT_DISP_PITCH<br><br>Display Pitch register for SDRAM linear address. In 16 pixels / unit |

### SPOT ADJUSTMENT ENABLE REGISTER – 0XF5F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SPOT_ADJUST_ENA<br><br>Display Adjust Enable register |

### SPOT2 CONTROL REGISTER – 0XF60

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP2_HTT [7:0], SPOT2 Horizontal Total Register<br><br>Lower 8 bits of the SP2_HTT |

### SPOT2 CONTROL REGISTER – 0XF61

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | N/A | | Reserved |
| 4:0 | RW | 0 | SP2_HTT [12:8], SPOT2 Horizontal Total Register<br><br>Upper 5 bits of the SP2_HTT |

### SPOT1 CONTROL REGISTER – 0XF62

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP2_HDE [7:0], SPOT2 Horizontal Display Enable Register<br><br>Lower 8 bits of the HDE register |

### SPOT2 CONTROL REGISTER – 0XF63

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | N/A | | Reserved |
| 3:0 | RW | 0 | SP2_HDE [11:8], SPOT2 Horizontal Display Enable Register<br><br>Upper 4 bit of the SP2_HDE |

### SPOT2 CONTROL REGISTER – 0XF64

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP2_VDE [7:0], SPOT2 Vertical Display Enable Register<br><br>Lower 8 bits of the VDE register |

### SPOT2 CONTROL REGISTER – 0XF65

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | N/A | 0 | BND_CH_EN[7:4] Border Channel Enable |
| 3 | N/A | 0 | Reserved |
| 2:0 | RW | 0 | SP2_VDE [11:8], SPOT2 Vertical Display Enable Register<br><br>Upper 3 bits of the Vertical Display Enable Register |

### SPOT2 CONTROL REGISTER – 0XF66

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **SP2_PHSYNC [7:0], SPOT2 Horizontal Position Sync Register**<br><br>Lower 8 bits of the SP2_PHSYNC<br><br>The register value determine when the horizontal sync start when the horizontal display counter reach to the sum of SP2_HDE and SP2_PHSYNC |

### SPOT2 CONTROL REGISTER – 0XF67

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | N/A | 0 | **BND_WIDTH_SEL in Vertical Border in Field Switching Mode**<br><br>[7] SP4 Border Select<br>[6] SP3 Border Select<br>[5] SP2 Border Select<br>[4] SP1 Border Select |
| 3:2 | RW | 0 | **SP2 FRSC Control**<br>11: bank_sel equals bank_selB<br>10: bank_sel equals {vblank, fld}<br>01: bank_sel equals {bank_selA[1], fld}<br>00: bank_sel equals bank_selA<br><br>11: wr_page equals wr_pageB<br>10: wr_page equals {~vblank, 0}<br>01: wr_page equals {!bank_selA[1], fld}<br>00: wr_page equals wr_pageA |
| 1 | RW | 0 | **Reserve** |
| 0 | RW | 0 | **SP2_PHSYNC[8], SPOT2 Horizontal Position Sync Register**<br><br>Upper 1 bit of the SP2_PHSYNC |

### SPOT2 CONTROL REGISTER – 0XF68

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **SP2_PVSYNC [7:0], SPOT2 Vertical Position Sync Register**<br><br>Lower 8 bits of the SP2_PVSYNC<br><br>The register value determine when the horizontal sync start when the horizontal display counter reach to the sum of SP2_VDE and SP2_PVSYNC |

### SPOT2 CONTROL REGISTER – 0XF69

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **SP2_VTT [7:0], SPOT2 VTT Register**<br><br>Lower 8 bits of the SP2_ VTT register |

### SPOT2 CONTROL REGISTER – 0XF6A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | [7] SP2_Half_D1 Select |
| 6 | RW | 0 | Reserve |
| 5:4 | RW | 0 | Select SPOT CVBS out to network port<br><br>00: Sel SPOT1<br>01: Sel SPOT2<br>10: Sel SPOT3<br>11: Sel SPOT4 |
| 3 | RW | 0 | Reserve |
| 2:0 | RW | 0 | [2:0] SP2_VTT[10:8] Upper 3 bits of the SP2_VTT |

### SPOT2 CONTROL REGISTER – 0XF6B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | SP2 VBLANK_SEL |
| 5 | RW | 0 | SP2 COLOR BAR ENA |
| 4 | RW | 0 | SP2_SWITCH_EN<br><br>0: Background enable<br>1: Background display disable |
| 3:2 | RW | 0 | SP2_FLD select<br><br>11: use bank0 from the FRSC as the field<br>10: use bank0 from the FRSC inverse as the filed<br>0x: use HVCNT generated field |
| 1:0 | RW | 0 | Reserve |

### SPOT2 CONTROL REGISTER – 0XF6C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SPOT Rotate Count<br><br>[7:6] SP4 rotate count<br>[5:4] SP3 rotate count<br>[3:2] SP2 rotate count<br>[1:0] SP1 rotate count |

### SPOT2 CONTROL REGISTER – 0XF6D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP2_DNS Upper Bits<br><br>[7:6] SP2 Destination Width upper bit [9:8]<br>[5:3] SP2 Source Height upper bit [10:8]<br>[2:0] SP2 Source Width upper bit [10:8] |

### SPOT2 CONTROL REGISTER – 0XF6E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | SP2 REC or Network port Enable<br>Enable the REC or Network port to use SP2 as display |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 6:4 | RW | 0 | SP2 color bar select |
| 3:1 | RW | 0 | SP2_Firsrt_line_control |
| 0 | RW | 0 | SP2 Destination Height upper bit [8] in down scale ratio |

**SPOT2 CONTROL REGISTER – 0XF6F**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 1 | SP2_ Field Mode Select<br>1: Field mode<br>0: Frame mode |
| 6 | | 0 | SP2_2ND_MEM Enable |
| 5 | | 0 | SP2_FRMCNT_EN SPOT2 Frame Count Enable |
| 4 | | 0 | SP2_ENA SPOT2 Enable |
| 3:2 | RW | 0 | SP2_ROT Rotation Enable |
| 1 | RW | 0 | SP2_VSPOL SPOT2 Vertical Polarity Control |
| 0 | RW | 0 | SP2_HSPOL SPOT2 Horizontal Polarity Control |

**SPOT2 CONTROL REGISTER – 0XF70**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP2_HSTART [7:0], SPOT2 Horizontal Start Register<br><br>This register determines the lower 8 bits of SPOT2 Horizontal start address |

**SPOT2 CONTROL REGISTER – 0XF71**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 01 | SP2_VSTART [7:0], SPOT2 Vertical Start Register<br><br>This register determines the lower 8 bits of SPOT2 Vertical start address |

**SPOT2 CONTROL REGISTER – 0XF72**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | N/A | 0 | SP2 Quad Mode Enable<br><br>When Enable, the SP2 memory controller will read half of each line data and start a new a new address calculation from second half line as well as half field for the vertical address to prevent whole screen jumping when weak video is presence.<br><br>This bit should set to one when screen image is set to quad mode. |
| 6 | N/A | | Reserved |
| 5:4 | RW | 0 | SP2_VSTART [9:8], SPOT2 Horizontal Start Register<br><br>Upper 2 bits of the SPOT2 Horizontal start register |
| 3:0 | RW | 0 | SP2_VSTART [11:8], SPOT2 Vertical Start Register<br><br>Upper 4 bits of the SPOT2 Vertical start register |

### SPOT2 CONTROL REGISTER – 0XF73

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP2_SRC_WID [7:0], SPOT2 Down Scale Source Width Register<br><br>This register determines the source width of the image. The upper 3 bits are in register 0xF6D. |

### SPOT2 CONTROL REGISTER – 0XF74

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP2_SRC_HI [7:0], SPOT2 Down Scale Source Height Register<br><br>This register determines the source height of the image. The upper 3 bits are in register 0xF6D. |

### SPOT2 CONTROL REGISTER – 0XF75

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP2_DST_WID [7:0], SPOT2 Down Scale Destination Width Register<br><br>This register determines the source width of the image. The upper 2 bits are in register 0xF6D. |

### SPOT2 CONTROL REGISTER – 0XF76

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP2_DST_HI [7:0], SPOT2 Down Scale Destination Height Register<br><br>This register determines the source height of the image. The upper 1 bit is in register 0xF6E. |

### SPOT2 CONTROL REGISTER – 0XF77

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | SP2_TVENC_FSC_MODE, SPOT2 TV FSCSEL<br><br>This register determines the TV modulation frequency standard select.<br><br>00 - NTSC M<br>01 - PAL B, D, G, H, I<br>10 - PAL M<br>11 - PAL N |
| 5 | RW | 0 | SP2_TVENC_SAT_LMT, SPOT2 TV video saturation limit<br><br>Control the data range of the ITU-R BT 656 output<br><br>0 - Not limit<br>1 – Data range is limited to 1 – 254 range |
| 4:1 | N/A | 0 | Reserved |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 0 | RW | 0 | **SP2_TV PAL or NTSC select**<br><br>This register determines the CVBS output Standard.<br><br>0: NSTC mode<br>1: PAL mode. |

**SPOT2 CONTROL REGISTER – 0XF78**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **SP2_TVENC_VSPOL, SPOT2 TVENC Vertical sync polarity**<br><br>0: Active Low<br>1: Active High |
| 6 | RW | 0 | **SP2_TVENC_HSPOL, SPOT2 TV Encoder Horizontal Sync Polarity**<br><br>0: Active Low<br>1: Active High |
| 5 | RW | 0 | **SP2_TVENC_VSDET, SPOT1 TV Encoder Vsync Det.** |
| 4 | RW | 0 | **SP2_TVENC_PED, SPOT2 TV Encoder Pedestal Enable**<br><br>Enable Pedestal for video output<br><br>0: 0 IRE for pedestal<br>1: 7.5 IRE pedestal level |
| 3 | RW | 0 | **SP2_TVENC_PDRST, SPOT2 TV Encoder Pedestal Reset**<br><br>Subcarrier Phase Alternation reset every 8 fields in PAL system<br><br>0: Constant relationship, reset every 8 fields<br>1: Free running |
| 2 | RW | 0 | **SP2_TVENC_PHALT, SPOT2 TV Encoder Phase Alternation**<br><br>0: Disable phase alternation for line by line<br>1: Enable phase alternation for line by line |
| 1 | RW | 0 | **SP2_TVENC_F_INV, SPOT2 TV Encoder Field Output Invert**<br><br>This register bit inverts the field output level if set. |
| 0 | RW | 0 | **SP2_TVENC_INTERLACE, SPOT2 TVENC Interlace Select**<br><br>This register bit determines the TV Standard in Interlace timing mode.<br><br>0 - Progressive<br>1 - Interlace |

### SPOT2 CONTROL REGISTER – 0XF79

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | N/A | | Reserved |
| 5:4 | RW | 0 | SP2_TVENC_YC_OS[1:0], SPOT2 TV Encoder Active Pixel Offset delay<br><br>00: No Active offset<br>01: Active pixel offset for 1/4 pixel unit<br>10: Active pixel offset for 1/2 pixel unit<br>11: Active pixel offset for 3/4 pixel unit |
| 3:2 | RW | 0 | SP2_TVENC_OSENC[1:0], SPOT2 TV Encoder Field Offset for the first video line<br><br>00: Apply same ENC_VDSEL for odd and even field<br>01: Apply same ENC_VDSEL for odd and odd field<br>10: Apply TVENC_VSDEL for odd field and TVENC_VSDEL+1 for even field<br>11: Apply TVENC_VSDEL for odd field and TVENC_VSDEL+2 for even field |
| 1 | RW | 0 | SP2_TVENC_FLDENCP, SPOT2 TV Encoder Field Polarity<br><br>This Register control the field polarity<br>0: Even field is high<br>1: Odd field is high |
| 0 | RW | 0 | SP2_TVENC_FLDDET, SPOT2 TV Encoder Field Detect |

### SPOT2 CONTROL REGISTER – 0XF7A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP2_TVENC_HSENC[7:0], SPOT2 TV Encoder Active pixel delay<br><br>Lower 8 bits of the pixel delay from active video by ½ pixel per step |

### SPOT2 CONTROL REGISTER – 0XF7B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | N/A | 0 | SP2_Rotate_Time_Offset |
| 1:0 | RW | 0 | SP2_TVENC_HSENC[9:8], SPOT2 TV Encoder Active pixel delay<br><br>Upper 2 bits of the pixel delay from active video by ½ pixel per step |

### SPOT2 CONTROL REGISTER – 0XF7C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | N/A | 1 | SP2_CVBS_SEL<br><br>11 : SP2 CVBS output is coming from SP4<br>10 : SP2 CVBS output is coming from SP3<br>01 : SP2 CVBS output is coming from SP2<br>00 : SP2 CVBS output is coming from SP1 |
| 5:0 | RW | 0 | SP2_TVENC_VSENC, SPOT2 TV Encoder Active line delay<br><br>6 bits of the line delay from active video by line per step |

### SPOT2 CONTROL REGISTER – 0XF7D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | N/A | | Reserved |
| 4:0 | RW | 0 | SP2_TVENC_LINE_OS, SPOT2 TV Encoder Active line offset<br><br>5 bits of the line offset |

### SPOT2 CONTROL REGISTER – 0XF7E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | N/A | 0 | [7;6] SP2_FLD_SW_MODE<br><br>[7] Field Switching Mode enable<br>[6] Field Switch Mode image height is 240 lines Enable |
| 5:0 | RW | 0 | SP2_TVENC_PIXEL_OS, SPOT2 TV Encoder Active Pixel offset<br><br>6 bits of the pixel offset |

### SPOT2 CONTROL REGISTER – 0XF7F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | N/A | | Reserved |
| 6 | RW | 0 | SP2_TVENC_C_OFF_D, SPOT2 TV Encoder C Off<br><br>Set the Color Off if set |
| 5:4 | RW | 0 | SP2_TVENC_CBW_D, SPOT2 TV Encoder CBCR delay<br><br>2 bits of the CbCr delay |
| 3:2 | RW | 0 | SP2_TVENC_YBW_D, SPOT2 TV Encoder Luma delay<br><br>2 bits of the Luminous delay |
| 1:0 | RW | 0 | SP2_TVENC_FLD_OS, SPOT2 TV Encoder Active Field offset<br><br>2 bits of the field offset |

### SPOT2 CONTROL REGISTER – 0XF80

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | SP2_TVENC_TST_FSC_FREE, SPOT2 TV Encoder Sub-Carrier is set to free run |
| 6 | RW | 0 | SP2_TVENC_T_CCIR_TIM, SPOT2 TV Encoder CCOR Timing<br><br>2 bits of the CbCr delay |
| 5 | RW | 0 | SP2_TVENC_T_656_STD, SPOT2 TV Encoder 656 Standard<br><br>2 bits of the Luminous delay |
| 4:0 | RW | 0 | SP2_TVENC_VIS_LINE_OS, SPOT2 TV Encoder line offset<br><br>5 bits Line offset |

### SPOT2 CONTROL REGISTER – 0XF81

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | RW | | Reserved |
| 4 | RW | 0 | **SP2_OSD_TITLE_EN**<br><br>Display title enable<br>1: enable<br>0: disable |
| 3 | RW | 0 | **SP2_OSD_TIME_EN**<br><br>Display time and date enable<br>1: enable<br>0: disable |
| 2 | RW | 0 | **SP2_OSD_CHPIC_EN**<br><br>Display picture for each channel<br>1: enable<br>0: disable |
| 1 | RW | 0 | **SP2_OSD_CHNUM_EN**<br><br>Display channel number for each channel<br>1: enable<br>0: disable |
| 0 | RW | 0 | **SP2_OSD_EN**<br><br>All OSD information enable or disable. If set to 1, each information is enabled by bit [4:1]. If set to 0, all OSD is disabled.<br>1: enable<br>0: disable |

### SPOT2 CONTROL REGISTER – 0XF82

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | **Reserved** |
| 5 | RW | 0 | **SP2_OSD_CHPIC_BLINK**<br><br>If set to 1, picture for each channel blink in picture index 2 and 3. Picture 0 and 1 will not blink<br>1: enable<br>0: disable |
| 4 | RW | 0 | **SP2_OSD_CHPIC_TRANS**<br><br>If set to 1, picture in color index “00” will be transparent<br>1: enable<br>0: disable |
| 3 | RW | 0 | **SP2_OSD_TITLE_MIX**<br><br>Display title mix enable bit. If set to 1, title will be 50% blending with video<br>1: enable<br>0: disable |
| 2 | RW | 0 | **SP2_OSD_TIME_MIX**<br><br>Display time and date mix enable bit. If set to 1, time/date will be 50% blending with video<br>1: enable<br>0: disable |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1 | RW | 0 | SP2_OSD_CHPIC_MIX<br><br>Display picture mix enable bit. If set to 1, picture will be 50% blending with video<br>1: enable<br>0: disable |
| 0 | RW | 0 | SP2_OSD_CHNUM_MIX<br><br>Display channel number information mix enable bit. If set to 1, channel information will be 50% blending with video<br>1: enable<br>0: disable |

**SPOT2 OSD FONT SIZE REGISTER – 0XF83**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0x1 | SP2_OSD_CHPIC_POS<br><br>Channel picture corner position.<br>00: left top<br>01: right top<br>10: left bottom<br>11: right bottom |
| 5:4 | RW | 0 | SP2_OSD_CHNUM_POS<br><br>Channel number corner position.<br>00: left top<br>01: right top<br>10: left bottom<br>11: right bottom |
| 3:2 | RW | 0x3 | SP2_OSD_FONT_VSIZE<br><br>Font vertical size for display. There are four choices.<br>00: 8<br>01: 10<br>10: 16, scale up from 8<br>11: 20, scale up from 10 |
| 1:0 | RW | 0x3 | SP2_OSD_FONT_HSIZE<br><br>Font horizontal size for display. There are four choices.<br>00: 6<br>01: 8<br>10: 12, scale up from 6<br>11: 16, scale up from 8 |

**SPOT2 OSD TIME AND DATE HORIZONTAL POSITION LOW BYTE REGISTER – 0XF84**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP2_OSD_TIME_HPOS[7:0]<br><br>Time and Date information horizontal position. It is one pixel unit. |

### SPOT2 OSD TIME AND DATE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XF85

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP2_OSD_TIME_HPOS[10:8]<br><br>Time and Date information horizontal position. It is one pixel unit. |

### SPOT2 OSD TIME AND DATE VERTICAL POSITION LOW BYTE REGISTER – 0XF86

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP2_OSD_TIME_VPOS[7:0]<br><br>Time and Date information vertical position. It is one pixel unit. |

### SPOT2 OSD TIME AND DATE VERTICAL POSITION HIGH BYTE REGISTER – 0XF87

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP2_OSD_TIME_VPOS[10:8]<br><br>Time and Date information vertical position. It is one pixel unit. |

### SPOT2 OSD TITLE HORIZONTAL POSITION LOW BYTE REGISTER – 0XF88

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP2_OSD_TITLE_HPOS[7:0]<br><br>Title information horizontal position. It is one pixel unit. |

### SPOT2 OSD TITLE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XF89

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | RW | 0 | Vertical Start Address offset [12:8] in 256 or 512 Mb memory configuration |
| 2:0 | RW | 0 | SP2_OSD_TITLE_HPOS[10:8]<br><br>Title information horizontal position. It is one pixel unit. |

### SPOT2 OSD TITLE VERTICAL POSITION LOW BYTE REGISTER – 0XF8A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP2_OSD_TITLE_VPOS[7:0]<br><br>Title information vertical position. It is one pixel unit. |

### SPOT2 OSD TITLE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XF8B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP2_OSD_TITLE_VPOS[10:8]<br><br>Title information vertical position. It is one pixel unit. |

### SPOT2 OSD FONT AND PICTURE RAM ADDRESS LOW BYTE REGISTER – 0XF8C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP2_OSD_FRAM_ADDR[7:0]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT2 OSD FONT AND PICTURE RAM ADDRESS HIGH BYTE REGISTER – 0XF8D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP2_OSD_FRAM_ADDR[10:8]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT2 OSD FONT AND PICTURE RAM DATA LOW BYTE REGISTER – 0XF8E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP2_OSD_FRAM_DATA[7:0]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT2 OSD FONT AND PICTURE RAM DATA HIGH BYTE REGISTER – 0XF8F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP2_OSD_FRAM_DATA[15:8]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT3 CONTROL REGISTER – 0XF90

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP3_HTT [7:0], SPOT3 Horizontal Total Register<br><br>Lower 8 bits of the SP3_HTT |

### SPOT3 CONTROL REGISTER – 0XF91

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | N/A | | Reserved |
| 4:0 | RW | 0 | SP3_HTT [12:8], SPOT3 Horizontal Total Register<br><br>Upper 5 bits of the SP3_HTT |

### SPOT3 CONTROL REGISTER – 0XF92

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **SP3_HDE [7:0], SPOT3 Horizontal Display Enable Register**<br><br>Lower 8 bits of the SP3_HDE register |

### SPOT3 CONTROL REGISTER – 0XF93

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | N/A | | Reserved |
| 3:0 | RW | 0 | **SP3_HDE [11:8], SPOT3 Horizontal Display Enable Register**<br><br>Upper 4 bits of the SP3_HDE |

### SPOT3 CONTROL REGISTER – 0XF94

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **SP3_VDE [7:0], SPOT3 Vertical Display Enable Register**<br><br>Lower 8 bits of the SP3_VDE register |

### SPOT3 CONTROL REGISTER – 0XF95

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | N/A | | **BND_CH_EN[10:8] Border Channel Enable** |
| 3 | N/A | | Reserved |
| 2:0 | RW | 0 | **SP3_VDE [11:8], SPOT3 Vertical Display Enable Register**<br><br>Upper 3 bits of the Vertical Display Enable Register |

### SPOT3 CONTROL REGISTER – 0XF96

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | **SP3_PHSYNC [7:0], SPOT3 Horizontal Position Sync Register**<br><br>Lower 8 bits of the SP3_PHSYNC<br><br>The register value determine when the horizontal sync start when the horizontal display counter reach to the sum of SP3_HDE and SP3_PHSYNC |

### SPOT3 CONTROL REGISTER – 0XF97

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | N/A | | Reserved |
| 3:2 | RW | 0 | SP3 FRSC Control<br>11: bank_sel equals bank_selB<br>10: bank_sel equals {vblank, fld}<br>01: bank_sel equals {bank_selA[1], fld}<br>00: bank_sel equals bank_selA<br><br>11: wr_page equals wr_pageB<br>10: wr_page equals {~vblank, 0}<br>01: wr_page equals {!bank_selA[1], fld}<br>00: wr_page equals wr_pageA |
| 1 | N/A | | Reserved |
| 0 | RW | 0 | SP3_PHSYNC [8], SPOT3 Horizontal Position Sync Register<br><br>Upper 1 bit of the SP3_PHSYNC |

### SPOT3 CONTROL REGISTER – 0XF98

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP3_PVSYNC [7:0], SPOT3 Vertical Position Sync Register<br><br>Lower 8 bits of the SP3_PVSYNC<br><br>The register value determine when the horizontal sync start when the horizontal display counter reach to the sum of SP3_VDE and SP3_PVSYNC |

### SPOT3 CONTROL REGISTER – 0XF99

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP3_VTT[7:0] |

### SPOT3 CONTROL REGISTER – 0XF9A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | [7] SP3_Half_D1 Select |
| 6:3 | RW | 0 | Reserve |
| 2:0 | RW | 0 | [2:0] SP3_VTT[10:8] |

### SPOT3 CONTROL REGISTER – 0XF9B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | SP3_VBLANK_SEL |
| 5 | RW | 0 | SP3_COLOR BAR ENA |
| 4 | RW | 0 | SP3_SWITCH_EN<br><br>0: Background enable<br>1: Background display disable |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 3:2 | RW | 0 | SP3_FLD select<br><br>11: use bank0 from the FRSC as the field<br>10: use bank0 from the FRSC inverse as the filed<br>0x: use HVCNT generated field |
| 1:0 | RW | 0 | SPOT_POWER_DOWN_EN<br><br>[1] SP2 PD enable when 0<br>[0] SP1 PD enable when 0 |

**SPOT3 CONTROL REGISTER – 0XF9C**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0 | REC, NETWORK port use SP1 as the display<br><br>[7] = 1, select Network port to display on SP1<br>= 0, [6:4] selects REC PORT 1 - 8 display on SP1<br>[6:4] = 0, selects REC PORT1,<br>[6:4] = 1, selects REC PORT2,<br>..... |
| 3:0 | RW | 0 | REC, NETWORK port use SP2 as the display<br><br>[3] = 1, select Network port to display on SP2<br>= 0, [2:0] selects REC PORT 1 - 8 display on SP2<br>[2:0] = 0, selects REC PORT1,<br>[2:0] = 1, selects REC PORT2,<br>..... |

**SPOT3 CONTROL REGISTER – 0XF9D**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP3_DNS Upper Bits<br>[7:6] SP3 Destination Width upper bit [9:8]<br>[5:3] SP3 Source Height upper bit [10:8]<br>[2:0] SP3 Source Width upper bit [10:8] |

**SPOT3 CONTROL REGISTER – 0XF9E**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | SP3 REC or Network port Enable<br>Enable the REC or Network port to use SP3 as display |
| 6:4 | RW | 0 | SP3 color bar select |
| 3:1 | RW | 0 | SP3_Firsrt_line_control |
| 0 | RW | 0 | SP3 Destination Height upper bit [8] |

