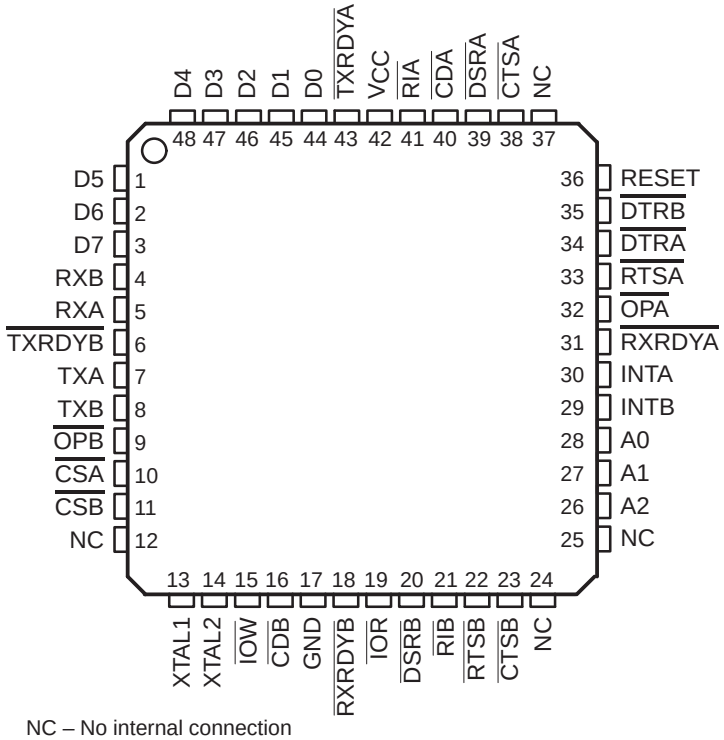
- Pin Compatible With ST16C2550 With Additional Enhancements
- Supports Up to 3.125 Mbps Baud Rate
  - Up to 2.1875 Mbps Baud Rate When Using Crystal (35 MHz Input Clock)
  - Up to 3.125 Mbps Baud Rate When Using Oscillator or Clock Source (50 MHz Input Clock)
- 64-Byte Transmit FIFO
- 64-Byte Receive FIFO With Error Flags
- Programmable and Selectable Transmit and Receive FIFO Trigger Levels for DMA and Interrupt Generation
- Programmable Receive FIFO Trigger Levels for Software/Hardware Flow Control
- Software/Hardware Flow Control
  - Programmable Xon/Xoff Characters
  - Programmable Auto-$\overline{\text{RTS}}$ and Auto-$\overline{\text{CTS}}$
- Software Flow Control Turned Off Optionally by Any Xon Rx Character
- DMA Signalling Capability for Both Received and Transmitted Data
- Supports 3.3-V Operation
- Software Selectable Baud Rate Generator Prescaleable Clock Rates of 1X and 4X
- Fast Access Time 2 Clock Cycle $\overline{\text{IOR}}$/$\overline{\text{IOW}}$ Pulse Width
- Programmable Sleep Mode
- Programmable Serial Interface Characteristics
  - 5, 6, 7, or 8Bit Characters
  - Even, Odd, or No Parity Bit Generation and Detection
  - 1, 1.5, or 2 Stop Bit Generation
- False Start Bit Detection
- Complete Status Reporting Capabilities in Both Normal and Sleep Mode
- Line Break Generation and Detection
- Internal Test and Loopback Capabilities
- Fully Prioritized Interrupt System Controls
- Modem Control Functions ($\overline{\text{CTS}}$, $\overline{\text{RTS}}$, $\overline{\text{DSR}}$, $\overline{\text{DTR}}$, $\overline{\text{RI}}$, and $\overline{\text{CD}}$)

**PACKAGE**
**(TOP VIEW)**




NC – No internal connection

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

## description

The TL16C752 is a dual universal asynchronous receiver/transmitter (UART) with 64-byte FIFOs, automatic hardware/software flow control, and data rates up to 3.125 Mbps. The ’752 offers enhanced features. It has a transmission control register (TCR) that stores received FIFO threshold level to start/stop transmission during hardware and sofware flow control. With the FIFO RDY register, the software gets the status of TXRDY/RXRDY for all four ports in one access. Onboard status registers provide the user with error indications and operational status, modem interface control. System interrupts may be tailored to meet user requirements. An internal loopback capability allows onboard diagnostics.

The UART transmits data sent to it from the peripheral 8-bit bus on the TX signal and receives characters on the RX signal. Characters can be programmed to be 5, 6, 7, or 8 bits. The UART has a 64-byte receive FIFO and transmit FIFO and can be programmed to interrupt at different trigger levels. The UART generates its own desired baud rate based upon a programmable divisor and its input clock. It can transmit even, odd, or no parity and 1, 1.5, or 2 stop bits. The receiver can detect break, idle, or framing errors, FIFO overflow, and parity errors. The transmitter can detect FIFO underflow. The UART also contains a software interface for modem control operations, and has software flow control and hardware flow control capabilities.

The TL16C752 is available in a 48-pin PT (LQFP) package. The 48-pin version offers three state interrupt control and provides constant active interrupt outputs.

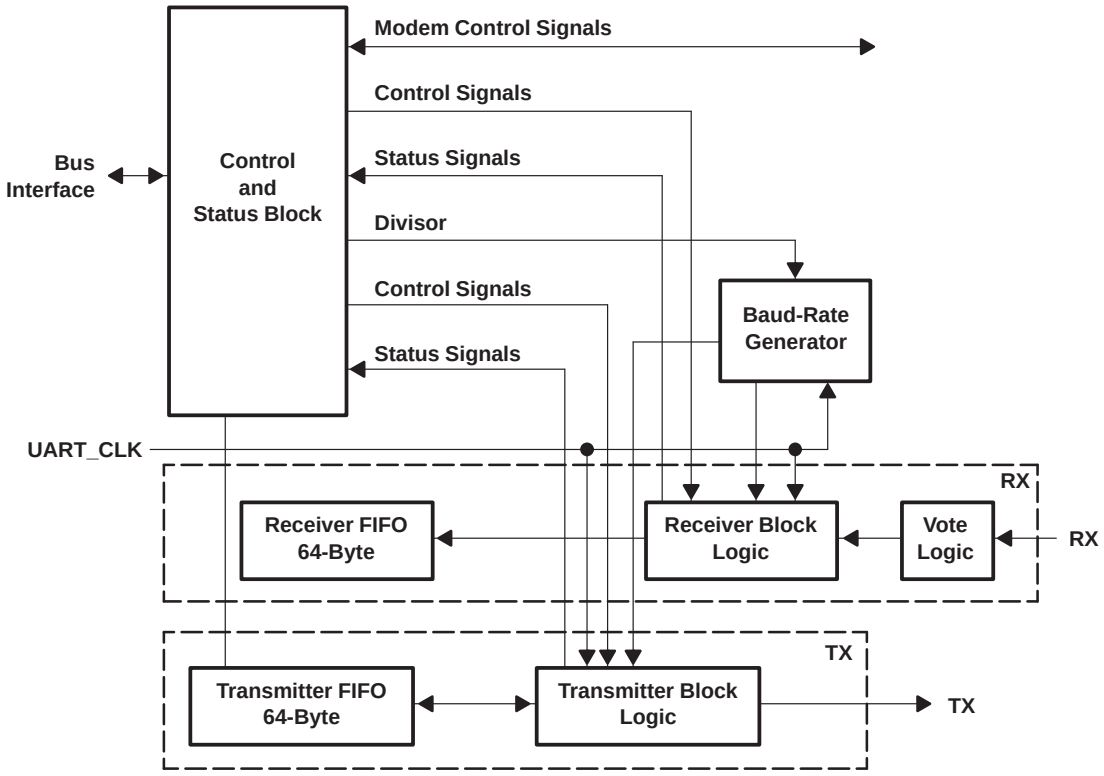
### Terminal Functions

| TERMINAL | | I/O | DESCRIPTION |
|---|---|---|---|
| NAME | NO. | | |
| A0 | 28 | I | Address 0 select bit. Internal registers address selection. |
| A1 | 27 | I | Address 1 select bit. Internal registers address selection. |
| A2 | 26 | I | Address 2 select bit. Internal registers address selection. |
| $\overline{\text{CDA}}$, $\overline{\text{CDB}}$ | 40, 16 | I | Carrier detect (active low). These inputs are associated with individual UART channels A through D. A 0 on these pins indicates that a carrier has been detected by the modem for that channel. |
| $\overline{\text{CSA}}$, $\overline{\text{CSB}}$ | 10, 11 | I | Chip select A and B (active low). These pins enable data transfers between the user CPU and the TL16C752 for the channel(s) addressed. Individual UART sections (A, B) are addressed by providing a 0 on the respective $\overline{\text{CS}}$ A and $\overline{\text{CS}}$ B pins. |
| $\overline{\text{CTSA}}$, $\overline{\text{CTSB}}$ | 38, 23 | I | Clear to send (active low). These inputs are associated with individual UART channels A and B. A logic 0 on the CTS pins indicates the modem or data set is ready to accept transmit data from the 752. Status can be tested by reading MSR bit 4. These pins only affect the transmit and receive operations when auto CTS function is enabled through the enhanced feature register (EFR) bit 7, for hardware flow control operation. |
| D0–D4<br>D5–D7 | 44–48,<br>1–3 | I/O | Data bus (bidirectional). These pins are the eight bit, three state-data bus for transferring information to or from the controlling CPU. D0 is the least significant bit and the first data bit in a transmit or receive serial data stream. |
| $\overline{\text{DSRA}}$, $\overline{\text{DSRB}}$ | 39, 20 | I | Data set ready (active low). These inputs are associated with individual UART channels A and B. A logic 0 on these pins indicates the modem or data set is powered on and is ready for data exchange with the UART. These pins have no effect on the UART transmit or receive operation. |
| $\overline{\text{DTRA}}$, $\overline{\text{DTRB}}$ | 34, 35 | O | Data terminal ready (active low). These outputs are associated with individual UART channels A and B. A logic 0 on these pins indicates that the 752 is powered on and ready. These pins can be controlled through the modem control register. Writing a 1 to MCR bit 0 sets the $\overline{\text{DTR}}$ output to 0, enabling the modem. These pins are a logic 1 after writing a 0 to MCR bit 0, or after a reset. These pins have no effect on the UART transmit or receive operation. |
| GND | 17 | Pwr | Signal and power ground |
| INTA, INTB | 30, 29 | O | Interrupt A and B (active high). These pins provide individual channel interrupts, INT A and B. INT A and B are enabled when MCR bit 3 is set to a logic 1, interrupts are enabled in the interrupt enable register (IER), and when an interrupt condition exists. Interrupt conditions include: receiver errors, available receiver buffer data, transmit buffer empty, or when a modem status flag is detected. |
| $\overline{\text{IOR}}$ | 19 | I | Read input (active low strobe). A high to low transition on $\overline{\text{IOR}}$ will load the contents of an internal register defined by address bits A0–A2 onto the TL16C752 data bus (D0–D7) for access by an external CPU. |

## Terminal Functions (Continued)

| TERMINAL | | I/O | DESCRIPTION |
|---|---|---|---|
| NAME | NO. | | |
| $\overline{IOW}$ | 15 | I | Write input (active low strobe). A high to low transition on $\overline{IOW}$ will transfer the contents of the data bus (D0–D7) from the external CPU to an internal register that is defined by address bits A0–A2 and $\overline{CSA}$ and $\overline{CSB}$ |
| $\overline{OPA}$, $\overline{OPB}$ | 32, 9 | 0 | User defined outputs. This function is associated with individual channels A and B. The state at these pins is defined by the user and through the software settings of the MCR register, bit 3. INTA–B are set to active mode and OP to a logic 0 when the MCR–3 is set to a logic 1. INTA–B are set to the thru state mode and OP is set to a logic 0. See bit 3, Modem Control Register (MCR bit 3). |
| RESET | 36 | I | Reset. RESET will reset the internal registers and all the outputs. The UART transmitter output and the receiver input will be disabled during reset time. See TL16C752 external reset conditions for initialization details. |
| $\overline{RIA}$, $\overline{RIB}$ | 41, 21 | I | Ring indicator (active low). These inputs are associated with individual UART channels A and B. A logic 0 on these pins indicates the modem has received a ringing signal from the telephone line. A logic 1 transition on these input pins generates an interrupt. |
| $\overline{RTSA}$, $\overline{RTSB}$ | 33, 22 | O | Request to send (active low). These outputs are associated with individual UART channels A and B. A logic 0 on the $\overline{RTS}$ pins indicates the transmitter has data ready and waiting to send. Writing a 1 in the modem control register (MCR bit 1) sets these pins to 0, indicating data is available. After a reset, these pins are set to 1. These pins only affects the transmit and receive operation when auto RTS function is enabled through the enhanced feature register (EFR) bit 6, for hardware flow control operation. |
| RXA, RXB | 5, 4 | I | Receive data input. These inputs are associated with individual serial channel data to the 752. During the local loopback mode, these RX input pins are disabled and TX data is internally connected to the UART RX input internally. |
| $\overline{RXRDYA}$, $\overline{RXRDYB}$ | 31, 18 | O | Receive ready (active low). RXRDY A and B goes low when the trigger level has been reached or a timeout interrupt occurs. They goes high when the RX FIFO is empty and there is an error in RX FIFO. |
| TXA, TXB | 7, 8 | O | Transmit data. These outputs are associated with individual serial transmit channel data from the 752. During the local loopback mode, the TX input pin is disabled and TX data is internally connected to the UART RX input. |
| $\overline{TXRDYA}$, $\overline{TXRDYB}$ | 43, 6 | O | Transmit ready (active low). $\overline{TXRDY}$ A and B go low when there trigger level numbers of spares available. They go high when the TX buffer is full. |
| $V_{CC}$ | 42 | I | Power supply inputs. |
| XTAL1 | 13 | I | Crystal or external clock input. XTAL1 functions as a crystal input or as an external clock input. A crystal can be connected between XTAL1 and XTAL2 to form an internal oscillator circuit (see Figures 9 and 10). Alternatively, an external clock can be connected to XTAL1 to provide custom data rates. |
| XTAL2 | 14 | O | Output of the crystal oscillator or buffered clock. See also XTAL1. XTAL2 is used as a crystal oscillator output or buffered a clock output. |

## functional block diagram




NOTE: The Vote logic determines whether the RX data is a logic 1 or 0. It takes three samples of the RX line, and uses a majority vote to determine the logic level received. The Vote logic operates on all bits received.

