SC16IS850L

Single UART with I²C-bus/SPI interface, 128 bytes of transmit and receive FIFOs, IrDA SIR built-in support

Rev. 2 — 18 July 2012 Product data sheet

1. General description

The SC16IS850L 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 device comes in very small HVQFN24 and TSSOP24 packages, which makes it ideally suitable for handheld, battery operated applications. It also enables seamless protocol conversion from I²C-bus or SPI to and RS-232/RS-485 and are fully bidirectional.

The SC16IS850L supports SPI clock speeds up to 12 Mbit/s, and it supports IrDA SIR up to 115.2 kbit/s. Its internal register set is backward-compatible with the widely used and widely popular 16C850. This allows the software to be easily written or ported from another platform.

The SC16IS850L 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
- 1.8 V operation
- Industrial temperature range: −40 °C to +85 °C
- 128 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 RTS/CTS
- Auto software flow control with programmable Xon/Xoff characters
- Single or double Xon/Xoff characters
- Automatic RS-485 support (automatic slave address detection)
- RS-485 driver direction control via $\overline{\text{RTS}}$ signal
- RS-485 driver direction control inversion
- Built-in IrDA encoder and decoder interface
- Supports IrDA SIR with speeds up to 115.2 kbit/s
- 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 ½, or 2 stop bits
- Line break generation and detection
- Internal Loopback mode
- Sleep current less than 5 μA at 1.8 V
- Industrial and commercial temperature ranges
- Available in HVQFN24 and TSSOP24 packages

2.2 I²C-bus features

- 400 kbit/s maximum speed
- Compliant with I²C-bus Fast-mode (Fm) speed
- Slave mode only

2.3 SPI features

- Supports 12 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

4. Ordering information

Table 1. Ordering information

Type number	Package		
	Name	Description	Version
SC16IS850LIBS	HVQFN24	plastic thermal enhanced very thin quad flat package; no leads; 24 terminals; body 4 × 4 × 0.85 mm	SOT616-3
SC16IS850LIPW	TSSOP24	plastic thin shrink small outline package; 24 leads; body width 4.4 mm	SOT355-1

