Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

Contact us

SC16C752

Dual UART with 64-byte FIFO

Rev. 04 — 20 June 2003 **Product data**

1. Description

The SC16C752 is a dual universal asynchronous receiver/transmitter (UART) with 64-byte FIFOs, automatic hardware/software flow control, and data rates up to 5 Mbits/s (3.3 V and 5 V). The SC16C752 offers enhanced features. It has a transmission control register (TCR) that stores receiver FIFO threshold levels to start/stop transmission during hardware and software flow control. With the FIFO RDY register, the software gets the status of $\overline{\text{TXRDY}}$/$\overline{\text{RXRDY}}$ for all four ports in one access. On-chip status registers provide the user with error indications, operational status, and modem interface control. System interrupts may be tailored to meet user requirements. An internal loop-back capability allows on-board diagnostics.

The UART transmits data, sent to it over 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 SC16C752 is available in a plastic LQFP48 package.

2. Features

- Pin compatible with SC16C2550 with additional enhancements
- Up to 5 Mbits/s baud rate (at 3.3 V and 5 V; at 2.5 V maximum baud rate is 3 Mbits/s)
- 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
- Software/hardware flow control
  - Programmable Xon/Xoff characters
  - Programmable auto-$\overline{\text{RTS}}$ and auto-$\overline{\text{CTS}}$
- Optional data flow resume by Xon any character
- DMA signalling capability for both received and transmitted data
- Supports 5 V, 3.3 V and 2.5 V operation
- Software selectable baud rate generator
- Prescaler provides additional divide-by-4 function

- Fast databus access time
- Programmable sleep mode
- Programmable serial interface characteristics
  - 5, 6, 7, or 8-bit 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 loop-back 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 $\overline{\text{CD}}$).

3. Ordering information

Table 1: Ordering information

Type number	Package		
	Name	Description	Version
SC16C752IB48	LQFP48	plastic low profile quad flat package; 48 leads; body 7 × 7 × 1.4 mm	SOT313-2

4. Block diagram

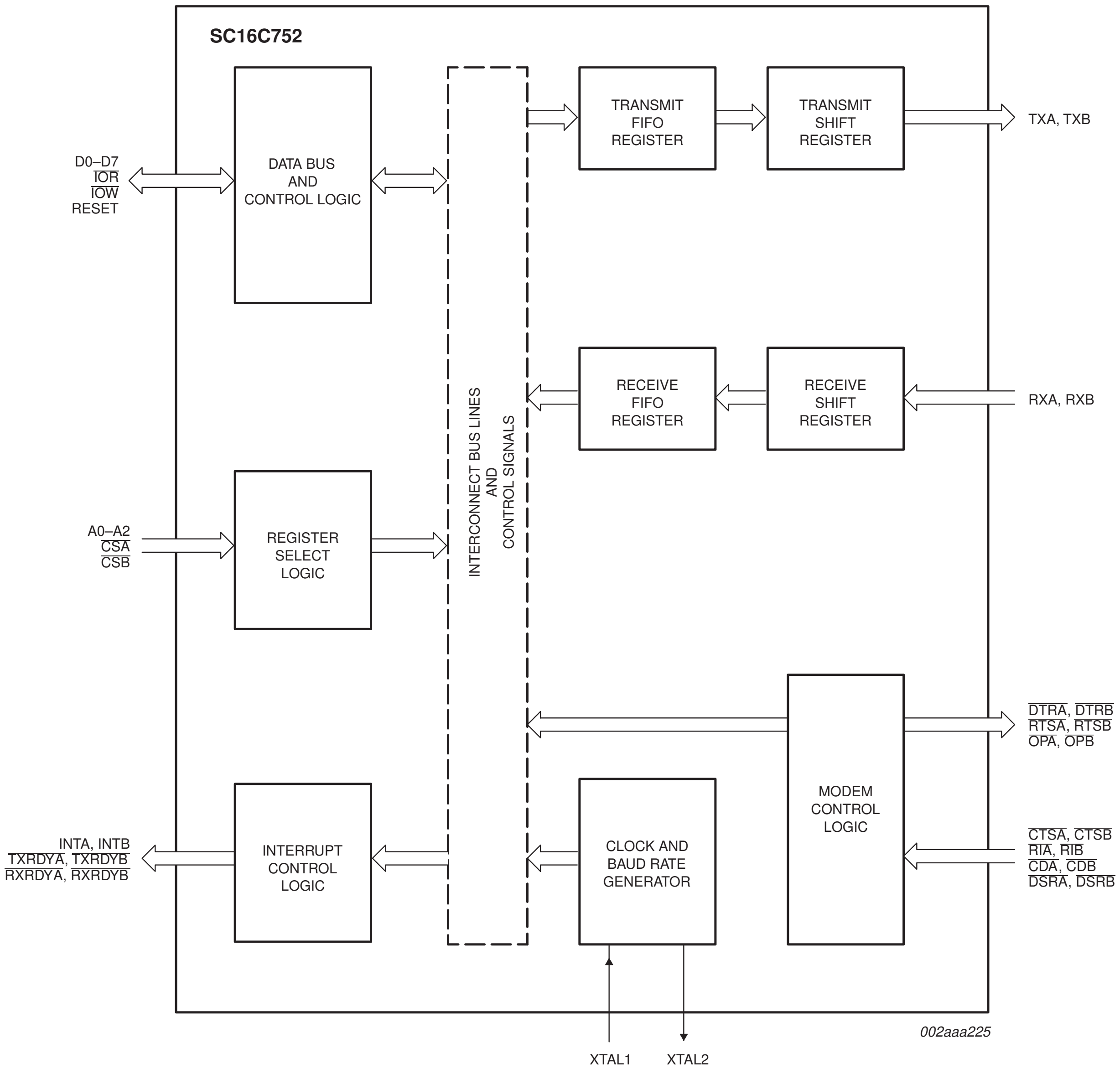


Fig 1. Block diagram.