## functional description

The TL16C752 UART is pin compatible with the ST16C2550 UART. It provides more enhanced features. All additional features are provided through a special enhanced feature register.

The UART will perform serial-to-parallel conversion on data characters received from peripheral devices or modems and parallel-to-parallel conversion on data characters transmitted by the processor. The complete status of each channel of the TL16C752 UART can be read at any time during functional operation by the processor.

The TL16C752 can be placed in an alternate mode (FIFO mode) relieving the processor of excessive software overhead by buffering received/transmitted characters. Both the receiver and transmitter FIFOs can store up to 64 bytes (including three additional bits of error status per byte for the receiver FIFO) and have selectable or programmable trigger levels. Primary outputs $\overline{\text{RXRDY}}$ and $\overline{\text{TXRDY}}$ allow signalling of DMA transfers.

The TL16C752 will have selectable hardware flow control and software flow control. Hardware flow control significantly reduces software overhead and increases system efficiency by automatically controlling serial data flow using the $\overline{\text{RTS}}$ output and $\overline{\text{CTS}}$ input signals. Software flow control automatically controls data flow by using programmable Xon/Xoff characters.

The UART will include a programmable baud rate generator that can divide the timing reference clock input by a divisor between 1 and ($2^{16}$–1).

## functional description (continued)

### trigger levels

The TL16C752 provides independent selectable and programmable trigger levels for both receiver and transmitter DMA and interrupt generation. After reset, both transmitter and receiver FIFOs are disabled and so, in effect, the trigger level is the default value of one byte. The selectable trigger levels are available via the FCR. The programmable trigger levels are available via the TLR.

### hardware flow control

Hardware flow control is composed on auto-$\overline{\text{CTS}}$ and auto-$\overline{\text{RTS}}$. Auto-$\overline{\text{CTS}}$ and auto-$\overline{\text{RTS}}$ can be enabled/disabled independently by programming EFR[7:6].

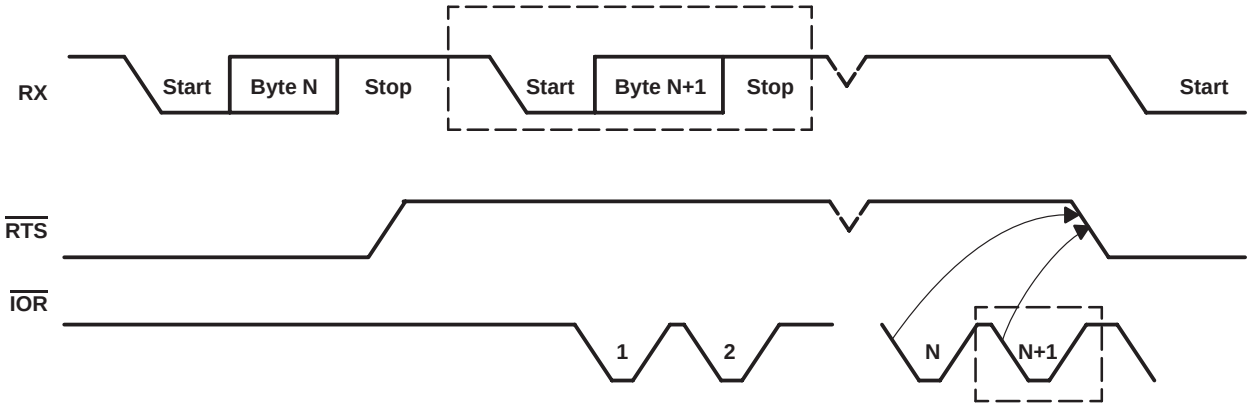
With auto-$\overline{\text{CTS}}$, $\overline{\text{CTS}}$ must be active before the UART can transmit data.

Auto-$\overline{\text{RTS}}$ only activates the RTS output when there is enough room in the FIFO to receive data and deactivates the RTS output when the RX FIFO is sufficiently full. The HALT and RESTORE trigger levels in the TCR determine the levels at which RTS is activated/deactivated.

If both auto-$\overline{\text{CTS}}$ and auto-$\overline{\text{RTS}}$ are enabled, when $\overline{\text{RTS}}$ is connected to $\overline{\text{CTS}}$, data transmission does not occur unless the receiver FIFO has empty space. Thus, overrun errors are eliminated during hardware flow control. If not enabled, overrun errors occur if the transmit data rate exceeds the receive FIFO latency.

### auto-$\overline{\text{RTS}}$

Auto-$\overline{\text{RTS}}$ data flow control originates in the receiver block (see functional block diagram). Figure 1 shows $\overline{\text{RTS}}$ functional timing. The receiver FIFO trigger levels used in Auto-$\overline{\text{RTS}}$ are stored in the TCR. RTS is active if the RX FIFO level is below the HALT trigger level in TCR[3:0]. When the receiver FIFO HALT trigger level is reached, $\overline{\text{RTS}}$ is deasserted. The sending device (e.g., another UART) may send an additional byte after the trigger level is reached (assuming the sending UART has another byte to send) because it may not recognize the deassertion of $\overline{\text{RTS}}$ until it has begun sending the additional byte. $\overline{\text{RTS}}$ is automatically reasserted once the receiver FIFO reaches the RESUME trigger level programmed via TCR[7:4]. This reassertion allows the sending device to resume transmission.



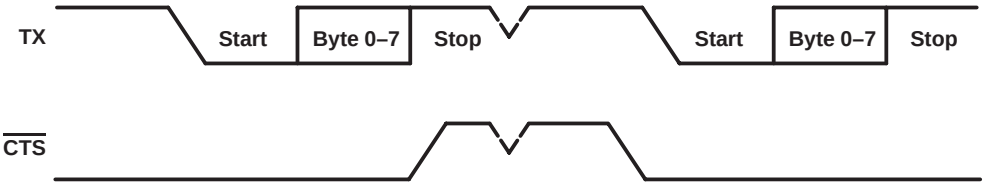

NOTES: 1. N = receiver FIFO trigger level
2. The two blocks in dashed lines cover the case where an additional byte is sent as described in Auto-$\overline{\text{RTS}}$.

**Figure 1. $\overline{\text{RTS}}$ Functional Timing**

## functional description (continued)

### auto-$\overline{\text{CTS}}$

The transmitter circuitry checks $\overline{\text{CTS}}$ before sending the next data byte. When $\overline{\text{CTS}}$ is active, the transmitter sends the next byte. To stop the transmitter from sending the following byte. $\overline{\text{CTS}}$ must be deasserted before the middle of the last stop bit that is currently being sent. The auto-$\overline{\text{CTS}}$ function reduces interrupts to the host system. When flow control is enabled, the $\overline{\text{CTS}}$ state changes and need not trigger host interrupts because the device automatically controls its own transmitter. Without auto-$\overline{\text{CTS}}$, the transmitter sends any data present in the transmit FIFO and a receiver overrun error can result. Figure 2 shows $\overline{\text{CTS}}$ functional timing, and Figure 3 shows an example of autoflow control.




NOTES: A. When $\overline{\text{CTS}}$ is low, the transmitter keeps sending serial data out
B. When $\overline{\text{CTS}}$ goes high before the middle of the last stop bit of the current byte, the transmitter finishes sending the current byte but it does not send the next byte.
C. When $\overline{\text{CTS}}$ goes from high to low, the transmitter begins sending data again.

**Figure 2. $\overline{\text{CTS}}$ Functional Timing**

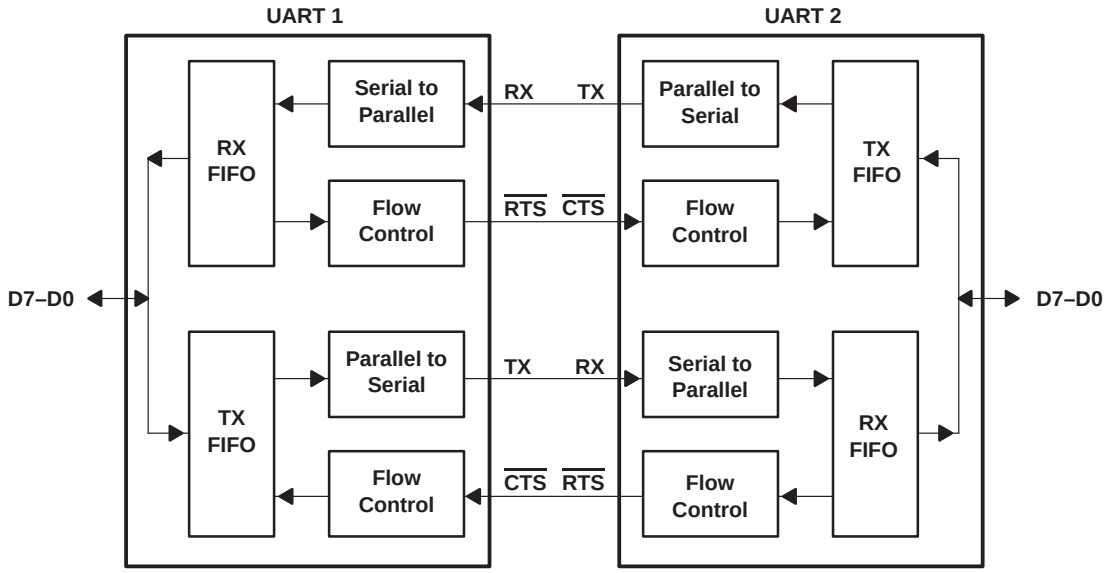



**Figure 3. Autoflow Control (Auto-$\overline{\text{RTS}}$ and Auto-$\overline{\text{CTS}}$) Example**

## functional description (continued)

### software flow control

Software flow control is enabled through the enhanced feature register and the modem control register. Different combinations of software flow control can be enabled by setting different combinations of EFR[3–0]. Table 1 shows software flow control options.

There are two other enhanced features relating to S/W flow control:

– **Xon Any Function [MCR950:** Operation will resume after receiving any character after recognizing the Xoff character.

**NOTE:**
It is possible that an Xon1 character is recognized as an Xon Any character which could cause an Xon2 character to be written to the RX FIFO.

– **Special Character [EFR(5)]:** Incoming data is compared to Xoff2. Detection of the special character sets the Xoff interrupt {IIR(4)] but does not halt transmission. The Xoff interrupt is cleared by a read of the IIR. The special character is transferred to the R FIFO.

**Table 1. Software Flow Control Options EFR[0:3]**

| BIT 3 | BIT 2 | BIT 1 | BIT 0 | Tx, Rx SOFTWARE FLOW CONTROLS |
|---|---|---|---|---|
| 0 | 0 | X | X | No transmit flow control |
| 1 | 0 | X | X | Transmit Xon1, Xoff1 |
| 0 | 1 | X | X | Transmit Xon2, Xoff2 |
| 1 | 1 | X | X | Transmit Xon1, Xon2: Xoff1, Xoff2 |
| X | X | 0 | 0 | No receive flow control |
| X | X | 1 | 0 | Receiver compares Xon1, Xoff1 |
| X | X | 0 | 1 | Receiver compares Xon2, Xoff2 |
| 1 | 0 | 1 | 1 | Transmit Xon1, Xoff1<br>Receiver compares Xon1 and Xon2, Xoff1 and Xoff2 |
| 0 | 1 | 1 | 1 | Transmit Xon2, Xoff2<br>Receiver compares Xon1 and Xon2, Xoff1 and Xoff2 |
| 1 | 1 | 1 | 1 | Transmit Xon1, Xon2: Xoff1, Xoff2Xoff1<br>Receiver compares Xon1 and Xon2: Xoff1 and Xoff2 |
| 0 | 0 | 1 | 1 | No transmit flow control<br>Receiver compares Xon1 and Xon2: Xoff1 and Xoff2 |

***RX***

When software flow control operation is enabled, the TL16C752 will compare incoming data with Xoff1/2 programmed characters (in certain cases Xoff1 and Xoff2 must be received sequentially[1]). When the correct Xoff characters are received, transmission is halted after completing transmission of the current character. Xoff detection also sets IIR[4] (if enabled via IER[5]) and causes INT to go high.

To resume transmission an Xon1/2 character must be received (in certain cases Xon1 and Xon2 must be received sequentially). When the correct Xon characters are received IIR[4] is cleared and the Xoff interrupt disappears.

**NOTE:**
If a Parity, Framing or Break error occurs while receiving a software flow control character, this character will be treated as normal data and will be written to the RCV FIFO.

1. When pairs of Xon/Xoff characters are programmed to occur sequentially, received Xon1/Xoff1 characters must be written to the Rx FIFO if the subsequent character is not Xon2/Xoff2.

## functional description (continued)

### *TX*

Xoff1/2 characters are transmitted when the RX FIFO has passed the programmed trigger level TCR[3:0].

Xon1/2 characters are transmitted when the RX FIFO reaches the trigger level programmed via TCR[7:4].