5. Block diagram

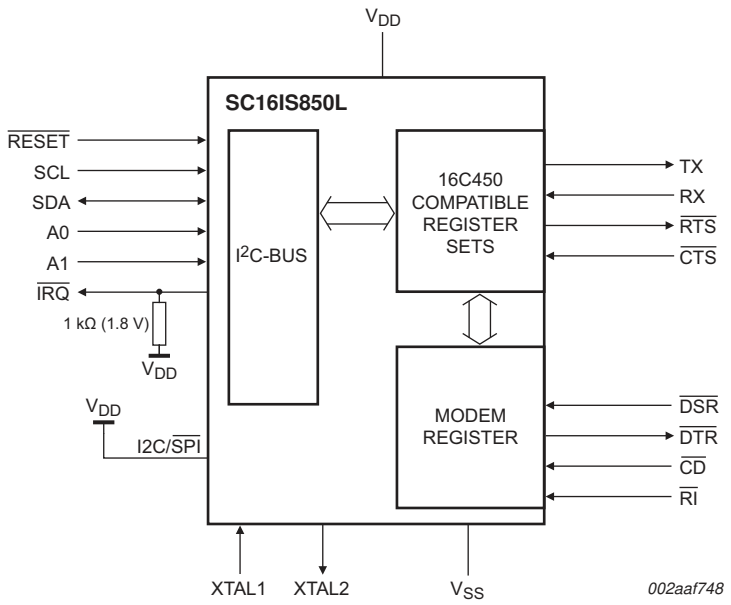


Fig 1. Block diagram of SC16IS850L I²C-bus interface

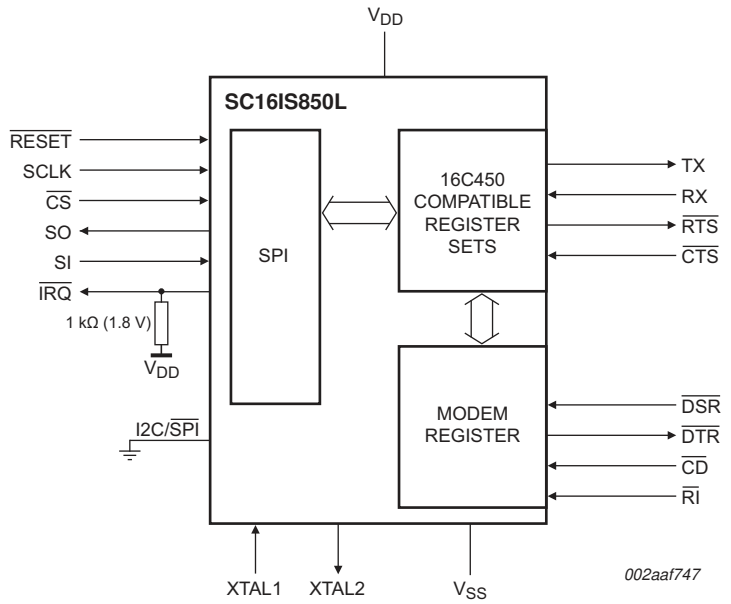


Fig 2. Block diagram of SC16IS850L SPI interface

6. Pinning information

6.1 Pinning

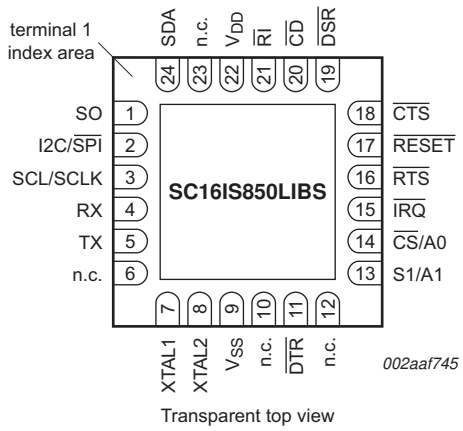


Fig 3. Pin configuration for HVQFN24

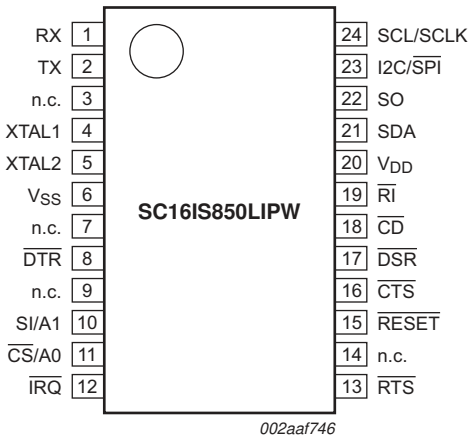


Fig 4. Pin configuration for TSSOP24

6.2 Pin description

Table 2. Pin description

Symbol	Pin		Type	Description
	HVQFN24	TSSOP24		
$\overline{CTS}$	18	16	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 SC16IS850L. Status can be tested by reading MSR[4]. This pin only affect 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	5	2	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	4	1	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.
$\overline{RESET}$	17	15	I	Device hardware reset (active LOW).
XTAL1	7	4	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 6). Alternatively, an external clock can be connected to this pin.
XTAL2	8	5	O	Crystal output or clock output. (See also XTAL1.) XTAL2 is used as a crystal oscillator output.
V_{DD}	22	20	-	Power supply.

Table 2. Pin description *...continued*

Symbol	Pin		Type	Description
	HVQFN24	TSSOP24		
V_{SS}	9[1]	6	-	Power ground.
I2C/$\overline{SPI}$	2	23	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. This pin has an internal pull-up resistor, and can be left unconnected if I²C-bus mode is selected.
$\overline{CS}$/A0	14	11	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. For I²C-bus slave address configuration, please refer to Table 33.
SI/A1	13	10	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. For I²C-bus slave address configuration, please refer to Table 33
SO	1	22	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	3	24	I	I²C-bus or SPI input clock.
SDA	24	21	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}.
$\overline{IRQ}$	15	12	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 10 kΩ resistor must be connected between this pin and V_{DD}.