**SPOT3 CONTROL REGISTER – 0XF9F**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 1 | SP3 Field Mode Select<br><br>1: Field mode<br>0: Frame mode |
| 6 | RW | 0 | SP3_2ND_MEM Enable |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 5 | RW | 0 | SP3 FRMCNT_EN SPOT3 Frame Count Enable |
| 4 | RW | 0 | SP3_ENA SPOT3 Enable |
| 3:2 | RW | 0 | SP3_ROT Rotation Enable |
| 1 | RW | 0 | SP3_VSPOL SPOT3 Vertical Polarity Control |
| 0 | RW | 0 | SP3_HSPOL SPOT3 Horizontal Polarity Control |

**SPOT3 CONTROL REGISTER – 0XFA0**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP3_HSTART [7:0], SPOT3 Horizontal Start Register<br><br>This register determines the lower 8 bits of SPOT3 Horizontal start address |

**SPOT3 CONTROL REGISTER – 0XFA1**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 01 | SP3_VSTART [7:0], SPOT3 Vertical Start Register<br><br>This register determines the lower 8 bits of SPOT3 Vertical start address |

**SPOT3 CONTROL REGISTER – 0XFA2**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | N/A | | SP3 Quad Mode Enable<br><br>When Enable, the SP3 memory controller will read half of each line data and start a new a new address calculation from second half line as well as half field for the vertical address to prevent whole screen jumping when weak video is presence.<br><br>This bit should set to one when screen image is set to quad mode. |
| 6 | N/A | | Reserve |
| 5:4 | RW | 0 | SP3_VSTART [9:8], SPOT3 Horizontal Start Register<br><br>Upper 2 bits of the SPOT3 Horizontal start register |
| 3:0 | N/A | | SP3_VSTART [11:8], SPOT3 Vertical Start Register<br><br>Upper 4 bits of the SPOT3 Vertical start register |

**SPOT3 CONTROL REGISTER – 0XFA3**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP3_SRC_WID [7:0], SPOT3 Down Scale Source Width Register<br><br>This register determines the source width of the image. The upper 3 bits are in register 0xF9D. |

**SPOT3 CONTROL REGISTER – 0XFA4**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | **SP3_SRC_HI [7:0], SPOT3 Down Scale source height Register**<br><br>This register determines the source height of the image. The upper 3 bits are in register 0xF9D. |

**SPOT3 CONTROL REGISTER – 0XFA5**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | **SP3_DST_WID [7:0], SPOT3 Down Scale Destination Width Register**<br><br>This register determines the destination width of the image. The upper 2 bits are in register 0xF6D. |

**SPOT3 CONTROL REGISTER – 0XFA6**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | **SP3_DST_HI [7:0], SPOT3 Down Scale Destination height Register**<br><br>This register determines the destination height of the image. The upper 1 bits are in register 0xF9E. |

**SPOT3 CONTROL REGISTER – 0XFA7**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | **SP3_TVENC_FSC_MODE, SPOT3 TV FSCSEL**<br><br>This register determines the TV modulation frequency standard select.<br><br>00 - NTSC M<br>01 - PAL B, D, G, H, I<br>10 - PAL M<br>11 - PAL N |
| 5 | RW | 0 | **SP3_TVENC_SAT_LMT, SPOT3 TV video saturation limit**<br><br>Control the data range of the ITU-R BT 656 output<br><br>0 - Not limit<br>1 – Data range is limited to 1 – 254 range |
| 4:1 | N/A | 0 | **Reserved** |
| 0 | RW | 0 | **SP3_TVENC_PALNT, SPOT3 TV PAL or NTSC select**<br><br>This register determines the CVBS output Standard.<br><br>0: NSTC mode<br>1: PAL mode. |

**SPOT3 CONTROL REGISTER – 0XFA8**

| Bit | R/W | Default | Description |
|---|---|---|---|

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | SP3_TVENC_VSPOL, SPOT3 TVENC Vertical sync polarity<br><br>0: Active Low<br>1: Active High |
| 6 | RW | 0 | SP3_TVENC_HSPOL, SPOT3 TV Encoder Horizontal Sync Polarity<br><br>0: Active Low<br>1: Active High |
| 5 | RW | 0 | SP3_TVENC_VSDET, SPOT3 TV Encoder Vsync Det. |
| 4 | RW | 0 | SP3_TVENC_PED, SPOT3 TV Encoder Pedestal Enable<br><br>Enable Pedestal for video output<br><br>0: 0 IRE for pedestal<br>1: 7.5 IRE pedestal level |
| 3 | RW | 0 | SP3_TVENC_PDRST, SPOT3 TV Encoder Pedestal Reset<br><br>Subcarrier Phase Alternation reset every 8 fields in PAL system<br><br>0: Constant relationship, reset every 8 fields<br>1: Free running |
| 2 | RW | 0 | SP3_TVENC_PHALT, SPOT3 TV Encoder Phase Alternation<br><br>0: Disable phase alternation for line by line<br>1: Enable phase alternation for line by line |
| 1 | RW | 0 | SP3_TVENC_F_INV, SPOT3 TV Encoder Field Output Invert<br><br>This register bit inverts the field output level if set. |
| 0 | RW | 0 | SP3_TVENC_INTERLACE, SPOT3 TVENC Interlace Select<br><br>This register bit determines the TV Standard in Interlace timing mode.<br><br>0 - Progressive<br>1 - Interlace |

**SPOT3 CONTROL REGISTER – 0XFA9**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | N/A | | Reserved |
| 5:4 | RW | 0 | SP3_TVENC_YC_OS[1:0], SPOT3 TV Encoder Active Pixel Offset delay<br><br>00: No Active offset<br>01: Active pixel offset for 1/4 pixel unit<br>10: Active pixel offset for 1/2 pixel unit<br>11: Active pixel offset for 3/4 pixel unit |
| 3:2 | RW | 0 | SP3_TVENC_OSENC[1:0], SPOT3 TV Encoder Field Offset for the first video line<br><br>00: Apply same ENC_VDSEL for odd and even field<br>01: Apply same ENC_VDSEL for odd and odd field<br>10: Apply TVENC_VSDEL for odd field and TVENC_VSDEL+1 for even field<br>11: Apply TVENC_VSDEL for odd field and TVENC_VSDEL+2 for even field |

| | | | |
|---|---|---|---|
| 1 | RW | 0 | SP3_TVENC_FLDENCP, SPOT3 TV Encoder Field Polarity<br><br>This Register control the field polarity<br>0: Even field is high<br>1: Odd field is high |
| 0 | RW | 0 | SP3_TVENC_FLDDET, SPOT3 TV Encoder Field Detect |

**SPOT3 CONTROL REGISTER – 0XFAA**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP3_TVENC_HSENC[7:0], SPOT3 TV Encoder Active pixel delay<br><br>Lower 8 bits of the pixel delay from active video by ½ pixel per step |

**SPOT3 CONTROL REGISTER – 0XFAB**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | N/A | 0 | SP2_Rotate_Time_Offset |
| 1:0 | RW | 0 | SP3_TVENC_HSENC[9:8], SPOT3 TV Encoder Active pixel delay<br><br>Upper 2 bits of the pixel delay from active video by ½ pixel per step |

**SPOT3 CONTROL REGISTER – 0XFAC**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | N/A | 10 | SP3_CVBS_SEL<br><br>11 : SP3 CVBS output is coming from SP4<br>10 : SP3 CVBS output is coming from SP3<br>01 : SP3 CVBS output is coming from SP2<br>00 : SP3 CVBS output is coming from SP1 |
| 5:0 | RW | 0 | SP3_TVENC_VSENC, SPOT3 TV Encoder Active line delay<br><br>6 bits of the line delay from active video by line per step |

**SPOT3 CONTROL REGISTER – 0XFAD**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | N/A | | Reserved |
| 4:0 | RW | 0 | SP3_TVENC_LINE_OS, SPOT3 TV Encoder Active line offset<br><br>5 bits of the line offset |

**SPOT3 CONTROL REGISTER – 0XFAE**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | N/A | 0 | [7;6] SP3_FLD_SW_MODE<br><br>[7] Field Switching Mode enable<br>[6] Field Switch Mode image height is 240 lines Enable |
| 5:0 | RW | 0 | SP3_TVENC_PIXEL_OS, SPOT3 TV Encoder Active Pixel offset<br><br>6 bits of the pixel offset |

**SPOT3 CONTROL REGISTER – 0XFAF**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | N/A | | Reserved |
| 6 | RW | 0 | SP3_TVENC_C_OFF_D, SPOT3 TV Encoder C Off<br><br>Set the Color Off if set |
| 5:4 | RW | 0 | SP3_TVENC_CBW_D, SPOT3 TV Encoder CBCR delay<br><br>2 bits of the CbCr delay |
| 3:2 | RW | 0 | SP3_TVENC_YBW_D, SPOT3 TV Encoder Luma delay<br><br>2 bits of the Luminous delay |
| 1:0 | RW | 0 | SP3_TVENC_FLD_OS, SPOT3 TV Encoder Active Field offset<br><br>2 bits of the field offset |

**SPOT3 CONTROL REGISTER – 0XFB0**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | SP3_TVENC_TST_FSC_FREE, SPOT3 TV Encoder Sub-Carrier is set to free run |
| 6 | RW | 0 | SP3_TVENC_T_CCIR_TIM, SPOT3 TV Encoder CCOR Timing<br><br>2 bits of the CbCr delay |
| 5 | RW | 0 | SP3_TVENC_T_656_STD, SPOT3 TV Encoder 656 Standard<br><br>2 bits of the Luminous delay |
| 4:0 | RW | 0 | SP3_TVENC_VIS_LINE_OS, SPOT2 TV Encoder line offset<br><br>5 bits Line offset |

**SPOT3 CONTROL REGISTER – 0XFB1**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | RW | 0 | Reserved |
| 4 | RW | 0 | SP3_OSD_TITLE_EN<br><br>Display title enable<br>1: enable<br>0: disable |
| 3 | RW | 0 | SP3_OSD_TIME_EN<br><br>Display time and date enable<br>1: enable<br>0: disable |
| 2 | RW | 0 | SP3_OSD_CHPIC_EN<br><br>Display picture for each channel<br>1: enable<br>0: disable |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1 | RW | 0 | **SP3_OSD_CHNUM_EN**<br><br>Display channel number for each channel<br>1: enable<br>0: disable |
| 0 | RW | 0 | **SP3_OSD_EN**<br><br>All OSD information enable or disable. If set to 1, each information is enabled by bit [4:1]. If set to 0, all OSD is disabled.<br>1: enable<br>0: disable |

**SPOT3 CONTROL REGISTER – 0XFB2**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | **Reserved** |
| 5 | RW | 0 | **SP3_OSD_CHPIC_BLINK**<br><br>If set to 1, picture for each channel blink in picture index 2 and 3. Picture 0 and 1 will not blink<br><br>1: enable<br>0: disable |
| 4 | RW | 0 | **SP3_OSD_CHPIC_TRANS**<br><br>If set to 1, picture in color index “00” will be transparent<br>1: enable<br>0: disable |
| 3 | RW | 0 | **SP3_OSD_TITLE_MIX**<br><br>Display title mix enable bit. If set to 1, title will be 50% blending with video<br>1: enable<br>0: disable |
| 2 | RW | 0 | **SP3_OSD_TIME_MIX**<br><br>Display time and date mix enable bit. If set to 1, time/date will be 50% blending with video<br>1: enable<br>0: disable |
| 1 | RW | 0 | **SP3_OSD_CHPIC_MIX**<br><br>Display picture mix enable bit. If set to 1, picture will be 50% blending with video<br>1: enable<br>0: disable |
| 0 | RW | 0 | **SP3_OSD_CHNUM_MIX**<br><br>Display channel number information mix enable bit. If set to 1, channel information will be 50% blending with video<br>1: enable<br>0: disable |

### SPOT3 OSD FONT SIZE REGISTER – 0XFB3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0x1 | SP3_OSD_CHPIC_POS<br><br>Channel picture corner position.<br>00: left top<br>01: right top<br>10: left bottom<br>11: right bottom |
| 5:4 | RW | 0 | SP3_OSD_CHNUM_POS<br><br>Channel number corner position.<br>00: left top<br>01: right top<br>10: left bottom<br>11: right bottom |
| 3:2 | RW | 0x3 | SP3_OSD_FONT_VSIZE<br><br>Font vertical size for display. There are four choices.<br>00: 8<br>01: 10<br>10: 16, scale up from 8<br>11: 20, scale up from 10 |
| 1:0 | RW | 0x3 | SP3_OSD_FONT_HSIZE<br><br>Font horizontal size for display. There are four choices.<br>00: 6<br>01: 8<br>10: 12, scale up from 6<br>11: 16, scale up from 8 |

### SPOT3 OSD TIME AND DATE HORIZONTAL POSITION LOW BYTE REGISTER – 0XFB4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP3_OSD_TIME_HPOS[7:0]<br><br>Time and Date information horizontal position. It is one pixel unit. |

### SPOT3 OSD TIME AND DATE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XFB5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP3_OSD_TIME_HPOS[10:8]<br><br>Time and Date information horizontal position. It is one pixel unit. |

### SPOT3 OSD TIME AND DATE VERTICAL POSITION LOW BYTE REGISTER – 0XFB6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP3_OSD_TIME_VPOS[7:0]<br><br>Time and Date information vertical position. It is one pixel unit. |

### SPOT3 OSD TIME AND DATE VERTICAL POSITION HIGH BYTE REGISTER – 0XFB7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP3_OSD_TIME_VPOS[10:8]<br><br>Time and Date information vertical position. It is one pixel unit. |

### SPOT3 OSD TITLE HORIZONTAL POSITION LOW BYTE REGISTER – 0XFB8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP3_OSD_TITLE_HPOS[7:0]<br><br>Title information horizontal position. It is one pixel unit. |

### SPOT3 OSD TITLE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XFB9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP3_OSD_TITLE_HPOS[10:8]<br><br>Title information horizontal position. It is one pixel unit. |

### SPOT3 OSD TITLE VERTICAL POSITION LOW BYTE REGISTER – 0XFBA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP3_OSD_TITLE_VPOS[7:0]<br><br>Title information vertical position. It is one pixel unit. |

### SPOT3 OSD TITLE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XFBB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP3_OSD_TITLE_VPOS[10:8]<br><br>Title information vertical position. It is one pixel unit. |

### SPOT3 OSD FONT AND PICTURE RAM ADDRESS LOW BYTE REGISTER – 0XFBC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP3_OSD_FRAM_ADDR[7:0]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT3 OSD FONT AND PICTURE RAM ADDRESS HIGH BYTE REGISTER – 0XFBD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP3_OSD_FRAM_ADDR[10:8]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT3 OSD FONT AND PICTURE RAM DATA LOW BYTE REGISTER – 0XFBE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP3_OSD_FRAM_DATA[7:0]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT3 OSD FONT AND PICTURE RAM DATA HIGH BYTE REGISTER – 0XFBF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP3_OSD_FRAM_DATA[15:8]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT4 CONTROL REGISTER – 0XFC0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_HTT [7:0], SPOT4 Horizontal Total Register<br><br>Lower 8 bits of the SP4_HTT |

### SPOT4 CONTROL REGISTER – 0XFC1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | N/A | | Reserved |
| 4:0 | RW | 0 | SP4_HTT [12:8], SPOT4 Horizontal Total Register<br><br>Upper 5 bits of the SP4_HTT |

### SPOT4 CONTROL REGISTER – 0XFC2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_HDE [7:0], SPOT4 Horizontal Display Enable Register<br><br>Lower 8 bits of the HDE register |

### SPOT4 CONTROL REGISTER – 0XFC3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | N/A | | Reserved |
| 3:0 | RW | 0 | SP4_HDE [11:8], SPOT4 Horizontal Display Enable Register<br><br>Upper 4 bits of the SP4_HDE |

### SPOT4 CONTROL REGISTER – 0XFC4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_VDE [7:0], SPOT4 Vertical Display Enable Register<br><br>Lower 8 bits of the VDE register |

### SPOT4 CONTROL REGISTER – 0XFC5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | N/A | | BND_CH_EN[15:11] Border Channel Enable |
| 3 | N/A | | Reserved |
| 2:0 | RW | 0 | SP4_VDE [11:8], SPOT4 Vertical Display Enable Register<br><br>Upper 3 bits of the Vertical Display Enable Register |

### SPOT4 CONTROL REGISTER – 0XFC6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_PHSYNC [7:0], SPOT4 Horizontal Position Sync Register<br><br>Lower 8 bits of the SP4_PHSYNC<br><br>The register value determine when the horizontal sync start when the horizontal display counter reach to the sum of SP4_HDE and SP4_PHSYNC |

### SPOT4 CONTROL REGISTER – 0XFC7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | N/A | | Reserved |
| 3:2 | RW | 0 | SP4 FRSC Control<br>11: bank_sel equals bank_selB<br>10: bank_sel equals {vblank, fld}<br>01: bank_sel equals {bank_selA[1], fld}<br>00: bank_sel equals bank_selA<br><br>11: wr_page equals wr_pageB<br>10: wr_page equals {~vblank, 0}<br>01: wr_page equals {!bank_selA[1], fld}<br>00: wr_page equals wr_pageA |
| 1 | RW | 0 | SP4_PHB [8], SPOT4 Horizontal Blank Register<br><br>Upper 1 bit of the SP4_PHSYNC |
| 0 | RW | 0 | SP4_PHSYNC [8], SPOT4 Horizontal Position Sync Register<br><br>Upper 1 bit of the SP4_PHSYNC |

### SPOT4 CONTROL REGISTER – 0XFC8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_PVSYNC [7:0], SPOT4 Vertical Position Sync Register<br><br>Lower 8 bits of the SP4_PVSYNC<br><br>The register value determine when the horizontal sync start when the horizontal display counter reach to the sum of SP4_VDE and SP4_PVSYNC |

**SPOT4 CONTROL REGISTER – 0XFC9**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_VTT [7:0], SPOT4 Vertical Total Register |

**SPOT4 CONTROL REGISTER – 0XFCA**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | [7] SP4_Half_D1 Select |
| 6:3 | RW | 0 | Reserve |
| 2:0 | RW | 0 | [2:0] SP4_VTT[10:8] |

**SPOT4 CONTROL REGISTER – 0XFCB**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | SP4_VBLANK_SEL |
| 5 | RW | 0 | SP4_COLOR BAR ENA |
| 4 | RW | 0 | SP4_SWITCH_EN |
| 3:2 | RW | 0 | SP4_FLD select<br><br>11: use bank0 from the FRSC as the field<br>10: use bank0 from the FRSC inverse as the filed<br>0x: use HVCNT generated field |
| 1:0 | RW | 0 | SPOT_POWER_DOWN_EN<br><br>[1] SP4 PD enable when 0<br>[0] SP3 PD enable when 0 |

**SPOT4 CONTROL REGISTER – 0XFCC**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | RW | 0 | REC, NETWORK port use SP3 as the display<br><br>[7] = 1, select Network port to display on SP3<br>= 0, [6:4] selects REC PORT 1 - 8 display on SP3<br>[6:4] = 0, selects REC PORT1,<br>[6:4] = 1, selects REC PORT2,<br>... |
| 3:0 | RW | 0 | REC, NETWORK port use SP4 as the display<br><br>[3] = 1, select Network port to display on SP4<br>= 0, [2:0] selects REC PORT 1 - 8 display on SP4<br>[2:0] = 0, selects REC PORT1,<br>[2:0] = 1, selects REC PORT2,<br>... |

**SPOT4 CONTROL REGISTER – 0XFCD**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_DNS Upper Bits<br>[7:6] SP4 Destination Width upper bit [9:8]<br>[5:3] SP4 Source Height upper bit [10:8]<br>[2:0] SP4 Source Width upper bit [10:8] |

**SPOT4 CONTROL REGISTER – 0XFCE**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | SP4 REC or Network port Enable<br>Enable the REC or Network port to use SP4 as display |
| 6:4 | RW | 0 | SP4 color bar select |
| 3:1 | RW | 0 | SP4_Firsrt_line_control |
| 0 | RW | 0 | SP4 Destination Height upper bit [8] |

**SPOT4 CONTROL REGISTER – 0XFCF**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 1 | SP4 Field Mode Select<br><br>1: Field mode<br>0: Frame mode |
| 6 | RW | 0 | SP4_2ND_MEM Enable |
| 5 | RW | 0 | SP4 FRMCNT_EN SPOT4 Frame Count Enable |
| 4 | RW | 0 | SP4_ENA SPOT4 Enable |
| 3:2 | RW | 0 | SP4_ROT Rotation Enable |
| 1 | RW | 0 | SP4_VSPOL SPOT4 Vertical Polarity Control |
| 0 | RW | 0 | SP4_HSPOL SPOT4 Horizontal Polarity Control |

**SPOT4 CONTROL REGISTER – 0XFD0**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP4_HSTART [7:0], SPOT4 Horizontal Start Register<br><br>This register determines the lower 8 bits of SPOT4 Horizontal start address |

**SPOT4 CONTROL REGISTER – 0XFD1**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 01 | SP4_VSTART [7:0], SPOT4 Vertical Start Register<br><br>This register determines the lower 8 bits of SPOT4 Vertical start address |

**SPOT4 CONTROL REGISTER – 0XFD2**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | N/A | | SP4 Quad Mode Enable<br><br>When Enable, the SP4 memory controller will read half of each line data and start a new a new address calculation from second half line as well as half field for the vertical address to prevent whole screen jumping when weak video is presence.<br><br>This bit should set to one when screen image is set to quad mode. |
| 6 | N/A | | Reserved |
| 5:4 | RW | 0 | SP4_VSTART [9:8], SPOT4 Horizontal Start Register<br><br>Upper 2 bits of the SPOT4 Horizontal start register in 4 pixels per unit |
| 3:0 | RW | 0 | SP4_VSTART [11:8], SPOT4 Vertical Start Register<br><br>Upper 4 bits of the SPOT4 Vertical start register in line per unit |

**SPOT4 CONTROL REGISTER – 0XFD3**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP4_SRC_WID [7:0], SPOT4 Down Scale source width Register<br><br>This register determines the source width of the image. The upper 3 bits are in register 0xFCD. |

**SPOT4 CONTROL REGISTER – 0XFD4**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP4_SRC_HI [7:0], SPOT4 Down Scale source height Register<br><br>This register determines the source height of the image. The upper 3 bits are in register 0xFCD. |

**SPOT4 CONTROL REGISTER – 0XFD5**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP4_DST_WID [7:0], SPOT4 Down Scale Destination width Register<br><br>This register determines the destination width of the image. The upper 2 bits are in register 0xFCD. |

**SPOT4 CONTROL REGISTER – 0XFD6**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 00 | SP4_DST_HI [7:0], SPOT4 Down Scale Destination height Register<br><br>This register determines the D height of the image. The upper 1 bit is in register 0xFCE. |

### SPOT4 CONTROL REGISTER – 0XFD7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | **SP4_TVENC_FSC_MODE, SPOT4 TV FSCSEL**<br><br>This register determines the TV modulation frequency standard select.<br><br>00 - NTSC M<br>01 - PAL B, D, G, H, I<br>10 - PAL M<br>11 - PAL N |
| 5 | RW | 0 | **SP4_TVENC_SAT_LMT, SPOT4 TV video saturation limit**<br><br>Control the data range of the ITU-R BT 656 output<br><br>0 - Not limit<br>1 – Data range is limited to 1 – 254 range |
| 4:1 | N/A | 0 | **Reserved** |
| 0 | RW | 0 | **SP4_TVENC_PALNT, SPOT4 TV PAL or NTSC select**<br><br>This register determines the CVBS output Standard.<br><br>0: NSTC mode<br>1: PAL mode. |

### SPOT4 CONTROL REGISTER – 0XFD8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **SP4_TVENC_VSPOL, SPOT4 TVENC Vertical sync polarity**<br><br>0: Active Low<br>1: Active High |
| 6 | RW | 0 | **SP4_TVENC_HSPOL, SPOT4 TV Encoder Horizontal Sync Polarity**<br><br>0: Active Low<br>1: Active High |
| 5 | RW | 0 | **SP4_TVENC_VSDET, SPOT4 TV Encoder Vsync Det.** |
| 4 | RW | 0 | **SP4_TVENC_PED, SPOT4 TV Encoder Pedestal Enable**<br><br>Enable Pedestal for video output<br><br>0: 0 IRE for pedestal<br>1: 7.5 IRE pedestal level |
| 3 | RW | 0 | **SP4_TVENC_PDRST, SPOT4 TV Encoder Pedestal Reset**<br><br>Subcarrier Phase Alternation reset every 8 fields in PAL system<br><br>0: Constant relationship, reset every 8 fields<br>1: Free running |
| 2 | RW | 0 | **SP4_TVENC_PHALT, SPOT4 TV Encoder Phase Alternation**<br><br>0: Disable phase alternation for line by line<br>1: Enable phase alternation for line by line |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1 | RW | 0 | SP4_TVENC_F_INV, SPOT4 TV Encoder Field Output Invert<br><br>This register bit inverts the field output level if set. |
| 0 | RW | 0 | SP4_TVENC_INTERLACE, SPOT4 TVENC Interlace Select<br><br>This register bit determines the TV Standard in Interlace timing mode.<br><br>0 - Progressive<br>1 - Interlace |