An important note here is that if, after an xoff character has been sent, software flow control is disabled the UART will transmit Xon characters automatically to enable normal transmission to proceed. A feature of the TL16C752 UART design is that if the software flow combination (EFR[3:0]) changes after an Xoff has been sent, the originally programmed Xon is automatically sent. If the RX FIFO is still above the trigger level, the newly programmed Xoff1/2 will be transmitted.

The transmission of Xoff/Xon(s) follows the exact same protocol as transmission of an ordinary byte from the FIFO. This means that even if the word length is set to be 5, 6, or 7 characters then the 5, 6, or 7 least significant bits of Xoff1,2/Xon1,2 will be transmitted. (Note that the transmission of 5, 6, or 7 bits of a character is seldom done, but this functionality is included to maintain compatibility with earlier designs.)

It is assumed that software flow control and hardware flow control will never be enabled simultaneously. Figure 4 shows an example of software flow control.

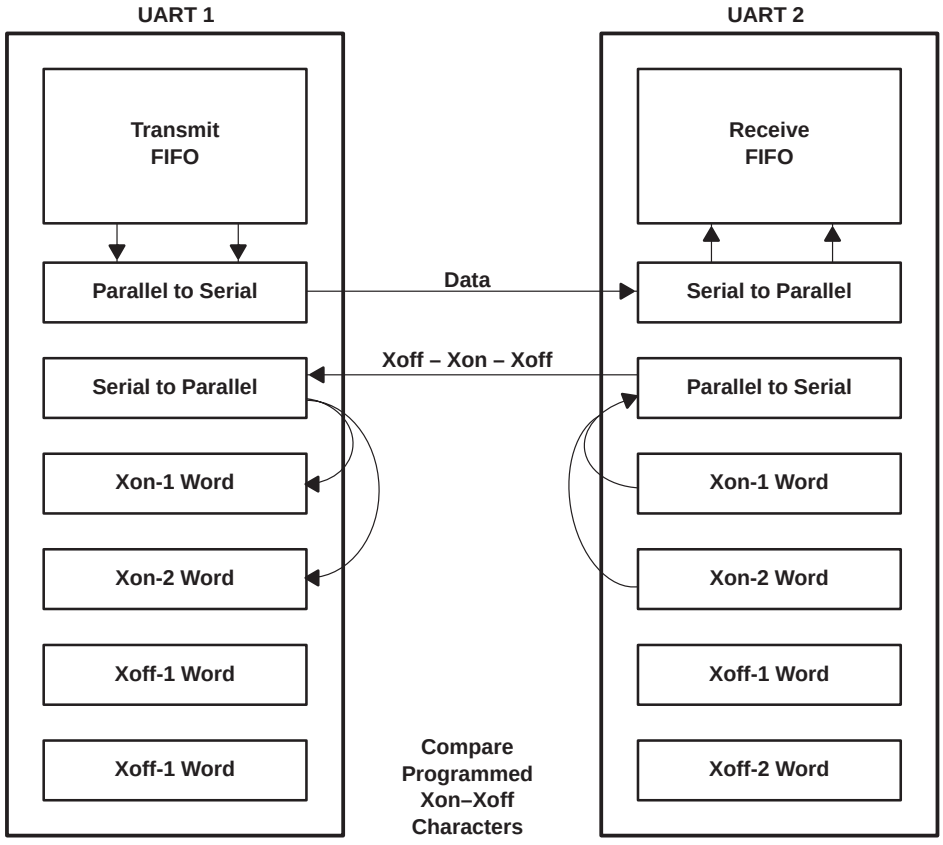



**Figure 4. Software Flow Control Example**

## functional description (continued)

### reset

Table 2 summarizes the state of registers after reset.

**Table 2. Register Reset Functions**

| REGISTER | RESET CONTROL | RESET STATE |
|---|---|---|
| Interrupt enable register | RESET | All bits cleared |
| Interrupt identification register | RESET | Bit 0 is set. All other bits cleared. |
| FIFO control register | RESET | All bits cleared |
| Line control register | RESET | Reset to 00011101 (1D hex). |
| Modem control register | RESET | All bits cleared |
| Line status register | RESET | Bits 5 and 6 set. All other bits cleared. |
| Modem status register | RESET | Bits 0–3 cleared. Bits 4–7 input signals. |
| Enhanced feature register | RESET | All bits cleared |
| Receiver holding register | RESET | Pointer logic cleared |
| Transmitter holding register | RESET | Pointer logic cleared |
| Transmission control register | RESET | All bits cleared |
| Trigger level register | RESET | All bits cleared |

NOTE: Registers DLL, DLH, SPR, Xon1, Xon2, Xoff1, Xoff2 are not reset by the top-level reset signal RESEST, i.e., they hold their initialization values during reset.

Table 3 summarizes the state of registers after reset.

**Table 3. Signal Reset Functions**

| SIGNAL | RESET CONTROL | RESET STATE |
|---|---|---|
| TX | RESET | High |
| $\overline{\text{RTS}}$ | RESET | High |
| $\overline{\text{DTR}}$ | RESET | High |
| $\overline{\text{RXRDY}}$ | RESET | High |
| $\overline{\text{TXRDY}}$ | RESET | Low |

### interrupts

The TL16C752 has interrupt generation and prioritization (6 prioritized levels of interrupts) capability. The Interrupt enable register (IER) enables each of the 6 types of interrupts and the INT signal in response to an interrupt generation. The IER can also disable the interrupt system by clearing bits 0–3, 5–7. When an interrupt is generated, the IIR indicates that an interrupt is pending and provides the type of interrupt through IIR[5–0]. Table 4 summarizes the interrupt control functions.

## functional description (continued)

**Table 4. Interrupt Control Functions**

| IIR[5–0] | PRIORITY LEVEL | INTERRUPT TYPE | INTERRUPT SOURCE | INTERRUPT RESET METHOD |
|---|---|---|---|---|
| 000001 | None | None | None | None |
| 000110 | 1 | Receiver line status | OE, FE, PE, or BI errors occur in characters in the RX FIFO | FE< PE< BI: All erroneous characters are read from the RX FIFO.<br>OE: Read LSR |
| 001100 | 2 | RX timeout | Stale data in RX FIFO | Read RHR |
| 000100 | 2 | RHR interrupt | DRDY (data ready)<br>(FIFO disable)<br>RX FIFO above trigger level (FIFO enable) | Read RHR |
| 000010 | 3 | THR interrupt | TFE (THR empty)<br>(FIFO disable)<br>TX FIFO passes below trigger level (FIFO enable) | Read IIR OR a write to the THR |
| 000000 | 4 | Modem status | MSR[3:0]/= 0 | Read MSR |
| 010000 | 5 | Xoff interrupt | Receive Xoff character(s)/special character | Receive Xon character(s)/Read of IIR |
| 100000 | 6 | CTS, RTS | $\overline{\text{RTS}}$ pin or $\overline{\text{CTS}}$ pin change state from active (low) to inactive (high) | Read IIR |

It is important to note that for the receiver line status interrupt, it is LSR[7] which generates the interrupt. LSR[4–2] are set when an erroneous character is read from the RX FIFO and they are cleared on a read of the LSR. LSR[7] is set when there is an error anywhere in the RX FIFO and is cleared only when there are no more errors remaining in the FIFO.

For the Xoff interrupt, if an Xoff flow character detection caused the interrupt, the interrupt is cleared by an Xon flow character detection. If a special character detection caused the interrupt, the interrupt is cleared by a read of the ISR.

### interrupt mode operation

In FIFO interrupt mode (FCR=1, IER[3:0] = 1) the processor is informed of the status of the receiver and transmitter by an interrupt signal, INT. Therefore, it is not necessary to continuously poll the line stats register (LSR) to see if any interrupts need to be serviced. Figure 5 shows FIFO interrupt mode operation.

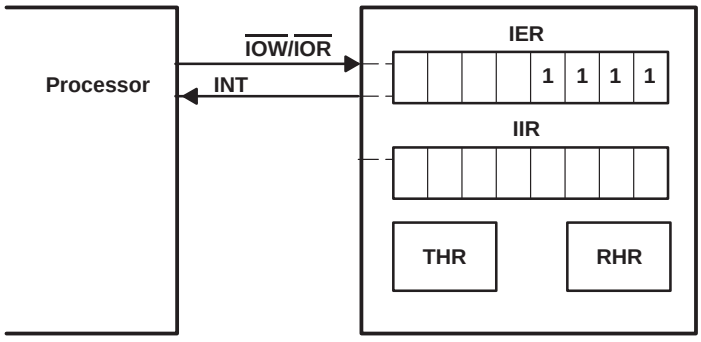



**Figure 5. FIFO Interrupt Mode Operation**

## functional description (continued)

### FIFO polled mode operation

In FIFO polled mode (FCR=1, IER[3:0] = 0) the status of the receiver and transmitter can then be checked by polling the line status register (LSR). This mode is an alternative to the FIFO interrupt mode of operation where the status of the receiver and transmitter is automatically known by means of interrupts sent to the CPU. Figure 6 shows FIFO polled mode operation.

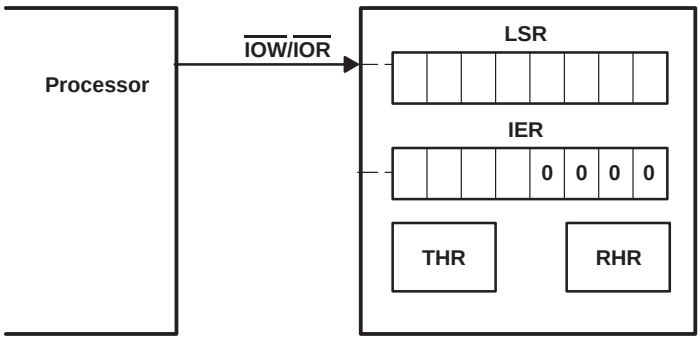



**Figure 6. FIFO Polled Mode Operation**

### DMA signalling

There are two modes of DMA operation: DMA mode 0 or 1, selected by FCR[3].

In DMA mode 0 or FIFO disable (FCR[0]=0) DMA occurs in single character transfers. In DMA mode 1 multi-character (or block) DMA transfers are catered for to relieve the processor for longer periods of time.

#### *single DMA transfers (DMA mode0/FIFO disable)*

Transmitter: When empty, the $\overline{\text{TXRDY}}$ signal becomes active. $\overline{\text{TXRDY}}$ will go inactive after one character has been loaded into it.

Receiver: $\overline{\text{RXRDY}}$ is active when there is at least one character in the FIFO. It becomes inactive when the receiver is empty.

Figure 7 shows $\overline{\text{TXRDY}}$ and $\overline{\text{RXRDY}}$ in DMA mode0/FIFO disable.

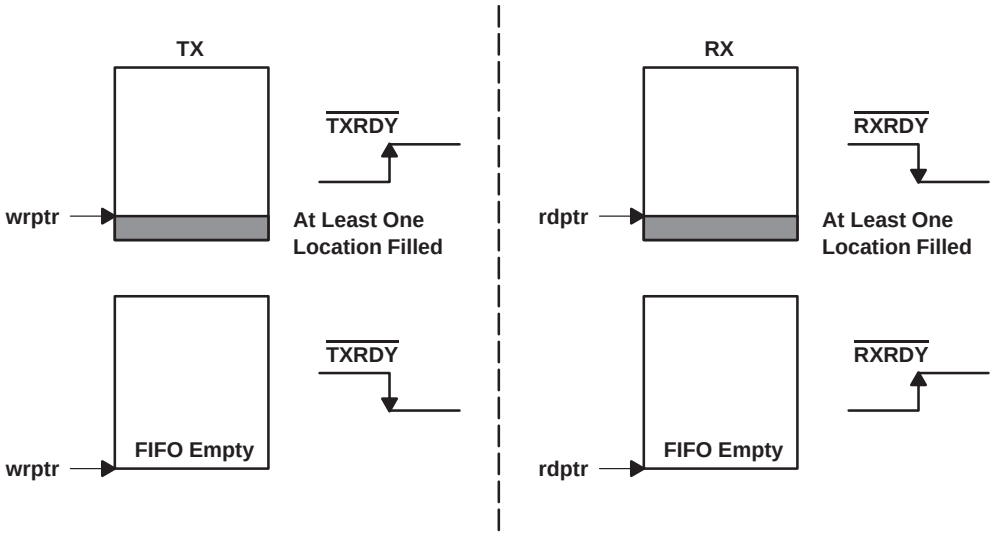



**Figure 7. $\overline{\text{TXRDY}}$ and $\overline{\text{RXRDY}}$ in DMA Mode 0/FIFO Disable**

## functional description (continued)

### *block DMA transfers (DMA mode 1)*

Transmitter: $\overline{\text{TXRDY}}$ is active when there is a trigger level number of spaces available. It becomes inactive when the FIFO is full.

Receiver: $\overline{\text{RXRDY}}$ becomes active when the trigger level has been reached OR when a timeout interrupt occurs. It will go inactive when the FIFO is empty OR an error in the RX FIFO is flagged by LSR(7)

Figure 8 shows $\overline{\text{TXRDY}}$ and $\overline{\text{RXRDY}}$ in DMA mode 1.