Table 2. Pin description *...continued*

Symbol	Pin		Type	Description
	HVQFN24	TSSOP24		
$\overline{RTS}$	16	13	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 reset, this pin is set to a logic 1. This pin only affect the transmit and receive operations when Auto-$\overline{RTS}$ function is enabled via the Enhanced Feature Register (EFR[6]) for hardware flow control operation.
$\overline{DSR}$	19	17	I	Data set ready. DSR is a modem status signal. Its condition can be checked by reading MSR[5]. MSR[1] indicates DSR has changed levels since the last read from the modem status register. If the modem status interrupt is enabled when DSR changes levels, an interrupt is generated.
$\overline{CD}$	20	18	I	Data carrier detect. $\overline{CD}$ is a modem status signal. Its condition can be checked by MSR[7]. MSR[3] indicates that $\overline{CD}$ has changed states since the last read from the modem status register. If the modem status interrupt is enabled when $\overline{CD}$ changes levels, an interrupt is generated.
$\overline{RI}$	21	19	I	Ring indicator. $\overline{RI}$ is a modem status signal. Its condition can be checked by reading MSR[6]. MSR[2] indicates that $\overline{RI}$ has transitioned from a LOW to a HIGH level since the last read from the modem status register. If the modem status interrupt is enabled when this transition occurs, an interrupt is generated.
$\overline{DTR}$	11	8	O	Data terminal ready. When active (LOW), $\overline{DTR}$ informs a modem or data set that the UART is ready to establish communication. $\overline{DTR}$ is placed in the active level by setting the DTR bit of the modem control register. $\overline{DTR}$ is placed in the inactive level either as a result of a Master Reset, during Loopback mode operation, or clearing the DTR bit.
n.c.	6, 10, 12, 23	3, 7, 9, 14	-	Not connected; these pins should be left open.

[1] HVQFN24 package die supply ground is connected to both V_{SS} pin and exposed center pad. V_{SS} pin 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 printed-circuit board in the thermal pad region.

7. Functional description

Please refer to Figure 1 “Block diagram of SC16IS850L I²C-bus interface”.

The SC16IS850L provides serial asynchronous receive data synchronization, serial-to-serial data conversions for both the transmitter and receiver sections. Synchronization for the serial data stream is accomplished by adding start and stop bits to the transmit data to form a data character (character orientated protocol). Data integrity is ensured by attaching a parity bit to the data character. The parity bit is checked by the receiver for any transmission bit errors. The electronic circuitry to provide all these functions is fairly complex, especially when manufactured on a single integrated silicon chip. The status of the UART can be read at any time during functional operation by the host through either I²C-bus or SPI interface.

The SC16IS850L represents such an integration with greatly enhanced features. The SC16IS850L is fabricated with an advanced CMOS process. The SC16IS850L provides a single UART capability with 128 bytes of transmit and receive FIFO memory, instead of 64 bytes for the SC16IS750. The SC16IS850L is designed to work with high speed modems and shared network environments that require fast data processing time. Increased performance is realized in the SC16IS850L by transmit and receive FIFOs. This allows the external processor to handle more networking tasks within a given time. In addition, the four selectable receive and transmit FIFO trigger interrupt levels are provided in 16C650 mode, or 128 programmable levels are provided in the extended mode for maximum data throughput performance especially when operating in a multi-channel environment (see ”Section 7.1 “Extended mode (128-byte FIFO)”). The FIFO memory greatly reduces the bandwidth requirement of the external controlling CPU and increases performance. Sleep mode function in the SC16IS850L allows the UART to be placed under low power mode when the serial data input line, RX, is idle, TX FIFO and Transmit Shift Registers are empty, and there is no interrupt pending except THR. The UART is capable of operation up to 5 Mbit/s with an external 80 MHz clock. With a crystal, the SC16IS850L is capable of operation up to 1.5 Mbit/s.