**SPOT4 CONTROL REGISTER – 0XFD9**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | N/A | | Reserved |
| 5:4 | RW | 0 | SP4_TVENC_YC_OS[1:0], SPOT4 TV Encoder Active Pixel Offset delay<br><br>00: No Active offset<br>01: Active pixel offset for 1/4 pixel unit<br>10: Active pixel offset for 1/2 pixel unit<br>11: Active pixel offset for 3/4 pixel unit |
| 3:2 | RW | 0 | SP4_TVENC_OSENC[1:0], SPOT4 TV Encoder Field Offset for the first video line<br><br>00: Apply same ENC_VDSEL for odd and even field<br>01: Apply same ENC_VDSEL for odd and odd field<br>10: Apply TVENC_VSDEL for odd field and TVENC_VSDEL+1 for even field<br>11: Apply TVENC_VSDEL for odd field and TVENC_VSDEL+2 for even field |
| 1 | RW | 0 | SP4_TVENC_FLDENCP, SPOT4 TV Encoder Field Polarity<br><br>This Register control the field polarity<br>0: Even field is high<br>1: Odd field is high |
| 0 | RW | 0 | SP4_TVENC_FLDDET, SPOT4 TV Encoder Field Detect |

**SPOT4 CONTROL REGISTER – 0XFDA**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_TVENC_HSENC[7:0], SPOT4 TV Encoder Active pixel delay<br><br>Lower 8 bits of the pixel delay from active video by ½ pixel per step |

**SPOT4 CONTROL REGISTER – 0XFDB**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | N/A | 0 | SP4_Rotate_Time_Offset |
| 1:0 | RW | 0 | SP4_TVENC_HSENC[9:8], SPOT4 TV Encoder Active pixel delay<br><br>Upper 2 bits of the pixel delay from active video by ½ pixel per step |

**SPOT4 CONTROL REGISTER – 0XFDC**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | N/A | 11 | SP4_CVBS_SEL<br><br>11 : SP4 CVBS output is coming from SP4<br>10 : SP4 CVBS output is coming from SP3<br>01 : SP4 CVBS output is coming from SP2<br>00 : SP4 CVBS output is coming from SP1 |
| 5:0 | RW | 0 | SP4_TVENC_VSENC, SPOT4 TV Encoder Active line delay<br><br>6 bits of the line delay from active video by line per step |

**SPOT4 CONTROL REGISTER – 0XFDD**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | N/A | | Reserved |
| 4:0 | RW | 0 | SP4_TVENC_LINE_OS, SPOT4 TV Encoder Active line offset<br><br>5 bits of the line offset |

**SPOT4 CONTROL REGISTER – 0XFDE**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | N/A | 0 | [7;6] SP4_FLD_SW_MODE<br><br>[7] Field Switching Mode enable<br>[6] Field Switch Mode image height is 240 lines Enable |
| 5:0 | RW | 0 | SP4_TVENC_PIXEL_OS, SPOT4 TV Encoder Active Pixel offset<br><br>6 bits of the pixel offset |

**SPOT4 CONTROL REGISTER – 0XFDF**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | N/A | | Reserved |
| 6 | RW | 0 | SP4_TVENC_C_OFF_D, SPOT4 TV Encoder C Off<br><br>Set the Color Off if set |
| 5:4 | RW | 0 | SP4_TVENC_CBW_D, SPOT4 TV Encoder CBCR delay<br><br>2 bits of the CbCr delay |
| 3:2 | RW | 0 | SP4_TVENC_YBW_D, SPOT4 TV Encoder Luma delay<br><br>2 bits of the Luminous delay |
| 1:0 | RW | 0 | SP4_TVENC_FLD_OS, SPOT4 TV Encoder Active Field offset<br><br>2 bits of the field offset |

**SPOT4 CONTROL REGISTER – 0XFE0**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | SP4_TVENC_TST_FSC_FREE, SPOT4 TV Encoder Sub-Carrier is set to free run |
| 6 | RW | 0 | SP4_TVENC_T_CCIR_TIM, SPOT4 TV Encoder CCOR Timing<br><br>2 bits of the CbCr delay |
| 5 | RW | 0 | SP4_TVENC_T_656_STD, SPOT4 TV Encoder 656 Standard<br><br>2 bits of the Luminous delay |
| 4:0 | RW | 0 | SP4_TVENC_VIS_LINE_OS, SPOT4 TV Encoder line offset<br><br>5 bits Line offset |

**SPOT4 CONTROL REGISTER – 0XFE1**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | RW | | Reserved |
| 4 | RW | 0 | SP4_OSD_TITLE_EN<br><br>Display title enable<br>1: enable<br>0: disable |
| 3 | RW | 0 | SP4_OSD_TIME_EN<br><br>Display time and date enable<br>1: enable<br>0: disable |
| 2 | RW | 0 | SP4_OSD_CHPIC_EN<br><br>Display picture for each channel<br>1: enable<br>0: disable |
| 1 | RW | 0 | SP4_OSD_CHNUM_EN<br><br>Display channel number for each channel<br>1: enable<br>0: disable |
| 0 | RW | 0 | SP4_OSD_EN<br><br>All OSD information enable or disable. If set to 1, each information is enabled by bit [4:1]. If set to 0, all OSD is disabled.<br>1: enable<br>0: disable |

**SPOT4 CONTROL REGISTER – 0XFE2**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0 | Reserved |
| 5 | RW | 0 | SP4_OSD_CHPIC_BLINK<br><br>If set to 1, picture for each channel blink in picture index 2 and 3. Picture 0 and 1 will not blink<br>1: enable<br>0: disable |
| 4 | RW | 0 | SP4_OSD_CHPIC_TRANS<br><br>If set to 1, picture in color index “00” will be transparent<br>1: enable<br>0: disable |
| 3 | RW | 0 | SP4_OSD_TITLE_MIX<br><br>Display title mix enable bit. If set to 1, title will be 50% blending with video<br>1: enable<br>0: disable |
| 2 | RW | 0 | SP4_OSD_TIME_MIX<br><br>Display time and date mix enable bit. If set to 1, time/date will be 50% blending with video<br>1: enable<br>0: disable |
| 1 | RW | 0 | SP4_OSD_CHPIC_MIX<br><br>Display picture mix enable bit. If set to 1, picture will be 50% blending with video<br>1: enable<br>0: disable |
| 0 | RW | 0 | SP4_OSD_CHNUM_MIX<br><br>Display channel number information mix enable bit. If set to 1, channel information will be 50% blending with video<br>1: enable<br>0: disable |

### SPOT4 OSD FONT SIZE REGISTER – 0XFE3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | RW | 0x1 | SP4_OSD_CHPIC_POS<br><br>Channel picture corner position.<br>00: left top<br>01: right top<br>10: left bottom<br>11: right bottom |
| 5:4 | RW | 0 | SP4_OSD_CHNUM_POS<br><br>Channel number corner position.<br>00: left top<br>01: right top<br>10: left bottom<br>11: right bottom |
| 3:2 | RW | 0x3 | SP4_OSD_FONT_VSIZE<br><br>Font vertical size for display. There are four choices.<br>00: 8<br>01: 10<br>10: 16, scale up from 8<br>11: 20, scale up from 10 |
| 1:0 | RW | 0x3 | SP4_OSD_FONT_HSIZE<br><br>Font horizontal size for display. There are four choices.<br>00: 6<br>01: 8<br>10: 12, scale up from 6<br>11: 16, scale up from 8 |

### SPOT4 OSD TIME AND DATE HORIZONTAL POSITION LOW BYTE REGISTER – 0XFE4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_OSD_TIME_HPOS[7:0]<br><br>Time and Date information horizontal position. It is one pixel unit. |

### SPOT4 OSD TIME AND DATE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XFE5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP4_OSD_TIME_HPOS[10:8]<br><br>Time and Date information horizontal position. It is one pixel unit. |

### SPOT4 OSD TIME AND DATE VERTICAL POSITION LOW BYTE REGISTER – 0XFE6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_OSD_TIME_VPOS[7:0]<br><br>Time and Date information vertical position. It is one pixel unit. |

### SPOT4 OSD TIME AND DATE VERTICAL POSITION HIGH BYTE REGISTER – 0XFE7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP4_OSD_TIME_VPOS[10:8]<br><br>Time and Date information vertical position. It is one pixel unit. |

### SPOT4 OSD TITLE HORIZONTAL POSITION LOW BYTE REGISTER – 0XFE8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_OSD_TITLE_HPOS[7:0]<br><br>Title information horizontal position. It is one pixel unit. |

### SPOT4 OSD TITLE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XFE9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP4_OSD_TITLE_HPOS[10:8]<br><br>Title information horizontal position. It is one pixel unit. |

### SPOT4 OSD TITLE VERTICAL POSITION LOW BYTE REGISTER – 0XFEA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_OSD_TITLE_VPOS[7:0]<br><br>Title information vertical position. It is one pixel unit. |

### SPOT4 OSD TITLE HORIZONTAL POSITION HIGH BYTE REGISTER – 0XFEB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP4_OSD_TITLE_VPOS[10:8]<br><br>Title information vertical position. It is one pixel unit. |

### SPOT4 OSD FONT AND PICTURE RAM ADDRESS LOW BYTE REGISTER – 0XFEC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_OSD_FRAM_ADDR[7:0]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT4 OSD FONT AND PICTURE RAM ADDRESS HIGH BYTE REGISTER – 0XFED

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0 | SP4_OSD_FRAM_ADDR[10:8]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT4 OSD FONT AND PICTURE RAM DATA LOW BYTE REGISTER – 0XFEE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_OSD_FRAM_DATA[7:0]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT4 OSD FONT AND PICTURE RAM DATA HIGH BYTE REGISTER – 0XFEF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | RW | 0 | SP4_OSD_FRAM_DATA[15:8]<br><br>Font and Picture SRAM address. When host write data to this SRAM, address is written first and then data. SRAM size is 1152x16. Font size is 640x16, Picture size is 512x16. |

### SPOT CHANNEL0 POSITION MAPPING REGISTER – 0XFF0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | CH0_WR_EN<br><br>Enable spot channel 0 when set |
| 6 | RW | 0 | CH0_VSIZE_SEL<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | CH0_VPOS_SEL, Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | CH0_HSIZE_SEL<br><br>0: CIF<br>1: D1 |
| 2:0 | RW | 0x0 | CH0_HPOS_SEL, Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

### SPOT CHANNEL1 POSITION MAPPING REGISTER – 0XFF1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH1_WR_EN**<br><br>Enable spot channel 1 when set |
| 6 | RW | 0 | **CH1_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH1_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH1_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 2:0 | RW | 0x0 | **CH1_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

### SPOT CHANNEL2 POSITION MAPPING REGISTER – 0XFF2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH2_WR_EN**<br><br>Enable spot channel 1 when set |
| 6 | RW | 0 | **CH2_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH2_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH2_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0x0 | **CH2_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

**SPOT CHANNEL3 POSITION MAPPING REGISTER – 0XFF3**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH3_WR_EN**<br><br>Enable spot channel 3 when set |
| 6 | RW | 0 | **CH3_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH3_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH3_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 2:0 | RW | 0x0 | **CH3_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

### SPOT CHANNEL4 POSITION MAPPING REGISTER – 0XFF4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH4_WR_EN**<br><br>Enable spot channel 4 when set |
| 6 | RW | 0 | **CH4_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH4_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH4_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 2:0 | RW | 0x0 | **CH4_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

### SPOT CHANNEL5 POSITION MAPPING REGISTER – 0XFF5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH5_WR_EN**<br><br>Enable spot channel 5 when set |
| 6 | RW | 0 | **CH5_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH5_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 350 |
| 3 | RW | 0x0 | **CH5_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0x0 | **CH5_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

**SPOT CHANNEL6 POSITION MAPPING REGISTER – 0XFF6**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH6_WR_EN**<br><br>Enable spot channel 6 when set |
| 6 | RW | 0 | **CH6_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH6_VPOS_SEL,** Set Vertical Display Address<br><br>0: 6<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH7_HSIZE_SEL**<br><br>0: C6F<br>1: D1 |
| 2:0 | RW | 0x0 | **CH7_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

### SPOT CHANNEL7 POSITION MAPPING REGISTER – 0XFF7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH7_WR_EN**<br><br>Enable spot channel 7 when set |
| 6 | RW | 0 | **CH7_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH7_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH7_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 2:0 | RW | 0x0 | **CH7_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

### SPOT CHANNEL8 POSITION MAPPING REGISTER – 0XFF8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH8_WR_EN**<br><br>Enable spot channel 8 when set |
| 6 | RW | 0 | **CH8_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH8_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH8_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0x0 | **CH8_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

**SPOT CHANNEL9 POSITION MAPPING REGISTER – 0XFF9**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH9_WR_EN**<br><br>Enable spot channel 9 when set |
| 6 | RW | 0 | **CH9_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH9_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH9_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 2:0 | RW | 0x0 | **CH13_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

### SPOT CHANNEL10 POSITION MAPPING REGISTER – 0XFFA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH10_WR_EN**<br><br>Enable spot channel 10 when set |
| 6 | RW | 0 | **CH10_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH10_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH10_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 2:0 | RW | 0x0 | **CH10_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

### SPOT CHANNEL11 POSITION MAPPING REGISTER – 0XFFB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH11_WR_EN**<br><br>Enable spot channel 11 when set |
| 6 | RW | 0 | **CH11_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH11_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH11_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0x0 | **CH11_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

**SPOT CHANNEL12 POSITION MAPPING REGISTER – 0XFFC**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH12_WR_EN**<br><br>Enable spot channel 12 when set |
| 6 | RW | 0 | **CH12_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH12_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH12_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 2:0 | RW | 0x0 | **CH12_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

### SPOT CHANNEL13 POSITION MAPPING REGISTER – 0XFFD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH13_WR_EN**<br><br>Enable spot channel 13 when set |
| 6 | RW | 0 | **CH13_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH13_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH13_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 2:0 | RW | 0x0 | **CH13_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5: 1680<br>6: 2160<br>7: 2520 |

### SPOT CHANNEL14 POSITION MAPPING REGISTER – 0XFFE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH14_WR_EN**<br><br>Enable spot channel 14 when set |
| 6 | RW | 0 | **CH14_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH14_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH14_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | RW | 0x0 | **CH14_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5:1680<br>6: 2160<br>7: 2520 |

**SPOT CHANNEL15 POSITION MAPPING REGISTER – 0XFFF**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | RW | 0 | **CH15_WR_EN**<br><br>Enable spot channel 15 when set |
| 6 | RW | 0 | **CH15_VSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 5:4 | RW | 0x0 | **CH15_VPOS_SEL,** Set Vertical Display Address<br><br>0: 0<br>1: 120<br>2: 240<br>3: 360 |
| 3 | RW | 0x0 | **CH15_HSIZE_SEL**<br><br>0: CIF<br>1: D1 |
| 2:0 | RW | 0x0 | **CH15_HPOS_SEL,** Set Horizontal Display Address<br><br>0: 0<br>1: 360<br>2: 720<br>3: 1080<br>4: 1440<br>5:1680<br>6: 2160<br>7: 2520 |

# Bitmap OSD for SPOT

## Introduction

**The simple OSD described in the last section is basically used for small icon and character based language caption. For example, a channel number or a sign to signaling user the channel is active, being recorded or just idle or disabled. TW2880 also provide a bitmap OSD for SPOT display to show complex Asian characters or complex graphics. This OSD will occupy display memory to its source data so it will consume bandwidth in the display area.**

## Features

- Supports 4 SPOT
- 1 bitmap pixel is made by 2 bits
- Each pixel has a 3 layer selection
  - 0 (Bitmap Off)
  - 1 layer
  - 2 layer
  - 3 layer
- Mixing support (video + Bitmap data)
- Blink support (0.5s/0.75s)
- 16 color support (13 fixed color and 3 user controllable color)

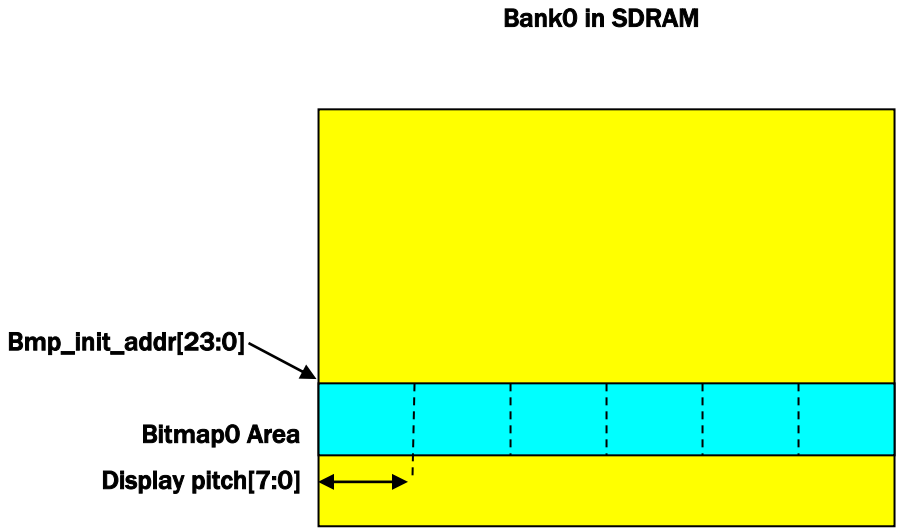
## Theory of Operation

**The TW2880's support Bitmap OSD for SPOT display. User writes bitmap data using host interface. User can control the area in Display RAM. In bitmap enable Bitmap engine reads bitmap data from SDRAM. Each bitmap data choices 4 layer, three characters expect "0" character choices 16 colors, 13 colors are picked and 3 color are control by user.**

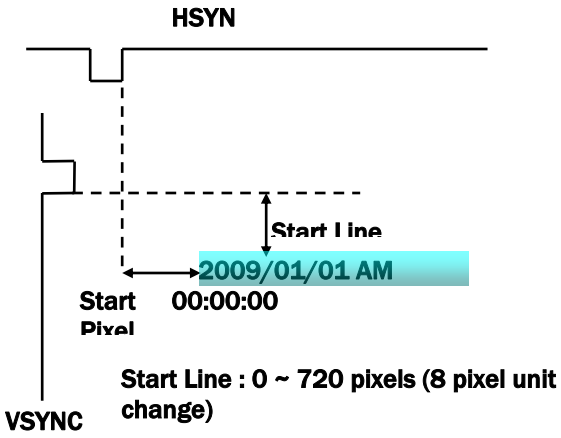
**The blinking period is controlled by the control register. Blinking time is 0.5/0.75sec in high duration and low duration. Blinking can be controlled by character unit.**

**Mixing can be controlled by character unit.**

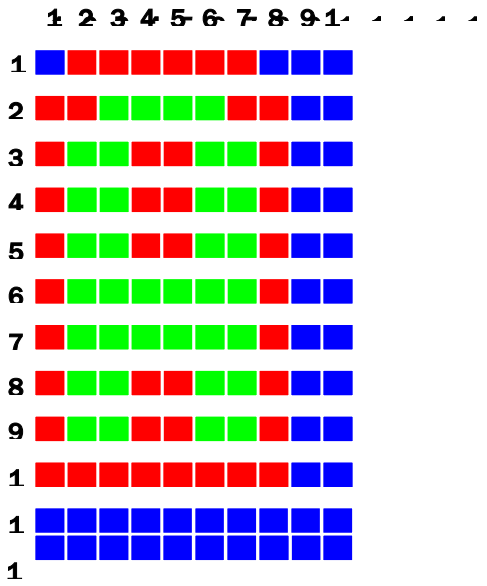
**Ex) Bitmap Area in SDRAM (bank0)**



**Ex) Bitmap Display**




**Ex) Bitmap Data (Font "A")**



**Each dot of font is composite 2 bits.**

**0 : bitmap off**

**1 : 1 layer (blue)**

**2 : 2 layer (red)**

**3 : 3 layer (green)**

## Register Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x080 | R/W | 0x00 | Bitmap source vertical position Register 1 |
| 0x081 | R/W | 0x00 | Bitmap source vertical position Register 2 |
| 0x082 | R/W | 0x00 | Bitmap source horizontal position Register 3 |
| 0x083 | R/W | 0x00 | Bitmap source horizontal position Register 4 |
| 0x088 | R/W | 0x00 | Bitmap source vertical Size Register 1 |
| 0x089 | R/W | 0x00 | Bitmap source vertical Size Register 2 |
| 0x08A | R/W | 0x00 | Bitmap source horizontal Size Register |
| 0x084 | R/W | 0x00 | SPOT1 Bitmap Destination vertical position Register 1 |
| 0x085 | R/W | 0x00 | SPOT1 Bitmap Destination vertical position Register 2 |
| 0x086 | R/W | 0x00 | SPOT1 Bitmap Destination horizontal position Register 3 |
| 0x087 | R/W | 0x00 | SPOT1 Bitmap Destination horizontal position Register 4 |
| 0x08B | R/W | 0x00 | SPOT1 Bitmap Destination vertical Size Register 1 |
| 0x08C | R/W | 0x00 | SPOT1 Bitmap Destination vertical Size Register 2 |
| 0x08D | R/W | 0x00 | SPOT1 Bitmap Destination horizontal Size Register 1 |
| 0x08E | R/W | 0x00 | SPOT1 Bitmap Start Line Register 1 |
| 0x08F | R/W | 0x00 | SPOT1 Bitmap Start Line Register 2 |
| 0x090 | R/W | 0x00 | SPOT1 Bitmap Start Pixel Register 1 |
| 0x091 | R/W | 0x00 | SPOT1 Bitmap Start Pixel Register 2 |
| 0x092 | R/W | 0x00 | SPOT1 Bitmap Control Register 1 |
| 0x093 | R/W | 0x00 | SPOT1 Bitmap Control Register 2 |
| 0x094 | R/W | 0x00 | SPOT1 Bitmap Control Register 3 |
| 0x095 | R/W | 0x00 | SPOT1 Bitmap DMA enable Control Register 3 |
| 0x096 | R/W | 0x00 | SPOT1 Bitmap Control Register 3 |
| 0x097 | R/W | 0x00 | SPOT1 Color look-up table1.Y |
| 0x098 | R/W | 0x00 | SPOT1 Color look-up table1.Cb |
| 0x099 | R/W | 0x00 | SPOT1 Color look-up table1.Cr |
| 0x09A | R/W | 0x00 | SPOT1 Color look-up table2.Y |
| 0x09B | R/W | 0x00 | SPOT1 Color look-up table2.Cb |
| 0x09C | R/W | 0x00 | SPOT1 Color look-up table2.Cr |
| 0x09D | R/W | 0x00 | SPOT1 Color look-up table3.Y |
| 0x09E | R/W | 0x00 | SPOT1 Color look-up table3.Cb |
| 0x09F | R/W | 0x00 | SPOT1 Color look-up table3.Cr |
| 0x0A4 | R/W | 0x00 | SPOT2 Bitmap Destination vertical position Register 1 |
| 0x0A5 | R/W | 0x00 | SPOT2 Bitmap Destination vertical position Register 2 |
| 0x0A6 | R/W | 0x00 | SPOT2 Bitmap Destination horizontal position Register 3 |
| 0x0A7 | R/W | 0x00 | SPOT2 Bitmap Destination horizontal position Register 4 |
| 0x0AB | R/W | 0x00 | SPOT2 Bitmap Destination vertical Size Register 1 |
| 0x0AC | R/W | 0x00 | SPOT2 Bitmap Destination vertical Size Register 2 |
| 0x0AD | R/W | 0x00 | SPOT2 Bitmap Destination horizontal Size Register 1 |
| 0x0AE | R/W | 0x00 | SPOT2 Bitmap Start Line Register 1 |
| 0x0AF | R/W | 0x00 | SPOT2 Bitmap Start Line Register 2 |
| 0x0B0 | R/W | 0x00 | SPOT2 Bitmap Start Pixel Register 1 |
| 0x0B1 | R/W | 0x00 | SPOT2 Bitmap Start Pixel Register 2 |
| 0x0B2 | R/W | 0x00 | SPOT2 Bitmap Control Register 1 |
| 0x0B3 | R/W | 0x00 | SPOT2 Bitmap Control Register 2 |
| 0x0B4 | R/W | 0x00 | SPOT2 Bitmap Control Register 3 |
| 0x0B5 | R/W | 0x00 | SPOT2 Bitmap DMA enable Control Register 3 |
| 0x0B6 | R/W | 0x00 | SPOT2 Bitmap Control Register 3 |
| 0x0B7 | R/W | 0x00 | SPOT2 Color look-up table1.Y |
| 0x0B8 | R/W | 0x00 | SPOT2 Color look-up table1.Cb |
| 0x0B9 | R/W | 0x00 | SPOT2 Color look-up table1.Cr |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x0BA | R/W | 0x00 | SPOT2 Color look-up table2.Y |
| 0x0BB | R/W | 0x00 | SPOT2 Color look-up table2.Cb |
| 0x0BC | R/W | 0x00 | SPOT2 Color look-up table2.Cr |
| 0x0BD | R/W | 0x00 | SPOT2 Color look-up table3.Y |
| 0x0BE | R/W | 0x00 | SPOT2 Color look-up table3.Cb |
| 0x0BF | R/W | 0x00 | SPOT2 Color look-up table3.Cr |
| 0x0C4 | R/W | 0x00 | SPOT3 Bitmap Destination vertical position Register 1 |
| 0x0C5 | R/W | 0x00 | SPOT3 Bitmap Destination vertical position Register 2 |
| 0x0C6 | R/W | 0x00 | SPOT3 Bitmap Destination horizontal position Register 3 |
| 0x0C7 | R/W | 0x00 | SPOT3 Bitmap Destination horizontal position Register 4 |
| 0x0CB | R/W | 0x00 | SPOT3 Bitmap Destination vertical Size Register 1 |
| 0x0CC | R/W | 0x00 | SPOT3 Bitmap Destination vertical Size Register 2 |
| 0x0CD | R/W | 0x00 | SPOT3 Bitmap Destination horizontal Size Register 1 |
| 0x0CE | R/W | 0x00 | SPOT3 Bitmap Start Line Register 1 |
| 0x0CF | R/W | 0x00 | SPOT3 Bitmap Start Line Register 2 |
| 0x0D0 | R/W | 0x00 | SPOT3 Bitmap Start Pixel Register 1 |
| 0x0D1 | R/W | 0x00 | SPOT3 Bitmap Start Pixel Register 2 |
| 0x0D2 | R/W | 0x00 | SPOT3 Bitmap Control Register 1 |
| 0x0D3 | R/W | 0x00 | SPOT3 Bitmap Control Register 2 |
| 0x0D4 | R/W | 0x00 | SPOT3 Bitmap Control Register 3 |
| 0x0D5 | R/W | 0x00 | SPOT3 Bitmap DMA enable Control Register 3 |
| 0x0D6 | R/W | 0x00 | SPOT3 Bitmap Control Register 3 |
| 0x0D7 | R/W | 0x00 | SPOT3 Color look-up table1.Y |
| 0x0D8 | R/W | 0x00 | SPOT3 Color look-up table1.Cb |
| 0x0D9 | R/W | 0x00 | SPOT3 Color look-up table1.Cr |
| 0x0DA | R/W | 0x00 | SPOT3 Color look-up table2.Y |
| 0x0DB | R/W | 0x00 | SPOT3 Color look-up table2.Cb |
| 0x0DC | R/W | 0x00 | SPOT3 Color look-up table2.Cr |
| 0x0DD | R/W | 0x00 | SPOT3 Color look-up table3.Y |
| 0x0DE | R/W | 0x00 | SPOT3 Color look-up table3.Cb |
| 0x0DF | R/W | 0x00 | SPOT3 Color look-up table3.Cr |
| 0x0E4 | R/W | 0x00 | SPOT4 Bitmap Destination vertical position Register 1 |
| 0x0E5 | R/W | 0x00 | SPOT4 Bitmap Destination vertical position Register 2 |
| 0x0E6 | R/W | 0x00 | SPOT4 Bitmap Destination horizontal position Register 3 |
| 0x0E7 | R/W | 0x00 | SPOT4 Bitmap Destination horizontal position Register 4 |
| 0x0EB | R/W | 0x00 | SPOT4 Bitmap Destination vertical Size Register 1 |
| 0x0EC | R/W | 0x00 | SPOT4 Bitmap Destination vertical Size Register 2 |
| 0x0ED | R/W | 0x00 | SPOT4 Bitmap Destination horizontal Size Register 1 |
| 0x0EE | R/W | 0x00 | SPOT4 Bitmap Start Line Register 1 |
| 0x0EF | R/W | 0x00 | SPOT4 Bitmap Start Line Register 2 |
| 0x0F0 | R/W | 0x00 | SPOT4 Bitmap Start Pixel Register 1 |
| 0x0F1 | R/W | 0x00 | SPOT4 Bitmap Start Pixel Register 2 |
| 0x0F2 | R/W | 0x00 | SPOT4 Bitmap Control Register 1 |
| 0x0F3 | R/W | 0x00 | SPOT4 Bitmap Control Register 2 |
| 0x0F4 | R/W | 0x00 | SPOT4 Bitmap Control Register 3 |
| 0x0F5 | R/W | 0x00 | SPOT4 Bitmap DMA enable Control Register 3 |
| 0x0F6 | R/W | 0x00 | SPOT4 Bitmap Control Register 3 |
| 0x0F7 | R/W | 0x00 | SPOT4 Color look-up table1.Y |
| 0x0F8 | R/W | 0x00 | SPOT4 Color look-up table1.Cb |
| 0x0F9 | R/W | 0x00 | SPOT4 Color look-up table1.Cr |
| 0x0FA | R/W | 0x00 | SPOT4 Color look-up table2.Y |
| 0x0FB | R/W | 0x00 | SPOT4 Color look-up table2.Cb |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x0FC | R/W | 0x00 | SPOT4 Color look-up table2.Cr |
| 0x0FD | R/W | 0x00 | SPOT4 Color look-up table3.Y |
| 0x0FE | R/W | 0x00 | SPOT4 Color look-up table3.Cb |
| 0x0FF | R/W | 0x00 | SPOT4 Color look-up table3.Cr |
| 0x23E | R/W | 0x00 | Bitmap Write Data |
| 0x23F | R/W | 0x00 | PB port Clock delay |
| 0x25D | R/W | 0x00 | Bitmap Display Pitch Register |
| 0x265 | R/W | 0x00 | Bitmap Initial Address Register 1 |
| 0x266 | R/W | 0x00 | Bitmap Initial Address Register 2 |
| 0x267 | R/W | 0x00 | Bitmap Initial Address Register 3 |
| 0x249 | R/W | 0x00 | Bitmap Wait Enable Register 1 |
| 0x24A | R/W | 0x00 | Bitmap Wait Enable Register 2 |
| 0x24B | R/W | 0x00 | OSG Bitmap Total Size Register 1 |
| 0x24C | R/W | 0x00 | OSG Bitmap Total Size Register 2 |
| 0x24D | R/W | 0x00 | OSG Bitmap Total Size Register 3 |
| 0x24E | R/W | 0x00 | OSG Bitmap Total Size Register 4 |
| 0x24F | R/W | 0x00 | OSG Bitmap Control Register |