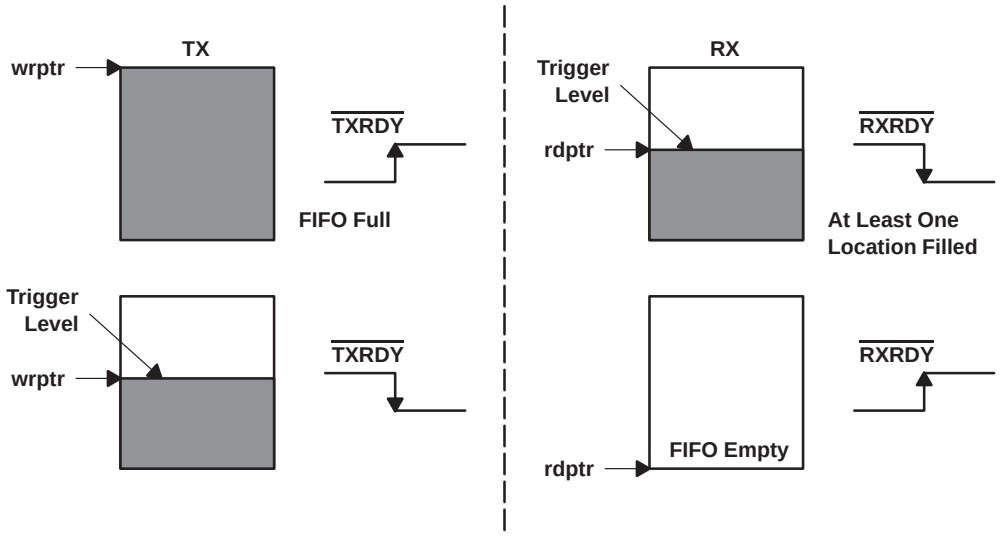



**Figure 8. $\overline{\text{TXRDY}}$ and $\overline{\text{RXRDY}}$ in DMA Mode 1**

### sleep mode

Sleep mode is an enhanced feature of the TL16C752 UART. It is enabled when EFR[4], the enhanced functions bit is set AND when IER[4] is set. Sleep mode is entered when

- The serial data input line, RX, is idle (see break and time-out conditions).
- The TX FIFO and TX shift register are empty
- There are no interrupts pending except THR and timeout interrupts.

**NOTE:**
Sleep mode will **not** be entered if there is data in the RX FIFO.

In sleep mode the UART clock and baud rate clock are stopped. Since most registers are clocked using these clocks, the power consumption is greatly reduced. The UART will wake up when any change is detected on the RX line, when there is any change in the state of the modem input pins, or if data is written to the TX FIFO.

**NOTE:**
Writing to the divisor latches, DLL and DLH, to set the baud clock, **MUST NOT** be done during sleep mode. Therefore it is advisable to disable sleep mode using IER[4] before writing to DLL or DLH.

### break and timeout conditions

An RX idle condition is detected when the receiver line, RX, has been high for a time equivalent to (4X programmed word length)+12 bits. The receiver line is sampled midway through each bit.

When a break condition occurs the TX line is pulled low. A break condition is activated by setting LCR[6].

## functional description (continued)

### programmable baud rate generator

The TL16C752 UART contains a programmable baud generator that takes any clock input and divides it by a divisor in the range between 1 and ($2^{16}$–1). Register bit MCR[7], can be used to select the 1X or 1X/4 clock to the internal baud rate generator. The output frequency of the baud rate generator is 16x the baud rate. The formula for the divisor is:

$$\text{divisor} = (\text{XTAL1 crystal input frequency/prescaler}) / (\text{desired baud rate} \times 16)$$

Where:

$$\text{prescaler} = \begin{cases} 1, & \text{when MCR[7] is set to 0 after reset (1X clock selected)} \\ 4, & \text{when MCR[7] is set to 1 after reset (1X/4 clock selected)} \end{cases}$$

Figure 9 shows the internal prescaler and baud rate generator circuitry.

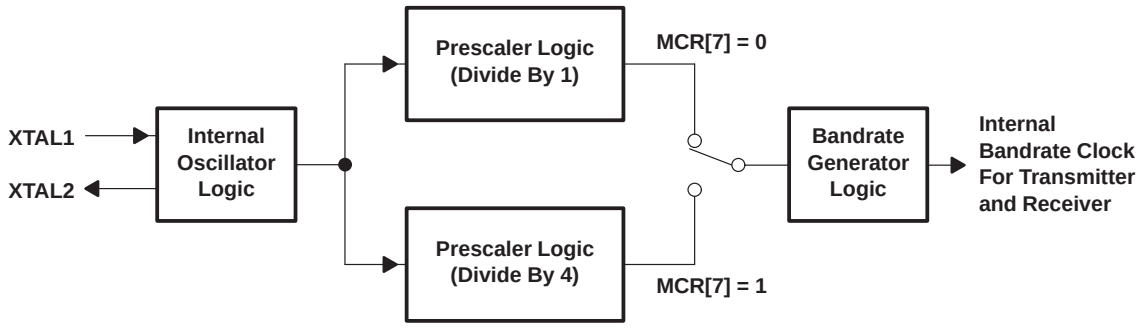



**Figure 9. Prescaler and Baud Rate Generator Block Diagram**

DLL and DLH must be written to in order to program the baud rate. DLL and DLH are the least significant and most significant byte of the baud rate divisor.

Writing to the divisor registers, DLL or DLH, may result in wait states being inserted during the write access while the baud rate generator is loaded with the new value. Note that if DLL and DLH are both zero, the UART is effectively disabled, as no baud clock will be generated.

**NOTE:**

The programmable baud rate generator is provided to select **both** the transmit and receive clock rates.

Table 5 and Table 6 show the baud rate and divisor correlation for crystal with frequency 1.8432 MHz and 3.072 MHz respectively.

Figure 10 and Figure 11 show the crystal clock circuit reference.

**programmable baud rate generator (continued)**

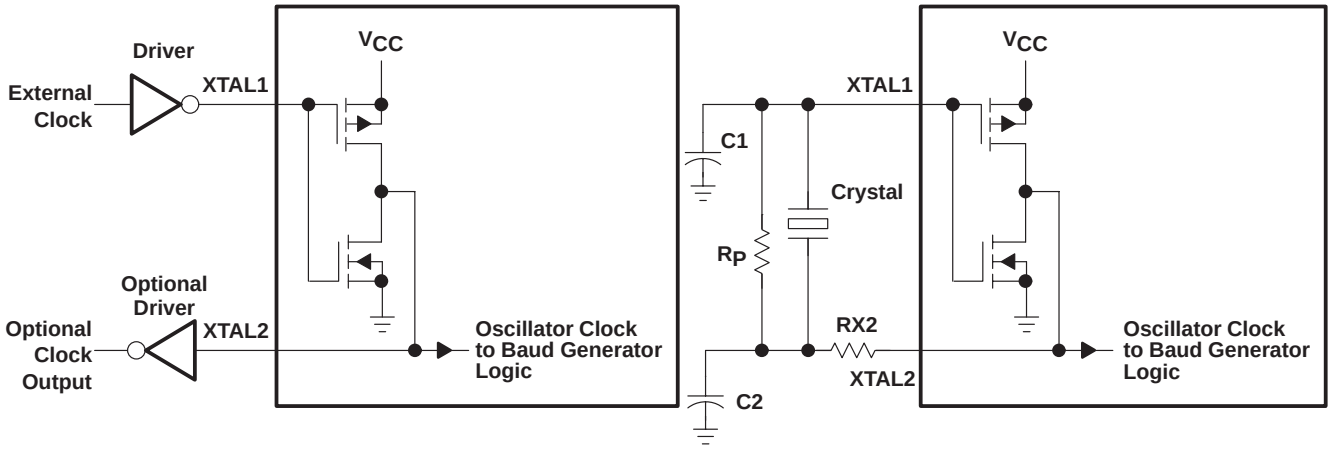
**Table 5. Baud Rates Using a 1.8432-MHz Crystal**

| DESIRED BAUD RATE | DIVISOR USED TO GENERATE 16 × CLOCK | PERCENT ERROR DIFFERENCE BETWEEN DESIRED AND ACTUAL |
|---|---|---|
| 50 | 2304 | |
| 75 | 1536 | |
| 110 | 1047 | 0.026 |
| 134.5 | 857 | 0.058 |
| 150 | 768 | |
| 300 | 384 | |
| 600 | 192 | |
| 1200 | 96 | |
| 1800 | 64 | |
| 2000 | 58 | 0.69 |
| 2400 | 48 | |
| 3600 | 32 | |
| 4800 | 24 | |
| 7200 | 16 | |
| 9600 | 12 | |
| 19200 | 6 | |
| 38400 | 3 | |
| 56000 | 2 | 2.86 |

**Table 6. Baud Rates Using a 3.072-MHz Crystal**

| DESIRED BAUD RATE | DIVISOR USED TO GENERATE 16 × CLOCK | PERCENT ERROR DIFFERENCE BETWEEN DESIRED AND ACTUAL |
|---|---|---|
| 50 | 3840 | |
| 75 | 2560 | |
| 110 | 1745 | 0.026 |
| 134.5 | 1428 | 0.034 |
| 150 | 1280 | |
| 300 | 640 | |
| 600 | 320 | |
| 1200 | 160 | |
| 1800 | 107 | 0.312 |
| 2000 | 96 | |
| 2400 | 80 | |
| 3600 | 53 | 0.628 |
| 4800 | 40 | |
| 7200 | 27 | 1.23 |
| 9600 | 20 | |
| 19200 | 10 | |
| 38400 | 5 | |

## programmable baud generator (continued)

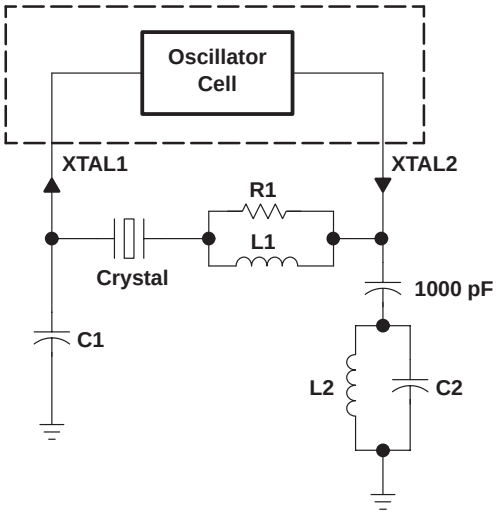



**TYPICAL CRYSTAL OSCILLATOR NETWORK**

| CRYSTAL | $R_P$ | RX2 | C1 | C2 |
|---|---|---|---|---|
| 3.072 MHz | 1 MΩ | 1.5 kΩ | 10–30 pF | 40–60 pF |
| 1.8432 MHz | 1 MΩ | 1.5 kΩ | 10–30 pF | 40–60 pF |

**Figure 10. Typical Crystal Clock Circuits†**

† For crystal with fundamental frequency from 1 MHz to 20 MHz




Recommended Trial Value for Third-Overtune Frequency Operation

| FREQUENCY (MHz) | C1 (pF) | C2 (pF) | L1 (mH) | L1 (pF) | R1 (kΩ) |
|---|---|---|---|---|---|
| 25 | 5 | 15 | 3.9 | 15 | 18 |
| 30 | 6 | 15 | 2.7 | 10 | 18 |
| 35 | 6 | 15 | 3.3 | 8.2 | 18 |

**Figure 11. Typical Third-Overtune Crystal Clock Circuits†**

† Recommend to use a series-mode crystal with series resistance no larger than 40 Ω.

### absolute maximum ratings over operating free-air temperature (unless otherwise noted)†

Supply voltage range, $V_{CC}$ . . . . . –0.5 V to 3.6 V
Input voltage range, $V_I$ . . . . . –0.5 V to $V_{CC}$ +0.5 V
Output voltage range, $V_O$ . . . . . –0.5 V to $V_{CC}$ +0.5 V
Operating free-air temperature range, $T_A$ . . . . . –40°C to 85°C
Storage temperature range, $T_{stg}$ . . . . . –65°C to 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

### recommended operating conditions

#### low voltage (3.3 V nominal)

| | | | MIN | NOM | MAX | UNIT |
|---|---|---|---|---|---|---|
| Supply voltage, $V_{CC}$ | | | 2.7 | 3.3 | 3.6 | V |
| Input voltage, $V_I$ | | | 0 | | $V_{CC}$ | V |
| High-level input voltage, $V_{IH}$ (see Note 3) | | | 0.7 $V_{CC}$ | | $V_{CC}$ | V |
| Low-level input voltage, $V_{IL}$ (see Note 3) | | | | | 0.3 $V_{CC}$ | V |
| Output voltage, $V_O$ (see Note 4) | | | 0 | | $V_{CC}$ | V |
| High-level output current, $V_{OH}$ | $I_{OH}$ = –8 mA, | See Note 6 | $V_{CC}$–0.8 | | | V |
| | $I_{OH}$ = –4 mA, | See Note 7 | $V_{CC}$–0.8 | | | |
| Low-level output current, $V_{OL}$ | $I_{OL}$ = –8 mA, | See Note 6 | | | 0.5 | V |
| | $I_{OL}$ = 4 mA, | See Note 7 | | | 0.5 | |
| Input capacitance, $C_I$ | | | | | 18 | pF |
| Operating free-air temperature, $T_A$ | | | –40 | 25 | 85 | °C |
| Virtual junction temperature range, $T_J$ (see Note 5) | | | 0 | 25 | 125 | °C |
| Oscillator/clock speed (see Note 8) | | | | | 50 | MHz |
| Clock duty cycle | | | | 50% | | |
| Supply current, $I_{CC}$ | 50 MHz, 3.6 V | | | 22 | | mA |
| | 5 MHz, 3.6 V | | | 6 | | |

NOTES: 3. Meets TTL levels, $V_{IO(min)}$ = 2 V and $V_{IH(max)}$ = 0.8 V on nonhysteresis inputs.
4. Applies for external output buffers.
5. These junction temperatures reflect simulated conditions. Absolute maximum junction temperature is 150°C. The customer is responsible for verifying junction temperature.
6. These parameters apply for $\overline{D7}$–$\overline{D0}$.
7. These parameters apply for $\overline{DTRA}$, $\overline{DTRB}$, $\overline{INIA}$, INTB, $\overline{RTSA}$, $\overline{RTSB}$, $\overline{RXRDYA}$, $\overline{RXRDYB}$, $\overline{TXRDYA}$, $\overline{TXRDYB}$, TXA, TXB.
8. The internal oscillator cell can only support up to 35 MHz clock frequency to make the crystal oscillating when crystal is used. If external oscillator or other on board clock source is used, the TL16C752 can work for input clock frequency up to 50 MHz.