5. Pinning information

5.1 Pinning

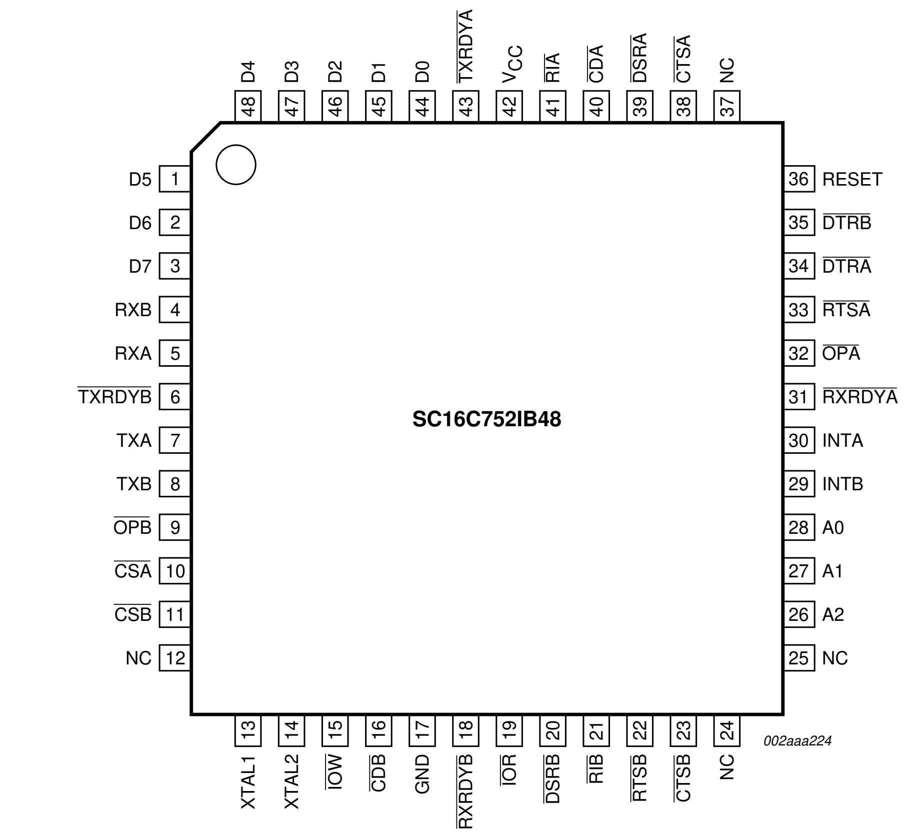


Fig 2. Pin configuration.

5.2 Pin description

Table 2: Pin description

Symbol	Pin	Type	Description
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 and B. A logic LOW on these pins indicates that a carrier has been detected by the modem for that channel. The state of these inputs is reflected in the modem status register (MSR).
$\overline{\text{CSA}}$, $\overline{\text{CSB}}$	10, 11	I	**Chip Select (Active-LOW).** These pins enable data transfers between the user CPU and the SC16C752 for the channel(s) addressed. Individual UART sections (A, B) are addressed by providing a logic LOW on the respective $\overline{\text{CSA}}$ and $\overline{\text{CSB}}$ pins.

Table 2: Pin description…*continued*

Symbol	Pin	Type	Description
$\overline{\mathrm{CTSA}}$, $\overline{\mathrm{CTSB}}$	38, 23	I	**Clear to Send (Active-LOW).** These inputs are associated with individual UART channels A and B. A logic 0 (LOW) on the $\overline{\mathrm{CTS}}$ pins indicates the modem or data set is ready to accept transmit data from the SC16C752. Status can be tested by reading MSR[4]. These pins only affect the transmit and receive operations when Auto-CTS function is enabled via the Enhanced Feature Register EFR[7] for hardware flow control operation.
D0-D4, D5-D7	44-48, 1-3	I/O	**Data bus (bi-directional).** These pins are the 8-bit, 3-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.
$\overline{\mathrm{DSRA}}$, $\overline{\mathrm{DSRB}}$	39, 20	I	**Data Set Ready (Active-LOW).** These inputs are associated with individual UART channels A and B. A logic 0 (LOW) on these pins indicates the modem or data set is powered-on and is ready for data exchange with the UART. The state of these inputs is reflected in the modem status register (MSR).
$\overline{\mathrm{DTRA}}$, $\overline{\mathrm{DTRB}}$	34, 35	O	**Data Terminal Ready (Active-LOW).** These outputs are associated with individual UART channels A and B. A logic 0 (LOW) on these pins indicates that the SC16C752 is powered-on and ready. These pins can be controlled via the modem control register. Writing a logic 1 to MCR[0] will set the $\overline{\mathrm{DTR}}$ output to logic 0 (LOW), enabling the modem. The output of these pins will be a logic 1 after writing a logic 0 to MCR[0], or after a reset.
GND	17	I	**Signal and power ground.**
INTA, INTB	30, 29	O	**Interrupt A and B (Active-HIGH).** These pins provide individual channel interrupts INTA and INTB. INTA and INTB are enabled when MCR[3] is set to a logic 1, interrupt sources are enabled in the interrupt enable register (IER). Interrupt conditions include: receiver errors, available receiver buffer data, available transmit buffer space, or when a modem status flag is detected. INTA, INTB are in the high-impedance state after reset.
$\overline{\mathrm{IOR}}$	19	I	**Input/Output Read strobe (Active-LOW).** A HIGH-to-LOW transition on $\overline{\mathrm{IOR}}$ will load the contents of an internal register defined by address bits A0-A2 onto the SC16C752 data bus (D0-D7) for access by external CPU.
$\overline{\mathrm{IOW}}$	15	I	**Input/Output Write strobe (Active-LOW).** A LOW-to-HIGH transition on $\overline{\mathrm{IOW}}$ 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 $\overline{\mathrm{CSA}}$ and $\overline{\mathrm{CSB}}$.
NC	12, 24, 25, 37	-	Not connected.
$\overline{\mathrm{OPA}}$, $\overline{\mathrm{OPB}}$	32, 9	O	**User defined outputs.** This function is associated with individual channels A and B. The state of these pins is defined by the user through the software settings of MCR[3]. INTA-INTB are set to active mode and $\overline{\mathrm{OPA}}$-$\overline{\mathrm{OPB}}$ to a logic 0 when MCR[3] is set to a logic 1. INTA-INTB are set to the 3-State mode and $\overline{\mathrm{OPA}}$-$\overline{\mathrm{OPB}}$ to a logic 1 when MCR[3] is set to a logic 0. The output of these two pins is HIGH after reset.
RESET	36	I	**Reset.** This pin will reset the internal registers and all the outputs. The UART transmitter output and the receiver input will be disabled during reset time. RESET is an active-HIGH input.
$\overline{\mathrm{RIA}}$, $\overline{\mathrm{RIB}}$	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 LOW-to-HIGH transition on these input pins generates a modem status interrupt, if enabled. The state of these inputs is reflected in the modem status register (MSR).

