Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

SC16IS740/750/760

Single UART with I²C-bus/SPI interface, 64 bytes of transmit and receive FIFOs, IrDA SIR built-in support

Rev. 7 — 9 June 2011 Product data sheet

1. General description

The SC16IS740/750/760 is a slave I²C-bus/SPI interface to a single-channel high performance UART. It offers data rates up to 5 Mbit/s and guarantees low operating and sleeping current. The SC16IS750 and SC16IS760 also provide the application with 8 additional programmable I/O pins. The device comes in very small HVQFN24, TSSOP24 (SC16IS750/760) and TSSOP16 (SC16IS740) packages, which makes it ideally suitable for handheld, battery operated applications. This family of products enables seamless protocol conversion from I²C-bus or SPI to and RS-232/RS-485 and are fully bidirectional.

The SC16IS760 differs from the SC16IS750 in that it supports SPI clock speeds up to 15 Mbit/s instead of the 4 Mbit/s supported by the SC16IS750, and in that it supports IrDA SIR up to 1.152 Mbit/s. In all other aspects, the SC16IS760 is functionally and electrically the same as the SC16IS750. The SC16IS740 is functionally and electrically identical to the SC16IS750, with the exception of the programmable I/O pins which are only present on the SC16IS750.

The SC16IS740/750/760's internal register set is backward-compatible with the widely used and widely popular 16C450. This allows the software to be easily written or ported from another platform.

The SC16IS740/750/760 also provides additional advanced features such as auto hardware and software flow control, automatic RS-485 support, and software reset. This allows the software to reset the UART at any moment, independent of the hardware reset signal.

2. Features and benefits

2.1 General features

- Single full-duplex UART
- Selectable I²C-bus or SPI interface
- 3.3 V or 2.5 V operation
- Industrial temperature range: −40 °C to +95 °C
- 64 bytes FIFO (transmitter and receiver)
- Fully compatible with industrial standard 16C450 and equivalent
- Baud rates up to 5 Mbit/s in 16× clock mode
- Auto hardware flow control using $\overline{\text{RTS}}$/$\overline{\text{CTS}}$
- Auto software flow control with programmable Xon/Xoff characters
- Single or double Xon/Xoff characters
- Automatic RS-485 support (automatic slave address detection)

- Up to eight programmable I/O pins (SC16IS750 and SC16IS760 only)
- RS-485 driver direction control via $\overline{RTS}$ signal
- RS-485 driver direction control inversion
- Built-in IrDA encoder and decoder interface
- SC16IS750 supports IrDA SIR with speeds up to 115.2 kbit/s
- SC16IS760 supports IrDA SIR with speeds up to 1.152 Mbit/s[1]
- Software reset
- Transmitter and receiver can be enabled/disabled independent of each other
- Receive and Transmit FIFO levels
- Programmable special character detection
- Fully programmable character formatting
  - 5-bit, 6-bit, 7-bit or 8-bit character
  - Even, odd, or no parity
  - 1, $1^1\diagup_2$, or 2 stop bits
- Line break generation and detection
- Internal Loopback mode
- Sleep current less than 30 μA at 3.3 V
- Industrial and commercial temperature ranges
- Available in HVQFN24, TSSOP24 (SC16IS750/760) and TSSOP16 (SC16IS740) packages

2.2 I²C-bus features

- Noise filter on SCL/SDA inputs
- 400 kbit/s maximum speed
- Compliant with I²C-bus fast speed
- Slave mode only

2.3 SPI features

- SC16IS750 supports 4 Mbit/s maximum SPI clock speed
- SC16IS760 supports 15 Mbit/s maximum SPI clock speed
- Slave mode only
- SPI Mode 0

3. Applications

- Factory automation and process control
- Portable and battery operated devices
- Cellular data devices

1. Please note that IrDA SIR at 1.152 Mbit/s is **not** compatible with IrDA MIR at that speed. Please refer to application notes for usage of IrDA SIR at 1.152 Mbit/s.

4. Ordering information

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
SC16IS740IPW	TSSOP16	plastic thin shrink small outline package; 16 leads; body width 4.4 mm	SOT403-1
SC16IS740IPW/Q900[1]	TSSOP16	plastic thin shrink small outline package; 16 leads; body width 4.4 mm	SOT403-1
SC16IS750IBS	HVQFN24	plastic thermal enhanced very thin quad flat package; no leads; 24 terminals; body 4 × 4 × 0.85 mm	SOT616-3
SC16IS750IPW	TSSOP24	plastic thin shrink small outline package; 24 leads; body width 4.4 mm	SOT355-1
SC16IS760IBS	HVQFN24	plastic thermal enhanced very thin quad flat package; no leads; 24 terminals; body 4 × 4 × 0.85 mm	SOT616-3
SC16IS760IPW	TSSOP24	plastic thin shrink small outline package; 24 leads; body width 4.4 mm	SOT355-1

[1] SC16IS740IPW/Q900 is AEC-Q100 compliant. Contact interface.support@nxp.com for PPAP.