The rich feature set of the SC16IS850L is available through internal registers. These features are: selectable and programmable receive and transmit FIFO trigger levels, selectable TX and RX baud rates, and modem interface controls, and are all standard features. Following a power-on reset, an external reset, or a software reset, the SC16IS850L is software compatible with the previous generation, SC16C550B, and SC16C650B.

The SC16IS850L 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 Extended mode (128-byte FIFO)

The device is in the extended mode when any of these four registers contains any value other than 0: FLWCNTH, FLWCNTL, TXINTLVL, RXINTLVL.

7.2 Internal registers

The SC16IS850L provides a set of 25 internal registers for monitoring and controlling the functions of the UART. These registers are shown in Table 3.

Table 3. Internal registers decoding

A2	A1	A0	Read mode	Write mode
General register set (THR/RHR, IER/ISR, MCR/MSR, FCR, LCR/LSR, EFCR, SPR)[1]				
0	0	0	Receive Holding Register	Transmit Holding Register
0	0	1	Interrupt Enable Register	Interrupt Enable Register
0	1	0	Interrupt Status Register	FIFO Control Register
0	1	1	Line Control Register	Line Control Register
1	0	0	Modem Control Register	Modem Control Register
1	0	1	Line Status Register	Extra Feature Control Register (EFCR)
1	1	0	Modem Status Register	n/a
1	1	1	Scratchpad Register	Scratchpad Register
Baud rate register set (DLL/DLM)[2]				
0	0	0	LSB of Divisor Latch	LSB of Divisor Latch
0	0	1	MSB of Divisor Latch	MSB of Divisor Latch
Second special register set (TXLVLCNT/RXLVLCNT)[3]				
0	1	1	Transmit FIFO Level Count	n/a
1	0	0	Receive FIFO Level Count	n/a
Enhanced feature register set (EFR, Xon1/Xon2, Xoff1/Xoff2)[4]				
0	1	0	Enhanced Feature Register	Enhanced Feature Register
1	0	0	Xon1 word	Xon1 word
1	0	1	Xon2 word	Xon2 word
1	1	0	Xoff1 word	Xoff1 word
1	1	1	Xoff2 word	Xoff2 word
First extra feature register set (TXINTLVL/RXINTLVL, FLWCNTH/FLWCNTL)[5]				
0	1	0	Transmit FIFO Interrupt Level	Transmit FIFO Interrupt Level
1	0	0	Receive FIFO Interrupt Level	Receive FIFO Interrupt Level
1	1	0	Flow Control Count High	Flow Control Count High
1	1	1	Flow Control Count Low	Flow Control Count Low
Second extra feature register set (CLKPRES, RS485TIME, AFCR2, AFCR1)[6]				
0	1	0	Clock Prescaler	Clock Prescaler
1	0	0	RS-485 turn-around Timer	RS-485 turn-around Timer
1	1	0	Additional Feature Control Register 2	Additional Feature Control Register 2
1	1	1	Additional Feature Control Register 1	Additional Feature Control Register 1

[1] These registers are accessible only when LCR[7] is a logic 0.

[2] These registers are accessible only when LCR[7] is a logic 1.

[3] Second Special registers are accessible only when EFCR[0] = 1.

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

[5] First Extra Feature Registers are only accessible when EFCR[2:1] = 01b.

[6] Second Extra Feature Registers are only accessible when EFCR[2:1] = 10b.

7.3 FIFO operation

7.3.1 32-byte FIFO mode

When all four of these registers (TXINTLVL, RXINTLVL, FLWCNTH, FLWCNTL) in the 'first extra feature register set' are empty (0x00) the transmit and receive trigger levels are set by FCR[7:4]. In this mode the transmit and receive trigger levels are backward compatible to the SC16C650B (see Table 4), and the FIFO sizes are 32 entries. The transmit and receive data FIFOs are enabled by the FIFO Control Register bit 0 (FCR[0]). It should be noted that the user can set the transmit trigger levels by writing to the FCR, but activation will not take place until EFR[4] is set to a logic 1. The receiver FIFO section includes a time-out function to ensure data is delivered to the external CPU (see Section 7.7). Please refer to Table 9 and Table 10 for the setting of FCR[7:4].