Table 2: Pin description…*continued*

Symbol	Pin	Type	Description
$\overline{\text{RTSA}}$, $\overline{\text{RTSB}}$	33, 22	O	**Request to Send (Active-LOW).** These outputs are associated with individual UART channels, A and B. A logic 0 on the $\overline{\text{RTS}}$ pin indicates the transmitter has data ready and waiting to send. Writing a logic 1 in the modem control register MCR[1] will set this pin to a logic 0, indicating data is available. After a reset these pins are set to a logic 1. These pins only affect the transmit and receive operations when Auto-$\overline{\text{RTS}}$ function is enabled via the Enhanced Feature Register (EFR[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 SC16C752. During the local loop-back mode, these RX input pins are disabled and TX data is connected to the UART RX input internally.
$\overline{\text{RXRDYA}}$, $\overline{\text{RXRDYB}}$	31, 18	O	**Receive Ready (Active-LOW).** $\overline{\text{RXRDYA}}$ or $\overline{\text{RXRDYB}}$ goes LOW when the trigger level has been reached or the FIFO has at least one character. It goes HIGH when the RX FIFO is empty.
TXA, TXB	7, 8	O	**Transmit data A, B.** These outputs are associated with individual serial transmit channel data from the SC16C752. During the local loop-back mode, the TX output pin is disabled and TX data is internally connected to the UART RX input.
$\overline{\text{TXRDYA}}$, $\overline{\text{TXRDYB}}$	43, 6	O	**Transmit Ready (Active-LOW).** $\overline{\text{TXRDYA}}$ or $\overline{\text{TXRDYB}}$ go LOW when there are at least a trigger level number of spaces available or when the FIFO is empty. It goes HIGH when the FIFO is full or not empty.
V_{CC}	42	I	**Power supply input.**
XTAL1	13	I	**Crystal or external clock input.** 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 Figure 12). Alternatively, an external clock can be connected to this pin 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 a buffered clock output.

6. Functional description

The SC16C752 UART is pin-compatible with the SC16C2550 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 SC16C752 UART can be read at any time during functional operation by the processor.

The SC16C752 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 SC16C752 has 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 includes a programmable baud rate generator that can divide the timing reference clock input by a divisor between 1 and (2^{16} – 1).

6.1 Trigger levels

The SC16C752 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.

6.2 Hardware flow control

Hardware flow control is comprised of 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 $\overline{\text{RTS}}$ output when there is enough room in the FIFO to receive data and de-activates the $\overline{\text{RTS}}$ output when the RX FIFO is sufficiently full. The halt and resume trigger levels in the TCR determine the levels at which $\overline{\text{RTS}}$ 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 servicing latency.

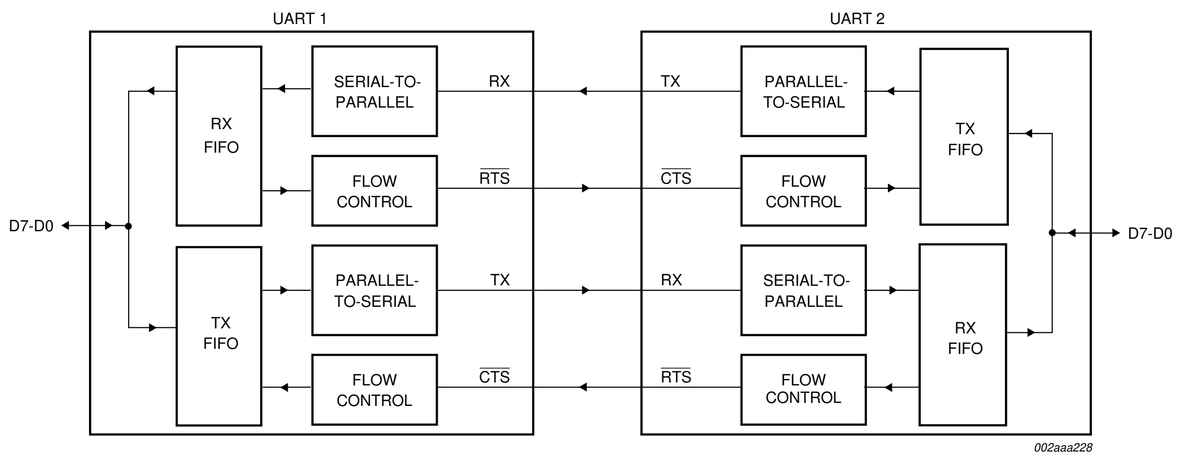
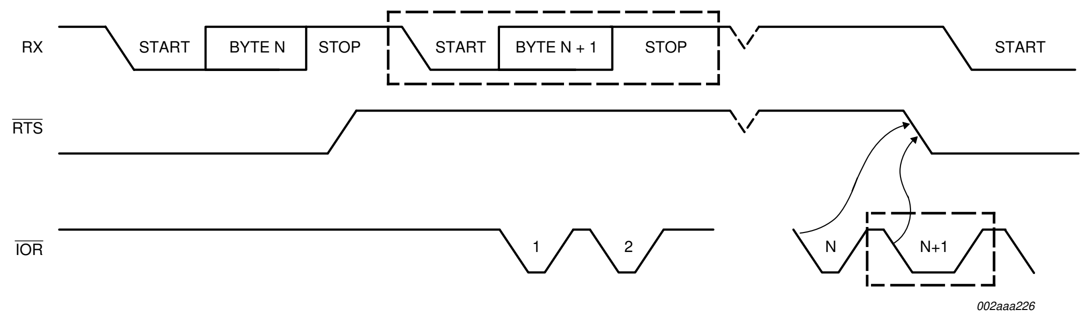


Fig 3. Autoflow control (Auto-$\overline{\text{RTS}}$ and Auto-$\overline{\text{CTS}}$) example.

6.2.1 Auto-$\overline{\text{RTS}}$

Auto-$\overline{\text{RTS}}$ data flow control originates in the receiver block (see Figure 1 “Block diagram.” on page 3). Figure 4 shows $\overline{\text{RTS}}$ functional timing. The receiver FIFO trigger levels used in Auto-$\overline{\text{RTS}}$ are stored in the TCR. $\overline{\text{RTS}}$ 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.