5. Block diagram

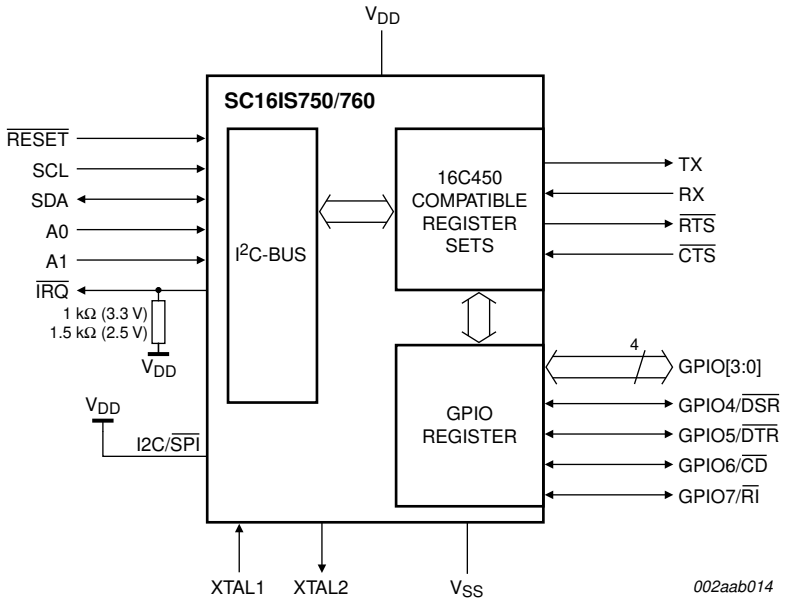


Fig 1. Block diagram of SC16IS750/760 I²C-bus interface

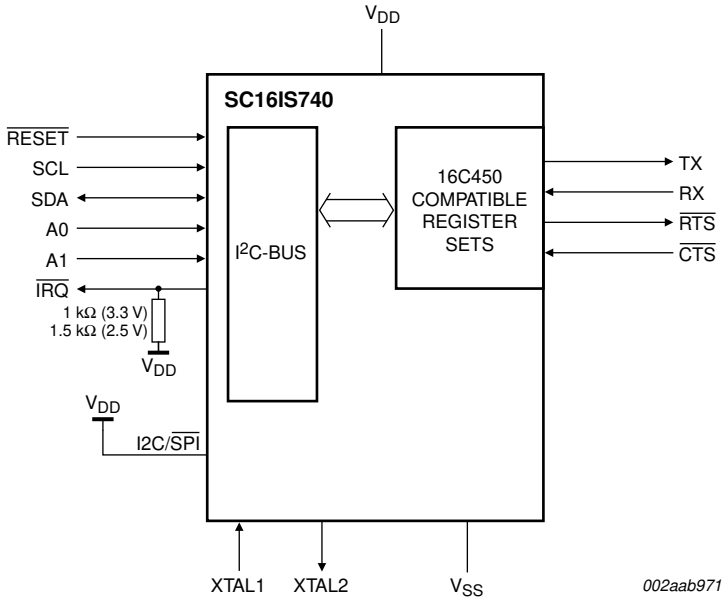


Fig 2. Block diagram of SC16IS740 I²C-bus interface

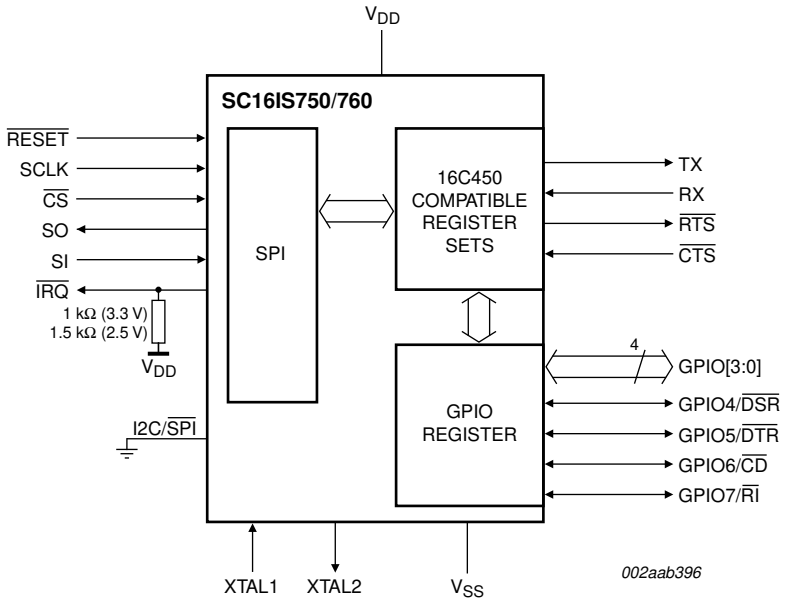


Fig 3. Block diagram of SC16IS750/760 SPI interface

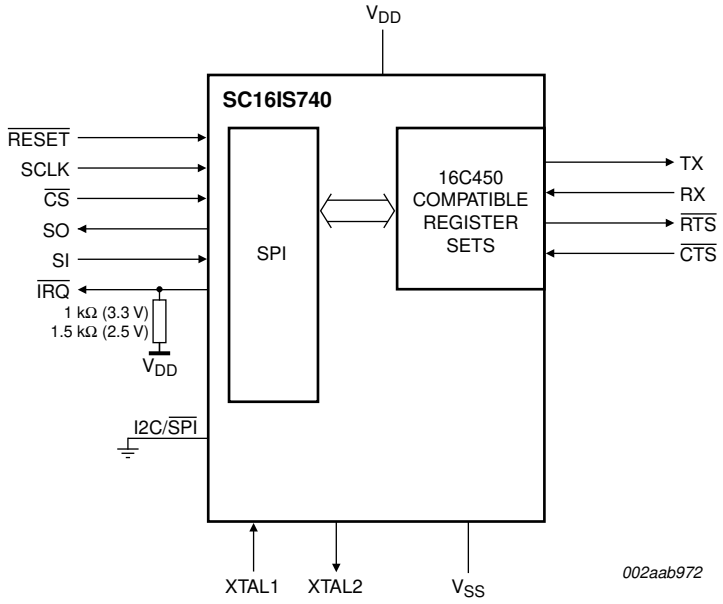


Fig 4. Block diagram of SC16IS740 SPI interface

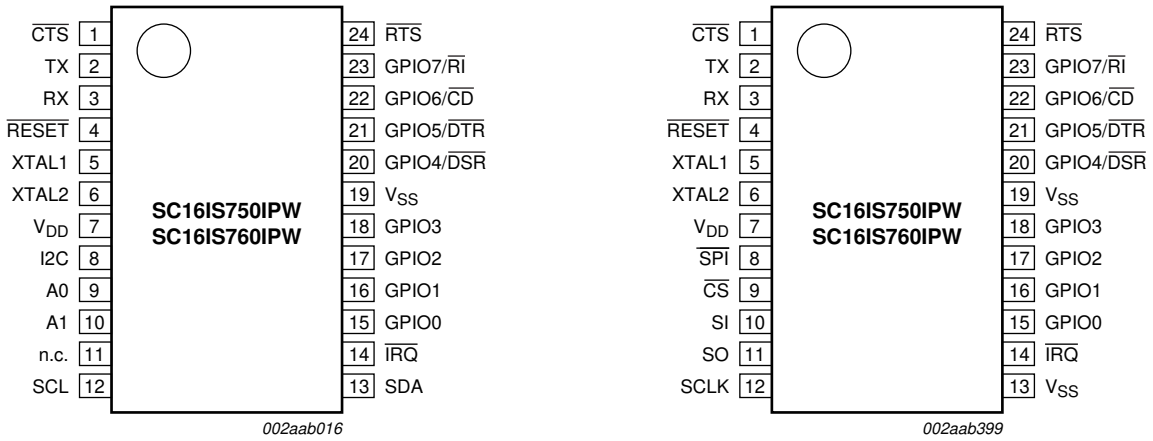
6. Pinning information

6.1 Pinning

a. I²C-bus interface

b. SPI interface

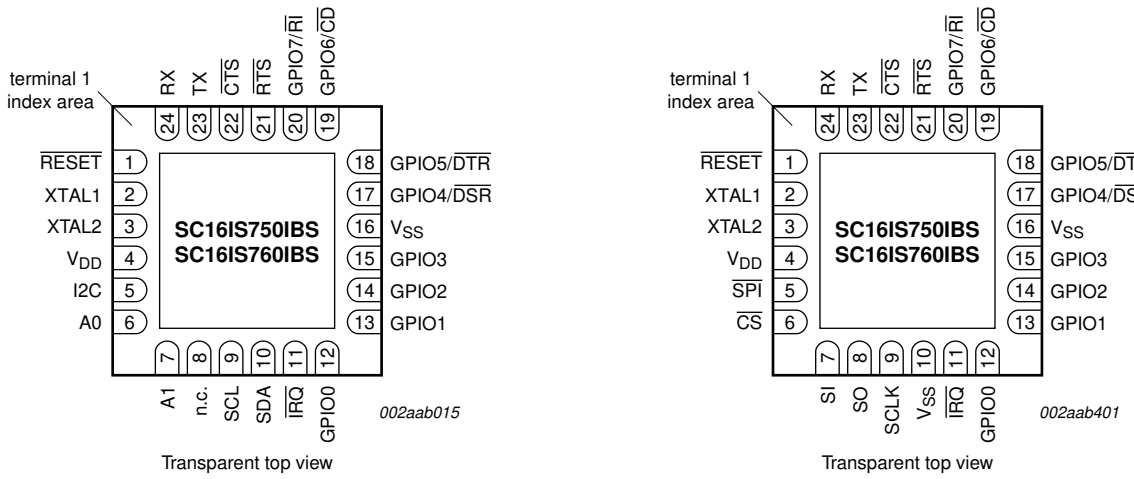
Fig 5. Pin configuration for TSSOP16



a. I²C-bus interface

b. SPI interface

Fig 6. Pin configuration for TSSOP24



a. I²C-bus interface

b. SPI interface

Fig 7. Pin configuration for HVQFN24

6.2 Pin description

Table 2. Pin description

Symbol	Pin			Type	Description
	TSSOP16	TSSOP24	HVQFN24		
$\overline{CTS}$	11	1	22	I	UART clear to send (active LOW). A logic 0 (LOW) on the $\overline{CTS}$ pin indicates the modem or data set is ready to accept transmit data from the SC16IS740/750/760. Status can be tested by reading MSR[4]. This pin only affects the transmit and receive operations when auto $\overline{CTS}$ function is enabled via the Enhanced Feature Register EFR[7] for hardware flow control operation.
TX	12	2	23	O	UART transmitter output. During the local Loopback mode, the TX output pin is disabled and TX data is internally connected to the UART RX input.
RX	13	3	24	I	UART receiver input. During the local Loopback mode, the RX input pin is disabled and TX data is connected to the UART RX input internally.
$\overline{RESET}$	14	4	1	I	device hardware reset (active LOW)[1]
XTAL1	15	5	2	I	Crystal input 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 16). Alternatively, an external clock can be connected to this pin.
XTAL2	16	6	3	O	Crystal output or clock output. (See also XTAL1.) XTAL2 is used as a crystal oscillator output.
V_{DD}	1	7	4	-	power supply
I2C/$\overline{SPI}$	8	8	5	I	I²C-bus or SPI interface select. I²C-bus interface is selected if this pin is at logic HIGH. SPI interface is selected if this pin is at logic LOW.