# Registers Definition

### BITMAP SOURCE VERTICAL POSTION REGISTER 1 – 0X080

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Source vertical position[7:0] (1 line) |

### BITMAP SOURCE VERTICAL POSTION REGISTER 2 – 0X081

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | W | 0 | Reserved |
| 4:0 | W | 0 | Source vertical position[12:8] |

### BITMAP SOURCE HORIZONTAL POSITION REGISTER 1 – 0X082

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Source horizontal position[7:0] (32 pixels) |

### BITMAP SOURCE HORIZONTAL POSITION REGISTER 2 – 0X083

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | W | 0 | Reserved |
| 5:4 | W | 0 | Source bank[1:0] |
| 3 | W | 0 | Reserved |
| 2:0 | W | 0 | Source horizontal position[10:8] |

### BITMAP SOURCE VERTICAL SIZE REGISTER 1 – 0X088

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0xf0 | Source vertical size[7:0] (1 line) |

### BITMAP SOURCE VERTICAL SIZE REGISTER 2 – 0X089

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | W | 0 | Reserved |
| 2:0 | W | 0 | Source vertical size [10:8] |

### BITMAP SOURCE HORIZONTAL SIZE REGISTER – 0X08A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | Reserved |
| 6:0 | W | 0x17 | Source horizontal size [6:0] (32 pixels) |

### SPOT1 BITMAP DESTINATION VERTICAL POSTION REGISTER 1 – 0X084

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Destination vertical position[7:0] (1 line) |

### SPOT1 BITMAP DESTINATION VERTICAL POSTION REGISTER 2 – 0X085

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | W | 0 | Reserved |
| 4:0 | W | 0 | Destination vertical position[12:8] |

### SPOT1 BITMAP DESTINATION HORIZONTAL POSITION REGISTER 3 – 0X086

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Destination horizontal position[7:0] (32 pixels) |

### SPOT1 BITMAP DESTINATION HORIZONTAL POSITION REGISTER 4 – 0X087

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | W | 0 | Reserved |
| 2:0 | W | 0 | Destination horizontal position[10:8] |

### SPOT1 BITMAP DESTINATION VERTICAL SIZE REGISTER 1 – 0X08B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0xf0 | Destination vertical size[7:0] (1 line) |

### SPOT1 BITMAP DESTINATION VERTICAL SIZE REGISTER 2 – 0X08C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | W | 0 | Reserved |
| 2:0 | W | 0 | Destination vertical size [10:8] |

### SPOT1 BITMAP DESTINATION HORIZONTAL SIZE REGISTER 1 – 0X08D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | Reserved |
| 6:0 | W | 0x17 | Destination horizontal size [6:0] (32 pixels) |

### SPOT1 BITMAP START LINE REGISTER 1 – 0X08E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Start Line [7:0] (1 line) |

### SPOT1 BITMAP START LINE REGISTER 2 – 0X08F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | W | 0 | Reserved |
| 1:0 | W | 0 | Start Line [9:8] |

### SPOT1 BITMAP START PIXEL REGISTER 1 – 0X090

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Start Pixel [7:0] (1 pixel) |

### SPOT1 BITMAP START PIXEL REGISTER 2 – 0X091

| Bit | R/W | Default | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 7:2 | W | 0 | Reserved |
| 1:0 | W | 0 | Start Pixel [9:8] |

### SPOT1 BITMAP CONTROL REGISTER 1 – 0X092

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | Reserved |
| 6:2 | W | 0 | Blink control<br>[6:5] : blink control<br>"0" : 0.5s/0.5s<br>"1" : 0.5s/0.75s<br>"2 & 3" : 0.75s/0.5s<br><br>[4] : 3 character blink<br>[3] : 2 character blink<br>[2] : 1 character blink |
| 1:0 | W | 0 | Bitmap bank |

### SPOT1 BITMAP CONTROL REGISTER 2 – 0X093

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | Reserved |
| 6:4 | W | 0 | [6] : 3 character mix<br>[5] : 2 character mix<br>[4] : 1 character mix |
| 3:0 | W | 0 | 1 character color select<br>"0" : White<br>"1" : Yellow<br>"2" : Cyan<br>"3" : Green<br>"4" : Magenta<br>"5" : Red<br>"6" : Blue<br>"7" : Black<br>"8" : strong white<br>"9" : light gray<br>"a" : gray<br>"b" : light green<br>"c" : purple<br>"d" : color look-up table1<br>"e" : color look-up table2<br>"f" : color look-up table3 |

### SPOT1 BITMAP CONTROL REGISTER 3 – 0X094

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | W | 2 | 3 character color select |
| 3:0 | W | 1 | 2 character color select |

### BITMAP DMA ENABLE CONTROL REGISTER 3 – 0X095

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | W | 0 | Reserved |

| | | | |
|---|---|---|---|
| 0 | W | 0 | DMA enable |

### SPOT1 BITMAP CONTROL REGISTER 3 – 0X096

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | W | 0 | Reserved |
| 0 | W | 0 | SPOT1 Bitmap enable |

### SPOT1 COLOR LOOK-UP TABLE1.Y – 0X097

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table1 Y value |

### SPOT1 COLOR LOOK-UP TABLE1.CB – 0X098

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table1 Cb value |

### SPOT1 COLOR LOOK-UP TABLE1.CR – 0X099

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table1 Cr value |

### SPOT1 COLOR LOOK-UP TABLE2.Y – 0X09A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table2 Y value |

### SPOT1 COLOR LOOK-UP TABLE2.CB – 0X09B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table2 Cb value |

### SPOT1 COLOR LOOK-UP TABLE2.CR – 0X09C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table2 Cr value |

### SPOT1 COLOR LOOK-UP TABLE3.Y – 0X09D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table3 Y value |

### SPOT1 COLOR LOOK-UP TABLE3.CB – 0X09E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table3 Cb value |

### SPOT1 COLOR LOOK-UP TABLE3.CR – 0X09F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table3 Cr value |

### SPOT2 BITMAP DESTINATION VERTICAL POSTION REGISTER 1 – 0X0A4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Destination vertical position[7:0] (1 line) |

### SPOT2 BITMAP DESTINATION VERTICAL POSTION REGISTER 2 – 0X0A5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | W | 0 | Reserved |
| 4:0 | W | 0 | Destination vertical position[12:8] |

### SPOT2 BITMAP DESTINATION HORIZONTAL POSITION REGISTER 1 – 0X0A6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Destination horizontal position[7:0] (32 pixels) |

### SPOT2 BITMAP DESTINATION HORIZONTAL POSITION REGISTER 2 – 0X0A7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | W | 0 | Reserved |
| 2:0 | W | 0 | Destination horizontal position[10:8] |

### SPOT2 BITMAP DESTINATION VERTICAL SIZE REGISTER 1 – 0X0AB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0xf0 | Destination vertical size[7:0] (1 line) |

### SPOT2 BITMAP DESTINATION VERTICAL SIZE REGISTER 2 – 0X0AC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | W | 0 | Reserved |
| 2:0 | W | 0 | Destination vertical size [10:8] |

### SPOT2 BITMAP DESTINATION HORIZONTAL SIZE REGISTER – 0X0AD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | Reserved |
| 6:0 | W | 0x17 | Destination horizontal size [6:0] (32 pixels) |

### SPOT2 BITMAP START LINE REGISTER 1 – 0X0AE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Start Line [7:0] (1 line) |

### SPOT2 BITMAP START LINE REGISTER 2 – 0X0AF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | W | 0 | Reserved |
| 1:0 | W | 0 | Start Line [9:8] |

### SPOT2 BITMAP START PIXEL REGISTER 1 – 0X0B0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Start Pixel [7:0] (1 pixel) |

### SPOT2 BITMAP START PIXEL REGISTER 2 – 0X0B1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | W | 0 | Reserved |
| 1:0 | W | 0 | Start Pixel [9:8] |

### SPOT2 BITMAP CONTROL REGISTER 1 – 0X0B2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | Reserved |
| 6:2 | W | 0 | Blink control<br>[6:5] : blink control<br>“0” : 0.5s/0.5s<br>“1” : 0.5s/0.75s<br>“2 & 3” : 0.75s/0.5s<br><br>[4] : 3 character blink<br>[3] : 2 character blink<br>[2] : 1 character blink |
| 1:0 | W | 0 | Bitmap bank |

### SPOT2 BITMAP CONTROL REGISTER 2 – 0X0B3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | Reserved |
| 6:4 | W | 0 | [6] : 3 character mix<br>[5] : 2 character mix<br>[4] : 1 character mix |
| 3:0 | W | 0 | 1 character color select<br>“0” : White<br>“1” : Yellow<br>“2” : Cyan<br>“3” : Green<br>“4” : Magenta<br>“5” : Red<br>“6” : Blue<br>“7” : Black<br>“8” : strong white<br>“9” : light gray<br>“a” : gray<br>“b” : light green<br>“c” : purple<br>“d” : color look-up table1<br>“e” : color look-up table2<br>“f” : color look-up table3 |

### SPOT2 BITMAP CONTROL REGISTER 3 – 0X0B4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | W | 2 | 3 character color select |
| 3:0 | W | 1 | 2 character color select |

### SPOT2 BITMAP CONTROL REGISTER 3 – 0X0B6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | W | 0 | Reserved |
| 0 | W | 0 | SPOT2 Bitmap enable |

### SPOT2 COLOR LOOK-UP TABLE1.Y – 0X0B7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table1 Y value |

### SPOT2 COLOR LOOK-UP TABLE1.CB – 0X0B8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table1 Cb value |

### SPOT2 COLOR LOOK-UP TABLE1.CR – 0X0B9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table1 Cr value |

### SPOT2 COLOR LOOK-UP TABLE2.Y – 0X0BA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table2 Y value |

### SPOT2 COLOR LOOK-UP TABLE2.CB – 0X0BB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table2 Cb value |

### SPOT2 COLOR LOOK-UP TABLE2.CR – 0X0BC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table2 Cr value |

### SPOT2 COLOR LOOK-UP TABLE3.Y – 0X0BD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table3 Y value |

### SPOT2 COLOR LOOK-UP TABLE3.CB – 0X0BE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table3 Cb value |

### SPOT2 COLOR LOOK-UP TABLE3.CR – 0X0BF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table3 Cr value |

### SPOT3 BITMAP DESTINATION VERTICAL POSTION REGISTER 1 – 0X0C4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Destination vertical position[7:0] (1 line) |

### SPOT3 BITMAP DESTINATION VERTICAL POSTION REGISTER 2 – 0X0C5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | W | 0 | Reserved |
| 4:0 | W | 0 | Destination vertical position[12:8] |

### SPOT3 BITMAP DESTINATION HORIZONTAL POSITION REGISTER 1 – 0X0C6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Destination horizontal position[7:0] (32 pixels) |

### SPOT3 BITMAP DESTINATION HORIZONTAL POSITION REGISTER 2 – 0X0C7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | W | 0 | Reserved |
| 2:0 | W | 0 | Destination horizontal position[10:8] |

### SPOT3 BITMAP DESTINATION VERTICAL SIZE REGISTER 1 – 0X0CB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0xf0 | Destination vertical size[7:0] (1 line) |

### SPOT3 BITMAP DESTINATION VERTICAL SIZE REGISTER 2 – 0X0CC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | W | 0 | Reserved |
| 2:0 | W | 0 | Destination vertical size [10:8] |

### SPOT3 BITMAP DESTINATION HORIZONTAL SIZE REGISTER – 0X0CD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | Reserved |
| 6:0 | W | 0x17 | Destination horizontal size [6:0] (32 pixels) |

### SPOT3 BITMAP START LINE REGISTER 1 – 0X0CE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Start Line [7:0] (1 line) |

### SPOT3 BITMAP START LINE REGISTER 2 – 0X0CF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | W | 0 | Reserved |
| 1:0 | W | 0 | Start Line [9:8] |

### SPOT3 BITMAP START PIXEL REGISTER 1 – 0X0D0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Start Pixel [7:0] (1 pixel) |

### SPOT3 BITMAP START PIXEL REGISTER 2 – 0X0D1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | W | 0 | Reserved |
| 1:0 | W | 0 | Start Pixel [9:8] |

### SPOT3 BITMAP CONTROL REGISTER 1 – 0X0D2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | Reserved |
| 6:2 | W | 0 | Blink control<br>[6:5] : blink control<br>"0" : 0.5s/0.5s<br>"1" : 0.5s/0.75s<br>"2 & 3" : 0.75s/0.5s<br><br>[4] : 3 character blink<br>[3] : 2 character blink<br>[2] : 1 character blink |
| 1:0 | W | 0 | Bitmap bank |

### SPOT3 BITMAP CONTROL REGISTER 2 – 0X0D3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | Reserved |
| 6:4 | W | 0 | [6] : 3 character mix<br>[5] : 2 character mix<br>[4] : 1 character mix |
| 3:0 | W | 0 | 1 character color select<br>"0" : White<br>"1" : Yellow<br>"2" : Cyan<br>"3" : Green<br>"4" : Magenta<br>"5" : Red<br>"6" : Blue<br>"7" : Black<br>"8" : strong white<br>"9" : light gray<br>"a" : gray<br>"b" : light green<br>"c" : purple<br>"d" : color look-up table1<br>"e" : color look-up table2<br>"f" : color look-up table3 |

### SPOT3 BITMAP CONTROL REGISTER 3 – 0X0D4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | W | 2 | 3 character color select |
| 3:0 | W | 1 | 2 character color select |

### SPOT3 BITMAP CONTROL REGISTER 3 – 0X0D6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | W | 0 | Reserved |
| 0 | W | 0 | SPOT3 Bitmap enable |

### SPOT3 COLOR LOOK-UP TABLE1.Y – 0X0D7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table1 Y value |

### SPOT3 COLOR LOOK-UP TABLE1.CB – 0X0D8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table1 Cb value |

### SPOT3 COLOR LOOK-UP TABLE1.CR – 0X0D9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table1 Cr value |

### SPOT3 COLOR LOOK-UP TABLE2.Y – 0X0DA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table2 Y value |

### SPOT3 COLOR LOOK-UP TABLE2.CB – 0X0DB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table2 Cb value |

### SPOT3 COLOR LOOK-UP TABLE2.CR – 0X0DC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table2 Cr value |

### SPOT3 COLOR LOOK-UP TABLE3.Y – 0X0DD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table3 Y value |

### SPOT3 COLOR LOOK-UP TABLE3.CB – 0X0DE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table3 Cb value |

### SPOT3 COLOR LOOK-UP TABLE3.CR – 0X0DF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table3 Cr value |

### SPOT4 BITMAP DESTINATION VERTICAL POSTION REGISTER 1 – 0X0E4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Destination vertical position[7:0] (1 line) |

### SPOT4 BITMAP DESTINATION VERTICAL POSTION REGISTER 2 – 0X0E5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | W | 0 | Reserved |
| 4:0 | W | 0 | Destination vertical position[12:8] |

### SPOT4 BITMAP DESTINATION HORIZONTAL POSITION REGISTER 3 – 0X0E6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Destination horizontal position[7:0] (32 pixels) |

### SPOT4 BITMAP DESTINATION HORIZONTAL POSITION REGISTER 4 – 0X0E7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | W | 0 | Reserved |
| 2:0 | W | 0 | Destination horizontal position[10:8] |

### SPOT4 BITMAP DESTINATION VERTICAL SIZE REGISTER 1 – 0X0EB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0xf0 | Destination vertical size[7:0] (1 line) |

### SPOT4 BITMAP DESTINATION VERTICAL SIZE REGISTER 2 – 0X0EC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | W | 0 | Reserved |
| 2:0 | W | 0 | Destination vertical size [10:8] |

### SPOT4 BITMAP DESTINATION HORIZONTAL SIZE REGISTER – 0X0ED

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | Reserved |
| 6:0 | W | 0x17 | Destination horizontal size [6:0] (32 pixels) |

### SPOT4 BITMAP START LINE REGISTER 1 – 0X0EE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Start Line [7:0] (1 line) |

### SPOT4 BITMAP START LINE REGISTER 2 – 0X0EF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | W | 0 | Reserved |
| 1:0 | W | 0 | Start Line [9:8] |

### SPOT4 BITMAP START PIXEL REGISTER 1 – 0X0F0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Start Pixel [7:0] (1 pixel) |

### SPOT4 BITMAP START PIXEL REGISTER 2 – 0X0F1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | W | 0 | Reserved |
| 1:0 | W | 0 | Start Pixel [9:8] |

### SPOT4 BITMAP CONTROL REGISTER 1 – 0X0F2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | Reserved |
| 6:2 | W | 0 | Blink control<br>[6:5] : blink control<br>“0” : 0.5s/0.5s<br>“1” : 0.5s/0.75s<br>“2 & 3” : 0.75s/0.5s<br><br>[4] : 3 character blink<br>[3] : 2 character blink<br>[2] : 1 character blink |
| 1:0 | W | 0 | Bitmap bank |

### SPOT4 BITMAP CONTROL REGISTER 2 – 0X0F3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | W | 0 | Reserved |
| 6:4 | W | 0 | [6] : 3 character mix<br>[5] : 2 character mix<br>[4] : 1 character mix |
| 3:0 | W | 0 | 1 character color select<br>“0” : White<br>“1” : Yellow<br>“2” : Cyan<br>“3” : Green<br>“4” : Magenta<br>“5” : Red<br>“6” : Blue<br>“7” : Black<br>“8” : strong white<br>“9” : light gray<br>“a” : gray<br>“b” : light green<br>“c” : purple<br>“d” : color look-up table1<br>“e” : color look-up table2<br>“f” : color look-up table3 |

### SPOT4 BITMAP CONTROL REGISTER 3 – 0X0F4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | W | 2 | 3 character color select |
| 3:0 | W | 1 | 2 character color select |

### SPOT4 BITMAP CONTROL REGISTER 3 – 0X0F6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | W | 0 | Reserved |
| 0 | W | 0 | SPOT4 Bitmap enable |

### SPOT4 COLOR LOOK-UP TABLE1.Y – 0X0F7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table1 Y value |

### SPOT4 COLOR LOOK-UP TABLE1.CB – 0X0F8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table1 Cb value |

### SPOT4 COLOR LOOK-UP TABLE1.CR – 0X0F9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table1 Cr value |

### SPOT4 COLOR LOOK-UP TABLE2.Y – 0X0FA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table2 Y value |

### SPOT4 COLOR LOOK-UP TABLE2.CB – 0X0FB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table2 Cb value |

### SPOT4 COLOR LOOK-UP TABLE2.CR – 0X0FC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table2 Cr value |

### SPOT4 COLOR LOOK-UP TABLE3.Y – 0X0FD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table3 Y value |

### SPOT4 COLOR LOOK-UP TABLE3.CB – 0X0FE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table3 Cb value |

### SPOT4 COLOR LOOK-UP TABLE3.CR – 0X0FF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Color look-up table3 Cr value |

### BITMAP WRITE DATA – 0X23E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Bitmap Write Data |

### PB INPUT CLOCK DELAY REGISTER – 0X23F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | W | 0 | PB Port 4 Clock Delay Control |
| 5:4 | W | 0 | PB Port 3 Clock Delay Control |
| 3:2 | W | 0 | PB Port 2 Clock Delay Control |
| 1:0 | W | 0 | PB Port 1 Clock Delay Control<br><br>“00” : No delay<br>“01” : 0.6 ns (typical)<br>“10” : 1.2 ns (typical)<br>“11” : 1.8 ns (typical) |

### BITMAP DISPLAY PITCH REGISTER 1 – 0X25D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Display pitch [7:0] |

### BITMAP INITIAL ADDRESS REGISTER 1 – 0X265

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Bmp_init_addr [7:0] |

### BITMAP INITIAL ADDRESS REGISTER 2 – 0X266

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Bmp_init_addr [15:8] |

### BITMAP INITIAL ADDRESS REGISTER 3 – 0X267

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | Bmp_init_addr [23:16] |

### BITMAP WAIT ENABLE REGISTER 1 – 0X249

| Bit | R/W | Default | Description |
|---|---|---|---|
| 4 | W | 0 | Bmp_wait_pin_en<br><br>Bitmap wait enable for interrupt Pin |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 1 | W | 0 | Bmp_wait_en<br><br>Bitmap wait enable for interrupt |
| 0 | W | 0 | Bmp_Big_Endian<br><br>"0" : Little Endian<br>"1" : Big Endian |

**This register is shared with Host DMA Transfer Mode Control Register at P.360**

### BITMAP WAIT ENABLE REGISTER 2 – 0X24A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R | 0 | Reserved |
| 3 | R | 0 | Write DMA done status |
| 1 | R | 0 | Bmp_waitb<br><br>"0" : busy status<br>"1" : non-busy status |

### OSG BITMAP TOTAL SIZE REGISTER 1 – 0X24B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | OSG_BMP_total_size [7:0] |

### OSG BITMAP TOTAL SIZE REGISTER 2 – 0X24C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | OSG_BMP_total_size [15:8] |

### OSG BITMAP TOTAL SIZE REGISTER 3 – 0X24D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | W | 0 | OSG_BMP_total_size [23:0] |

### OSG BITMAP TOTAL SIZE REGISTER 4 – 0X24E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 2:0 | W | 0 | OSG_BMP_total_size [26:24] |

### OSG BITMAP CONTROL REGISTER – 0X24F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | R | 0 | Reserved |
| 0 | R | 0 | Vavb : Vertical sync inverse signal |

# Host Interface

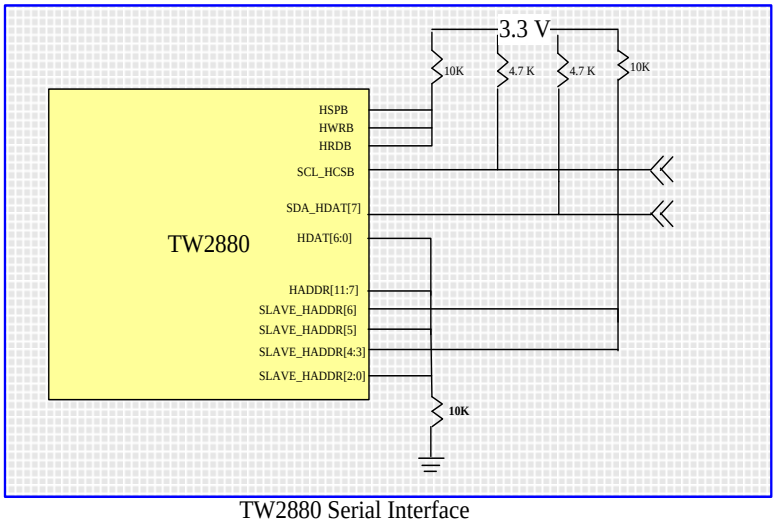
## Introduction

TW2880 provides three kind of interfaces to meet different needs of the customers. There are serial host interface, parallel host interface and DMA interface. The selection is done by setting HSPB pin high or low. When HSPB is low, the parallel interface is selected, while the EN16B pin is used for 16 bit indication or not. When HSPB pin is set to high, the serial interface is selected. Some of the interface pins served as dual purpose pins depending on the working mode. The pin HCSB and the HDAT[7] in the parallel mode become SCLK and SDAT pins in serial mode and the pins HADDR[6:0] in parallel mode become slave address in the serial mode. Each interface protocol is shown in the following figures.