Table 4. Interrupt trigger level and flow control mechanism

FCR[7:6]	FCR[5:4]	INT pin activation		Negate $\overline{RTS}$ or send Xoff	Assert $\overline{RTS}$ or send Xon
		RX	TX		
00	00	8	16	8	0
01	01	16	8	16	7
10	10	24	24	24	15
11	11	28	30	28	23

7.3.2 128-byte FIFO mode

When either TXINTLVL, RXINTLVL, FLWCNTH or FLWCNTL in the 'first extra feature register set' contains any value other than 0x00, the transmit and receive trigger levels are set by TXINTLVL and RXINTLVL registers. TXINTLVL sets the trigger levels for the transmit FIFO, and the transmit trigger levels can be set to any value between 1 and 128 with granularity of 1. RXINTLVL sets the trigger levels for the receive FIFO, the receive trigger levels can be set to any value between 1 and 128 with granularity of 1.

When the effective FIFO size changes (that is, when FCR[0] toggles or when the combined content of TXINTLVL, RXINTLVL, FLWCNTH and FLWCNTL changes between equal and unequal to 0x00), both RX FIFO and TX FIFO will be reset (data in the FIFO will be lost).

7.4 Hardware flow control

When automatic hardware flow control is enabled, the SC16IS850L monitors the $\overline{CTS}$ pin for a remote buffer overflow indication and controls the $\overline{RTS}$ pin for local buffer overflows. Automatic hardware flow control is selected by setting EFR[6] (RTS) and EFR[7] (CTS) to a logic 1. If $\overline{CTS}$ transitions from a logic 0 to a logic 1 indicating a flow control request, ISR[5] will be set to a logic 1 (if enabled via IER[7:6]), and the SC16IS850L will suspend TX transmissions as soon as the stop bit of the character in process is shifted out. Transmission is resumed after the $\overline{CTS}$ input returns to a logic 0, indicating more data may be sent.

When AFCR1[2] is set to logic 1 then the function of $\overline{CTS}$ pin is mapped to the $\overline{DSR}$ pin, and the function of $\overline{RTS}$ is mapped to $\overline{DTR}$ pin. $\overline{DSR}$ and $\overline{DTR}$ pins will behave as described above for $\overline{CTS}$ and $\overline{RTS}$.

With the automatic hardware flow control function enabled, an interrupt is generated when the receive FIFO reaches the programmed trigger level. The $\overline{\text{RTS}}$ (or $\overline{\text{DTR}}$) pin will not be forced to a logic 1 ($\overline{\text{RTS}}$ off), until the receive FIFO reaches the next trigger level. However, the $\overline{\text{RTS}}$ (or $\overline{\text{DTR}}$) pin will return to a logic 0 after the receive buffer (FIFO) is unloaded to the next trigger level below the programmed trigger level. Under the above described conditions, the SC16IS850L will continue to accept data until the receive FIFO is full.

When the TXINTLVL, RXINTLVL, FLWCNTH and FLWCNTL in the 'first extra feature register set' are all zeroes, the hardware and software flow control trigger levels are set by FCR[7:4]; see Table 4.

When the TXINTLVL, RXINTLVL, FLWCNTH or FLWCNTL in the 'first extra feature register set' contain any value other than 0x00, the hardware and software flow control trigger levels are set by FLWCNTH and FLWCNTL. The content in FLWCNTH determines how many bytes are in the receive FIFO before $\overline{\text{RTS}}$ (or $\overline{\text{DTR}}$) is de-asserted or Xoff is sent. The content in FLWCNTL determines how many bytes are in the receive FIFO before $\overline{\text{RTS}}$ (or $\overline{\text{DTR}}$) is asserted, or Xon is sent.