## timing requirements $T_A$ = –40°C to 85°C, $V_{CC}$ = 3.3 V ± 10% (unless otherwise noted) (see Figures 12–19)

| PARAMETER | | TEST CONDITIONS | LIMITS | | UNIT |
|---|---|---|---|---|---|
| | | | MIN | MAX | |
| $t_{d1}$ | $\overline{IOR}$ delay from chip select | | 10 | | ns |
| $t_{d2}$ | Read cycle delay | | $2t_{p(I)}$‡ | | ns |
| $t_{d3}$ | Delay from $\overline{IOR}$ to data | | 28.5 | | ns |
| $t_{d4}$ | Data disable time | | | 15 | ns |
| $t_{d5}$ | $\overline{IOW}$ delay from chip select | | 10 | | ns |
| $t_{d6}$ | Write cycle delay | | $2t_{p(I)}$‡ | | ns |
| $t_{d7}$ | Delay from $\overline{IOW}$ to output | 100 pF load | | 50 | ns |
| $t_{d8}$ | Delay to set interrupt from MODEM input | 100 pF load | | 70 | ns |
| $t_{d9}$ | Delay to reset interrupt from $\overline{IOR}$ | 100 pF load | | 70 | ns |
| $t_{d10}$ | Delay from stop to set interrupt | | | $1_{Rclk}$ | † |
| $t_{d11}$ | Delay from $\overline{IOR}$ to reset interrupt | 100 pF load | | 70 | ns |
| $t_{d12}$ | Delay from stop to interrupt | | | 100 | ns |
| $t_{d13}$ | Delay from initial INT reset to transmit start | | 8 | 24 | † |
| $t_{d14}$ | Delay from $\overline{IOW}$ to reset interrupt | | | 70 | ns |
| $t_{d15}$ | Delay from stop to set $\overline{RXRDY}$ | | | 1 | Clock |
| $t_{d16}$ | Delay from $\overline{IOR}$ to reset $\overline{RXRDY}$ | | | 1 | µs |
| $t_{d17}$ | Delay from $\overline{IOW}$ to set $\overline{TXRDY}$ | | | 70 | ns |
| $t_{d18}$ | Delay from start to reset $\overline{TXRDY}$ | | | 16 | † |
| $t_{h1}$ | Chip select hold time from $\overline{IOR}$ | | 0 | | ns |
| $t_{h2}$ | Chip select hold time from $\overline{IOW}$ | | 0 | | ns |
| $t_{h3}$ | Data hold time | | 15 | | ns |
| $t_{p1}$, $t_{p2}$ | Clock pulse duration | | 10 | | ns |
| $t_{p3}$ | Oscillator/Clock speed | $V_{CC}$ = 3 V | | 50 | MHz |
| $t_{su1}$ | Address setup time | | 10 | | ns |
| $t_{su2}$ | Data setup time | | 16 | | ns |
| $t_{su3}$ | Address setup time | | 10 | | ns |
| $t_{w1}$ | $\overline{IOR}$ strobe width | | $2t_{p(I)}$‡ | | ns |
| $t_{w2}$ | $\overline{IOW}$ strobe width§ | | $2t_{p(I)}$‡ | | ns |

† Baudrate
‡ $t_{p(I)}$ = input clock period
§ The $\overline{IOW}$ strobe width for the DLL and DLH accesses must be two clock periods.