## Pin Assignments for Serial and Parallel Interface

| Pin Name | Serial Mode | Parallel Mode | DMA MODE | |
|---|---|---|---|---|
| HSPB | HIGH | LOW | LOW | Function select |
| EN16B | LOW | EN16B | EN16B | 8/16 select |
| HCSB | SCLK | PCSB | PCSB | Chip Select |
| HWRB | Not Used (Set High) | PWRB | PWRB | Write pulse |
| HRDB | Not Used (Set High) | PRDB | PRDB | Read pulse |
| HADDR [6:0] | Slave Addr[6:0] | Parallel Addr[6:0] | Parallel Addr[6:0] | Address |
| HADDR [11:7] | Not Used (Set Low) | Parallel Addr[11:7] | Parallel Addr[11:7] | Address |
| HDAT [6:0] | Not Used (Set Low) | Parallel D[6:0] | DMA D[6:0] | Data |
| HDAT [7] | SDAT | Parallel D[7] | DMA D[7] | Data |
| HDAT [15:8] | Not Used (Set Low) | Parallel D[15:8] | DMA D[15:8] | Data |
| BT601_V1 | Not Used (Set Low) | Not Used (Set Low) | HDONE (Set Low if not used) | Done |
| BT601_H1 | Not Used | Not Used | HDACK | Acknowledge |
| HDAT_IOB | Not Used | Not Used | HDREQ | Request |
| P_WAIT_ST | Not Used | P_WAIT_ST | Not Used | Wait signal |

# Serial Interface

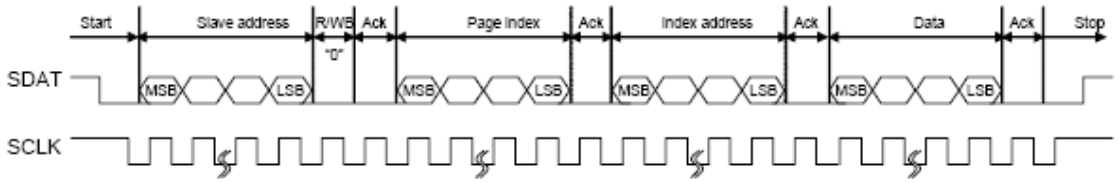



TW2880 Serial Interface

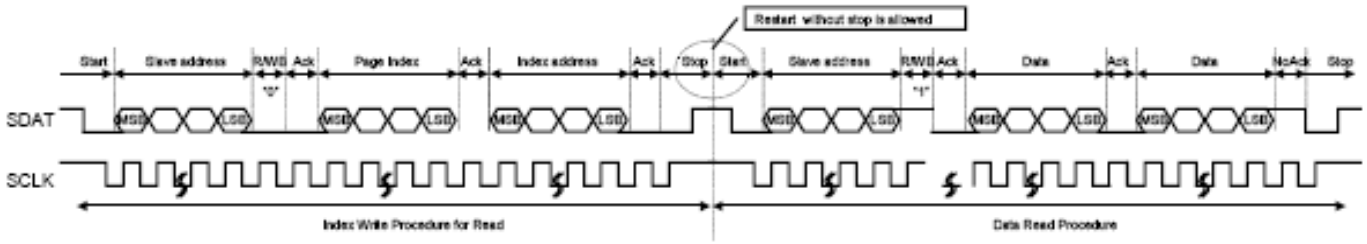
**In the Serial Mode, there are seven address pins can be assigned to any combination as the I²C slave address for full flexibility. The figure below shows an illustration of the serial interface for the case of the slave address set for Read as "0xB1" and Write as "0xB0".**

**TW2880 has total of 16 pages for registers with each page contains 256 eight bit registers. So that the page index[3:0] is used to select page of registers, Indexed address is used to select the registers in each page. For each page assignment, please refer to the correspondent module. Following is the detail timing diagram.**

**TW2880 also supports automatic index increment so that it can read or write continuous multi-bytes without restart. Therefore, The host can read or write multiple bytes in sequential order without writing additional slave address, page index and index address. With the automatic index increment function, the data transfer performance is highly improved and transfer rate on the bus is up t0 400 Kbits per second.**




**WRITE TIMING OF THE SERIAL INTERFACE**



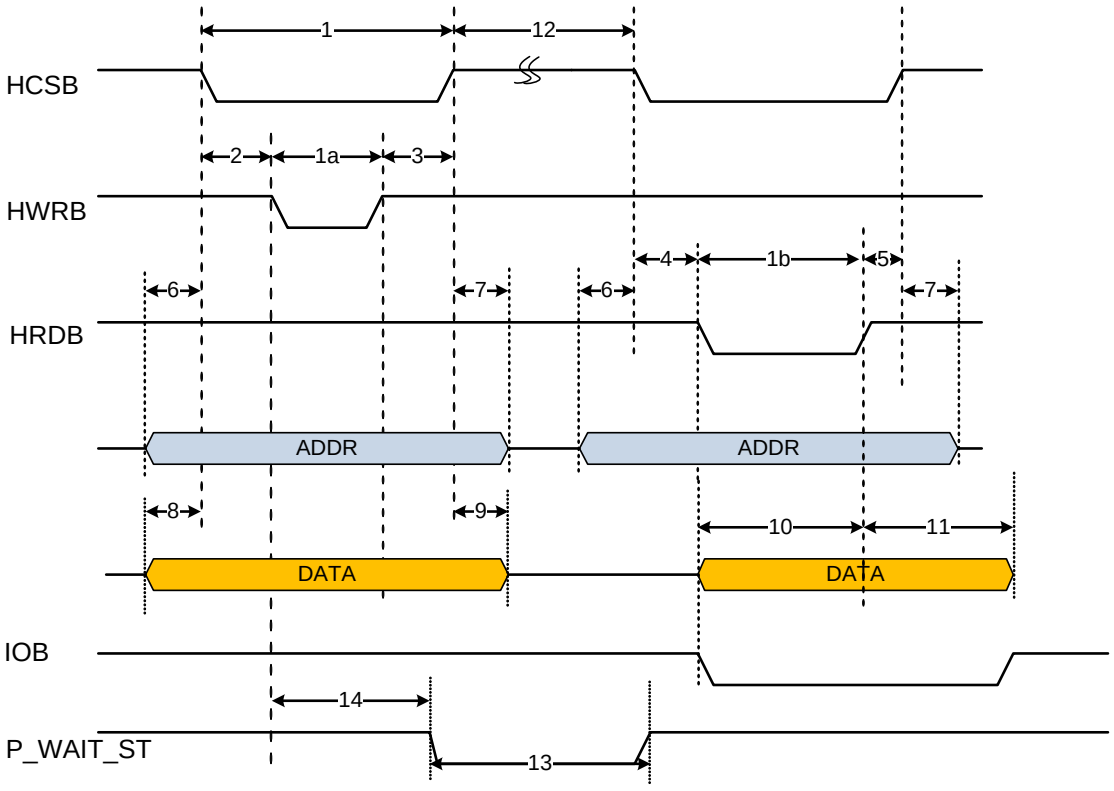
## Parallel Interface

The parallel interface consists of 16 bi-directional data pins (HDAT[15:0]), read /write signal (HWRB, HRDB), and chip select signal (HCSB). When parallel interface is the intended interface, the HSPB pin must set to low. All bus transaction cycles are initiated by pulling HCSB is low. For a write cycle, HWRB is pulled low the whole cycle and the HRDB should maintain high, then TW2880 is to set in write mode and is ready to receive data from coming from the HDAT[7:0]. The data latching is done in the beginning of the HWRB pulse. If the HWRB is set to HIGH and HRDB is set to low, the TW2880 is set in read mode and ready to send data through HDAT[7:0] to the host. If reading TW2880's registers, the read data will become valid after a fixed access time when the HRDB pulse go high. If reading TW2880's SDRAM content, the access time will depend on the activities on the DRAM and is not a constant. User need to use HDAT_IOB as a data valid signal.

Address space arrangement in parallel interface mode looks like this: twelve bit address with the upper four bits served as page selector and the lower eight bits served as indexed address. Altogether there are 16 pages and each with 256 (0x00-0xFF) selections. For all registers definition, please refer to the related module to in the TW2880 specification.

### SINGLE TRANSFER

The single transfer throughput of the parallel host port is about 20MB/sec (sustained). This is calculated from the waveform below. One transaction is one HCSB pulse and the waiting time is 70+30 = 100 nS. So 8 bit input is 10 MB/sec and 16 bit input is 20 MB/sec. Some host CPU has wait signal in the design, P_WAIT_ST is designed to use in this situation as it will act as a throttle to control the data write action.



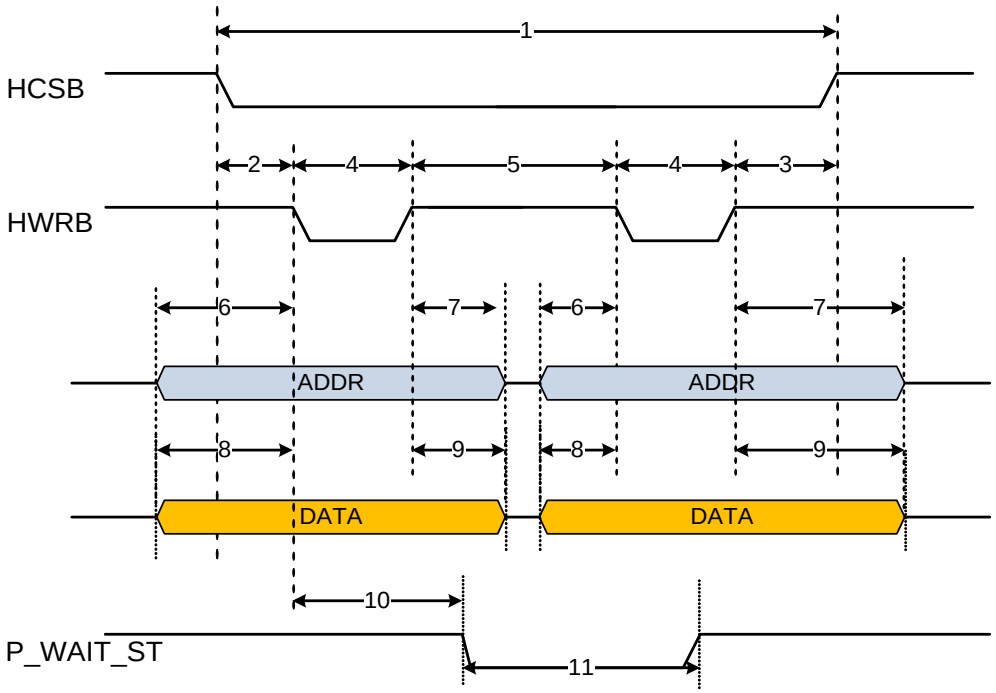
| | Parameter | Symbol | Min | Max | Units |
|---|---|---|---|---|---|
| 1, 1a, 1b | HCSB, HWRB, HRDB pulse width<br>Address Data latching time | Tcs | 30 | N/A | ns |
| 2 | HCSB low to HWRB low | Hwrb_tsu | 0 | N/A | ns |
| 3 | HWRB high to HCSB high | Hwrb_th | 0 | N/A | ns |
| 4 | HCSB low to HRDB low | Hrdb_tsu | 0 | N/A | ns |
| 5 | HRDB high to HCSB high | Hrdb_th | 0 | N/A | ns |
| 6 | HADDR setup to HCSB low | Haddr_tsu | 2 | N/A | ns |
| 7 | HADDR hold after HCSB high | Haddr_th | 0 | N/A | ns |
| 8 | HDAT write setup to HCSB low | Hdat_wr_tsu | 5 | N/A | ns |
| 9 | HDAT write hold after HCSB high | Hdat_wr_th | 0 | N/A | ns |
| 10 | HDAT read delay to HRDB inactive | Hdat_trds | 10 | 30 | ns |
| 11 | HDAT read hold after HRDB inactive | Hdat_trdh | 0 | 60 | ns |
| 12 | HCSB inactive pulse width | Tcsn | 70 | - | ns |
| 13 | CPU Halt to TW2880 access | TcpuH | 30 | - | ns |
| 14 | Write pulse low to wait signal low | Twait | 30 | 35 | ns |

**TIMING PARAMETERS OF THE PARALLEL INTERFACE**

**BURST TRANSFER**

**For the OSG write data transfer (write through register port 0x13A), TW2880 provides a burst write mode to increase the transfer speed. The transfer throughput at this mode can reach 30 MB/sec (sustained). This is calculated from the waveform below. After HCSB is pulled low to initiate the transaction, the HWRB pulse can be repeal toggled to perform the write operations while the HCSB signal is kept low. One transaction is one HWRB pulse and the waiting time is 33+33 = 66 nS. So 8 bit input is 15 MB/sec and 16 bit input is 30 MB/sec.**

**In this mode P_WAIT_ST can also be used to halt the write operation once the FIFO inside is full. Burst read operation is also available but not as important as write operation.**




| | Parameter | Symbol | Min | Max | Units |
|---|---|---|---|---|---|
| 1 | HCSB pulse width | Tcs | 33 | N/A | ns |
| 2 | HCSB low to HWRB low | Hwrb_tsu | 0 | N/A | ns |
| 3 | HWRB high to HCSB high | Hwrb_th | 0 | N/A | ns |
| 4 | HWRB pulse width | Hwr | 33 | N/A | ns |
| 5 | Spacing between HWRB pulses | Hwr_sp | 33 | N/A | ns |
| 6 | HADDR setup to HWRB low | Haddr_tsu | 2 | N/A | ns |
| 7 | HADDR hold after HWRB high | Haddr_th | 0 | N/A | ns |
| 8 | HDAT write setup to HWRB low | Hdat_wr_tsu | 5 | N/A | ns |
| 9 | HDAT write hold after HWRB high | Hdat_wr_th | 0 | N/A | ns |
| 10 | Write pulse low to wait signal low | Twait | 30 | 35 | ns |
| 11 | CPU Halt to TW2880 access | TcpuH | 30 | - | ns |

## External Host Interface Connection

As stated in the previous section there are two data width connection methods in TW2880: 8 bit and 16 bit connection. 8 bit connection is rather straight forward, user only needs to tie EN16B to low and connect CPU address [11:0] to HADDR[11:0], CPU data [7:0] to HDAT[7:0] and start accessing but basically treat TW2880 as an 8 bit peripheral.

On the other hand, when high OSG data throughput is needed and user may elect to connect TW2880 as a 16 bit device, In this usage user needs to connect EN16B along with other control signals, connect CPU address [12:1] to HADDR[11:0], CPU data [15:0] to HDAT[15:0]. In this way, the CPU address used to access TW2880 needs to be multiplied by two. For TW2880 registers access, CPU should set EN16B to low, data is to / from HDAT[7:0], upper byte is ignored (except 0x13A). For memory access, CPU needs to set EN16B to high, and data is to / from hdat[15:0]. EN16B timing requirement follows the requirement of HCSB.

## External Host to Display Memory access Interface

TW2880 allows the external host get access to the display memory. This feature is provided mainly for debugging purpose because in normal operations user does not need to read or write the content of the DRAM. To use this feature user only need to program the 24 bit linear starting address into the three registers (0x000, 0x001, 0x002), then program the DRAM access control register 0x003 to specify the read or the write operation and the burst length. Once these registers are programmed, for write operations users need to write to port register 0x004 continuously until the number of bytes written equal to the burst length. For read operation user needs to read the port register continuously to get all the data requested.

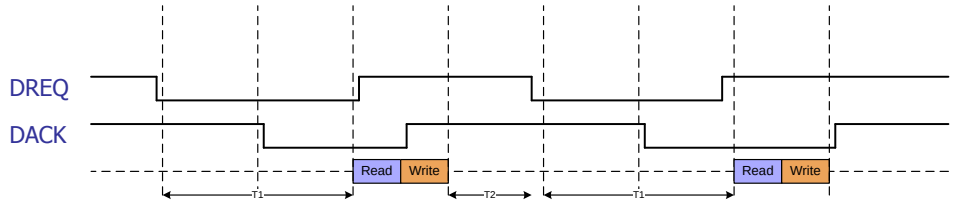
## DMA Interface

The DMA interface in TW2880 is designed to facilitate the high speed transfer of the bitmap and menu data from the host to TW2880. TW2880 DMA interface served as a master when doing DMA transfer. Two asynchronously sampled pin: HDREQ, HDACK and the 16 bit data bus are used.

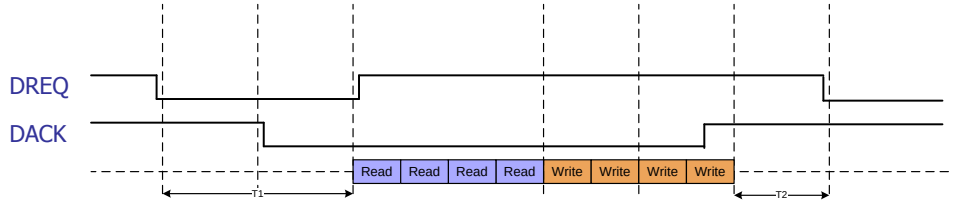
To start a DMA transaction, user first set the proper timing parameters, transaction length and direction in the DMAC register (page 2). When enable the DMA channel, HDREQ will go active to signaling the host a DMA request is made, DMAC then will wait for HDACK to go active, meaning after some time valid data will appear or we need to supply data to the data bus. After the first transaction is done, TW2880 will need to wait for a while before starting the sequence again.

The burst size we supported can be one or four. Also if setting the total count larger than one, TW2880 can accept multiple data stream into the TW2880. Please also see the “DMAC” chapter for the destination arrangement.

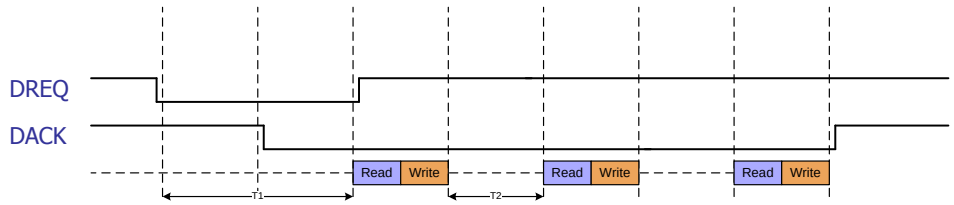
Unit Transfer, Burst Size = 1




Unit Transfer, Burst Size = 4

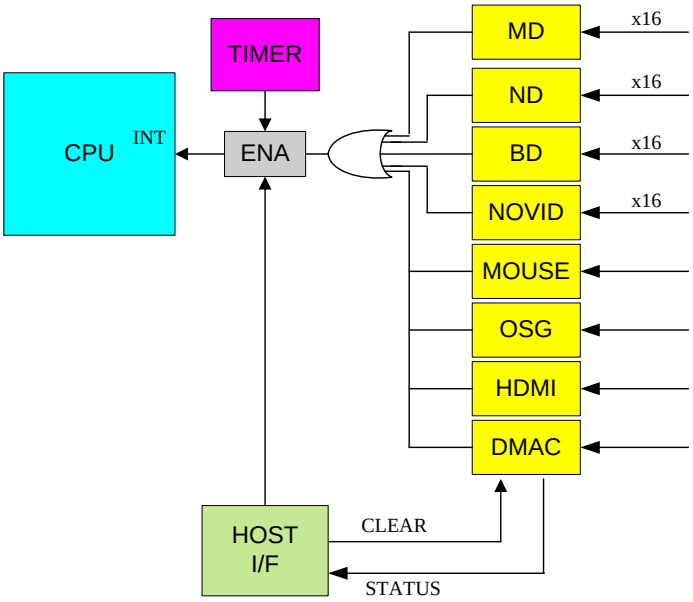



Multiple Transfer, Burst Size = 1




## Interrupt Interface

**The TW2880 provides a very sophisticated interrupt request function for user to inter-react with the host CPU. Any video loss, motion, blind, or night detection in every channel will generate an interrupt request to the host CPU. The polarity of the interrupt is selectable by the user. The user can disable the one single interrupt function for each channel and category. After receiving the interrupt, the host can distinguish which functional unit generated the interrupt by writing to the interrupt status registers. User can choose to read the real time detection status of the all functional units by changing a bit. The interrupt can be cleared by reading the interrupt clearing registers.**



**A set of idle and resend counters is incorporated in the interrupt generation process to help easing the burden of the CPU is responding and switching between different interrupt service routines. Once an interrupt is raised and does not get attention of CPU for certain period of time, the interrupt of TW2880 will become inactive for a while and become active again. This is process will go on indefinably until the unit get reset. This function can be disabled by user.**