Table 2. Pin description *...continued*

Symbol	Pin			Type	Description
	TSSOP16	TSSOP24	HVQFN24		
$\overline{CS}$/A0	2	9	6	I	SPI chip select or I²C-bus device address select A0. If SPI configuration is selected by I2C/$\overline{SPI}$ pin, this pin is the SPI chip select pin (Schmitt-trigger, active LOW). If I²C-bus configuration is selected by I2C/$\overline{SPI}$ pin, this pin along with A1 pin allows user to change the device's base address.
SI/A1	3	10	7	I	SPI data input pin or I²C-bus device address select A1. If SPI configuration is selected by I2C/$\overline{SPI}$ pin, this is the SPI data input pin. If I²C-bus configuration is selected by I2C/$\overline{SPI}$ pin, this pin along with A0 pin allows user to change the device's base address. To select the device address, please refer to Table 32.
SO	4	11	8	O	SPI data output pin. If SPI configuration is selected by I2C/$\overline{SPI}$ pin, this is a 3-stateable output pin. If I²C-bus configuration is selected by I2C/$\overline{SPI}$ pin, this pin function is undefined and must be left as n.c. (not connected).
SCL/SCLK	5	12	9	I	I²C-bus or SPI input clock.
SDA	6	13	10	I/O	I²C-bus data input/output, open-drain if I²C-bus configuration is selected by I2C/$\overline{SPI}$ pin. If SPI configuration is selected then this pin is an undefined pin and must be connected to V_{SS}.
$\overline{IRQ}$	7	14	11	O	Interrupt (open-drain, active LOW). Interrupt is enabled when interrupt sources are enabled in the Interrupt Enable Register (IER). Interrupt conditions include: change of state of the input pins, receiver errors, available receiver buffer data, available transmit buffer space, or when a modem status flag is detected. An external resistor (1 kΩ for 3.3 V, 1.5 kΩ for 2.5 V) must be connected between this pin and V_{DD}.
GPIO0	-	15	12	I/O	programmable I/O pin[2]
GPIO1	-	16	13	I/O	programmable I/O pin[2]
GPIO2	-	17	14	I/O	programmable I/O pin[2]
GPIO3	-	18	15	I/O	programmable I/O pin[2]
GPIO4/$\overline{DSR}$	-	20	17	I/O	programmable I/O pin or modem's $\overline{DSR}$ pin[2][3]
GPIO5/$\overline{DTR}$	-	21	18	I/O	programmable I/O pin or modem's $\overline{DTR}$ pin[2][3]
GPIO6/$\overline{CD}$	-	22	19	I/O	programmable I/O pin or modem's $\overline{CD}$ pin[2][3]
GPIO7/$\overline{RI}$	-	23	20	I/O	programmable I/O pin or modem's $\overline{RI}$ pin[2][3]
$\overline{RTS}$	10	24	21	O	UART request to send (active LOW). A logic 0 on the $\overline{RTS}$ 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 this pin is set to a logic 1. This pin only affects the transmit and receive operations when auto $\overline{RTS}$ function is enabled via the Enhanced Feature Register (EFR[6]) for hardware flow control operation.
V_{SS}	9	19	16[4]	-	ground
V_{SS}	-	-	center pad[4]	-	The center pad on the back side of the HVQFN24 package is metallic and should be connected to ground on the printed-circuit board.

[1] See Section 7.4.1 "Hardware reset, Power-On Reset (POR) and software reset"

[2] These pins have an active pull-up resistor at their inputs. See Table 36.

[3] Selectable with IOControl register bit 1.

[4] HVQFN24 package die supply ground is connected to both V_{SS} pins and exposed center pad. V_{SS} pins must be connected to supply ground for proper device operation. For enhanced thermal, electrical, and board level performance, the exposed pad needs to be soldered to the board using a corresponding thermal pad on the board and for proper heat conduction through the board, thermal vias need to be incorporated in the PCB in the thermal pad region.

7. Functional description

The UART will perform serial-to-I²C conversion on data characters received from peripheral devices or modems, and I²C-to-serial conversion on data characters transmitted by the host. The complete status the SC16IS740/750/760 UART can be read at any time during functional operation by the host.

The SC16IS740/750/760 can be placed in an alternate mode (FIFO mode) relieving the host of excessive software overhead by buffering received/transmitted characters. Both the receiver and transmitter FIFOs can store up to 64 characters (including three additional bits of error status per character for the receiver FIFO) and have selectable or programmable trigger levels.

The SC16IS740/750/760 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 $\overline{RTS}$ output and $\overline{CTS}$ 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 ($2^{16} - 1$).

7.1 Trigger levels

The SC16IS740/750/760 provides independently selectable and programmable trigger levels for both receiver and transmitter interrupt generation. After reset, both transmitter and receiver FIFOs are disabled and so, in effect, the trigger level is the default value of one character. The selectable trigger levels are available via the FCR. The programmable trigger levels are available via the TLR. If TLR bits are cleared then selectable trigger level in FCR is used. If TLR bits are not cleared then programmable trigger level in TLR is used.

7.2 Hardware flow control

Hardware flow control is comprised of auto $\overline{CTS}$ and auto $\overline{RTS}$ (see Figure 8). Auto $\overline{CTS}$ and auto $\overline{RTS}$ can be enabled/disabled independently by programming EFR[7:6].

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

Auto $\overline{RTS}$ only activates the $\overline{RTS}$ output when there is enough room in the FIFO to receive data and de-activates the $\overline{RTS}$ output when the RX FIFO is sufficiently full. The halt and resume trigger levels in the TCR determine the levels at which $\overline{RTS}$ is activated/deactivated. If TCR bits are cleared then selectable trigger levels in FCR are used in place of TCR.

If both auto $\overline{CTS}$ and auto $\overline{RTS}$ are enabled, when $\overline{RTS}$ is connected to $\overline{CTS}$, 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.

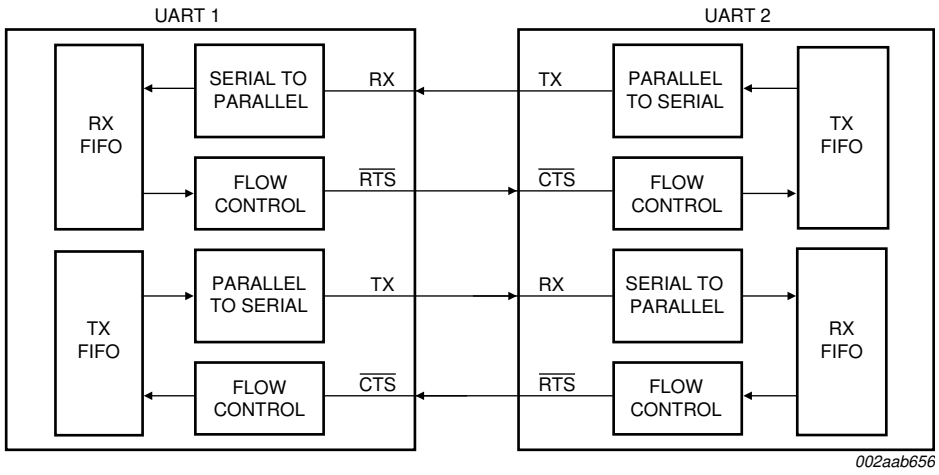
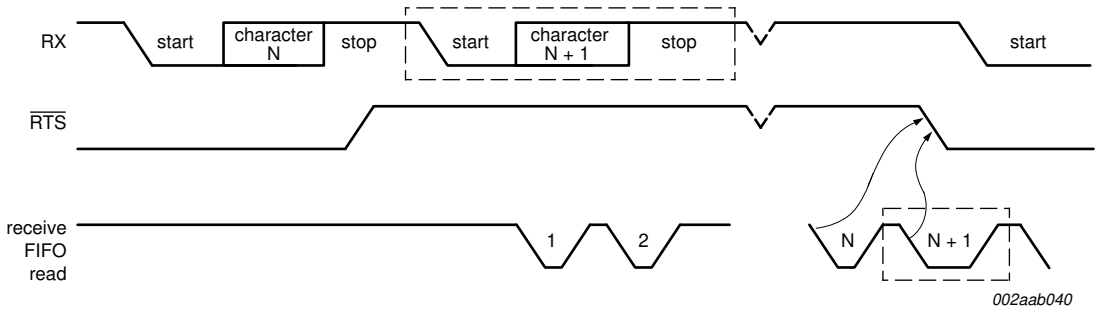