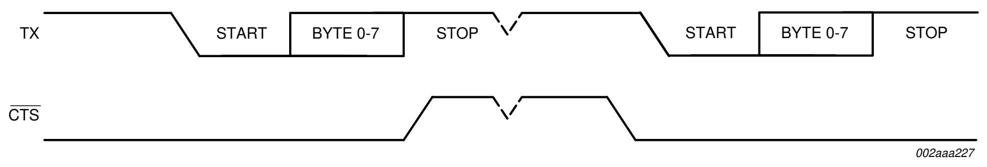
(1) N = receiver FIFO trigger level.

(2) The two blocks in dashed lines cover the case where an additional byte is sent, as described in Section 6.2.1.

Fig 4. $\overline{\text{RTS}}$ functional timing.

6.2.2 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, $\overline{\text{CTS}}$ level changes do 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 may result.



(1) When $\overline{\text{CTS}}$ is LOW, the transmitter keeps sending serial data out.

(2) 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 is does not send the next byte.

(3) When $\overline{\text{CTS}}$ goes from HIGH to LOW, the transmitter begins sending data again.

Fig 5. $\overline{\text{CTS}}$ functional timing.

6.3 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 3 shows software flow control options.

Table 3: Software flow control options (EFR[0:3])

EFR[3]	EFR[2]	EFR[1]	EFR[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 compared Xon1, Xoff1
X	X	0	1	receiver compares Xon2, Xoff2
1	0	1	1	transmit Xon1, Xoff1 receiver compares Xon1 and Xon2, Xoff1 and Xoff2
0	1	1	1	transmit Xon2, Xoff2 receiver compares Xon1 and Xon2, Xoff1 and Xoff2
1	1	1	1	transmit Xon1, Xon2, Xoff1, Xoff2 receiver compares Xon1 and Xon2, Xoff1 and Xoff2

Remark: When using software flow control, the Xon/Xoff characters cannot be used for data characters.

There are two other enhanced features relating to software flow control:

- **Xon Any function (MCR[5]):** Operation will resume after receiving any character after recognizing the Xoff character. 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 RX FIFO.

6.3.1 RX

When software flow control operation is enabled, the SC16C752 will compare incoming data with Xoff1,2 programmed characters (in certain cases, Xoff1 and Xoff2 must be received sequentially). When the correct Xoff character 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.

6.3.2 TX

Xoff1/2 character is transmitted when the RX FIFO has passed the HALT trigger level programmed in TCR[3:0].

Xon1/2 character is transmitted when the RX FIFO reaches the RESUME trigger level programmed in TCR[7:4].

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 6 shows an example of software flow control.

6.3.3 Software flow control example

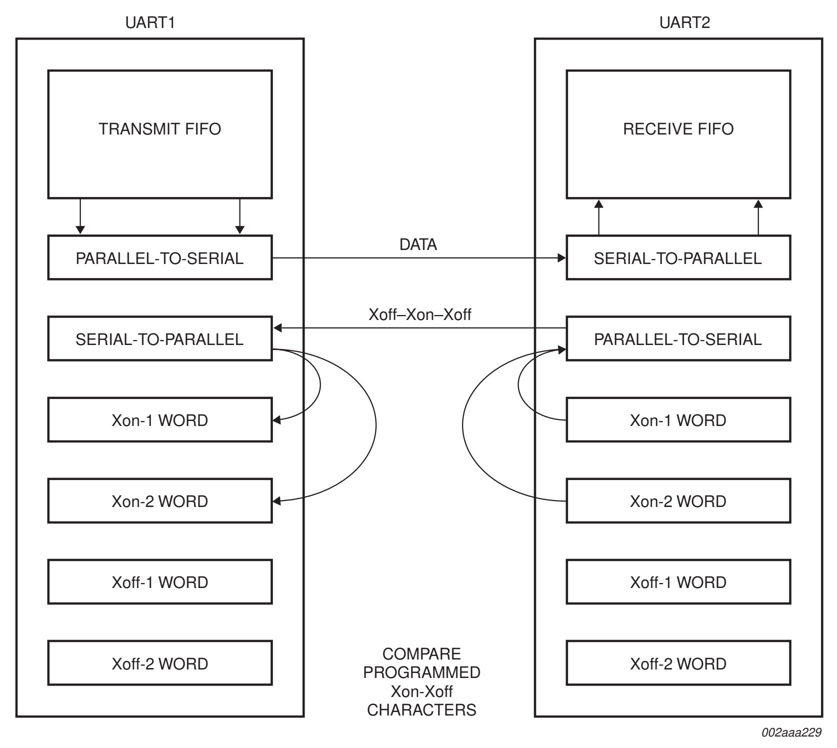


Fig 6. Software flow control example.

Assumptions: UART1 is transmitting a large text file to UART2. Both UARTs are using software flow control with single character Xoff (0F) and Xon (0D) tokens. Both have Xoff threshold (TCR[3:0] = F) set to 60, and Xon threshold (TCR[7:4] = 8) set to 32. Both have the interrupt receive threshold (TLR[7:4] = D) set to 52.

UART 1 begins transmission and sends 52 characters, at which point UART2 will generate an interrupt to its processor to service the RCV FIFO, but assume the interrupt latency is fairly long. UART1 will continue sending characters until a total of

60 characters have been sent. At this time, UART2 will transmit a 0F to UART1, informing UART1 to halt transmission. UART1 will likely send the 61st character while UART2 is sending the Xoff character. Now UART2 is serviced and the processor reads enough data out of the RX FIFO that the level drops to 32. UART2 will now send a 0D to UART1, informing UART1 to resume transmission.

6.4 Reset

Table 4 summarizes the state of register after reset.

Table 4: 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.

[1] Registers DLL, DLH, SPR, Xon1, Xon2, Xoff1, Xoff2 are not reset by the top-level reset signal RESET, i.e., they hold their initialization values during reset.

Table 5 summarizes the state of registers after reset.

Table 5: 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

6.5 Interrupts

The SC16C752 has interrupt generation and prioritization (six prioritized levels of interrupts) capability. The interrupt enable register (IER) enables each of the six 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 6 summarizes the interrupt control functions.

Table 6: 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. OE: read LSR
001100	2	RX time-out	stale data in RX FIFO	read RHR
000100	2	RHR interrupt	DRDY (data ready) (FIFO disable) RX FIFO above trigger level (FIFO enable)	read RHR
000010	3	THR interrupt	TFE (THR empty) (FIFO disable) TX FIFO passes above 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 framing error, parity error, and break conditions, LSR[7] generates the interrupt. 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. LSR[4:2] always represent the error status for the received character at the top of the RX FIFO. Reading the RX FIFO updates LSR[4:2] to the appropriate status for the new character at the top of the FIFO. If the RX FIFO is empty, then LSR[4:2] are all zeros.