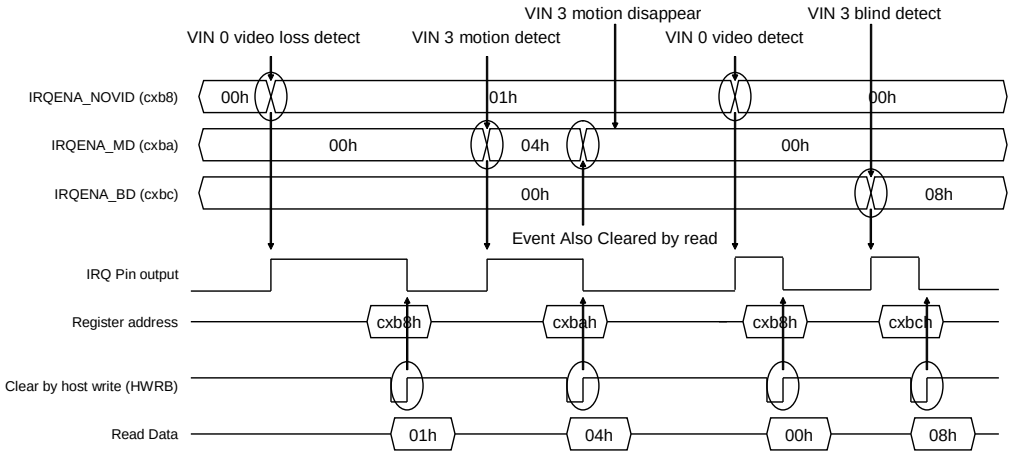



**ILLUSTRATION OF THE INTERRUPT GENERATED AND CLEARED SEQUENCE**

## Register Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0xCB0 | R/W | 0x00 | No video IRQ Enable Register 1 |
| 0xCB1 | R/W | 0x00 | No video IRQ Enable Register 2 |
| 0xCB2 | R/W | 0x00 | Motion Detection IRQ Enable Register 1 |
| 0xCB3 | R/W | 0x00 | Motion Detection IRQ Enable Register 2 |
| 0xCB4 | R/W | 0x00 | Blind Detection IRQ Enable Register 1 |
| 0xCB5 | R/W | 0x00 | Blind Detection IRQ Enable Register 2 |
| 0xCB6 | R/W | 0x00 | Night Detection IRQ Enable Register 1 |
| 0xCB7 | R/W | 0x00 | Night Detection IRQ Enable Register 2 |
| 0xCB8 | R/W | 0x00 | No video IRQ Status Register 1 |
| 0xCB9 | R/W | 0x00 | No video IRQ Status Register 2 |
| 0xCBA | R/W | 0x00 | Motion Detection IRQ Status Register 1 |
| 0xCBB | R/W | 0x00 | Motion Detection IRQ Status Register 2 |
| 0xCBC | R/W | 0x00 | Blind Detection IRQ Status Register 1 |
| 0xCBD | R/W | 0x00 | Blind Detection IRQ Status Register 2 |
| 0xCBE | R/W | 0x00 | Night Detection IRQ Status Register 1 |
| 0xCBF | R/W | 0x00 | Night Detection IRQ Status Register 2 |
| 0xCC0 | R/W | 0x00 | IRQ Main Control Register |
| 0xCC1 | R/W | 0x00 | IRQ Spacing Register |
| 0xCC2 | R/W | 0x00 | IRQ Report Counter Register 1 |
| 0xCC3 | R/W | 0x00 | IRQ Report Counter Register 2 |
| 0xCC4 | R/W | 0x00 | IRQ Report Counter Register 3 |
| 0xCC6 | R/W | 0x00 | AUX. IRQ Status Register |
| 0xCC7 | R/W | 0x00 | MPOUT Selection Register |
| 0x000 | R/W | 0x00 | Display DRAM Address Register 0 |
| 0x001 | R/W | 0x00 | Display DRAM Address Register 1 |
| 0x002 | R/W | 0x00 | Display DRAM Address Register 2 |
| 0x003 | R/W | 0x00 | Display DRAM Access Control Register |
| 0x004 | R/W | 0x00 | Display DRAM Data Port Register |

# Registers Description

### NO VIDEO IRQ ENABLE REGISTER 1 – 0XCB0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | IRQENA_NOVID[7:0] for channel 7:0<br><br>1: enable<br>0: disable |

### NO VIDEO IRQ ENABLE REGISTER 2 – 0XCB1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | IRQENA_NOVID[15:8] for channel 15:8<br><br>1: enable<br>0: disable |

### MOTION DETECTION IRQ ENABLE REGISTER 1 – 0XCB2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | IRQENA_MD[7:0] for channel 7:0<br><br>1: enable<br>0: disable |

### MOTION DETECTION IRQ ENABLE REGISTER 2 – 0XCB3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | IRQENA_MD[15:8] for channel 15:8<br><br>1: enable<br>0: disable |

### BLIND DETECTION IRQ ENABLE REGISTER 1 – 0XCB4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | IRQENA_BD[7:0] for channel 7:0<br><br>1: enable<br>0: disable |

### BLIND DETECTION IRQ ENABLE REGISTER 2 – 0XCB5

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | IRQENA_BD[15:8] for channel 15:8<br><br>1: enable<br>0: disable |

### NIGHT DETECTION IRQ ENABLE REGISTER 1 – 0XCB6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | IRQENA_ND[7:0] for channel 7:0<br><br>1: enable<br>0: disable |

### NIGHT DETECTION IRQ ENABLE REGISTER 2 – 0XCB7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | IRQENA_ND[15:8] for channel 15:8<br><br>1: enable<br>0: disable |

### NO VIDEO IRQ STATUS REGISTER 1 – 0XCB8

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0 | Irq_rd_novid[7:0] for channel 7:0<br><br>1: Interrupt<br>0: idle |

### NO VIDEO IRQ STATUS REGISTER 2 – 0XCB9

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0 | Irq_rd_novid[15:8] for channel 15:8<br><br>1: interrupt<br>0: idle |

### MOTION DETECTION IRQ STATUS REGISTER 1 – 0XCBA

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0 | Irq_rd_md[7:0] for channel 7:0<br><br>1: interrupt<br>0: idle |

### MOTION DETECTION IRQ STATUS REGISTER 2 – 0XCBB

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0 | Irq_rd_md[15:8] for channel 15:8<br><br>1: interrupt<br>0: idle |

### BLIND DETECTION IRQ STATUS REGISTER 1 – 0XCBC

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0 | **Irq_rd_bd[7:0] for channel 7:0**<br><br>1: interrupt<br>0: idle |

### BLIND DETECTION IRQ STATUS REGISTER 2 – 0XCBD

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0 | **Irq_rd_bd[15:8] for channel 15:8**<br><br>1: interrupt<br>0: idle |

### NIGHT DETECTION IRQ STATUS REGISTER 1 – 0XCBE

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0 | **Irq_rd_nd[7:0] for channel 7:0**<br><br>1: interrupt<br>0: idle |

### NIGHT DETECTION IRQ STATUS REGISTER 2 – 0XCBF

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0 | **Irq_rd_nd[15:8] for channel 15:8**<br><br>1: interrupt<br>0: idle |

### IRQ MAIN CONTROL REGISTER —OFFSET 0XCC0

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | **IRQENA_RD_MD**<br><br>0 = normal<br>1 = When read interrupt status line become interrupt enable line |
| 6 | R/W | 0 | **Interrupt Polarity Select**<br><br>0 positive 1 negative |
| 5 | R/W | 0 | **Main Interrupt line enable**<br><br>0 no report 1 report |
| 4:0 | R/W | 0 | **Control Line Select for reserved, nd, bd, md, novid**<br><br>0 real detect 1 irq register |

### IRQ SPACING REGISTER – 0XCC1

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | **Irq period[7:0]** |

### IRQ REPORT COUNTER REGISTER 1 – 0XCC2

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | IRQ Report Counter[7:0] |

### IRQ REPORT COUNTER REGISTER 2 – 0XCC3

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x0 | IRQ Report Counter[15:8] |

### IRQ REPORT COUNTER REGISTER 3 – 0XCC4

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0x0 | Reserved |
| 3:0 | R/W | 0x0 | IRQ Report Counter[19:16] |

### AUX. IRQ STATUS REGISTER —OFFSET 0XCC6

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0 | Reserved |
| 6 | R | 0 | DMAC Interrupt Status<br><br>0: idle 1: active |
| 5 | R | 0 | HDMI Interrupt Status<br><br>0: idle 1: active |
| 4 | R | 0 | OSG1 Interrupt Status<br><br>0: idle 1: active |
| 3 | R | 0 | OSG2 Interrupt Status<br><br>0: idle 1: active |
| 2 | R | 0 | OSG3 Interrupt Status<br><br>0: idle 1: active |
| 1 | R | 0 | MOUSE Interrupt Status<br><br>0: idle 1: active |
| 0 | R | 0 | I²C Master Interrupt Status<br><br>0: idle 1: active |

### MPOUT SELECTION REGISTER —OFFSET 0XCC7

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | Reserved |
| 6:4 | | | Port Select<br><br>111: Select port8<br>110: Select port7<br>101: Select port6<br>100: Select port5<br>011: Select port4<br>010: Select port3<br>001: Select port2<br>000: Select port1 |
| 3:0 | R/W | 0 | Select Essential Output Sync. Pulse for User<br><br>1111: Select reserved<br>1110: Select reserved<br>1101: Select VIS_VOS<br>1100: Select VIS_HOS<br>1011: Select VIS_FOS<br>1010: Select even_rosd<br>1001: Select fld_rosd<br>1000: Select vblank_rosd<br>0111: Select hblank_rosd<br>0110: Select FLD_MODE<br>0101: Select even_toggle<br>0100: Select even<br>0011: Select fm_even<br>0010: Select ft_even<br>0001: Select hblank<br>0000: Select vblank |

### DISPLAY DRAM ADDRESS REGISTER 0 – 0X000

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | Display Memory Address Lo Byte of the tress byte address |

### DISPLAY DRAM ADDRESS REGISTER 1– 0X001

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | Display Memory Address Mid Byte of the tress byte address |

### DISPLAY DRAM ADDRESS REGISTER 2– 0X002

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | Display Memory Address High Byte of the tress byte address |

### DISPLAY DRAM ACCESS CONTROL REGISTER —OFFSET 0X003

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | **Display DRAM read / write enable**<br><br>00 = Reserved<br>01 = Enable read access<br>10 = Reserved<br>11 = Enable write access |
| 5:0 | R/W | 0 | **Burst Width**<br><br>Numbers of bytes to be written or read from memory |

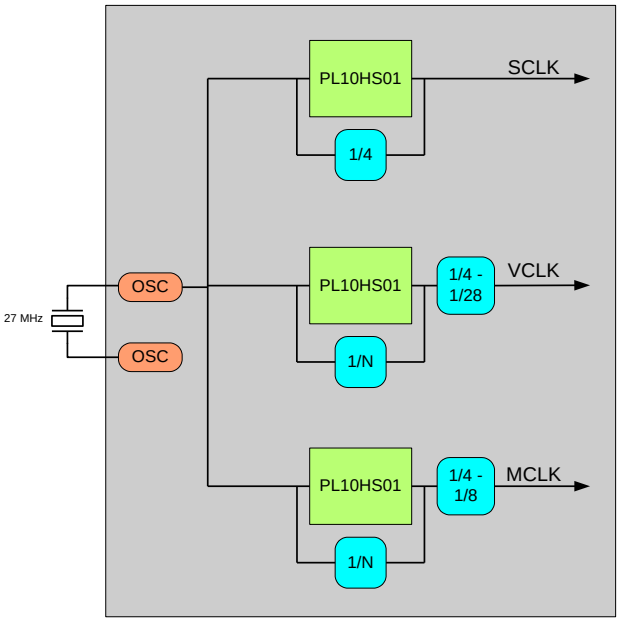
### DISPLAY DRAM DATA PORT REGISTER– 0X004

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | **Display Memory Data[7:0]** |

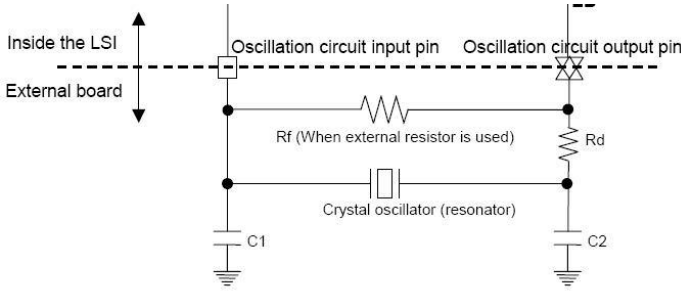
# Clock Synthesizer and Pin MUX

## Introduction

**TW2880 has an integrated PLL system to generate all the clocks that are needed. Follow the block diagram, a 27 MHz crystal is the only external component that are needed. The crystal and the on-chip oscillator supply the base frequency to the on-chip PLL system. The base frequency is fed into three PLLs: One for generating a fixed system clock at 108 MHz, one for generating display clock for the display unit and is changeable. The last PLL is used for supplying memory clock to the outside SDRAM and is also changeable.**




## Clock Synthesizer Circuit




| Symbol | Explanation |
|---|---|
| **C1, C2** | **External load capacitance** |
| **Rf** | **Feedback resistor** |
| **Rd** | **Dumping resistor (only when necessary)** |

| Frequency | C1, C2 (pf) | Rf (Mohm) | Rd (ohm) | Order |
|---|---|---|---|---|
| **27 MHz** | **10** | **1** | **470** | **Fundamental** |

| | Freq | Tolerance @25°C | Stability @25°C | Duty Cycle | Remark |
|---|---|---|---|---|---|
| OSC | 27 MHz | ±50 ppm | ±50 ppm | 55% | |

## Output Clock Summary

| From | Frequency | Duty cycle | Jitter | Skew (from rising edge of reference clock) | Remark |
|---|---|---|---|---|---|
| SCLK | 108 MHz | 50% | 250ps | 1000ps | |
| VCLK | 25 MHz | 50% | 250ps | 1000ps | D=16 N=Eh |
| | 108 MHz | 50% | 250ps | 1000ps | D=04 N=Fh |
| | 126 MHz | 50% | 250ps | 1000ps | D=06 N=1Bh |
| | 148.5 MHz | 50% | 250ps | 1000ps | D=08 N=2Bh |
| MCLK | 198.5 MHz | 50% | 250ps | 1000ps | D=06 N=2Bh |
| | 184.5 MHz | 50% | 250ps | 1000ps | D=06 N=28h |
| | 166.5 MHz | 50% | 250ps | 1000ps | D=06 N=24h |
| | 144 MHz | 50% | 250ps | 1000ps | D=06 N=1Fh |
| | 135 MHz | 50% | 250ps | 1000ps | D=06 N=1Dh |

## Initial Conditions and Changing PLL values

The 108 MHz clock is used as system clock. It will power the register block and most of the recording and CPU related blocks. The frequency of this clock will not get changed during the normal operation. The other two PLLs has initial setting and frequencies when power up, their frequencies will get changed on the fly during subsequent operations. The dividers of these two PLLs can be selected through registers programming. The available dividers to MCLK PLL are: 4, 6, 7, 8. VCLK PLL are: 4, 6, 7, 8, 10, 12, 16, 28.

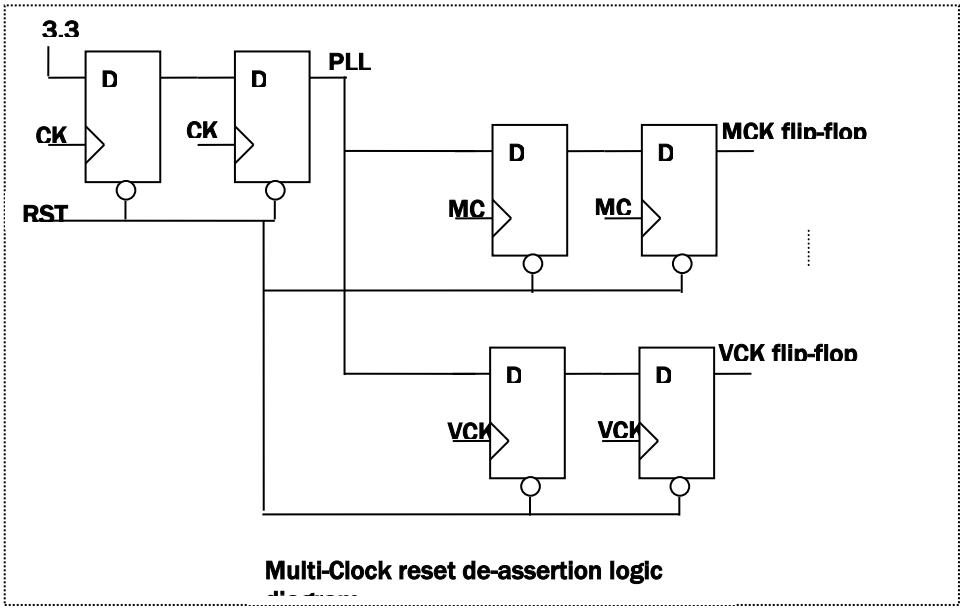
The formula used to calculate one particular frequency is:

F = 27 * (N+1) div4 (or div5 ...., div10)

## RESET Logic

TW2880's reset source comes from an external reset signal running asynchronously for at least 10 uS. This signal reset the whole chip logic by resets a pair of master reset flips- flops, which in turn drive the master reset signal asynchronously through the reset buffer tree to the rest of the flip-flops in the design. The entire chip will be asynchronously reset.

To come out of reset mode we need to de-asserting the reset signal pin. Which then permits the d-input of the first master reset flip-flop to be clocked through a reset synchronizer. It takes two rising edges after the reset removal to synchronize removal of the master reset. TW2880 has individual reset control to all the major functioning blocks and is controlled by user through register programming.



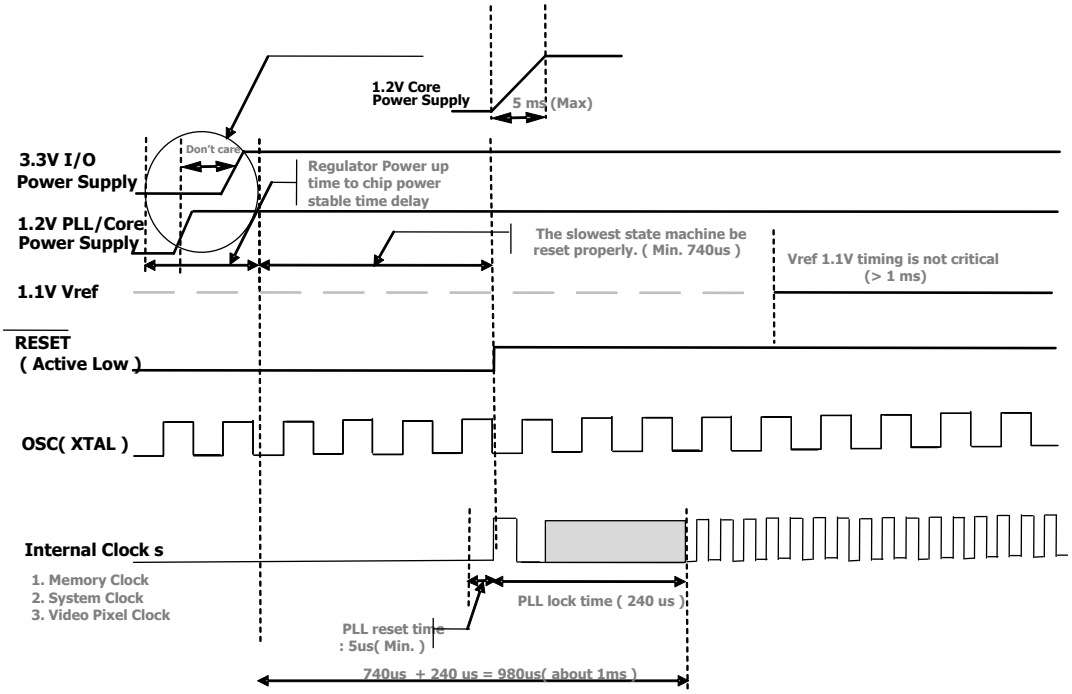

Multi-Clock reset de-assertion logic

## Playback Pin Mux

TW2880's playback port can support three kinds of input formats: BT656, BT1120 and 16 bit RGB. It is controlled by register 0x371 bit 0-1 and 2-3. If bit 0-1 are set to 0 then the two ports are used as two separate BT656 ports. Set to one will let these two ports become a 16 bit BT1120 port running at 1080i resolution (74.25MHz). One clock input is unused at this configuration. Bit 2-3 control the digital RGB input of PB ports.

## Power On Sequence

**There are six power rails in TW2880, Core 1.2V, PLL 1.2V, I/O 3.3V, DAC 3.3V, Vref 1.1v, HDMI 3.3V. The recommended power sequence is showed in the following drawing:**




**The power to the crystal circuit needs to be turned on first followed by the power to 1.2V core supply rail.**

**Then it is followed by the 3.3V I/O power to the TW2880. However, 3.3V I/O power can be turned on first**

**as long as the 1.2V core power rail is also turned on within 100 mS. The master reset signal needs to last at least 740 µs to truly reset internal PLL circuitry. After master reset signal, TW2880 will generate many internal reset signals to reset the chip using the stable clocks. The 1.1V DAC reference voltage has a direct impact on the video quality of the output image so it has to either come from a voltage refernce chip or from a well designed CCS circuit. Please see the example on the EVB. The TW2880 has to follow certain power off sequence to guarantee working 3.3V I/O power should be turned off first, then the 1.2V core and PLL power rail should be turned off.**

## Live Video Pin Mux

**TW2880's live video port have other definition when running at 108 MHz mode. Please refer to control register 0x201 bit 2. When this bit equal to 0, the inputs are used for live video channel running at 54 MHz. When this bit equal to 1, the port 5. 6. 7. 8 are defined as output pins and is used to output digital R, G, B of the main display. But if this bit equals to 1 and the external OSD option is selected, this extra pins are used as input pin to accept OSD data from an external chip in master mode. See the following chart as reference.**

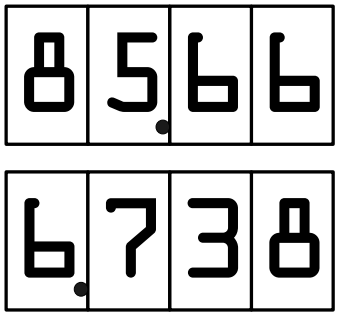
| | 0x201 Bit [2] = 0 | 0x201 Bit [2] = 1 | |
|---|---|---|---|
| INPUT | LV clock12 | LV clock12 | |
| IN1 | BT656[7:0] | BT656[7:0] | |
| IN2 | BT656[7:0] | BT656[7:0] | |
| INPUT | LV clock34 | LV clock34 | |

| | | | |
|---|---|---|---|
| IN3 | BT656[7:0] | BT656[7:0] | |
| IN4 | BT656[7:0] | BT656[7:0] | |
| IN/OUT | LV clock56 | VCLK out | Pixel clock |
| IN/OUT 5 | BT656[7:0] | R[7:0] | |
| IN/OUT 6 | BT656[7:0] | G[7:0] | |
| IN/OUT | LV clock78 | DEN / SWITCH | Display enable |
| IN/OUT 7 | BT656[7:0] | B[7:0] | |
| IN/OUT 8 | BT656[7:0] | | |

## Port 80 Display

Port 80 display subsystem is a very handy tool in debugging and diagnostic the TW2880-based system. User can access to port 80 display registers by the normal CPU read / write. Together with this capability and the on board 7-segement decoder and LDE, use can "print out" 2 or 4 digits of alphanumeric code and read it. This is the exact equivalent of the 2 hex digit LED display system normally seen in an PC where each booting step is displayed thought the LED. This is useful to determine how far the system boot process go if you have a system hang and can also be served as a process indicator when the system is running normally. A two digit or four digit systems option is supported.

User can choose lit up the two decimals to further classify the booting code or communication message as one big category and several small categories.