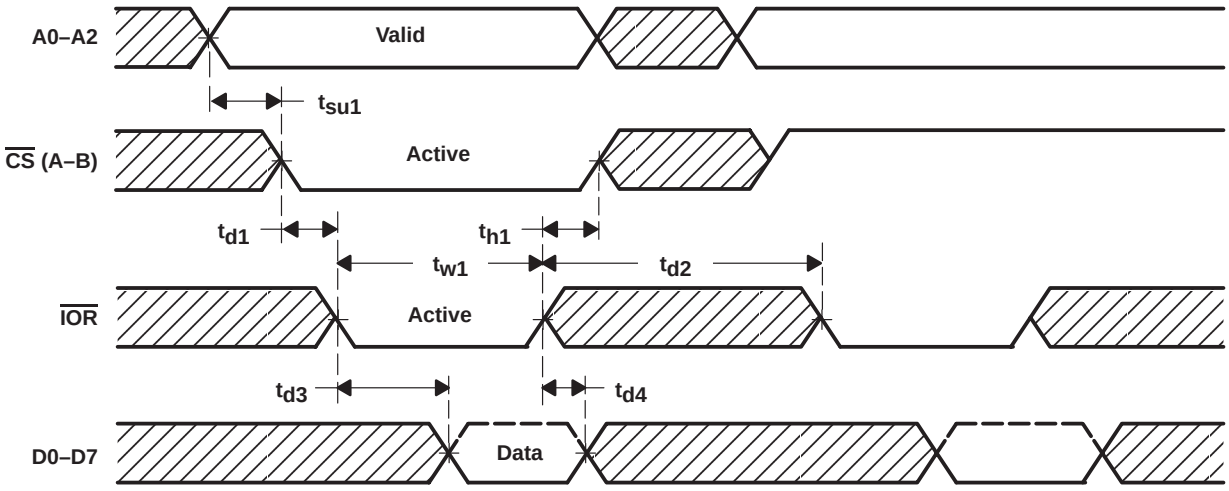



**Figure 12. General Read Timing**

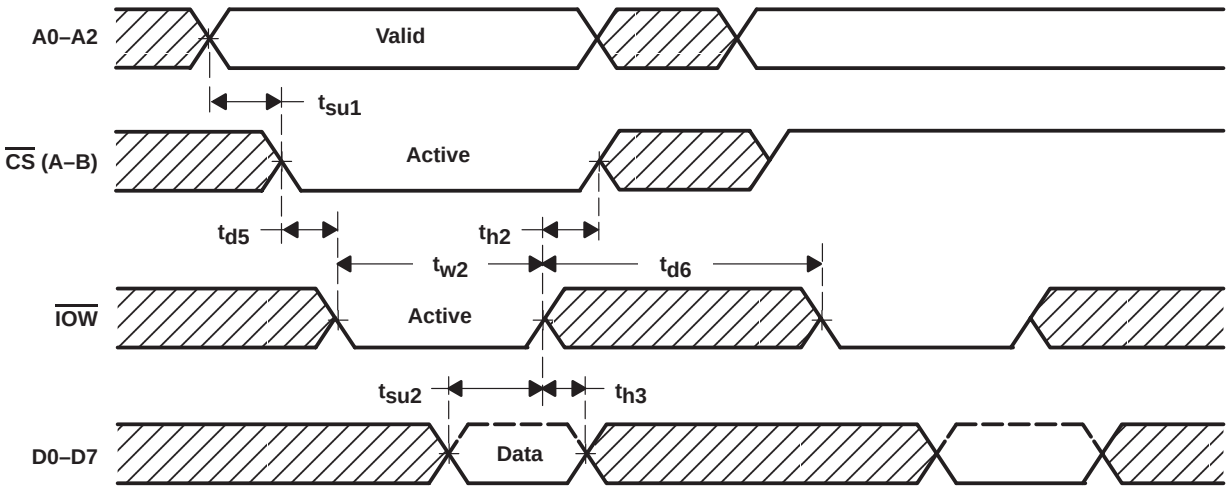



**Figure 13. General Write Timing**

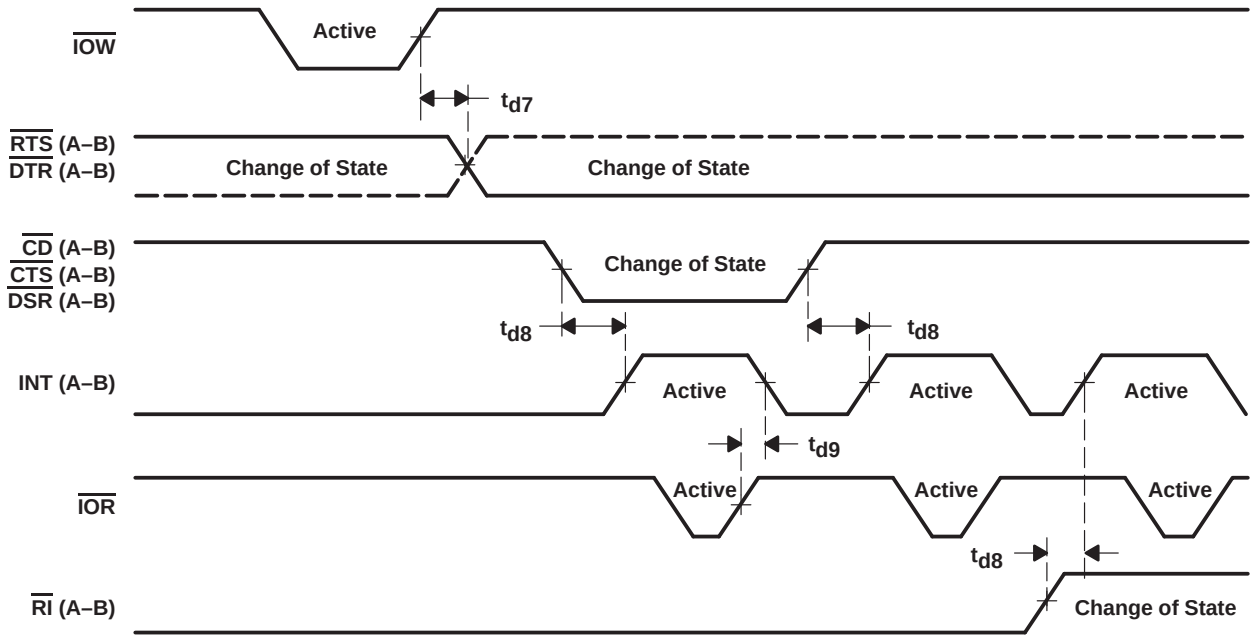



**Figure 14. Modem Input/Output Timing**

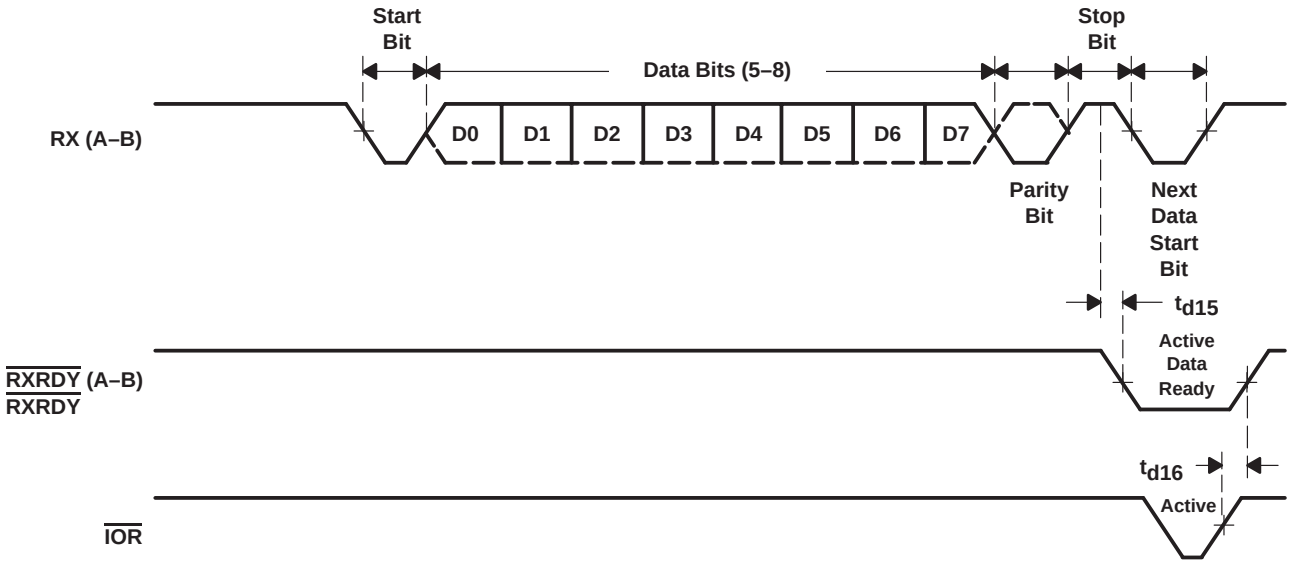



**Figure 15. Receive Ready Timing in None FIFO Mode**

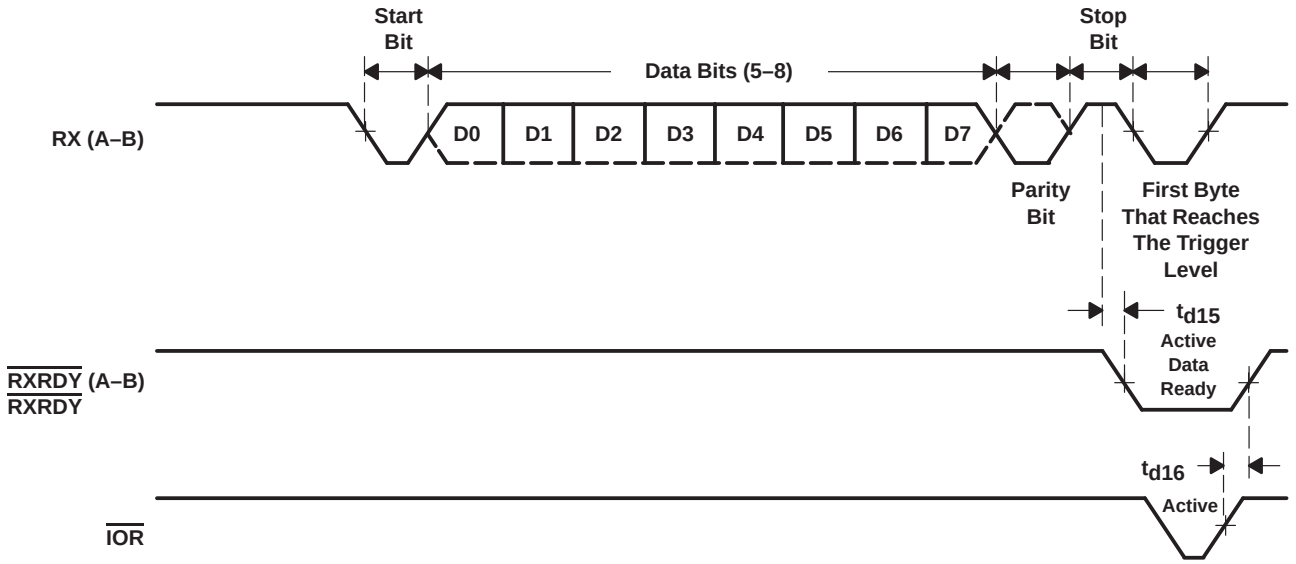



**Figure 16. Receive Timing in FIFO Mode**

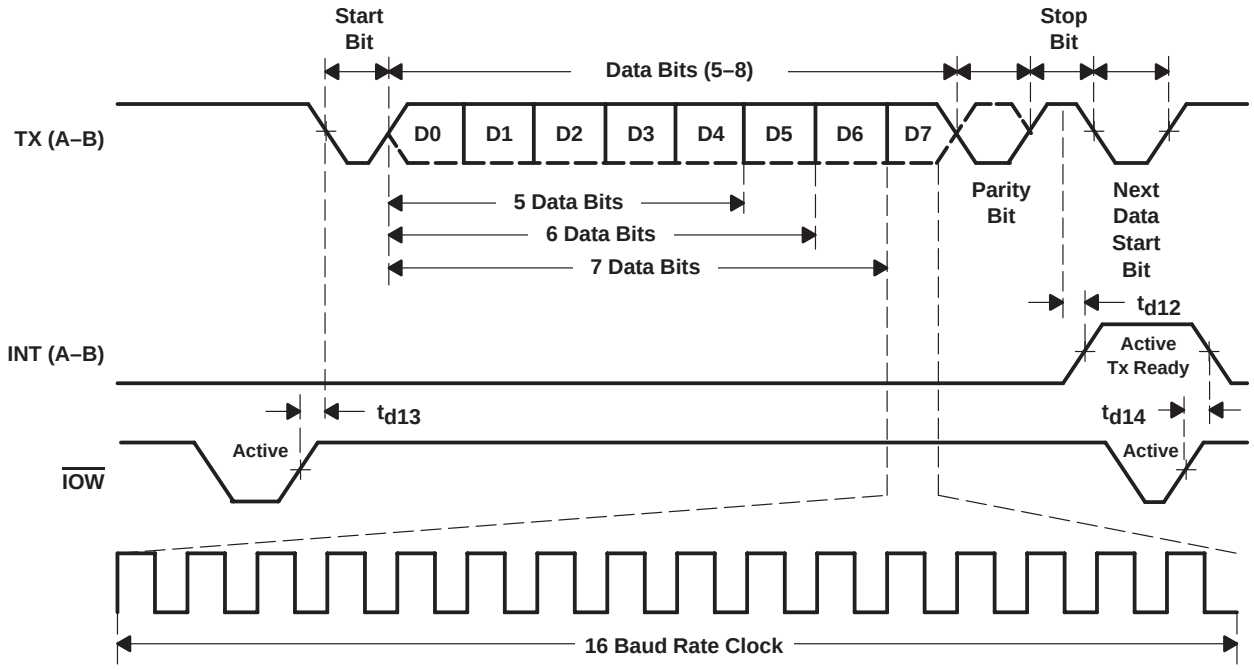



**Figure 17. Transmit Timing**

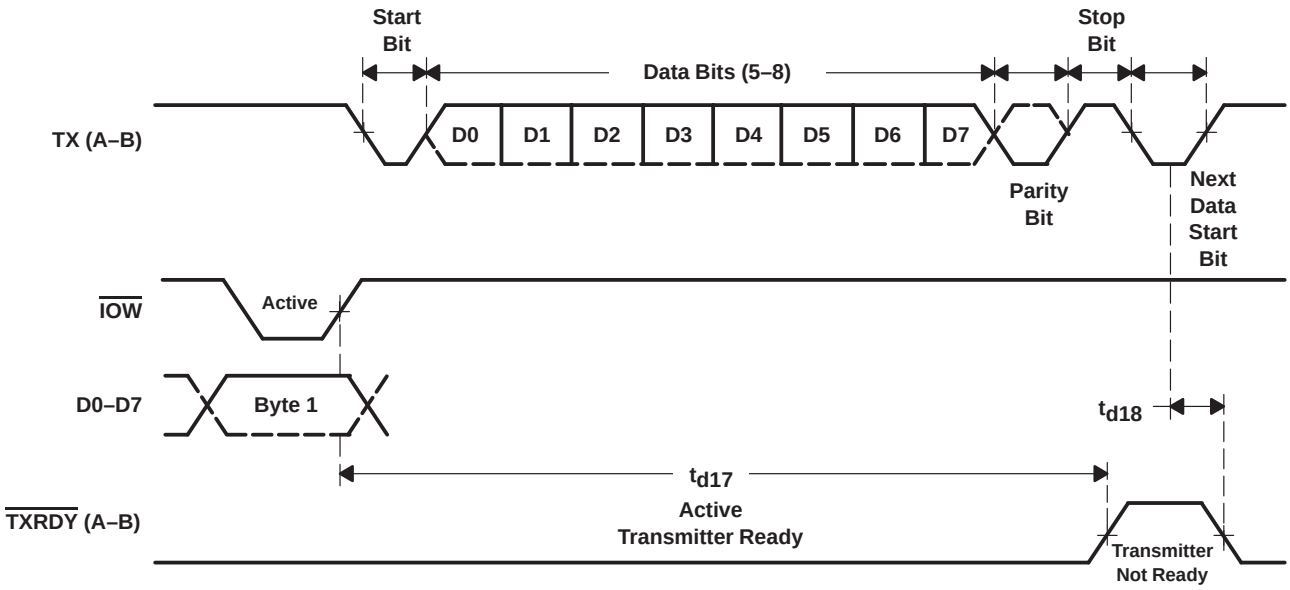



**Figure 18. Transmit Ready Timing in None FIFO Mode**

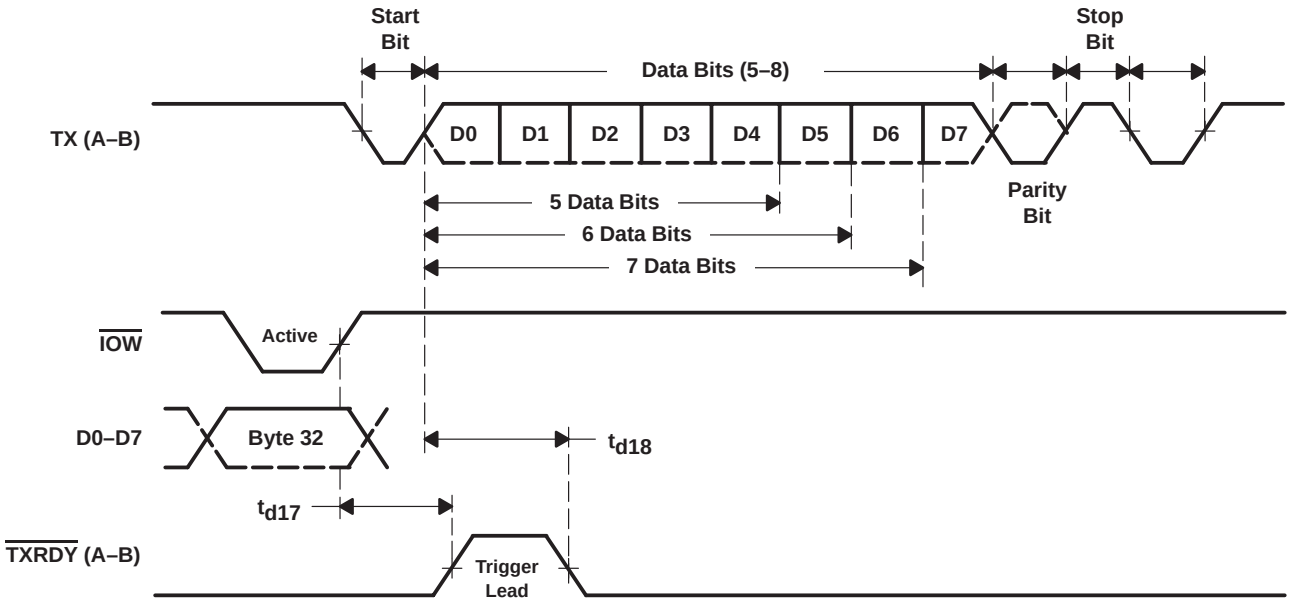



**Figure 19. Transmit Ready Timing in FIFO Mode**

## PRINCIPLES OF OPERATION

### register map†

Each register is selected using it's own combination of address lines A[0], A[1], A[2]. The programming combinations for register selection are shown in Table 7.

**Table 7. Register Map – Read/Write Properties**

| A[2] | A[1] | A[0] | READ MODE | WRITE MODE |
|---|---|---|---|---|
| 0 | 0 | 0 | Receive holding register (RHR) | Transmit holding register (THR) |
| 0 | 0 | 1 | Interrupt enable register (IER) | Interrupt enable register |
| 0 | 1 | 0 | Interrupt identification register (IIR) | FIFO control register (FCR) |
| 0 | 1 | 1 | Line control register (LCR) | Line control register |
| 1 | 0 | 0 | Modem control register (MCR) | Modem control register |
| 1 | 0 | 1 | Line status register (LSR) | |
| 1 | 1 | 0 | Modem status register (MSR) | |
| 1 | 1 | 1 | Scratch register (SPR) | Scratch register (SPR) |
| | | | | |
| 0 | 0 | 0 | Divisor latch LSB (DLL) | Divisor latch LSB (DLL) |
| 0 | 0 | 1 | Divisor latch MSB (DLH) | Divisor latch MSB (DLH |
| 0 | 1 | 0 | Enhanced feature register (EFR) | Enhanced feature register |
| 1 | 0 | 0 | Xon-1 word | Xon-1 word |
| 1 | 0 | 1 | Xon-2 word | Xon-2 word |
| 1 | 1 | 0 | Xoff-1 word | Xoff-1 word |
| 1 | 1 | 1 | Xoff-2 word | Xoff-2 word |
| 1 | 1 | 0 | Transmission control register (TCR) | Transmission control register |
| 1 | 1 | 1 | Trigger level register (TLR) | Trigger level register |
| | | | | |
| 1 | 1 | 1 | FIFO ready register | |

† DLL and DLH are accessible only when LCR bit-7, is 1.
Enhanced feature register, Xon1, 2 and Xoff1, 2 are accessible only when LCR is set to 10111111 (8hBF).
Transmission control register and trigger level register are accessible only when EFR[4] = 1 and MCR[6] = 1, i.e.. EFR[4] and MCR[6] are read/write enables.
FIFORdy register is accessible only when CSA and CSB = 0, MCR [bit 2] = 1 and loopback is disabled.

## PRINCIPLES OF OPERATION

Table 8 lists and describes the TL16C752 internal registers.

**Table 8. TL16C752A Internal Registers**

| Addr | REGISTER | BIT 7 | BIT 6 | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 | READ/ WRITE |
|---|---|---|---|---|---|---|---|---|---|---|
| 000 | RHR | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | Read |
| 000 | THR | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | Write |
| 001 | IER | 0/$\overline{\text{CTS}}$ interrupt enable† | 0/$\overline{\text{RTS}}$ interrupt enable† | 0/$\overline{\text{Xoff}}$ sleep mode† | 0/X Sleep mode† | Modem status interrupt | Rx line status interrupt | THR empty interrupt | Rx data available interrupt | Read/Write |
| 010 | FCR | Rx trigger level | Rx trigger level | 0/TX trigger level† | 0/TX trigger level† | DMA mode select | Resets Tx FIFO | Resets Rx FIFO | Enables FIFOs | Write |
| 010 | IIR | FCR(0) | FCR(0) | 0/$\overline{\text{CTS}}$, $\overline{\text{RTS}}$† | 0/Xoff† | Interrupt priority Bit 2 | Interrupt priority Bit 1 | Interrupt priority Bit 0 | Interrupt status | Read |
| 011 | LCR | DLAB and EFR enable | Break control bit | Sets parity | Parity type select | Parity enable | No. of stop bits | Word length | Word length | Read/Write |
| 100 | MCR | 1x or 1x/4 clock | TCR and TLR enable | 0/Xon Any | 0/Enable loopback | IRQ enable | $\overline{\text{FIFO Rdy}}$ enable | $\overline{\text{RTS}}$ | $\overline{\text{DTR}}$ | Read/Write |
| 101 | LSR | 0/Error in Rx FIFO | THR and TSR empty | THR empty | Break interrupt | Framing error | Parity error | Over-run error | Data in receiver | Read |
| 110 | MSR | $\overline{\text{CD}}$ | $\overline{\text{RI}}$ | $\overline{\text{DSR}}$ | $\overline{\text{CTS}}$ | $\Delta\overline{\text{CD}}$ | $\Delta\overline{\text{RI}}$ | $\Delta\overline{\text{DSR}}$ | $\Delta\overline{\text{CTS}}$ | Read |
| 111 | SPR | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | Read/Write |
| 000 | DLL | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | Read/Write |
| 001 | DLH | bit 15 | bit 14 | bit 13 | bit 12 | bit 11 | bit 10 | bit 9 | bit 8 | Read/Write |
| 010 | EFR | Auto-$\overline{\text{CTS}}$ | Auto-$\overline{\text{RTS}}$ | Special character detect | Enable enhanced functions† | S/W flow control Bit 3 | S/W flow control Bit 2 | S/W flow control Bit 1 | S/W flow control Bit 0 | Read/Write |
| 100 | Xon1 | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | Read/Write |
| 101 | Xon2 | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | Read/Write |
| 110 | Xoff1 | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | Read/Write |
| 111 | Xoff2 | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | Read/Write |
| 110 | TCR | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | Read/Write |
| 111 | TLR | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | Read/Write |
| 111 | FIFO Rdy | 0 | 0 | RX FIFO B status | RX FIFO A status | 0 | 0 | TX FIFO B status | TX FIFO A status | Read |

† The shaded bits in the above table can only be modified if register bit EFR[4] is enabled, i.e., if enhanced functions are enabled.