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 LSR.

6.5.1 Interrupt mode operation

In interrupt mode (if any bit of IER[3:0] is 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 status register (LSR) to see if any interrupt needs to be serviced. Figure 7 shows interrupt mode operation.

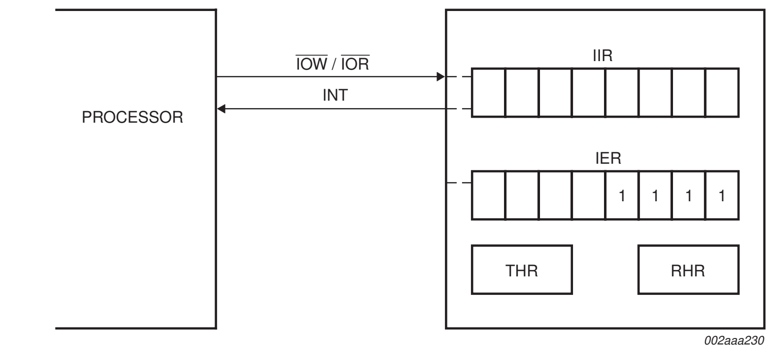


Fig 7. Interrupt mode operation.

6.5.2 Polled mode operation

In polled mode (IER[3:0] = 0000) the status of the receiver and transmitter can 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 8 shows FIFO polled mode operation.

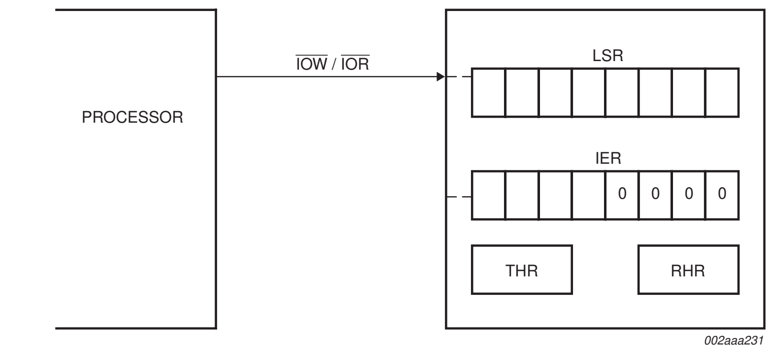


Fig 8. FIFO polled mode operation.

6.6 DMA operation

There are two modes of DMA operation, DMA mode 0 or DMA mode 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 managed to relieve the processor for longer periods of time.

6.6.1 Single DMA transfers (DMA mode 0/FIFO disable)

Figure 9 shows $\overline{\text{TXRDY}}$ and $\overline{\text{RXRDY}}$ in DMA mode 0/FIFO disable.

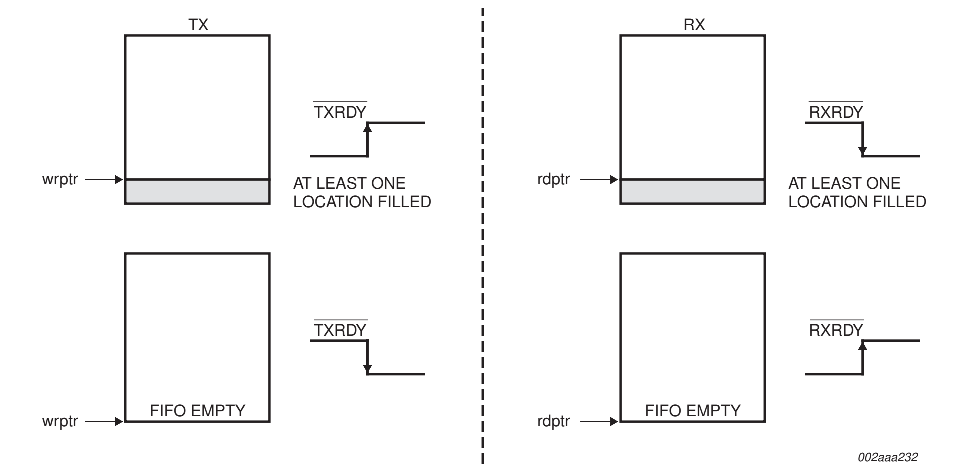

TX
$\overline{\text{TXRDY}}$
wrptr
AT LEAST ONE LOCATION FILLED
$\overline{\text{TXRDY}}$
wrptr
FIFO EMPTY
RX
$\overline{\text{RXRDY}}$
rdptr
AT LEAST ONE LOCATION FILLED
$\overline{\text{RXRDY}}$
rdptr
FIFO EMPTY
002aaa232

Fig 9. $\overline{\text{TXRDY}}$ and $\overline{\text{RXRDY}}$ in DMA mode 0/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.

6.6.2 Block DMA transfers (DMA mode 1)

Figure 10 shows $\overline{TXRDY}$ and $\overline{RXRDY}$ in DMA mode 1.

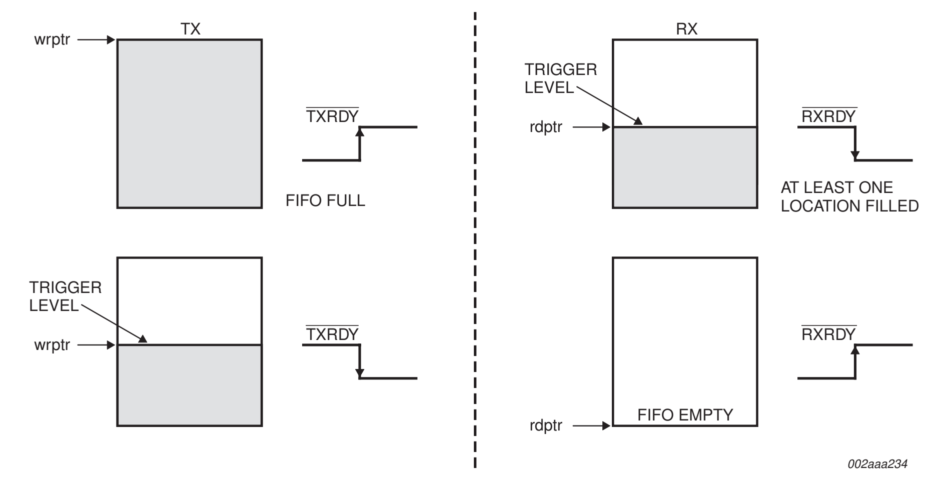


Fig 10. $\overline{TXRDY}$ and $\overline{RXRDY}$ in DMA mode 1.