Fig 8. Autoflow control (auto $\overline{RTS}$ and auto $\overline{CTS}$) example

7.2.1 Auto $\overline{RTS}$

Figure 9 shows $\overline{RTS}$ functional timing. The receiver FIFO trigger levels used in auto $\overline{RTS}$ are stored in the TCR or FCR. $\overline{RTS}$ 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, $\overline{RTS}$ is deasserted. The sending device (for example, another UART) may send an additional character after the trigger level is reached (assuming the sending UART has another character to send) because it may not recognize the deassertion of $\overline{RTS}$ until it has begun sending the additional character. 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.



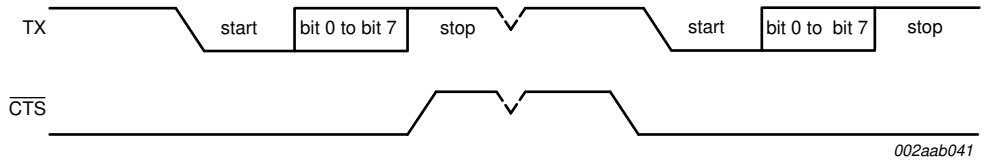
(1) N = receiver FIFO trigger level.

(2) The two blocks in dashed lines cover the case where an additional character is sent, as described in Section 7.2.1

Fig 9. $\overline{RTS}$ functional timing

7.2.2 Auto $\overline{\text{CTS}}$

Figure 10 shows $\overline{\text{CTS}}$ functional timing. 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 character, the transmitter finishes sending the current character, but it does not send the next character.

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

Fig 10. $\overline{\text{CTS}}$ functional timing

7.3 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[3:0])

EFR[3]	EFR[2]	EFR[1]	EFR[0]	TX, RX software flow control
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 and Xon2, Xoff1 and 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 receiver compares Xon1 or Xon2, Xoff1 or Xoff2
0	1	1	1	transmit Xon2, Xoff2 receiver compares Xon1 or Xon2, Xoff1 or Xoff2
1	1	1	1	transmit Xon1 and Xon2, Xoff1 and Xoff2 receiver compares Xon1 and Xon2, Xoff1 and Xoff2
0	0	1	1	no transmit flow control receiver compares Xon1 and Xon2, Xoff1 and Xoff2

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

- **Xon Any function (MCR[5]):** Receiving any character will resume operation 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.

7.3.1 RX

When software flow control operation is enabled, the SC16IS740/750/760 will compare incoming data with Xoff1/Xoff2 programmed characters (in certain cases, Xoff1 and Xoff2 must be received sequentially). 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 $\overline{\text{IRQ}}$ to go LOW.

To resume transmission, an Xon1/Xon2 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.

7.3.2 TX

Xoff1/Xoff2 character is transmitted when the RX FIFO has passed the HALT trigger level programmed in TCR[3:0] or the selectable trigger level in FCR[7:6]

Xon1/Xoff2 character is transmitted when the RX FIFO reaches the RESUME trigger level programmed in TCR[7:4] or RX FIFO falls below the lower selectable trigger level in FCR[7:6].

The transmission of Xoff/Xon(s) follows the exact same protocol as transmission of an ordinary character from the FIFO. This means that even if the word length is set to be 5, 6, or 7 bits, then the 5, 6, or 7 least significant bits of XOFF1/XOFF2 or XON1/XON2 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 11 shows an example of software flow control.

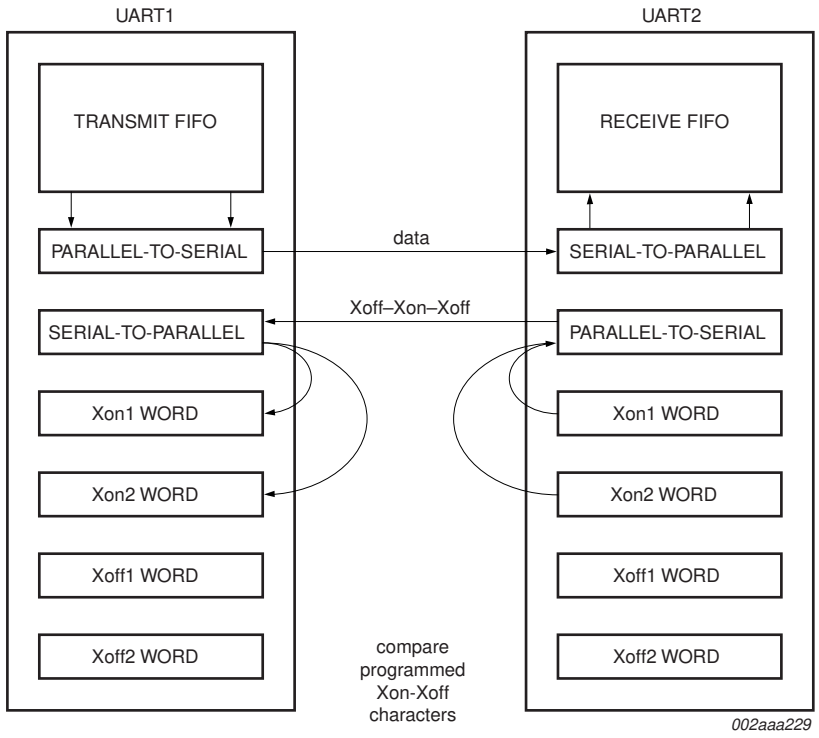


Fig 11. Example of software flow control

7.4 Reset and power-on sequence

7.4.1 Hardware reset, Power-On Reset (POR) and software reset

These three reset methods are identical and will reset the internal registers as indicated in Table 4.