## PRINCIPLES OF OPERATION

### *receiver holding register (RHR)*

The receiver section consists of the receiver holding register (RHR) and the receiver shift register (RSR). The RHR is actually a 64-byte FIFO. The RSR receives serial data from RX terminal. The data is converted to parallel data and moved to the RHR. The receiver section is controlled by the line control register. If the FIFO is disabled location zero of the FIFO is used to store the characters. (NOTE: In this case characters are overwritten if overflow occurs.)

### *transmit holding register (THR)*

The transmitter section consists of the transmit holding register (THR) and the transmit shift register (TSR). The transmit holding register is actually a 64-byte FIFO. The THR receives data and shifts it into the TSR where it is converted to serial data and moved out on the TX terminal. If the FIFO is disabled the FIFO is still used to store the byte. (NOTE: In this case characters are overwritten if overflow occurs.)

### *FIFO control register (FCR)*

This is a write-only register which is used for enabling the FIFOs, clearing the FIFOs, setting transmitter and receiver trigger levels and selecting the type of DMA signalling. Table 9 shows FIFO control register bit settings.

**Table 9. FIFO Control Register (FCR) Bit Settings**

| BIT NO. | BIT SETTINGS |
|---|---|
| 0 | 0 = Disable the transmit and receive FIFOs<br>1 = Enable the transmit and receive FIFOs |
| 1 | 0 = No change<br>1 = Clears the receive FIFO and resets it's counter logic to zero. Will return to zero after clearing FIFO. |
| 2 | 0 = No change<br>1 = Clears the transmit FIFO and resets it's counter logic to zero. Will return to zero after clearing FIFO. |
| 3 | 0 = DMA Mode 0<br>1 = DMA MOde 1 |
| 5:4 | Sets the trigger level for the TX FIFO:<br>00 – 8 spaces<br>01 – 16 spaces<br>10 – 32 spaces<br>11 – 56 spaces |
| 7:6 | Sets the trigger level for the RX FIFO:<br>00 – 8 characters<br>01 – 16 characters<br>10 – 56 characters<br>11 – 60 characters |

NOTE: FCR[5–4] can only be modified and enabled when EFR[4] is set. This is because the transmit trigger level is regarded as an enhanced function.

## PRINCIPLES OF OPERATION

### *line control register (LCR)*

This register controls the data communication format. The word length, number of stop bits, and parity type are selected by writing the appropriate bits to the LCR. Table 10 shows line control register bit settings.

**Table 10. Line Control Register (LCR) Bit Settings**

| BIT NO. | BIT SETTINGS |
|---|---|
| 1:0 | Specifies the word length to be transmitted or received.<br>00 – 5 bits<br>01 – 6 bits<br>10 – 7 bits<br>11 – 8 bits |
| 2 | Specifies the number of stop bits:<br>0 – 1 stop bits (Word length = 5, 6, 7, 8)<br>1 – 1.5 stop bits (Word length = 5)<br>1 – 2 stop bits (Word length = 6, 7, 8) |
| 3 | 0 = No parity<br>1 = A parity bit is generated during transmission and the receiver checks for received parity. |
| 4 | 0 = Odd parity is generated (if LCR(3) = 1)<br>1 = Even parity is generated (if LCR(3) = 1) |
| 5 | Selects the forced parity format (if LCR(3) = 1)<br>If LCR(5) = 1 and LCR(4) = 0 = the parity bit is forced to 1 in the transmitted and received data.<br>If LCR(5) = 1 and LCR(4) = 1 = the parity bit is forced to 0 in the transmitted and received data. |
| 6 | Break control bit.<br>0 = Normal operating condition<br>1 = Forces the transmitter output to go low to alert the communication terminal. |
| 7 | 0 = Normal operating condition<br>1 = Divisor latch enable |

## PRINCIPLES OF OPERATION

### *line status register (LSR)*

Table 11 shows line status register bit settings.

**Table 11. Line Status Register (LSR) Bit Settings**

| BIT NO. | BIT SETTINGS |
|---|---|
| 0 | 0 = No data in the receive FIFO<br>1 = At least one character in the RX FIFO |
| 1 | 0 = No overrun error<br>1 = Overrun error has occurred. |
| 2 | 0 = No parity error in data being read from RX FIFO<br>1 = Parity error in data being read from RX FIFO. |
| 3 | 0 = No framing error in data being read from RX FIFO<br>1 = Framing error occurred in data being read from RX FIFO (i.e., received data did not have a valid stop bit) |
| 4 | 0 = No break condition<br>1 = A break was detected while the data being read from the RX FIFO was being received. (i.e., RX i/p was low for one character time frame). |
| 5 | 0 = Transmit hold register is NOT empty<br>1 = Transmit hold register is empty. The processor can now load up to 64 bytes of data into the THR if the TX FIFO is enabled. |
| 6 | 0 = Transmitter hold AND shift registers are not empty.<br>1 = Transmitter hold AND shift registers are empty. |
| 7 | 0 = Normal operation<br>1 = At least one parity error, framing error or break indication in the receiver FIFO. BIt 7 is cleared when no more errors are present in the FIFO. |

When the LSR is read, LSR[4:2] reflect the error bits [BI, FE, PE] of the character at the top of the RX FIFO (next character to be read). The LSR[4:2] registers don't physically exist as the data read from the RX FIFO is output directly onto the output data-bus, DI[4:2], when the LSR is read. Therefore errors in a character are identified by reading the LSR and then reading the RHR.

LSR[7] is set when there is an error anywhere in the RX FIFO and is cleared only when there are no more errors remaining in the FIFO.

**NOTE:**

Reading the LSR does not cause an increment of the RX FIFO read pointer. The RX FIFO read pointer is incremented by reading the RHR. Reading the RX FIFO when it is empty could cause LSR[7] to be set, which can only be reset by clearing the RX FIFO via FCR[1]. It is strongly recommended that the RX FIFO be read only when data is present.

## PRINCIPLES OF OPERATION

### *modem control register (MCR)*

The MCR controls the interface with the modem, data set or peripheral device that is emulating the modem. Table 12 shows modem control register bit settings.

**Table 12. Modem Control Register (MCR) Bit Settings**

| BIT NO. | BIT SETTINGS |
|---|---|
| 0 | 0 = Force $\overline{\text{DTR}}$ output to inactive (high)<br>1 = Force $\overline{\text{DTR}}$ output to active (low).<br>In loopback controls MSR[5]. |
| 1 | 0 = Force $\overline{\text{RTS}}$ output to inactive (high)<br>1 = Force $\overline{\text{RTS}}$ output to active (low).<br>In loopback controls MSR[4].<br>If Auto-$\overline{\text{RTS}}$ is enabled the $\overline{\text{RTS}}$ output is controlled by hardware flow control |
| 2 | 0 Disables the FIFO Rdy register<br>1 Enable the FIFO Rdy register.<br>In loopback controls MSR[6]. |
| 3 | 0 = Forces the IRQ(A-D) outputs to 3-state<br>1 = Forces the IRQ(A-D) outputs to the active state.<br>In loopback controls MSR[7]. |
| 4 | 0 = Normal operating mode<br>1 = Enable local loopback mode (internal)<br>In this mode the MCR[3:0] signals are looped back into MSR[3:0] and the TX output is looped back to the RX input internally. |
| 5 | 0 = Disable Xon Any function<br>1 = Enable Xon Any function |
| 6 | 0 = No action<br>1 = Enable access to the TCR and TLR registers. |
| 7 | 0 = Normal or divide by one clock input (CLKSEL)<br>1 = Divide by four clock input |

NOTE: MCR[7:5] can only be modified when EFR[4] is set i.e., EFR[4] is a write enable.

### *modem status register (MSR)*

This 8-bit register provides information about the current state of the control lines from the modem, data set or peripheral device to the processor. It also indicates when a control input from the modem changes state. Table 13 shows modem status register bit settings.

**Table 13. Modem Status Register (MSR) Bit Settings**

| BIT NO. | BIT SETTINGS |
|---|---|
| 0 | Indicates that $\overline{\text{CTS}}$ input (or MCR[1] in loopback) has changed state. Cleared on a read. |
| 1 | Indicates that $\overline{\text{DSR}}$ input (or MCR[0] in loopback) has changed state. Cleared on a read. |
| 2 | Indicates that $\overline{\text{RI}}$ input (or MCR[2] in loopback) has changed state from low to high. Cleared on a read. |
| 3 | Indicates that $\overline{\text{CD}}$ input (or MCR[3] in loopback) has changed state. Cleared on a read. |
| 4 | This bit is equivalent to MCR[1] during local loop-back mode. It is the compliment to the $\overline{\text{CTS}}$ input. |
| 5 | This bit is equivalent to MCR[0] during local loop-back mode. It is the compliment to the $\overline{\text{DSR}}$ input. |
| 6 | This bit is equivalent to MCR[2] during local loop-back mode. It is the compliment to the $\overline{\text{RI}}$ input. |
| 7 | This bit is equivalent to MCR[3] during local loop-back mode. It is the compliment to the $\overline{\text{CD}}$ input. |

NOTE: The primary inputs $\overline{\text{RI}}$, $\overline{\text{CD}}$, $\overline{\text{CTS}}$, $\overline{\text{DSR}}$ are all active low but their registered equivalents in the MSR and MCR (in loopback) registers are active high.

## PRINCIPLES OF OPERATION

### *interrupt enable register (IER)*

The interrupt enable register (IER) enables each of the six types of interrupt, receiver error, RHR interrupt, THR interrupt, Xoff received or CTS/RTS change of state from low to high. The INT output signal is activated in response to interrupt generation. Table 14 shows interrupt enable register bit settings.

**Table 14. Interrupt Enable Register (IER) Bit Settings**

| BIT NO. | BIT SETTINGS |
|---|---|
| 0 | 0 = Disable the RHR interrupt<br>1 = Enable the RHR interrupt |
| 1 | 0 = Disable the THR interrupt<br>1 = Enable the THR interrupt |
| 2 | 0 = Disable the receiver line status interrupt<br>1 = Enable the receiver line status interrupt |
| 3 | 0 = Disable the modem status register interrupt<br>1 = Enable the modem status register interrupt |
| 4 | 0 = Disable sleep mode<br>1 = Enable sleep mode |
| 5 | 0 = Disable the Xoff interrupt<br>1 = Enable the Xoff interrupt. |
| 6 | 0 = Disable the $\overline{\text{RTS}}$ interrupt<br>1 = Enable the $\overline{\text{RTS}}$ interrupt |
| 7 | 0 = Disable the $\overline{\text{CTS}}$ interrupt<br>1 = Enable the $\overline{\text{CTS}}$ interrupt |

NOTE: IER[7:4] can only be modified if EFR[4] is set, i.e., EFR[4] is a write enable. Re-enabling IER[1] will not cause a new interrupt until new data has been written to the TX FIFO.

### *interrupt identification register (IIR)*

The IIR is a read-only 8-bit register which provides the source of the interrupt in a prioritized manner. Table 15 shows interrupt identification register bit settings.

**Table 15. Interrupt Identification Register (IIR) Bit Settings**

| BIT NO. | BIT SETTINGS |
|---|---|
| 0 | 0 = A interrupt is pending<br>1 = No interrupt is pending |
| 3:1 | 3-Bit encoded interrupt. See Table 14. |
| 4 | 1 = Xoff/Special character has been detected. |
| 5 | $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ low to high change of state. |
| 7:6 | Mirror the contents of FCR[0] |

## PRINCIPLES OF OPERATION

### *interrupt identification register (IIR) (continued)*

The interrupt priority list is illustrated in Table 16.

**Table 16. Interrupt Priority List**

| PRIORITY LEVEL | BIT 5 | BIT 4 | BIT 3 | BIT 2 | BIT 1 | BIT 0 | INTERRUPT SOURCE |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 | 1 | 0 | Receiver line status error. |
| 2 | 0 | 0 | 1 | 1 | 0 | 0 | Receiver timeout interrupt |
| 2 | 0 | 0 | 0 | 1 | 0 | 0 | RHR interrupt |
| 3 | 0 | 0 | 0 | 0 | 1 | 0 | THR interrupt |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | Modem interrupt |
| 5 | 0 | 1 | 0 | 0 | 0 | 0 | Received Xoff signal/special character |
| 6 | 1 | 0 | 0 | 0 | 0 | 0 | $\overline{\text{CTS}}$, $\overline{\text{RTS}}$ change of state from active (low) to inactive (high). |

### *enhanced feature register (EFR)*

This 8-bit register enables or disables the enhanced features of the UART. Table 17 shows the enhanced feature register bit settings.