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

Receiver: $\overline{RXRDY}$ becomes active when the trigger level has been reached, or when a time-out interrupt occurs. It will go inactive when the FIFO is empty or an error in the RX FIFO is flagged by LSR[7].

6.7 Sleep mode

Sleep mode is an enhanced feature of the SC16C752 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 Section 6.8 “Break and time-out conditions”).
- The TX FIFO and TX shift register are empty.
- There are no interrupts pending except THR and time-out interrupts.

Remark: 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.

Remark: 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.

6.8 Break and time-out conditions

An RX idle condition is detected when the receiver line, RX, has been HIGH for 4 character time. 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].

6.9 Programmable baud rate generator

The SC16C752 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). An additional divide-by-4 prescaler is also available and can be selected by MCR[7], as shown in Figure 11. The output frequency of the baud rate generator is 16× the baud rate. The formula for the divisor is:

$$\text{divisor} = \frac{\left(\dfrac{\text{XTAL1 crystal input frequency}}{\text{prescaler}}\right)}{(\text{desired baud rate} \times 16)}$$

Where:

prescaler = 1, when MCR[7] is set to 0 after reset (divide-by-1 clock selected)

prescaler = 4, when MCR[7] is set to 1 after reset (divide-by-4 clock selected).

Remark: The default value of prescaler after reset is divide-by-1.

Figure 11 shows the internal prescaler and baud rate generator circuitry.

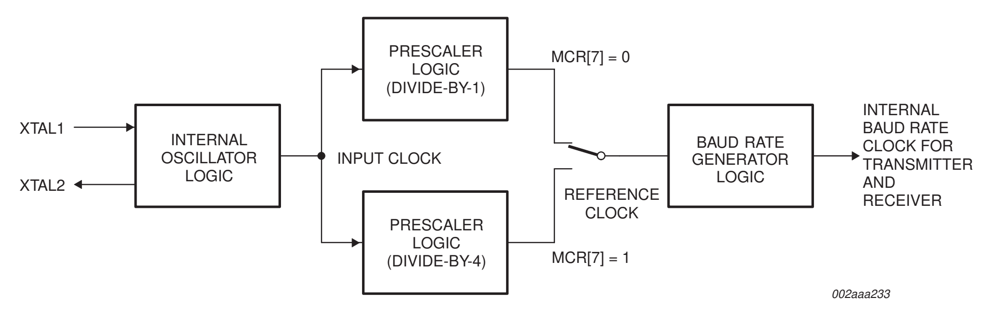

XTAL1
XTAL2
INTERNAL OSCILLATOR LOGIC
INPUT CLOCK
PRESCALER LOGIC (DIVIDE-BY-1)
PRESCALER LOGIC (DIVIDE-BY-4)
MCR[7] = 0
MCR[7] = 1
REFERENCE CLOCK
BAUD RATE GENERATOR LOGIC
INTERNAL BAUD RATE CLOCK FOR TRANSMITTER AND RECEIVER
002aaa233

Fig 11. 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. If DLL and DLH are both zero, the UART is effectively disabled, as no baud clock will be generated.

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

Table 7 and Table 8 show the baud rate and divisor correlation for crystal with frequency 1.8432 MHz and 3.072 MHz, respectively.

Figure 12 shows the crystal clock circuit reference.

Table 7: 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 8: 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	2304	
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	

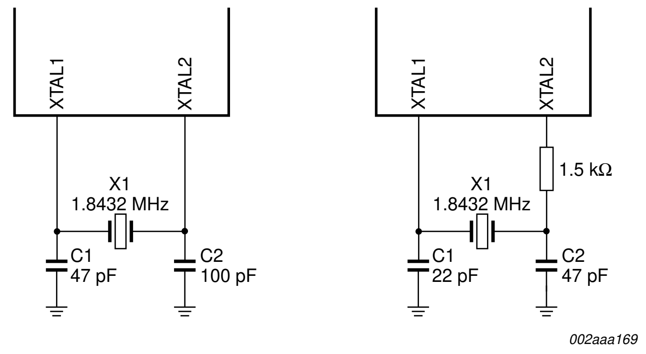


Fig 12. Crystal oscillator connections.

7. Register descriptions

Each register is selected using address lines A0, A1, A2, and in some cases, bits from other registers. The programming combinations for register selection are shown in Table 9.

Table 9: Register map - read/write properties

A2	A1	A0	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)[1]	modem control register[1]
1	0	1	line status register (LSR)	
1	1	0	modem status register (MSR)	
1	1	1	scratchpad register (SPR)	scratchpad register
0	0	0	divisor latch LSB (DLL)[2], [3]	divisor latch LSB[2], [3]
0	0	1	divisor latch MSB (DLH)[2], [3]	divisor latch MSB[2], [3]
0	1	0	enhanced feature register (EFR)[2], [4]	enhanced feature register[2], [4]
1	0	0	Xon1 word[2], [4]	Xon1 word[2], [4]
1	0	1	Xon2 word[2], [4]	Xon2 word[2], [4]
1	1	0	Xoff1 word[2], [4]	Xoff1 word[2], [4]
1	1	1	Xoff2 word[2], [4]	Xoff2 word[2], [4]
1	1	0	transmission control register (TCR)[2], [5]	transmission control register[2], [5]
1	1	1	trigger level register (TLR)[2], [5]	trigger level register[2], [5]
1	1	1	FIFO ready register[2], [6]	

[1] MCR[7] can only be modified when EFR[4] is set.

[2] Accessed by a combination of address pins and register bits.

[3] Accessible only when LCR[7] is logic 1.

[4] Accessible only when LCR is set to 10111111 (8hBF).

[5] Accessible only when EFR[4] = 1 and MCR[6] = 1, i.e., EFR[4] and MCR[6] are read/write enables.

[6] Accessible only when $\overline{\text{CSA}}$ or $\overline{\text{CSB}}$ = 0, MCR[2] = 1, and loop-back is disabled (MCR[4] = 0).

Table 10 lists and describes the SC16C752 internal registers.

Table 10: SC16C752 internal registers

Shaded bits are only accessible when EFR[4] is set.