## HDMI Transmitter Signal Adjustment

TW2880's HDMI transmitter support four output modes to fit different situation. It is listed in the below table.

| IODCNT[2:0] | IOPCNT[2:0] | DRVCNT | OUTPUT MODE |
|---|---|---|---|
| 101 | 011 | 1 | A: Normal |
| 110 | 000 | 1 | B: Lower Amplitude |
| 110 | 010 | 0 | C: Medium De-Emphasis |
| 101 | 011 | 0 | D: Deep De-Emphasis |

**The major difference between the four output modes are in the voltage swing. It is tabulated in Table 26.**

TABLE 26. SWING CONTROL

| Symbol | Parameter | Conditions | mode | Min | Typ | Max | Unit |
|---|---|---|---|---|---|---|---|
| Vod | Differential Outputs single ended swing amplitude (Emphasis bit) | Rload=50ohm | A,C,D | | 500 | | mV |
| | | | B | | 250 | | mV |
| Vod-de | Differential Outputs single ended swing amplitude (De-Emphasis bit) | Rload=50ohm | A | | 500 | | mV |
| | | | C | | 350 | | mV |
| | | | B,D | | 250 | | mV |
| | | | | | | | |
| De-Ratio | De-Emphasis ratio | Rload=50ohm | A,B | | 0 | | dB |
| | | | C | | 3 | | dB |
| | | | D | | 6 | | dB |

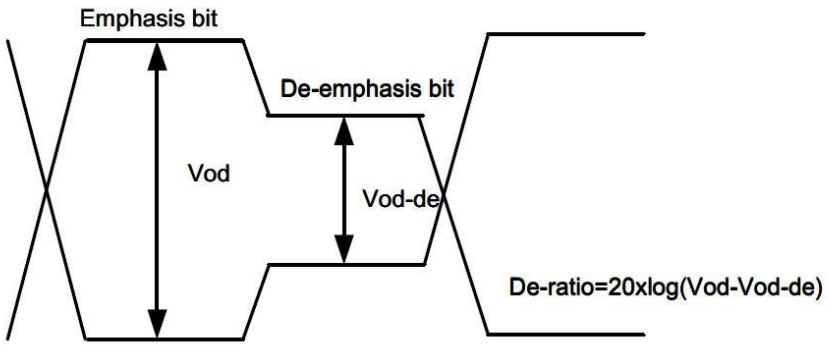



FIGURE 25. DE-EMPHASIS RATIO

**Figure 25 shows the De-emphasis function. In this chart, the Emphasis bit means the first bit of the series of same value bits. The Emphasis bit has full swing amplitude (see "Vod" in Table 26). The De-emphasis bit follows the Emphasis bit. The De-emphasis ratio/amplitude can be changed as described in "Vod-de" and "De-Ratio" in Table 26.**

## Register Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x200 | R/W | 0x00 | Reserved |
| 0x201 | R/W | 0x00 | MISC. Control Register |
| 0x202 | R/W | 0x00 | MPLL M Duty Cycle Control Register (MPLLMR) |
| 0x203 | R/W | 0x00 | MPLL N Multiplier Factor Register (MPLLNR) |
| 0x204 | R/W | 0x00 | NPLL M Duty Cycle Control Register (NPLLMR) |
| 0x205 | R/W | 0x00 | NPLL N Multiplier Factor Register (NPLLNR) |
| 0x206 | R/W | 0x00 | SPLL M Duty Cycle Control Register (SPLLMR) |
| 0x207 | R/W | 0x00 | SPLL N Multiplier Factor Register (SPLLNR) |
| 0x208 | R/W | 0x00 | MCLK PLL Divider Control Register 1 |
| 0x209 | R/W | 0x00 | VCLK PLL Divider Control Register 1 |
| 0x20A | R/W | 0x00 | IN Rate Select Register |
| 0x20B | R/W | 0x00 | Digital Block Reset Control Register 1 |
| 0x20C | R/W | 0x00 | Digital Block Reset Control Register 2 |
| 0x20D | R/W | 0x00 | Digital Block Reset Control Register 3 |
| 0x20E | R/W | 0x00 | Digital Block Reset Control Register 4 |
| 0x20F | R/W | 0x00 | Digital Block Reset Control Register 5 |
| 0x210 | R/W | 0x00 | LCD Display Pitch Register |
| 0x211 | R/W | 0x00 | Display DRAM Controller Configuration Register |
| 0x212 | R/W | 0x00 | Recording DRAM Controller Configuration Register |
| 0x213 | R/W | 0x00 | Port 80 Control Register |
| 0x214 | R/W | 0x00 | Port 80 Content Register 1 |
| 0x215 | R/W | 0x00 | Port 80 Content Register 2 |
| 0x216 | R/W | 0x00 | SPLL Phase Control Register 1 |
| 0x217 | R/W | 0x00 | PLL Soft Reset Control Register |
| 0x218 | R/W | 0x00 | REC Output Clock Phase Control Register 1 |
| 0x219 | R/W | 0x00 | REC Output Clock Phase Control Register 2 |
| 0x21A | R/W | 0x00 | TV Decoder Output Clock Select Register |
| 0x21B | R/W | 0x00 | HDMI physical Interface Control Register |
| 0x21C | R/W | 0x00 | Live Input Clock Phase Select Register |
| 0x21D | R/W | 0x00 | DRAM Space Access Select Register |
| 0x21E | R/W | 0x00 | Network Port Data Source Select Register |
| 0x21F | R/W | 0x00 | SCLK Phase Control Register 2 |
| 0x220 | R/W | 0x00 | MCLK Phase Control Register |
| 0x221 | R/W | 0x00 | VCLK PLL Divider Control Register 2 |
| 0x222 | R/W | 0x00 | VCLK PLL Divider Control Register 3 |
| 0x223 | R/W | 0x00 | VCLK PLL Divider Control Register 4 |
| 0x224 | R/W | 0x48 | LP_CON1 x Control Register |
| 0x225 | R/W | 0x5a | LP_CON2 x Control Register |
| 0x226 | R/W | 0x6c | LP_CON3 x Control Register |
| 0x227 | R/W | 0x7e | LP_CON4 x Control Register |
| 0x228 | R/W | 0x00 | Audio Control 1 Register |
| 0x229 | R/W | 0x00 | Audio Control 2 Register |
| 0x22C | R/W | 0x00 | CPU Fast Access Control Register |
| 0x22D | R/W | 0x00 | CPU Freeze Display Pitch Register |
| 0x22E | R/W | 0x00 | SCLK Phase Control Register 3 |
| 0x26D | R/W | 0x00 | DRAM Address Mapping Register |
| 0x270 | R/W | 0x00 | Chip ID Register |
| 0x280 | R/W | 0x00 | Display DRAM Priority Arbitration Register 1 |
| 0x281 | R/W | 0x00 | Display DRAM Priority Arbitration Register 2 |
| 0x282 | R/W | 0x00 | Display DRAM Priority Arbitration Register 3 |
| 0x284 | R/W | 0x03 | Record DRAM Priority Arbitration Register 1 |

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x285 | R/W | 0x00 | Record DRAM Priority Arbitration Register 2 |
| 0x286 | R/W | 0x00 | Record DRAM Priority Arbitration Register 3 |

# Registers Description

### AUDIO CONTROL REGISTER —OFFSET 0X200

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R | 0 | Reserved |

### MISC. CONTROL REGISTER —OFFSET 0X201

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | Digital Output Select<br><br>0: select main<br>1: select dual monitor |
| 6 | R | 0 | Reserved |
| 5 | R/W | 0 | External_CLKIN enable<br><br>0: no external clock in<br>1: external clock in for BT.1120 mode |
| 4 | R/W | 0 | OSG output clock enable<br><br>0: No external OSG clock output<br>1: external OSG clock output through din8[1] port |
| 3 | R/W | 0 | REC_ENC_OEN<br><br>0: REC output enable<br>1: REC output off |
| 2 | R/W | 0 | External_RGB24_OEN<br><br>0: Vin port 5,6,7 is set as inputs<br>1: RGB24 output enable control, MP_out as clock source |
| 1 | R/W | 0 | Reserved |
| 0 | R/W | 0 | LCD display power down<br><br>0: Enable analog VGA display<br>1: VGA display power off |

### MPLL M DUTY CYCLE CONTROL REGISTER (MPLLMR) —OFFSET 0X202

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | Reserve |
| 5:0 | R/W | 0x0d | MPLL M factor<br><br>MPLL FB duty control |

### MPLL N MULTIPLER FACTOR REGISTER (MPLLNR) —OFFSET 0X203

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | Reserve |
| 5:0 | R/W | 0x24<br><br>166.6MHz | MPLL N factor<br><br>MPLL multiply factor. Mclk = 27MHz * (N+1) / Mclk_DivQ |

### VPLL M DUTY CYCLE CONTROL REGISTER (VPLLMR) —OFFSET 0X204

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | Reserve |
| 5:0 | R/W | 0x0d | VPLL M factor<br><br>VPLL FB duty control |

### VPLL N MULTIPLER FACTOR REGISTER (VPLLNR) —OFFSET 0X205

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | Reserve |
| 5:0 | R/W | 0x27<br><br>108MHz | VPLL N factor<br><br>VPLL multiply factor. Vclk = 27MHz * (N+1) / Vclk_DivQ |

### SPLL M DUTY CYCLE CONTROL REGISTER (SPLLMR) —OFFSET 0X206

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | Reserve |
| 5:0 | R/W | 0x0f | SPLL M factor<br><br>SPLL feedback duty control, can be set to 0 |

### SPLL N MULTIPLY FACTOR REGISTER (SPLLNR) —OFFSET 0X207

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | Reserve |
| 5:0 | R/W | 0x1f | SPLL N factor<br><br>SPLL multiply factor. sclk = 27MHz * (N+1) / 8. When normal operation, don't change this register. |

### MCLK PLL DIVIDER CONTROL REGISTER 1 —OFFSET 0X208 (UPPER BYTE AT 0X220)

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | Reserve |
| 5 | R/W | 0 | REC Port DRAM Clk Invert control<br><br>0 : not invert<br>1 : invert |
| 4 | R/W | 0 | LCD Display DRAM Clk Invert control<br><br>0 : not invert<br>1 : invert |
| 3:1 | R/W | 0 | Internal Mclk Divide Control<br><br>100 : Mclk_divQ equals 8<br>010 : Mclk_divQ equals 7<br>001 : Mclk_divQ equals 6<br>000 : Mclk_divQ equals 4 |
| 0 | R/W | 0 | Mclk phase Control<br><br>1 : invert<br>0 : not invert |

### VCLK PLL DIVIDER CONTROL REGISTER 1—OFFSET 0X209 (UPPER BYTE AT X221 .. X223)

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R/W | 3'h4 | **Internal Rclk Divide Control**<br><br>3'h7 : Vclk_divQ equals 16<br>3'h6 : Vclk_divQ equals 28<br>3'h5 : Vclk_divQ equals 12<br>3'h4 : Vclk_divQ equals 10<br>3'h3 : Vclk_divQ equals 8<br>3'h2 : Vclk_divQ equals 7<br>3'h1 : Vclk_divQ equals 6<br>3'h0 : V_divQ equal 4 |
| 4 | R/W | 0 | **Rclk phase control**<br><br>1 : invert<br>0 : not invert |
| 3:1 | R/W | 3'h4 | **Internal Vclock Divide Control**<br><br>3'h7 : Vclk_divQ equals 16<br>3'h6 : Vclk_divQ equals 28<br>3'h5 : Vclk_divQ equals 12<br>3'h4 : Vclk_divQ equals 10<br>3'h3 : Vclk_divQ equals 8<br>3'h2 : Vclk_divQ equals 7<br>3'h1 : Vclk_divQ equals 6<br>3'h0 : Vclk_divQ equals 4 |
| 0 | R/W | 0 | **Vclk phase control**<br><br>1 : invert<br>0 : not invert |

### IN RATE SELECT REGISTER —OFFSET 0X20A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | - | - | **Reserved** |
| 1:0 | R/W | 0 | **Input data frequency selection**<br><br>00: live video input is set to 108 MHz<br>01: live video input is set to 54 MHz<br>1X: Live video input is set at 27 MHz |

### DIGITAL BLOCK RESET CONTROL REGISTER 1 —OFFSET 0X20B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:2 | R | 01 | **Reserved** |
| 1 | R/W | 0 | **Soft Reset Control RGB controller**<br><br>0: normal operation<br>1: reset RGB controller |
| 0 | R/W | 0 | **Soft Reset Control PB in RGB interface controller**<br><br>0: normal operation<br>1: reset PB controller |

### DIGITAL BLOCK RESET CONTROL REGISTER 2 —OFFSET 0X20C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | - | - | Reserved |
| 4 | R/W | 0 | **Soft Reset OSGW controller**<br><br>0: normal operation<br>1: reset OSGW controller |
| 3 | R/W | 0 | **Soft Reset OSG controller**<br><br>0: normal operation<br>1: reset OSG controller |
| 2 | R/W | 0 | **Soft Reset 2D/3D de-interlace controller**<br><br>0: normal operation<br>1: reset 2D/3D de-interlace controller |
| 1 | R/W | 0 | **Soft Reset LCD DRAM controller**<br><br>0: normal operation<br>1: reset LCD DRAM controller |
| 0 | R/W | 0 | **Soft Reset the whole LCD controller**<br><br>0: normal operation<br>1: reset LCD controller |

### DIGITAL BLOCK RESET CONTROL REGISTER 3 —OFFSET 0X20D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | **Soft Reset OSG BITMAP controller**<br><br>0 normal operation<br>1 reset osg BITMAP controller |
| 6 | R/W | 0 | **Soft Reset OSG DMA controller**<br><br>0 normal operation<br>1 reset osg DMA controller |
| 5 | R/W | 0 | **Soft Reset Freeze DMA controller**<br><br>0 normal operation<br>1 reset freeze DMA controller |
| 4 | R/W | 0 | **Soft Reset I²C master controller**<br><br>0 normal operation<br>1 reset I²C master controller |
| 3 | R/W | 0 | **Soft Reset the SPOT controller SCLK domain**<br><br>0 normal operation<br>1 reset spot ck108 control modules |
| 2 | R/W | 0 | **Soft Reset the SPOT controller MCLK domain**<br><br>0: normal operation<br>1: reset spot mclk domain logic |
| 1 | R/W | 0 | **Soft Reset the Dual Monitor controller SCLK domain**<br><br>0 normal operation<br>1 reset dual monitor ck108 control modules |
| 0 | R/W | 0 | **Soft Reset the Dual Monitor controller MCLK domain** |

| Bit | R/W | Default | Description |
|---|---|---|---|
| | | | 0: normal operation<br>1: reset dual monitor mclk domain logic |

**DIGITAL BLOCK RESET CONTROL REGISTER 4 —OFFSET 0X20E**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R | 0 | Reserved |
| 3 | R/W | 0 | Soft Reset IRQ controller<br><br>0 normal operation<br>1 reset control modules |
| 2 | R/W | 0 | Soft Reset MD controller MCLK & SCLK domain<br><br>0 normal operation<br>1 reset ENC MCLK & SCLK control modules |
| 1 | R/W | 0 | Soft Reset Encoder controller in SCLK, VCLK, MCLK domain<br><br>0 normal operation<br>1 reset modules |
| 0 | R/W | 0 | Soft Reset Decoder controller SCLK, VCLK, MCLK domain<br><br>0 normal operation<br>1 reset modules |

**DIGITAL BLOCK RESET CONTROL REGISTER 5 —OFFSET 0X20F**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | - | - | Reserved |
| 6 | R/W | 0 | Soft Reset Host to Memory controller<br><br>0 normal operation<br>1 reset Host to Memory controller |
| 5 | R/W | 0 | Soft Reset DCU2 controller<br><br>0 normal operation<br>1 reset DCU2 controller |
| 4 | R/W | 0 | Soft Reset DCU controller<br><br>0 normal operation<br>1 reset DCU controller |
| 3 | R/W | 0 | Soft Reset Down Scaler controller SCLK domains<br><br>0 normal operation<br>1 reset Down Scaler in sck control modules |
| 2 | R/W | 0 | Soft Reset Down Scaler controller PB channels<br><br>0 normal operation<br>1 reset Down Scaler in PBCLK control modules |
| 1 | R/W | 0 | Soft Reset Down Scaler controller CK27VIN domains<br><br>0 normal operation<br>1 reset Down Scaler in CK27VIN control modules |
| 0 | R/W | 0 | Soft Reset the Down Scaler controller<br><br>0 normal operation<br>1 reset all Down Scaler controllers |

### LCD DISPLAY PITCH REGISTER — OFFSET 0X210

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0x80 | **Horizontal Display Pitch for LCD**<br><br>This is SDRAM logic horizontal size. The unit is 16 pixels / 32 bytes. Default is 2048/16=128. If 2x128Mbit SDRAM, one page SDRAM can be mapped to 2048x2048 pixels logic size. There are four pages in one SDRAM. |

### DISPLAY DRAM CONTROLLER CONFIGURATION REGISTER — 0X211

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0 | Reserved |
| 6 | R/W | 0 | 1 = Extra pre-charge cycle (default 0) |
| 5 | R/W | 0 | 1 = Disable refresh |
| 4 | R/W | 0 | Refresh speed select<br><br>1 = fast speed<br>0 = Slow speed |
| 3 | R/W | 0 | 0 = Extra RAS2CAS cycle (default 0) |
| 2:1 | R/W | 1 | DRAM Type Select<br><br>00 = 64 Mbit<br>01 = 128 Mbit<br>10 = 256 Mbit<br>11 = Reserved |
| 0 | R/W | 1 | DRAM Data Width Select<br><br>0 = 32 bit, 1 = 64 bit (default) |

### RECORDING DRAM CONTROLLER CONFIGURATION REGISTER — 0X212

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0 | Reserved |
| 6 | R/W | 0 | 1 = Extra pre-charge cycle (default 0) |
| 5 | R/W | 0 | 1 = Disable refresh |
| 4 | R/W | 0 | Refresh speed select<br><br>1 = fast speed<br>0 = Slow speed |
| 3 | R/W | 0 | 0 = Extra RAS2CAS cycle (default 0) |
| 2:1 | R/W | 1 | DRAM Type Select |

| Bit | R/W | Default | Description |
|---|---|---|---|
| | | | 00 = 64 Mbit<br>01 = 128 Mbit<br>10 = 256 Mbit<br>11 = Reserved |
| 0 | R/W | 1 | DRAM Data Width Select<br>0 = 32 bit, 1 = 64 bit (default) |

**PORT 80 CONTROL REGISTER — 0X213**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | Port 80 function control<br>1 = enable<br>0 = disable |
| 6 | R/W | 0 | Port 80 digit select<br>0 = output two digit to port 80 (default)<br>1 = output four digit to port 80 |
| 5-4 | R | 0 | Reserved |
| 3 | R/W | 0 | Digit 3 Decimal Control, 1 = enable |
| 2 | R/W | 0 | Digit 2 Decimal Control, 1 = enable |
| 1 | R/W | 0 | Digit 1 Decimal Control, 1 = enable |
| 0 | R/W | 0 | Digit 0 Decimal Control, 1 = enable |

**PORT 80 CONTENT REGISTER 1 — 0X214**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0 | Port 80 digit 1 |
| 3:0 | R/W | 0 | Port 80 digit 0 |

**PORT 80 CONTENT REGISTER 2 — 0X215**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R/W | 0 | Port 80 digit 3 |
| 3:0 | R/W | 0 | Port 80 digit 2 |

### SPLL PHASE CONTROL REGISTER I —OFFSET 0X216

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | **DM_ENC_CLK clock phase select**<br><br>1: Invert the DM_ENC_CLK<br>0: Normal |
| 6 | R/W | 0 | **DM_VCLK clock phase select**<br><br>1: Invert the DM_VCLK<br>0: Normal |
| 5 | R/W | 0 | **DM_FCLK clock phase select**<br><br>1: Invert the DM_FCLK<br>0: Normal |
| 4 | R/W | 0 | **Select the 27 MHz CLK Phase**<br><br>1: Invert the 27 MHz CLK<br>0: Normal |
| 3 | R/W | 0 | **Select the 54 MHz CLK Phase**<br><br>1: Invert the 54 MHz CLK<br>0: Normal |
| 2 | R/W | 0 | **Select 108 MHz phase**<br><br>1: Invert the 108 MHz<br>0: Normal |
| 1:0 | R/W | 0 | **Select 108 MHz phase control**<br><br>11: 270° phase shift<br>10: 180° phase shift<br>01: 90° phase shift<br>00: Normal |

### PLL SOFT RESET CONTROL REGISTER —OFFSET 0X217

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | **reserve** |
| 6 | R/W | 0 | **Soft reset port80 in ck108 domain** |
| 5 | R/W | 0 | **Soft reset extclk** |
| 4 | R/W | 0 | **Soft reset HDMI in vclk domain** |
| 3 | R/W | 0 | **Soft reset audio in mclk domain** |
| 2 | R/W | 0 | **Soft reset VCK PLL** |
| 1 | R/W | 0 | **Soft reset MCK PLL** |
| 0 | R/W | 0 | **Soft reset SCK PLL** |

### REC OUTPUT CLOCK PHASE CONTROL REGISTER 1 —OFFSET 0X218

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | **REC1_CLKP_SEL**<br><br>11: Select inverted DM_VCLK<br>10: Select Normal DM_VCLK<br>0X: Select CLK defined by P_CON1 register |
| 5:3 | R/W | 0 | **Select DM_VCLK phase as the rec1_clkp, rec3_clkp source**<br><br>111: 315 degree phase shift |

| Bit | R/W | Default | Description |
|---|---|---|---|
| | | | 110: 270 degree phase shift<br>101: 225 degree phase shift<br>100: 180 degree phase shift<br>011: 135 degree phase shift<br>010: 90 degree phase shift<br>001: 45 degree phase shift<br>000: No degree phase shift |
| 2:0 | R/W | 0 | **Select rclk phase when vclk is the rclk source**<br><br>111: 315 degree phase shift<br>110: 270 degree phase shift<br>101: 225 degree phase shift<br>100: 180 degree phase shift<br>011: 135 degree phase shift<br>010: 90 degree phase shift<br>001: 45 degree phase shift<br>000: No degree phase shift |

**REC OUTPUT CLOCK PHASE CONTROL REGISTER 2 —OFFSET 0X219**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | **Rec4_clkn , Rec4_clkp phase control**<br><br>11: 270 degree phase shift<br>10: 180 degree phase shift<br>01: 90 degree phase shift<br>00: No phase shift |
| 5:4 | | 0 | **Rec3_clkn, Rec3_clkp phase control**<br><br>11: 270 degree phase shift<br>10: 180 degree phase shift<br>01: 90 degree phase shift<br>00: No phase shift |
| 3:2 | R/W | 0 | **Rec2_clkn, Rec2_clkp phase control**<br><br>11: 270 degree phase shift<br>10: 180 degree phase shift<br>01: 90 degree phase shift<br>00: No phase shift |
| 1:0 | R/W | 0 | **Rec1_clkn, Rec1_clkp phase control**<br><br>11: 270 degree phase shift<br>10: 180 degree phase shift<br>01: 90 degree phase shift<br>00: No phase shift |

**TV DECODER OUTPUT CLOCK SELECT REGISTER —OFFSET 0X21A**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R/W | 0 | **REC3_CLKP_SEL**<br><br>11: Select inverted DM_VCLK<br>10: Select Normal DM_VCLK<br>0x: Select CLK defined by P_CON1_reg |
| 4 | R/W | 0 | **HIFREQ_CLK_SEL2**<br><br>1: Select VPLL Hi_freq_clk as the delay control clk for REC |

| Bit | R/W | Default | Description |
|---|---|---|---|
| | | | 0: Select SPLL Hi_freq_clk as the delay control clk for REC |
| 3 | R/W | 0 | **HIFREQ_CLK_SEL1**<br><br>1: Select VPLL Hi_freq_clk as the delay control clk for RCLK<br>0: Select SPLL Hi_freq_clk as the delay control clk for RCLK |
| 2:1 | R/W | 0 | **HDMI_CLK_SEL**<br><br>11: Select Inverted DM_VCLK<br>10: Select Normal DM_VCLK<br>0x: Select Normal VCLK |
| 0 | R/W | 0 | **Select TV decoder clock frequency**<br><br>1: 54 MHz clock is sent to TV decoder<br>0: 108 MHz clock is sent to TV decoder |

### HDMI PHYSICAL INTERFACE CONTROL REGISTER —OFFSET 0X21B

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | | 0 | **Reserve** |
| 6:4 | R/W | 101 | **IODCNT Output Current Control**<br><br>111: Max<br>000: Min |
| 3:1 | R/W | 011 | **IOPCNT De-Emphasis Control**<br><br>111: Max<br>000: Min |
| 0 | R/W | 1 | **IDRVCNT De-Emphasis Disable**<br><br>1: De-Emphasis Disable<br>0: De-Emphasis Enable |

### LIVE INPUT CLOCK PHASE SELECT REGISTER —OFFSET 0X21C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | **CK27PB4 input phase Select**<br><br>1: Inverts the input clock phase<br>0: Normal, no phase invert |
| 6 | R/W | 0 | **CK27PB3 input phase Select**<br><br>1: Inverts the input clock phase<br>0: Normal, no phase invert |
| 5 | R/W | 0 | **CK27PB2 input phase Select**<br><br>1: Inverts the input clock phase<br>0: Normal, no phase invert |
| 4 | R/W | 0 | **CK27PB1 input phase Select**<br><br>1: Inverts the input clock phase<br>0: Normal, no phase invert |
| 3 | R/W | 0 | **CKVin78 input phase Select**<br><br>1: Inverts the input clock phase<br>0: Normal, no phase invert |
| 2 | R/W | 0 | **CKVin56 input phase Select** |

| Bit | R/W | Default | Description |
|---|---|---|---|
| | | | 1: Inverts the input clock phase<br>0: Normal, no phase invert |
| 1 | R/W | 0 | **CKVin34 input phase Select**<br><br>1: Inverts the input clock phase<br>0: Normal, no phase invert |
| 0 | R/W | 0 | **CKVin12 input phase Select**<br><br>1: Inverts the input clock phase<br>0: Normal, no phase invert |

### DRAM SPACE ACCESS SELECT REGISTER —OFFSET 0X21D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:1 | R/W | 0 | **Reserve** |
| 0 | R/W | 0 | **Select DRAM space access from host**<br><br>1: Host access REC DRAM space<br>0: Host access LCD DRAM space |

### NETWORK PORT DATA SOURCE SELECT REGISTER —OFFSET 0X21E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | **Select HDMI TX source**<br><br>1: from Dual display<br>0: from Main display |
| 6:4 | R/W | 0 | **REC2NetWork Port data Source**<br><br>111: rec8_data<br>110: rec7_data<br>111: rec6_data<br>110: rec5_data<br>011: rec4_data<br>010: rec3_data<br>001: rec2_data<br>000: rec1_data |
| 3 | 0 | 0 | **Reserved** |
| 2 | R/W | 0 | **Rec4_by601_H,V,F mux**<br><br>1: rec4_by601_H4 as the mclk_pll locked output<br>rec4_by601_V4 as the vclk_pll locked output<br>rec4_by601_F4 as the sclk_pll locked output<br><br>0: if P80_CON1[7] is 1<br>rec4_by601_H4 as the port80_out output<br>rec4_by601_V4 as the port80_clko output<br>rec4_by601_F4 as the port80_en output<br><br>if P80_CON1[7] is 0, normal H, V, F output |
| 1:0 | R/W | 0 | **Select Network port data source**<br><br>11: REC2NetWork<br>10: DM_vdout<br>01: SPOT_vdout<br>00: NET_ENC_DATA |

### SCLK PHASE CONTROL REGISTER 2 —OFFSET 0X21F

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | **DM_VCLK_EXT clock phase select**<br><br>1: Inverts the DM_VCLK_EXT<br>0: Normal |
| 6:5 | R/W | 0 | **DM_FCLK select when SCLK is the clock source**<br><br>11: 108 MHz<br>10: 54 MHz<br>01: 27 MHz<br>00: 13.5 MHz |
| 4:3 | R/W | 0 | **DM_VCLK select when SCLK is the clock source**<br><br>11: 108 MHz<br>10: 54 MHz<br>01: 27 MHz<br>00: 13.5 MHz |
| 2 | R/W | 0 | **DM_VCLK clock source select**<br><br>1: DM_VCLK is derived from VCLK<br>0: DM_VCLK is derived from SCLK |
| 1 | R/W | 0 | **DM_FCLK clock source select**<br><br>1: DM_FCLK is derived from VCLK<br>0: DM_FCLK is derived from SCLK |
| 0 | R/W | 0 | **DM_DAC_CLK Phase Select**<br><br>1: Inverts the DM_DAC_CLK<br>0: Normal |