Table 4 summarizes the state of register.

Table 4. Register reset[1]

Register	Reset state
Interrupt Enable Register	all bits cleared
Interrupt Identification Register	bit 0 is set; all other bits cleared
FIFO Control Register	all bits cleared
Line Control Register	reset to 0001 1101 (0x1D)
Modem Control Register	all bits cleared
Line Status Register	bit 5 and bit 6 set; all other bits cleared
Modem Status Register	bits 0:3 cleared; bits 4:7 input signals
Enhanced Feature Register	all bits cleared
Receiver Holding Register	pointer logic cleared
Transmitter Holding Register	pointer logic cleared
Transmission Control Register	all bits cleared.
Trigger Level Register	all bits cleared.

Table 4. Register reset[1]

Register	Reset state
Transmit FIFO level	reset to 0100 0000 (0x40)
Receive FIFO level	all bits cleared
I/O direction[2]	all bits cleared
I/O interrupt enable[2]	all bits cleared
I/O control[3]	all bits cleared
Extra Feature Register	all bits cleared

[1] Registers DLL, DLH, SPR, XON1, XON2, XOFF1, XOFF2 are not reset by the top-level reset signal $\overline{\text{RESET}}$, POR or Software Reset, that is, they hold their initialization values during reset.

[2] This register is not supported in SC16IS740.

[3] Only UART Software Reset bit is supported in this register.

Table 5 summarizes the state of registers after reset.

Table 5. Output signals after reset

Signal	Reset state
TX	HIGH
$\overline{\text{RTS}}$	HIGH
I/Os	inputs
$\overline{\text{IRQ}}$	HIGH by external pull-up

7.4.2 Power-on sequence

After power is applied, the device is reset by the internal POR. The host must wait at least 3 μs before initializing a communication with the device.

An external reset pulse (see Figure 26) can also be used to reset the device after power is applied.

Once the device is reset properly, the host processor can start to communicate with the device. Internal registers can be accessed (read and write), however, at this time the UART transmitter and receiver cannot be used until there is a stable clock at XTAL1 pin. Normally, if an external clock such as a system clock or an external oscillator is used to supply a clock to XTAL1 pin, the clock should be stable at this time. But if a crystal is used, the host processor must wait until the crystal is generating a stable clock before accessing the UART transmitter or receiver.

The crystal's start-up time depends on the crystal being used, V_{CC} ramp-up time and the loading capacitor values. The start-up time can be as long as a few milliseconds.

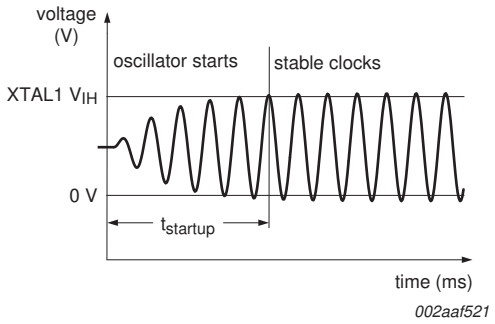


Fig 12. Start-up time

7.5 Interrupts

The SC16IS740/750/760 has interrupt generation and prioritization (seven prioritized levels of interrupts) capability. The interrupt enable registers (IER and IOIntEna) enable each of the seven types of interrupts and the $\overline{IRQ}$ signal in response to an interrupt generation. 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. Summary of interrupt control functions

IIR[5:0]	Priority level	Interrupt type	Interrupt source
00 0001	none	none	none
00 0110	1	receiver line status	OE, FE, PE, or BI errors occur in characters in the RX FIFO
00 1100	2	RX time-out	Stale data in RX FIFO
00 0100	2	RHR interrupt	Receive data ready (FIFO disable) or RX FIFO above trigger level (FIFO enable)
00 0010	3	THR interrupt	Transmit FIFO empty (FIFO disable) or TX FIFO passes above trigger level (FIFO enable)
00 0000	4	modem status[1]	Change of state of modem input pins
11 0000	5	I/O pins[1]	Input pins change of state
01 0000	6	Xoff interrupt	Receive Xoff character(s)/ special character
10 0000	7	$\overline{CTS}$, $\overline{RTS}$	$\overline{RTS}$ pin or $\overline{CTS}$ pin change state from active (LOW) to inactive (HIGH)

[1] Available only on SC16IS750/SC16IS760.

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 IIR.

7.5.1 Interrupt mode operation

In Interrupt mode (if any bit of IER[3:0] is 1) the host is informed of the status of the receiver and transmitter by an interrupt signal, $\overline{IRQ}$. Therefore, it is not necessary to continuously poll the Line Status Register (LSR) to see if any interrupt needs to be serviced. Figure 13 shows Interrupt mode operation.

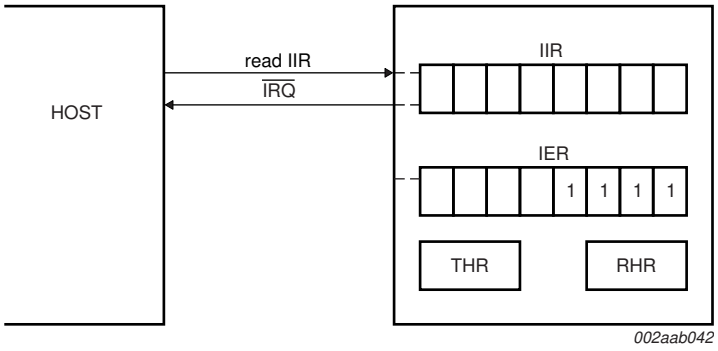


Fig 13. Interrupt mode operation

7.5.2 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 14 shows FIFO Polled mode operation.