A2	A1	A0	Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Read/ Write
General Register Set[1]												
0	0	0	RHR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R
0	0	0	THR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	W
0	0	1	IER	0/$\overline{\text{CTS}}$ interrupt enable[2]	0/$\overline{\text{RTS}}$ interrupt enable[2]	0/$\overline{\text{Xoff}}$[2]	0/X sleep mode[2]	modem status interrupt	receive line status interrupt	THR empty interrupt	Rx data available interrupt	R/W
0	1	0	FCR	RX trigger level (MSB)	RX trigger level (LSB)	0/TX trigger level (MSB)[2]	0/TX trigger level (LSB)[2]	DMA mode select	TX FIFO reset	RX FIFO reset	FIFO enable	W
0	1	0	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	R
0	1	1	LCR	DLAB	break control bit	set parity	parity type select	parity enable	number of stop bits	word length bit 1	word length bit 0	R/W
1	0	0	MCR	1× or 1×/4 clock	TCR and TLR enable	0/Xon Any	0/enable loop-back	IRQ enable OP	FIFO ready enable	RTS	DTR	R/W
1	0	1	LSR	0/error in RX FIFO	THR and TSR empty	THR empty	break interrupt	framing error	parity error	overrun error	data in receiver	R
1	1	0	MSR	CD	RI	DSR	CTS	ΔCD	ΔRI	ΔDSR	ΔCTS	R
1	1	1	SPR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
1	1	0	TCR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
1	1	1	TLR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
1	1	1	FIFO Rdy	0	0	RX FIFO B status	RX FIFO A status	0	0	TX FIFO B status	TX FIFO A status	R
Special Register Set[3]												
0	0	0	DLL	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0	0	1	DLH	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
Enhanced Register Set[4]												
0	1	0	EFR	Auto $\overline{\text{CTS}}$	Auto $\overline{\text{RTS}}$	Special character detect	Enable enhanced functions [2]	software flow control bit 3	software flow control bit 2	software flow control bit 1	software flow control bit 0	R/W
1	0	0	Xon1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
1	0	1	Xon2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
1	1	0	Xoff1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
1	1	1	Xoff2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W

[1] These registers are accessible only when LCR[7] = 0.

[2] 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.

[3] The Special Register set is accessible only when LCR[7] is set to a logic 1.

[4] Enhanced Feature Register; Xon-1,2 and Xoff-1,2 are accessible only when LCR is set to ‘BF_{Hex}’.

Remark: Refer to the notes under Table 9 for more register access information.

7.1 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 the 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.

Remark: In this case, characters are overwritten if overflow occurs.

If overflow occurs, characters are lost. The RHR also stores the error status bits associated with each character.

7.2 Transmit holding register (THR)

The transmitter section consists of the transmit holding register (THR) and the transmit shift register (TSR). The THR 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. Characters are lost if overflow occurs.

7.3 FIFO control register (FCR)

This is a write-only register that is used for enabling the FIFOs, clearing the FIFOs, setting transmitter and receiver trigger levels, and selecting the type of DMA signalling. Table 11 shows FIFO control register bit settings.

Table 11: FIFO Control Register bits description

Bit	Symbol	Description
7-6	FCR[7] (MSB), FCR[6] (LSB)	RCVR trigger. Sets the trigger level for the RX FIFO. 00 - 8 characters 01 - 16 characters 10 - 56 characters 11 - 60 characters
5-4	FCR[5] (MSB), FCR[4] (LSB)	TX trigger. Sets the trigger level for the TX FIFO. 00 - 8 spaces 01 - 16 spaces 10 - 32 spaces 11 - 56 spaces 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.
3	FCR[3]	DMA mode select. Logic 0 = Set DMA mode ‘0’ Logic 1 = Set DMA mode ‘1’
2	FCR[2]	Reset TX FIFO. Logic 0 = No FIFO transmit reset (normal default condition). Logic 1 = Clears the contents of the transmit FIFO and resets the FIFO counter logic (the transmit shift register is not cleared or altered). This bit will return to a logic 0 after clearing the FIFO.
1	FCR[1]	Reset RX FIFO. Logic 0 = No FIFO receive reset (normal default condition). Logic 1 = Clears the contents of the receive FIFO and resets the FIFO counter logic (the receive shift register is not cleared or altered). This bit will return to a logic 0 after clearing the FIFO.
0	FCR[0]	FIFO enable. Logic 0 = Disable the transmit and receive FIFO (normal default condition). Logic 1 = Enable the transmit and receive FIFO.

7.4 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 12 shows the line control register bit settings.

Table 12: Line Control Register bits description

Bit	Symbol	Description
7	LCR[7]	Divisor latch enable. Logic 0 = Divisor latch disabled (normal default condition). Logic 1 = Divisor latch enabled.
6	LCR[6]	Break control bit. When enabled, the Break control bit causes a break condition to be transmitted (the TX output is forced to a logic 0 state). This condition exists until disabled by setting LCR[6] to a logic 0. Logic 0 = no TX break condition (normal default condition). Logic 1 = forces the transmitter output (TX) to a logic 0 to alert the communication terminal to a line break condition.
5	LCR[5]	Set parity. LCR[5] selects the forced parity format (if LCR[3] = 1). Logic 0 = parity is not forced (normal default condition). LCR[5] = logic 1 and LCR[4] = logic 0: parity bit is forced to a logical 1 for the transmit and receive data. LCR[5] = logic 1 and LCR[4] = logic 1: parity bit is forced to a logical 0 for the transmit and receive data.
4	LCR[4]	Parity type select. Logic 0 = ODD Parity is generated (if LCR[3] = 1). Logic 1 = EVEN Parity is generated (if LCR[3] = 1).
3	LCR[3]	Parity enable. Logic 0 = no parity (normal default condition). Logic 1 = a parity bit is generated during transmission and the receiver checks for received parity.
2	LCR[2]	Number of Stop bits. Specifies the number of stop bits. 0 - 1 stop bit (word length = 5, 6, 7, 8) 1 - 1.5 stop bits (word length = 5) 1 = 2 stop bits (word length = 6, 7, 8)
1-0	LCR[1-0]	Word length bits 1, 0. These two bits specify the word length to be transmitted or received. 00 - 5 bits 01 - 6 bits 10 - 7 bits 11 - 8 bits

7.5 Line status register (LSR)

Table 13 shows the line status register bit settings.