### MCLK PHASE CONTROL REGISTER 2 —OFFSET 0X220

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | **D_REC_MCLK Phase control to latch REC DRAM data in**<br><br>11: 135 deg phase shift<br>10: 90 deg phase shift<br>01: 45 deg phase shift<br>00: No phase shift |
| 5:4 | R/W | 0 | **D_LCD_MCLK Phase control to latch LCD DRAM data in**<br><br>11: 135 deg phase shift<br>10: 90 deg phase shift<br>01: 45 deg phase shift<br>00: No phase shift |
| 3:2 | R/W | 0 | **MCLK Phase control for REC DRAM controller**<br><br>11: 135 deg phase shift<br>10: 90 deg phase shift<br>01: 45 deg phase shift<br>00: No phase shift |
| 1:0 | R/W | 0 | **MCLK Phase control for LCD DRAM controller**<br><br>11: 135 deg phase shift<br>10: 90 deg phase shift<br>01: 45 deg phase shift<br>00: No phase shift |

### VCLK PLL DIVIDER CONTROL REGISTER 2 —OFFSET 0X221

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:5 | R/W | 3'h4 | **DM_FCLK Divide Control**<br><br>3'h7 : Vclk_divQ equal 16<br>3'h6 : Vclk_divQ equal 28<br>3'h5 : Vclk_divQ equal 12<br>3'h4 : Vclk_divQ equal 10<br>3'h3 : Vclk_divQ equal 8<br>3'h2 : Vclk_divQ equal 7<br>3'h1 : Vclk_divQ equal 6<br>3'h0 : Vclk_divQ equal 4 |
| 4 | R/W | 0 | **DM_FCLK phase control**<br><br>1 : invert<br>0 : not invert |
| 3:1 | R/W | 3'h4 | **VCLK2 Divide Control and phase control by VCLK phase control register 0x223**<br><br>3'h7 : Vclk_divQ equal 16<br>3'h6 : Vclk_divQ equal 28<br>3'h5 : Vclk_divQ equal 12<br>3'h4 : Vclk_divQ equal 10<br>3'h3 : Vclk_divQ equal 8<br>3'h2 : Vclk_divQ equal 7<br>3'h1 : Vclk_divQ equal 6<br>3'h0 : Vclk_divQ equal 4 |
| 0 | R/W | 0 | **VCLK2 phase control used for lcd_dac_clk and HDMI clk**<br><br>1 : invert<br>0 : not invert |

### VCLK PLL DIVIDER CONTROL REGISTER 3 —OFFSET 0X222

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | **VCLK1 source from SCLK**<br><br>11 : SCK108 MHz<br>10 : SCK54 MHz<br>01 : SCK13.5 MHz<br>00 : use VCLK PLL to generate the VCLK1 frequency |
| 5:4 | R/W | 0 | **VCLK source from SCLK**<br><br>11 : SCK108 MHz<br>10 : SCK27 MHz<br>01 : SCK13.5 MHz<br>00 : use VCLK PLL to generate the VCLK frequency |
| 3:1 | R/W | 3'h4 | **DM_VCLK Divide Control**<br><br>3'h7 : Vclk_divQ equal 16<br>3'h6 : Vclk_divQ equal 28<br>3'h5 : Vclk_divQ equal 12<br>3'h4 : Vclk_divQ equal 10<br>3'h3 : Vclk_divQ equal 8<br>3'h2 : Vclk_divQ equal 7<br>3'h1 : Vclk_divQ equal 6<br>3'h0 : Vclk_divQ equal 4 |
| 0 | R/W | 0 | **DM_VCLK phase control** |

| Bit | R/W | Default | Description |
|---|---|---|---|
| | | | 1 : invert<br>0 : not invert |

### VCLK PLL DIVIDER CONTROL REGISTER 4 —OFFSET 0X223

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | **VCLK1, DM_VCLK phase control**<br><br>11: 135 deg phase shift<br>10: 90 deg phase shift<br>01: 45 deg phase shift<br>00: No phase shift |
| 5:4 | R/W | 0 | **VCLK2, DM_FCLK phase control**<br><br>11: 135 deg phase shift<br>10: 90 deg phase shift<br>01: 45 deg phase shift<br>00: No phase shift |
| 3 | R/W | 0 | **HDMI_CLK Phase Control**<br><br>1: Invert VCLK as the HDMI_CLK<br>0: Use VCLK as the HDMI_CLK |
| 2 | R/W | 0 | **LCD_DAC_CLK Phase Control**<br><br>1: Invert VCLK as the LCD_DAC_CLK<br>0: Use VCLK as the LCD _DAC_CLK |
| 1:0 | R/W | 0 | **VCLK2 source from SCLK**<br><br>11 : SCK108 MHz<br>10 : SCK54 MHz<br>01 : SCK27 MHz<br>00 : use VCLK PLL to generate the VCLK frequency |

### LP_CON1 X CONTROL REGISTER —OFFSET 0X224

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | | 0 | **Reserve** |
| 6:4 | R/W | 100 | **Playback Port 1 data is from:**<br><br>111: rec4 port<br>110: rec3 port<br>101: rec2 port<br>100: rec1 port<br>011: PB4 port<br>010: PB3 port<br>001: PB2 port<br>000: PB1 port |
| 3:0 | R/W | 1000 | **Playback port 1 clock is coming from:**<br><br>1111: rec4_clkn<br>1110: rec4_clkp<br>1101: rec3_clkn<br>1100: rec3_clkp<br>1011: rec2_clkn<br>1010: rec2_clkp<br>1001: rec1_clkn<br>1000: rec1_clkp |

| Bit | R/W | Default | Description |
|---|---|---|---|
| | | | 0111: pb4_clkn<br>0110: pb3_clkn<br>0101: pb2_clkn<br>0100: pb1_clkn<br>0011: pb4_clkp<br>0010: pb3_clkp<br>0001: pb2_clkp<br>0000: pb1_clkp |

**LP_CON2 X CONTROL REGISTER —OFFSET 0X225**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | | 0 | Reserve |
| 6:4 | R/W | 101 | Playback Port 2 data is from:<br><br>111: rec4 port<br>110: rec3 port<br>101: rec2 port<br>100: rec1 port<br>011: PB4 port<br>010: PB3 port<br>001: PB2 port<br>000: PB1 port |
| 3:0 | R/W | 1010 | Playback port 2 clock is coming from:<br><br>1111: rec4_clkn<br>1110: rec4_clkp<br>1101: rec3_clkn<br>1100: rec3_clkp<br>1011: rec2_clkn<br>1010: rec2_clkp<br>1001: rec1_clkn<br>1000: rec1_clkp<br>0111: pb4_clkn<br>0110: pb3_clkn<br>0101: pb2_clkn<br>0100: pb1_clkn<br>0011: pb4_clkp<br>0010: pb3_clkp<br>0001: pb2_clkp<br>0000: pb1_clkp |

### LP_CON3 X CONTROL REGISTER —OFFSET 0X226

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | | 0 | Reserve |
| 6:4 | R/W | 110 | Playback Port 3 data is from:<br><br>111: rec4 port<br>110: rec3 port<br>101: rec2 port<br>100: rec1 port<br>011: PB4 port<br>010: PB3 port<br>001: PB2 port<br>000: PB1 port |
| 3:0 | R/W | 1100 | Playback port 3 clock is coming from:<br><br>1111: rec4_clkn<br>1110: rec4_clkp<br>1101: rec3_clkn<br>1100: rec3_clkp<br>1011: rec2_clkn<br>1010: rec2_clkp<br>1001: rec1_clkn<br>1000: rec1_clkp<br>0111: pb4_clkn<br>0110: pb3_clkn<br>0101: pb2_clkn<br>0100: pb1_clkn<br>0011: pb4_clkp<br>0010: pb3_clkp<br>0001: pb2_clkp<br>0000: pb1_clkp |

### LP_CON4 X CONTROL REGISTER —OFFSET 0X227

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | | 0 | Reserve |
| 6:4 | R/W | 111 | Playback Port 4 data is from:<br><br>111: rec4 port<br>110: rec3 port<br>101: rec2 port<br>100: rec1 port<br>011: PB4 port<br>010: PB3 port<br>001: PB2 port<br>000: PB1 port |
| 3:0 | R/W | 1110 | Playback port 4 clock is coming from:<br><br>1111: rec4_clkn<br>1110: rec4_clkp<br>1101: rec3_clkn<br>1100: rec3_clkp<br>1011: rec2_clkn<br>1010: rec2_clkp<br>1001: rec1_clkn<br>1000: rec1_clkp<br>0111: pb4_clkn<br>0110: pb3_clkn<br>0101: pb2_clkn |

| Bit | R/W | Default | Description |
|---|---|---|---|
| | | | 0100: pb1_clkn<br>0011: pb4_clkp<br>0010: pb3_clkp<br>0001: pb2_clkp<br>0000: pb1_clkp |

### AUDIO CONTROL 1 REGISTER – 0X228

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:6 | R/W | 0 | **Port_sel[1:0]**<br><br>There are four $i^2s$ data input pin. This register select which pin will be sent to HDMI.<br>00: port 1<br>01: port 2<br>10: port 3<br>11: port 4 |
| 5 | R/W | 0 | **Bypass_I2s**<br><br>When enable this bit, $i^2s$ data will be directly sent to HDMI. For TW2864 $I^2S$ data, this bit must be set to 0 |
| 4 | R/W | 0 | **Mute**<br><br>0: normal sound<br>1: mute |
| 3:0 | R/W | 0 | **Ch_sel**<br><br>Select one of the 16 channels from TW2864 |

### AUDIO CONTROL 2 REGISTER – 0X229

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | **Asclk_sel**<br><br>ASCLK is for SPDIF format. This clock can be selected from external clock or from ck108 on chip.<br><br>0: external pin<br>1: internal 108MHz |
| 6 | R/W | 0 | **Audio_master**<br><br>0: slave mode, TW2864 provide clock<br>1: master mode, TW2880 provide clock |
| 5 | R/W | 0 | **Ups_en**<br><br>0: no up sample<br>1: up sample from 6k or 8k to 32k, only useful in master mode |
| 4 | R/W | 0 | **Pat_en**<br><br>This bit enable internal pattern |

| Bit | R/W | Default | Description |
|---|---|---|---|
| 3:2 | R/W | 0 | **Rate_sel**<br><br>Select sample rate, only useful in master mode<br>00: 8k<br>01: 16k<br>10, 11: 32k |
| 1:0 | R/W | 0 | **Pat_freq_sel**<br><br>Pattern selection<br>00: 1k<br>01: 2k<br>10: 4k<br>11: 8k |

### RECORDING PIN PORT DATA SELECT REGISTER — 0X22A

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | **Recording Port 1,2 Source Select**<br><br>1 = data is from main display / dual monitor<br>0 = data is from recording unit |
| 6 | R/W | 0 | **Recording Port 3,4 Source Select**<br><br>1 = data is from main display / dual monitor<br>0 = data is from recording unit |
| 5 | R/W | 0 | **Recording Port 1,2 Source Select**<br><br>1 = data is from dual monitor<br>0 = data is from main monitor |
| 4 | R/W | 0 | **Recording Port 3,4 Source Select**<br><br>1 = data is from dual monitor<br>0 = data is from main monitor |
| 3:0 | R | 00 | **Reserved** |

### CPU FAST ACCESS CONTROL REGISTER — 0X22C

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | | 0 | **Freeze Display Pitch register**<br><br>Bit 8 of the Freeze display pitch |
| 6:1 | R | 0 | **Reserved** |
| 1:0 | R/W | 11 | **CPU IO read access to control**<br><br>11: finish read in seven sclk<br>10: finish read in six sclk<br>01: finish read in five sclk<br>00: finish read in four sclk |

### CPU FREEZE DISPLAY PITCH REGISTER — 0X22D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | | 0 | **Freeze Display Pitch register [8:0]**<br><br>Bit [7:0] of the Freeze display pitch register |

### SCLK PHASE CONTROL REGISTER 3 —OFFSET 0X22E

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | | 0 | **Reserved** |
| 2:1 | R/W | 0 | **NETWORK_CLK Phase shift select**<br><br>11: $270^0$ phase shift<br>10: $180^0$ phase shift<br>01: $90^0$ phase shift<br>00: No phase shift |
| 0 | R/W | 0 | **NETWORK_CLK Phase Select**<br><br>1: Inverts the NETWORK_CLK<br>0: Normal |

### DRAM ADDRESS MAPPING REGISTER — 0X26D

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:3 | R | 0 | **Reserved** |
| 2 | R/W | 0 | **Record DRAM Controller Address Mapping Option**<br><br>1 = New<br>0 = Original |
| 1 | R | 0 | **Reserved** |
| 0 | R/W | 0 | **Display DRAM Controller Address Mapping Option**<br><br>1 = New<br>0 = Original |

### CHIP ID REGISTER — 0X270

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:4 | R | 0 | **28 Series ID**<br><br>Other = Reserved<br>0000 = TW2880 |
| 3:0 | R | 0 | **Stepping**<br><br>Other = Reserved<br>0001 = B1<br>0000 = A2 |

### DISPLAY DRAM PRIORITY ARBITRATION REGISTER 1 — 0X280

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | **Priority Arbitration[7:0]**<br><br>Set 1 to enable Clients arbitrate with high priority<br><br>Following table show client round robin number for each module<br><br>0 : lcd<br>1 : dmon<br>2 : di_wr<br>3 : rgbw<br>4 : di_rd<br>5 : spot_osd<br>6 : osgrd1<br>7 : osgrd2<br>8 : osgrd3<br>9 : dm_osgrd1<br>10 : dm_osgrd2<br>11 : host_dma<br>12 : osgw<br>13 : freeze<br>14 : cpu<br>15 : lcd_mouse<br>16 : dm_mouse |

### DISPLAY DRAM PRIORITY ARBITRATION REGISTER 2 — 0X281

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | **Priority Arbitration[15:8]**<br><br>See above. |

### DISPLAY DRAM PRIORITY ARBITRATION REGISTER 3 — 0X282

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | **Priority Arbitration[23:16]**<br><br>See above. |

### RECORD DRAM PRIORITY ARBITRATION REGISTER 1 — 0X284

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 3 | **Priority Arbitration[7:0]**<br><br>See below. |

**RECORD DRAM PRIORITY ARBITRATION REGISTER 2 — 0X285**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | **Priority Arbitration[15:8]**<br><br>See below. |

**RECORD DRAM PRIORITY ARBITRATION REGISTER 3 — 0X286**

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7:0 | R/W | 0 | **Priority Arbitration[23:16]**<br><br>Set 1 to enable Clients arbitrate with high priority<br><br>Following table show client round robin number for each module<br><br>0 : recw<br>1 : spw<br>2 : qcif<br>3 : rout1<br>4 : rout2<br>5 : rout3<br>6 : rout4<br>7 : rout5<br>8 : rout6<br>9 : rout7<br>10 : rout8<br>11 : net<br>12 : spot1<br>13 : spot2<br>14: spot3<br>15 : spot4<br>16 : mdw<br>17 : mdr<br>18 : dma_cpu2 |

# I²C MASTER CONTROL

## Introduction

TW2880 I²C master controller is the only interface to get access to the internal HDMI transmitter following the I²C bus specification. This is an embedded design within the TW2880 device. To access the HDMI transmitter slave, the firmware needs to follow a simple procedure routine to write to or read from The HDMI device.

## Features

- Only two pair of data and clock lines.
- The operating clock frequency is software programmable.
- Serial, 8-bit oriented data transfers can be made at fast-mode.
- No need to design bus interface because the I²C control is already integrated on-chip.
- Addressing protocol is flexible by register program
- To start the bus, only three registers need to be written in parallel interface.

## Operation

By accessing the parallel host registers in the register page 7, the I²C master controller will convert the registers data to serial data according to the I²C specification. For the registers definition, please refer to the register section.

To write to the HDMI registers, The following steps need to be followed:

Program the register I²C master registers 'X70, 71 to determine the I²C bus frequency.

To write data to the HDMI registers.

First select the I²C master address 'X72 with data 'X72 which is the HDMI slave address, write to the I²C master address 'X74 with data 'X96 which will kick off the I²C START operation. The I²C master initiate the START command and put `X72 to the serial data bus SDA along with the SCL clock. If the I²C master receive the ACK signal from the HDMI slave, it will turn on the INT and set the proper bit in the status register.

Clear the interrupt flag, write data 'X0f to I²C master controller command register address 'X74 to clear the interrupt flag.

Write 'X72 with the HDMI register address 'X_addr

Write 'X74 with the 'X16 which will continue put 'Xdd on the I²C bus

wait for the HDMI ACK

Clear the interrupt as 3.

Write 'X72 with the HDMI register data 'X_data

Wait for the HDMI ACK

Clear the interrupt by 'X4F to finish the I²C STOP operation.

**For the next HDMI register write, repeat numbers 2 to 10 again.**

**To read from the HDMI registers, The following steps need to be followed:**

**Write to I²C master address ‘X72 with HDMI slave address ‘X72**

**Wait for the HDMI ACK respond**

**Clear the interrupt by write ‘X74 with ‘X0F**

**Write to I²C master address ‘X72 with HDMI register address ‘X_addr**

**Wait for the HDMI ACK respond**

**Write to the I²C master address ‘X72 with HDMI slave address + R ‘X73**

**Wait for the HDMI ACK respond**

**Clear the interrupt by writing ‘X74 with ‘X0F**

**Write to I²C master ‘X74 with the value ‘X26**

**Wait for the interrupt flag**

**READ I²C master register ‘X73 which stored the read HDMI value.**

## Register Table

| Address | R/W | Default | Description |
|---|---|---|---|
| 0x770 | R/W | 0x00 | I²C Bus Frequency Control Register (Lower) |
| 0x771 | R/W | 0x00 | I²C Bus Frequency Control Register (Higher) |
| 0x772 | R/W | 0x00 | I²C Command / Data Register |
| 0x773 | R/W | 0x00 | I²C Data Read Register |
| 0x774 | R/W | 0x00 | I²C Command Register |
| 0x775 | R/W | 0x00 | I²C Command Status Register |

## Registers

### I²C BUS FREQUENCY CONTROL REGISTER (LOWER) —OFFSET 0X770

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7-0 | R/W | dd | I²C bus frequency control lower 8 bits |

### I²C BUS FREQUENCY CONTROL REGISTER (UPPER) —OFFSET 0X771

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7-0 | R/W | 00 | I²C bus frequency control upper 8 bits |

### I²C COMMAND / DATA REGISTER —OFFSET 0X772

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7-0 | R/W | 0 | [7:0] I2C slave address [7:1] and R/W command [0] control<br>Or use as data register<br>To access HDMI write, the value should be program 'X72 / 'x7A<br>To access HDMI read, the value should be program 'X73 / 'x7B<br>B0 = 0 write command<br>B0 = 1 Read command |

### I²C DATA READ REGISTER —OFFSET 0X773

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7-0 | R/W | dd | I²C Data Read Register |

### I²C COMMAND REGISTER —OFFSET 0X774

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R/W | 0 | $I^2C$ START |
| 6 | R/W | 0 | $I^2C$ STOP |
| 5 | R/W | 0 | Read the value from the HDMI and put it in the $I^2C$ data read register |
| 4 | R/W | 0 | Write the value in the $I^2C$ slave register or Data register |
| 3 | R/W | 0 | Send the ACK to the $I^2C$ bus when read |
| 2 | R/W | 0 | Clock count enable |
| 1 | R/W | 0 | Interrupt enable |
| 0 | R/W | 0 | Interrupt acknowledge |

### I²C COMMAND STATUS REGISTER —OFFSET 0X775

| Bit | R/W | Default | Description |
|---|---|---|---|
| 7 | R | 0 | I²C START |
| 6 | R | 0 | I²C STOP |
| 5 | R | 0 | Read the value from the HDMI and put it in the I²C data read register |
| 4 | R | 0 | Write the value in the I²C slave register or Data register |
| 3 | R | 0 | Send the ACK to the I²C bus when read |
| 2 | R | 0 | Clock count enable |
| 1 | R | 0 | Interrupt enable |
| 0 | R | 0 | Interrupt acknowledge |

# Revision History

| SECTION | REVISION SUMMARY | DATE |
|---|---|---|
| V0.9 | First Setup document directory and copy data from Rev. B 0508 file | 07/15/10 |
| V0.91 | Modified Privacy windows size info. | 07/26/10 |
| | Modified display image to record using BT.1120 | 07/26/10 |
| | Add Live channel correction registers | 07/26/10 |
| | Add non standard definition registers | 07/29/10 |
| | Add new rgb_interface registers | 07/30/10 |
| | Replace recording section with new file | 07/31/10 |
| | Add stop-n-go interrupt register | 08/03/10 |
| | Modify 0x3db register description | 08/04/10 |
| | Add Privacy counter overflow bit for H and V | 08/05/10 |
| | Add PHSYNC[9] overflow bit for VESA 50 Hz mode | 08/06/10 |
| | Add user color register to privacy window | 08/12/10 |
| | Add OSG Windows address update | 08/12/10 |
| | Add Intersil logo | 08/12/10 |
| | Delete Gamma Section | 08/15/10 |
| | Add overflow bit in privacy windows | 08/15/10 |
| | Add ignored channel definition in CHID | 08/16/10 |
| | Add register table description for rgb and PB | 08/20/10 |
| | Modified DC characteristics | 10/11/10 |
| | Modified input and output Min. / Max. value | 10/11/10 |
| | Delete HCLK from DMA diagram | 10/11/10 |
| | Change host interface read latching diagram | 10/28/10 |
| | Fixed 0x4c3.4 definition typo | 11/09/10 |
| | Add 0x3ff bit 7 description | 12/03/10 |
| | Add DRAM priority control | 12/22/10 |
| | Modify PB path selection explanation | 12/23/10 |
| | Add 0xc68 record port clock description | 01/03/11 |
| | Add 0x201 bit 7 description | 01/03/11 |
| | Modified 0x3ff bit 6, 5 meaning | 01/06/11 |
| | Modified 0x6fd, 0x6fe register wording | 01/14/11 |
| | Correction 0x6f1, 0x6f0 PB mode definition | 01/17/11 |

| SECTION | REVISION SUMMARY | DATE |
|---|---|---|
| | Correct SPOT register definition for C | 02/09/11 |
| | Add DRAM priority register into register table | 02/15/11 |
| | Add Channel Ignore description in 0x6fa | 02/18/11 |
| | Modified 0x3f2 definition | 02/22/11 |
| | Change DMON BT.1120 source | 02/23/11 |
| | Fix HDMI Amplitude Control Description typo | 03/11/11 |
| | Fix power on sequence voltage / time requirement | 03/21/11 |
| | Add 0x6fc [2] description | 03/30/11 |
| | Change the wording on PB to include 4 HD port | 04/04/11 |
| | Change 0xF18 [5] definition | 06/01/11 |
| | Put in more SPOT section register changes | 06/02/11 |
| | Perform spelling check | 06/24/11 |
| | Added Ordering Information table.<br>Applied Intersil template. Assigned Intersil File Number.<br>Applied Intersil standard watermarks that are required for pre-release datasheet.<br>Removed Techwell Disclaimer on page 1 per Intersil Legal Dept. Replaced by Intersil's standard copyright notice and disclaimer.<br>Updated Spec tables to add Intersil standard note to MIN MAX column (Compliance to datasheet limits is assured by one or more methods: production test, characterization and/or design)<br>Updated Caution statement in Absolute Maximum Ratings table per Intersil Legal.<br>Added Intersil disclaimer to page 635. | 6/30/11 |
| | Modified P.184 HDMI I2C port address selection | 7/15/11 |
| | Change P.36 resolution road map | 7/15/11 |
| | Change summary page description | 7/16/11 |
| | Change Page 1 description | 8/08/11 |
| | Change SPOT bits description | 8/08/11 |

| SECTION | REVISION SUMMARY | DATE |
|---|---|---|
| | Add back Host interface table outline | 8/08/11 |
| | Change 0x3f8 – 0x3fb default values. | 11/10/11 |
| | Change 0xc4[1:0] = 2'b11 definition | 11/11/11 |
| | Change the wording and symbol used in 0x3c6 and 0x3ff to make it more clear | 11/14/11 |
| | Added Power off sequence to last paragraph in Power On Sequence section on page 604 | 6/4/13 |

© Copyright Intersil Americas LLC 2010-2013. All Rights Reserved.
All trademarks and registered trademarks are the property of their respective owners.

For additional products, see www.intersil.com/product_tree
Intersil products are manufactured, assembled and tested utilizing ISO9001 quality systems as noted in the quality certifications found at http://www.intersil.com/en/support/qualandreliability.html

*Intersil products are sold by description only. Intersil may modify the circuit design and/or specifications of products at any time without notice, provided that such modification does not, in Intersil's sole judgment, affect the form, fit or function of the product. Accordingly, the reader is cautioned to verify that datasheets are current before placing orders. Information furnished by Intersil is believed to be accurate and reliable. However, no responsibility is assumed by Intersil or its subsidiaries for its use; nor for any infringements of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Intersil or its subsidiaries.*

For information regarding Intersil Corporation and its products, see http://www.intersil.com