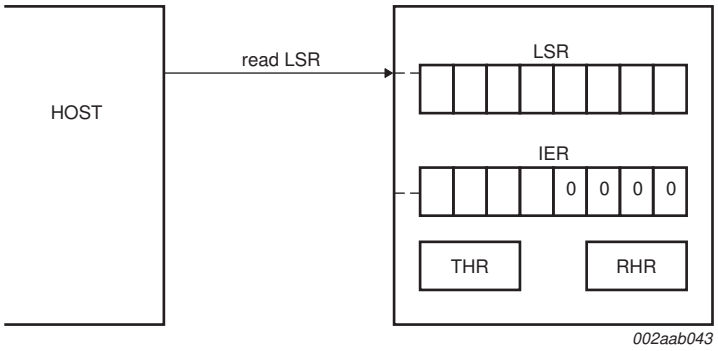


Fig 14. FIFO Polled mode operation

7.6 Sleep mode

Sleep mode is an enhanced feature of the SC16IS740/750/760 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 7.7 “Break and time-out conditions”).
- The TX FIFO and TX shift register are empty.
- There are no interrupts pending except THR.

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

In Sleep mode, the clock to the UART is 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.

Wake-up by serial data on RX input pin is supported in UART mode but not in IrDA mode in Rev. C and Rev. D of the device. Refer to application note *AN10964, “How to wake up SC16IS/740/750/760 in IrDA mode”* for a software procedure to wake up the device by receiving data in the IrDA mode.

Wake-up by serial data on RX input pin is supported in both UART mode and IrDA mode in Rev. E of the device.

The device will not wake up by GPIO pin transition, but GPIO pin input state can be read, and GPIO interrupt is working normally during Sleep mode.

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.

7.7 Break and time-out conditions

When the UART receives a number of characters and these data are not enough to set off the receive interrupt (because they do not reach the receive trigger level), the UART will generate a time-out interrupt instead, 4 character times after the last character is received. The time-out counter will be reset at the center of each stop bit received or each time the receive FIFO is read.

A break condition is detected when the RX pin is pulled LOW for a duration longer than the time it takes to send a complete character plus Start, Stop and Parity bits. A break condition can be sent by setting LCR[6]. When this happens the TX pin will be pulled LOW until LSR[6] is cleared by the software.

7.8 Programmable baud rate generator

The SC16IS740/750/760 UART contains a programmable baud rate 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 15. The output frequency of the baud rate generator is 16 × the baud rate. The formula for the divisor is given in Equation 1:

$$divisor = \frac{\left(\frac{XTAL1\ crystal\ input\ frequency}{prescaler}\right)}{desired\ baud\ rate \times 16} \quad (1)$$

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 15 shows the internal prescaler and baud rate generator circuitry.

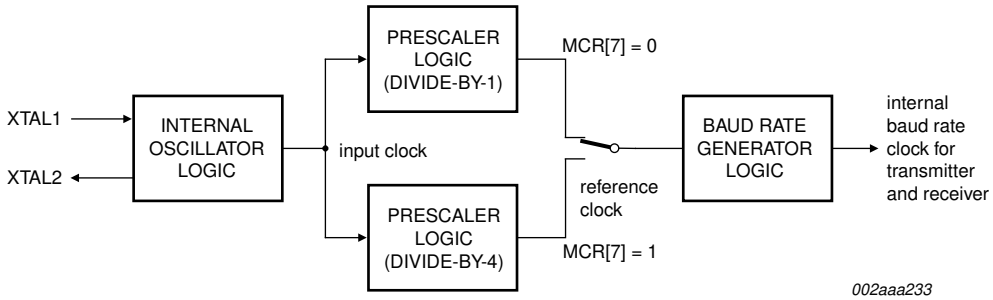


Fig 15. Prescaler and baud rate generator block diagram

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. The crystal's frequency tolerance should be selected as such to keep the baud rate error to be below 1 % for reliable operation with other UARTs. Crystals with ±100 ppm is generally recommended.

Figure 16 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	0
75	1536	0
110	1047	0.026
134.5	857	0.058
150	768	0
300	384	0
600	192	0
1200	96	0
1800	64	0
2000	58	0.69
2400	48	0
3600	32	0
4800	24	0
7200	16	0
9600	12	0
19200	6	0
38400	3	0
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	0
75	2560	0
110	1745	0.026
134.5	1428	0.034
150	1280	0
300	640	0
600	320	0
1200	160	0
1800	107	0.312
2000	96	0
2400	80	0
3600	53	0.628
4800	40	0
7200	27	1.23
9600	20	0
19200	10	0
38400	5	0

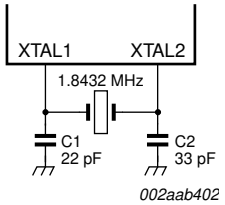


Fig 16. Crystal oscillator circuit reference

8. Register descriptions

The programming combinations for register selection are shown in Table 9.

Table 9. Register map - read/write properties

Register name	Read mode	Write mode
RHR/THR	Receive Holding Register (RHR)	Transmit Holding Register (THR)
IER	Interrupt Enable Register (IER)	Interrupt Enable Register
IIR/FCR	Interrupt Identification Register (IIR)	FIFO Control Register (FCR)
LCR	Line Control Register (LCR)	Line Control Register
MCR	Modem Control Register (MCR)[1]	Modem Control Register[1]
LSR	Line Status Register (LSR)	n/a
MSR	Modem Status Register (MSR)	n/a
SPR	Scratchpad Register (SPR)	Scratchpad Register
TCR	Transmission Control Register (TCR)[2]	Transmission Control Register[2]
TLR	Trigger Level Register (TLR)[2]	Trigger Level Register[2]
TXLVL	Transmit FIFO Level Register	n/a
RXLVL	Receive FIFO Level Register	n/a
IODir[3]	I/O pin Direction Register	I/O pin Direction Register
IOState[3]	I/O pin States Register	n/a
IOIntEna[3]	I/O Interrupt Enable Register	I/O Interrupt Enable Register
IOControl[3]	I/O pins Control Register	I/O pins Control Register
EFCR	Extra Features Register	Extra Features Register
DLL	divisor latch LSB (DLL)[4]	divisor latch LSB[4]
DLH	divisor latch MSB (DLH)[4]	divisor latch MSB[4]
EFR	Enhanced Feature Register (EFR)[5]	Enhanced Feature Register[5]
XON1	Xon1 word[5]	Xon1 word[5]
XON2	Xon2 word[5]	Xon2 word[5]
XOFF1	Xoff1 word[5]	Xoff1 word[5]
XOFF2	Xoff2 word[5]	Xoff2 word[5]

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

[2] Accessible only when ERF[4] = 1 and MCR[2] = 1, that is, EFR[4] and MCR[2] are read/write enables.