Table 13: Line Status Register bits description

Bit	Symbol	Description
7	LSR[7]	FIFO data error. Logic 0 = No error (normal default condition). Logic 1 = At least one parity error, framing error, or break indication is in the receiver FIFO. This bit is cleared when no more errors are present in the FIFO.
6	LSR[6]	THR and TSR empty. This bit is the Transmit Empty indicator. Logic 0 = Transmitter hold **and** shift registers are not empty. Logic 1 = Transmitter hold **and** shift registers are empty.
5	LSR[5]	THR empty. This bit is the Transmit Holding Register Empty indicator. Logic 0 = Transmit hold register is **not** empty. Logic 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.
4	LSR[4]	Break interrupt. Logic 0 = No break condition (normal default condition). Logic 1 = A break condition occurred and associated byte is 00, i.e., RX was LOW for one character time frame.
3	LSR[3]	Framing error. Logic 0 = No framing error in data being read from RX FIFO (normal default condition). Logic 1 = Framing error occurred in data being read from RX FIFO, i.e., received data did not have a valid stop bit.
2	LSR[2]	Parity error. Logic 0 = No parity error (normal default condition). Logic 1 = Parity error in data being read from RX FIFO.
1	LSR[1]	Overrun error. Logic 0 = No overrun error (normal default condition). Logic 1 = Overrun error has occurred.
0	LSR[0]	Data in receiver. Logic 0 = No data in receive FIFO (normal default condition). Logic 1 = At least one character in the RX 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 do not 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.

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.

Remark: The three error bits (parity, framing, break) may not be updated correctly in the first read of the LSR when the input clock (XTAL1) is running faster than 36 MHz. However, the second read is always correct. It is strongly recommended that when using this device with a clock faster than 36 MHz, that the LSR be read twice and only the second read be used for decision making. All other bits in the LSR are correct on all reads.

7.6 Modem control register (MCR)

The MCR controls the interface with the mode, data set, or peripheral device that is emulating the modem. Table 14 shows modem control register bit settings.

Table 14: Modem Control Register bits description

Bit	Symbol	Description
7	MCR[7] [1]	Clock select. Logic 0 = Divide-by-1 clock input. Logic 1 = Divide-by-4 clock input.
6	MCR[6] [1]	TCR and TLR enable. Logic 0 = no action. Logic 1 = Enable access to the TCR and TLR registers.
5	MCR[5] [1]	Xon Any. Logic 0 = Disable Xon Any function. Logic 1 = Enable Xon Any function.
4	MCR[4]	Enable loop-back. Logic 0 = Normal operating mode. Logic 1 = Enable local loop-back mode (internal). In this mode the MCR[3:0] signals are looped back into MSR[7:4] and the TX output is looped back to the RX input internally.
3	MCR[3]	IRQ enable OP. Logic 0 = Forces INTA-INTB outputs to the 3-State mode and OP output to HIGH state. Logic 1 = Forces the INTA-INTB outputs to the active state and OP output to LOW state. In loop-back mode, controls MSR[7].
2	MCR[2]	FIFO Ready enable. Logic 0 = Disable the FIFO Rdy register. Logic 1 = Enable the FIFO Rdy register. In loop-back mode, controls MSR[6].
1	MCR[1]	$\overline{\text{RTS}}$ Logic 0 = Force $\overline{\text{RTS}}$ output to inactive (HIGH). Logic 1 = Force $\overline{\text{RTS}}$ output to active (LOW). In loop-back mode, controls MSR[4]. If Auto-$\overline{\text{RTS}}$ is enabled, the $\overline{\text{RTS}}$ output is controlled by hardware flow control.
0	MCR[0]	$\overline{\text{DTR}}$ Logic 0 = Force $\overline{\text{DTR}}$ output to inactive (HIGH). Logic 1 = Force $\overline{\text{DTR}}$ output to active (LOW). In loop-back mode, controls MSR[5].

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

7.7 Modem status register (MSR)

This 8-bit register provides information about the current state of the control lines from the mode, data set, or peripheral device to the processor. It also indicates when a control input from the modem changes state. Table 15 shows modem status register bit settings per channel.

Table 15: Modem Status Register bits description

Bit	Symbol	Description
7	MSR[7]	CD (Active-HIGH, logical 1). This bit is the complement of the $\overline{\text{CD}}$ input during normal mode. During internal loop-back mode, it is equivalent to MCR[3].
6	MSR[6]	RI (Active-HIGH, logical 1). This bit is the complement of the $\overline{\text{RI}}$ input during normal mode. During internal loop-back mode, it is equivalent to MCR[2].
5	MSR[5]	DSR (Active-HIGH, logical 1). This bit is the complement of the $\overline{\text{DSR}}$ input during normal mode. During internal loop-back mode, it is equivalent MCR[0].
4	MSR[4]	CTS (Active-HIGH, logical 1). This bit is the complement of the $\overline{\text{CTS}}$ input during normal mode. During internal loop-back mode, it is equivalent to MCR[1].
3	MSR[3]	ΔCD. Indicates that $\overline{\text{CD}}$ input (or MCR[3] in loop-back mode) has changed state. Cleared on a read.
2	MSR[2]	ΔRI. Indicates that $\overline{\text{RI}}$ input (or MCR[2] in loop-back mode) has changed state from LOW to HIGH. Cleared on a read.
1	MSR[1]	ΔDSR. Indicates that $\overline{\text{DSR}}$ input (or MCR[0] in loop-back mode) has changed state. Cleared on a read.
0	MSR[0]	ΔCTS. Indicates that CTS input (or MCR[1] in loop-back mode) has changed state. Cleared on a read.

[1] 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 loop-back) registers are Active-HIGH.

7.8 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 16 shows interrupt enable register bit settings.

Table 16: Interrupt Enable Register bits description

Bit	Symbol	Description
7	IER[7] [1]	$\overline{\text{CTS}}$ interrupt enable. Logic 0 = Disable the $\overline{\text{CTS}}$ interrupt (normal default condition). Logic 1 = Enable the $\overline{\text{CTS}}$ interrupt.
6	IER[6] [1]	$\overline{\text{RTS}}$ interrupt enable. Logic 0 = Disable the $\overline{\text{RTS}}$ interrupt (normal default condition). Logic 1 = Enable the $\overline{\text{RTS}}$ interrupt.
5	IER[5] [1]	Xoff interrupt. Logic 0 = Disable the Xoff interrupt (normal default condition). Logic 1 = Enable the Xoff interrupt.