**Table 17. Enhanced Feature Register (EFR) Bit Settings**

| BIT NO. | BIT SETTINGS |
|---|---|
| 3:0 | Combinations of software flow control can be selected by programming bit 3–bit 0. See Table 1. |
| 4 | Enhanced functions enable bit.<br>0 = Disables enhanced functions and writing to IER bits 4–7, FCR bits 4–5, MCR bits 5–7.<br>1 = Enables the enhanced function IER bits 4–7, FCR bit 4–5, and MCR bits 5–7 can be modified, i.e., this bit is therefore a write enable. |
| 5 | 0 = Normal operation<br>1 = Special character detect. Received data is compared with Xoff-2 data. If a match occurs the received data is transferred to FIFO and IIR bit 4 is set to 1 to indicate a special character has been detected. |
| 6 | $\overline{\text{RTS}}$ flow control enable bit<br>0 = Normal operation<br>1 = $\overline{\text{RTS}}$ flow control is enabled i.e., $\overline{\text{RTS}}$ pin goes high when the receiver FIFO HALT trigger level TCR[3:0] is reached, and goes low when the receiver FIFO RESTORE transmission trigger level TCR[7:4] is reached. |
| 7 | $\overline{\text{CTS}}$ flow control enable bit<br>0 = Normal operation<br>1 = $\overline{\text{CTS}}$ flow control is enabled i.e., transmission is halted when a high signal is detected on the $\overline{\text{CTS}}$ pin. |

### *divisor latches (DLL, DLH)*

These are two 8-bit registers which store the 16-bit divisor for generation of the baud clock in the baud rate generator. DLH, stores the most significant part of the divisor. DLL stores the least significant part of the division.

Note that DLL and DLH can only be written to before sleep mode is enabled (i.e., before IER[4] is set).

### *transmission control register (TCR)*

This 8-bit register is used to store the receive FIFO threshold levels to start/stop transmission during hardware/software flow control. Table 18 shows transmission control register bit settings.

## PRINCIPLES OF OPERATION

**Table 18. Transmission Control Register (TCR) Bit Settings**

| BIT NO. | BIT SETTINGS |
|---|---|
| 3:0 | RCV FIFO trigger level to HALT transmission (0–60) |
| 7:4 | RCV FIFO trigger level to RESTORE transmission (0–60) |

TCR trigger levels are available from 0–60 bytes with a granularity of four.

**NOTE:**

TCR can only be written to when EFR[4] = 1 and MCR[6] = 1. The programmer must program the TCR such that TCR[3:0] > TCR[7:4]. There is no built-in hardware check to make sure this condition is met. Also, the TCR must be programmed with this condition before Auto-$\overline{\text{RTS}}$ or software flow control is enabled to avoid spurious operation of the device.

### *trigger level register (TLR)*

This 8-bit register is pulsed to store the transmit and received FIFO trigger levels used for DMA and interrupt generation. Trigger levels from 4–60 can be programmed with a granularity of 4. Table 19 shows trigger level register bit settings.

**Table 19. Trigger Level Register (TLR) Bit Settings**

| BIT NO. | BIT SETTINGS |
|---|---|
| 3:0 | Transmit FIFO trigger levels (4–60) |
| 7:4 | RCV FIFO trigger levels (4–60) |

**NOTE:**

TLR can only be written to when EFR[4] = 1 and MCR[6] = 1. If TLR[3:0] or TLR[7:4] = 1, the selectable trigger levels using the FIFO control register (FCR) are used for the transmit and receive FIFO trigger levels. Trigger levels from 4–60 bytes are available with a granularity of four. The TLR should be programmed for N/4, where N is the desired trigger level.

### *FIFO ready register*

The FIFO ready register provides real-time status of the transmit and receive FIFOs. Table 20 shows the FIFO ready register bit settings.

**Table 20. FIFO Ready Register**

| BIT NO. | BIT SETTINGS |
|---|---|
| 3:0 | 0 = There are less than a TX trigger level number of spaces available in the TX FIFO.<br>1 = There are at least a TX trigger level number of spaces available in the TX FIFO |
| 7:4 | 0 = There are less than a RX trigger level number of characters in the RX FIFO.<br>1 = The RX FIFO has more than a RX trigger level number of characters available for reading OR a timeout condition has occurred. |

The FIFORdy register is a read only register and can be accessed when any of the two UARTs are selected CAS A-B = 0, MCR[2] (FIFO Rdy Enable) is a logic 1 and loopback is disabled. The address space is 111.

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 110 | TCR | bit 7 | bit 6 | bit 5 | bit 4 | bit 3 | bit 2 | bit 1 | bit 0 | Read/Write |

## PRINCIPLES OF OPERATION

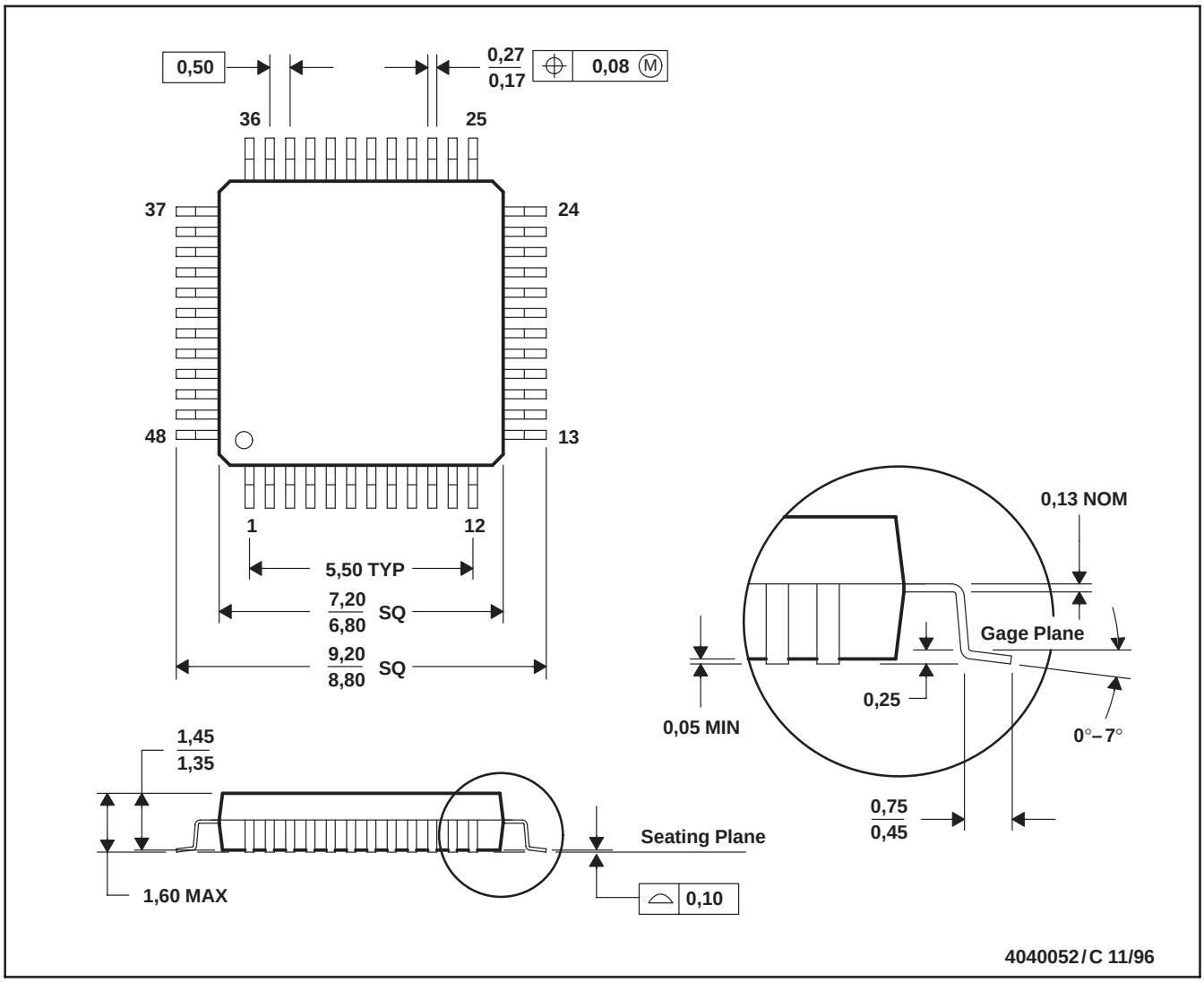
### TL16C752 programmer's guide

The base set of registers that is used during high speed data transfer have a straightforward access method. The extended function registers require special access bits to be decoded along with the address lines. The following guide will help with programming these registers. Note that the descriptions below are for individual register access. Some streamlining through interleaving can be obtained when programming all the registers.

| | |
|---|---|
| Set baud rate to VALUE1, VALUE2 | Read LCR (03), save in temp<br>Set LCR (03) to 80<br>Set DLL (00) to VALUE1<br>Set DLM (01) to VALUE2<br>Set LCR (03) to temp |
| Set Xoff1, Xon1 to VALUE1, VALUE2 | Read LCR (03), save in temp<br>Set LCR (03) to BF<br>Set Xoff1 (06) to VALUE1<br>Set Xon1 (04) to VALUE2<br>Set LCR (03) to temp |
| Set Xoff2, Xon2 to VALUE1, VALUE2 | Read LCR (03), save in temp<br>Set LCR (03) to BF<br>Set Xoff2 (07) to VALUE1<br>Set Xon2 (05) to VALUE2<br>Set LCR (03) to temp |
| Set software flow control mode to VALUE | Read LCR (03), save in temp<br>Set LCR (03) to BF<br>Set EFR (02) to VALUE<br>Set LCR (03) to temp |
| Set flow control threshold to VALUE | Read LCR (03), save in temp1<br>Set LCR (03) to BF<br>Read EFR (02), save in temp2<br>Set EFR (02) to 10 + temp2<br>Set LCR (03) to 00<br>Read MCR (04), save in temp3<br>Set MCR (04) to 40 + temp3<br>Set TCR (06) to VALUE<br>Set LCR (03) to BF<br>Set EFR (02) to temp2<br>Set LCR (03) to temp1<br>Set MCR (04) to temp3 |
| Set xmt and rcv FIFO thresholds to VALUE | Read LCR (03), save in temp1<br>Set LCR (03) to BF<br>Read EFR (02), save in temp2<br>Set EFR (02) to 10 + temp2<br>Set LCR (03) to 00<br>Read MCR (04), save in temp3<br>Set MCR (04) to 40 + temp3<br>Set TLR (07) to VALUE<br>Set LCR (03) to BF<br>Set EFR (02) to temp2<br>Set LCR (03) to temp1<br>Set MCR (04) to temp2 |
| Read FIFORdy register | Read MCR (04), save in temp1<br>Set temp2 = temp1 × EF<br>Set MCR (04) = 04 + temp2<br>Read FRR (07), save in temp2<br>Pass temp2 back to host<br>Set MCR (04) to temp1 |

## MECHANICAL DATA

**PT (S-PQFP-G48)** **PLASTIC QUAD FLATPACK**




NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-026
D. This may also be a thermally enhanced plastic package with leads conected to the die pads.

## PACKAGING INFORMATION

| Orderable Device | Status (1) | Package Type | Package Drawing | Pins | Package Qty | Eco Plan (2) | Lead/Ball Finish | MSL Peak Temp (3) |
|---|---|---|---|---|---|---|---|---|
| TL16C752PT | OBSOLETE | LQFP | PT | 48 | | TBD | Call TI | Call TI |

(1) The marketing status values are defined as follows:
**ACTIVE:** Product device recommended for new designs.
**LIFEBUY:** TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
**NRND:** Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
**PREVIEW:** Device has been announced but is not in production. Samples may or may not be available.
**OBSOLETE:** TI has discontinued the production of the device.

(2) Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS) or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
**TBD:** The Pb-Free/Green conversion plan has not been defined.
**Pb-Free (RoHS):** TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
**Green (RoHS & no Sb/Br):** TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

(3) MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

**Important Information and Disclaimer:**The information provided on this page represents TI's knowledge and belief as of the date that it is provided. TI bases its knowledge and belief on information provided by third parties, and makes no representation or warranty as to the accuracy of such information. Efforts are underway to better integrate information from third parties. TI has taken and continues to take reasonable steps to provide representative and accurate information but may not have conducted destructive testing or chemical analysis on incoming materials and chemicals. TI and TI suppliers consider certain information to be proprietary, and thus CAS numbers and other limited information may not be available for release.

In no event shall TI's liability arising out of such information exceed the total purchase price of the TI part(s) at issue in this document sold by TI to Customer on an annual basis.

## IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

| **Products** | | **Applications** | |
|---|---|---|---|
| Amplifiers | amplifier.ti.com | Audio | www.ti.com/audio |
| Data Converters | dataconverter.ti.com | Automotive | www.ti.com/automotive |
| DSP | dsp.ti.com | Broadband | www.ti.com/broadband |
| Interface | interface.ti.com | Digital Control | www.ti.com/digitalcontrol |
| Logic | logic.ti.com | Military | www.ti.com/military |
| Power Mgmt | power.ti.com | Optical Networking | www.ti.com/opticalnetwork |
| Microcontrollers | microcontroller.ti.com | Security | www.ti.com/security |
| | | Telephony | www.ti.com/telephony |
| | | Video & Imaging | www.ti.com/video |
| | | Wireless | www.ti.com/wireless |

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2005, Texas Instruments Incorporated