[3] Available only on SC16IS750/SC16IS760.

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

[5] Accessible only when LCR is set to 1011 1111b (0xBF).

Table 10. SC16IS740/750/760 internal registers

Register address	Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R/W
General register set[1]										
0x00	RHR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R
0x00	THR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	W
0x01	IER	$\overline{\text{CTS}}$ interrupt enable[2]	$\overline{\text{RTS}}$ interrupt enable[2]	$\overline{\text{Xoff}}$[2]	Sleep mode[2]	modem status interrupt	receive line status interrupt	THR empty interrupt	RX data available interrupt	R/W
0x02	FCR	RX trigger level (MSB)	RX trigger level (LSB)	TX trigger level (MSB)[2]	TX trigger level (LSB)[2]	reserved[3]	TX FIFO reset[4]	RX FIFO reset[4]	FIFO enable	W
0x02	IIR[5]	FIFO enable	FIFO enable	interrupt priority bit 4[2]	interrupt priority bit 3[2]	interrupt priority bit 2	interrupt priority bit 1	interrupt priority bit 0	interrupt status	R
0x03	LCR	Divisor Latch Enable	set break	set parity	even parity	parity enable	stop bit	word length bit 1	word length bit 0	R/W
0x04	MCR	clock divisor[2]	IrDA mode enable[2]	Xon Any[2]	loopback enable	reserved[3]	TCR and TLR enable[2]	$\overline{\text{RTS}}$	$\overline{\text{DTR}}$/(IO5)[6]	R/W
0x05	LSR	FIFO data error	THR and TSR empty	THR empty	break interrupt	framing error	parity error	overrun error	data in receiver	R
0x06	MSR	CD/(IO6)[6]	RI/(IO7)[6]	DSR/ (IO4)[6]	CTS	ΔCD/ (IO6)[6]	ΔRI/(IO7)[6]	ΔDSR/ (IO4)[6]	ΔCTS	R
0x07	SPR	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x06	TCR[7]	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x07	TLR[7]	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x08	TXLVL	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R
0x09	RXLVL	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R
0x0A	IODir[6]	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x0B	IOState[6]	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x0C	IOIntEna[6]	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x0D	reserved[3]	reserved[3]	reserved[3]	reserved[3]	reserved[3]	reserved[3]	reserved[3]	reserved[3]	reserved[3]	
0x0E	IOControl[6]	reserved[3]	reserved[3]	reserved[3]	reserved[3]	UART software reset[8]	reserved[3]	I/O[7:4] or $\overline{\text{RI}}$, $\overline{\text{CD}}$, $\overline{\text{DTR}}$, $\overline{\text{DSR}}$	latch	R/W
0x0F	EFCR	IrDA mode (slow/ fast)[9]	reserved[3]	auto RS-485 RTS output inversion	auto RS-485 RTS direction control	reserved[3]	transmitter disable	receiver disable	9-bit mode enable	R/W

Table 10. SC16IS740/750/760 internal registers *…continued*

Register address	Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R/W
Special register set[10]										
0x00	DLL	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x01	DLH	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
Enhanced register set[11]										
0x02	EFR	Auto $\overline{CTS}$	Auto $\overline{RTS}$	special character detect	enable enhanced functions	software flow control bit 3	software flow control bit 2	software flow control bit 1	software flow control bit 0	R/W
0x04	XON1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x05	XON2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x06	XOFF1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0x07	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] These bits in can only be modified if register bit EFR[4] is enabled.

[3] These bits are reserved and should be set to 0.

[4] After Receive FIFO or Transmit FIFO reset (through FCR[1:0]), the user must wait at least $2 \times T_{clk}$ of XTAL1 before reading or writing data to RHR and THR, respectively.

[5] Burst reads on the serial interface (that is, reading multiple elements on the I²C-bus without a STOP or repeated START condition, or reading multiple elements on the SPI bus without de-asserting the $\overline{CS}$ pin), should not be performed on the IIR register.

[6] Only available on the SC16IS750/SC16IS760.

[7] These registers are accessible only when MCR[2] = 1 and EFR[4] = 1.

[8] Device returns NACK on I²C-bus when this bit is written.

[9] IrDA mode slow/fast for SC16IS760, slow only for SC16IS750.

[10] The special register set is accessible only when LCR[7] = 1 and not 0xBF.

[11] Enhanced Feature Registers are only accessible when LCR = 0xBF.

8.1 Receive 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 pin. 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.

8.2 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 pin. If the FIFO is disabled, the FIFO is still used to store the byte. Characters are lost if overflow occurs.

8.3 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. 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)	RX 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]	reserved
2	FCR[2][1]	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 level 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][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 level 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

[1] FIFO reset requires at least two XTAL1 clocks, therefore, they cannot be reset without the presence of the XTAL1 clock.

8.4 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 to 1 stop bit (word length = 5, 6, 7, 8) 1 to 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; see Table 15.

Table 13. LCR[5] parity selection

LCR[5]	LCR[4]	LCR[3]	Parity selection
X	X	0	no parity
0	0	1	odd parity
0	1	1	even parity
1	0	1	forced parity ‘1’
1	1	1	forced parity ‘0’

Table 14. LCR[2] stop bit length

LCR[2]	Word length (bits)	Stop bit length (bit times)
0	5, 6, 7, 8	1
1	5	$1\frac{1}{2}$
1	6, 7, 8	2

Table 15. LCR[1:0] word length

LCR[1]	LCR[0]	Word length (bits)
0	0	5
0	1	6
1	0	7
1	1	8