In 128-byte FIFO mode, hardware and software flow control trigger levels can be set to any value between 1 and 128 in granularity of 1. The value of FLWCNTH should always be greater than FLWCNTL. The UART does not check for this condition automatically, and if this condition is not met, spurious operation of the device might occur. When using FLWCNTH and FLWCNTL, these registers must be initialized to proper values before hardware or software flow control is enabled via the EFR register.

7.5 Software flow control

When software flow control is enabled, the SC16IS850L compares one or two sequentially received data characters with the programmed Xon or Xoff character value(s). If the received character(s) match the programmed Xoff values, the SC16IS850L will halt transmission (TX) as soon as the current character(s) has completed transmission. When a match occurs, ISR bit 4 will be set (if enabled via IER[5]) and the interrupt output pin (if receive interrupt is enabled) will be activated. Following a suspension due to a match of the Xoff characters' values, the SC16IS850L will monitor the receive data stream for a match to the Xon1/Xon2 character value(s). If a match is found, the SC16IS850L will resume operation and clear the flags (ISR[4]).

Reset initially sets the contents of the Xon/Xoff 8-bit flow control registers to a logic 0. Following reset, the user can write any Xon/Xoff value desired for software flow control. Different conditions can be set to detect Xon/Xoff characters and suspend/resume transmissions (see Table 22). When double 8-bit Xon/Xoff characters are selected, the SC16IS850L compares two consecutive receive characters with two software flow control 8-bit values (Xon1, Xon2, Xoff1, Xoff2) and controls TX transmissions accordingly. Under the above described flow control mechanisms, flow control characters are not placed (stacked) in the receive FIFO. When using software flow control, the Xon/Xoff characters cannot be used for data transfer.

In the event that the receive buffer is overfilling, the SC16IS850L automatically sends an Xoff character (when enabled) via the serial TX output to the remote UART. The SC16IS850L sends the Xoff1/Xoff2 characters as soon as the number of received data in

the receive FIFO passes the programmed trigger level. To clear this condition, the SC16IS850L will transmit the programmed Xon1/Xon2 characters as soon as the number of characters in the receive FIFO drops below the programmed trigger level.

7.6 Special character detect

A special character detect feature is provided to detect an 8-bit character when EFR[5] is set. When an 8-bit character is detected, it will be placed on the user-accessible data stack along with normal incoming RX data. This condition is selected in conjunction with EFR[3:0] (see Table 22). Note that software flow control should be turned off when using this special mode by setting EFR[3:0] to all zeroes.

The SC16IS850L compares each incoming receive character with Xoff2 data. If a match occurs, the received data will be transferred to the FIFO, and ISR[4] will be set to indicate detection of a special character. Although Table 6 “SC16IS850L internal registers” shows Xon1, Xon2, Xoff1, Xoff2 with eight bits of character information, the actual number of bits is dependent on the programmed word length. Line Control Register bits LCR[1:0] define the number of character bits, that is, either 5 bits, 6 bits, 7 bits or 8 bits. The word length selected by LCR[1:0] also determines the number of bits that will be used for the special character comparison. Bit 0 in Xon1, Xon2, Xoff1, Xoff2 corresponds with the LSB bit for the received character.

7.7 Interrupt priority and time-out interrupts

The interrupts are enabled by IER[7:0]. Care must be taken when handling these interrupts. Following a reset, if Interrupt Enable Register (IER) bit 1 = 1, the SC16IS850L will issue a Transmit Holding Register interrupt. This interrupt must be serviced prior to continuing operations. The ISR indicates the current singular highest priority interrupt only. A condition can exist where a higher priority interrupt masks the lower priority interrupt(s) (see Table 11). Only after servicing the higher pending interrupt will the lower priority interrupt(s) be reflected in the status register. Servicing the interrupt without investigating further interrupt conditions can result in data errors.

Receive Data Ready and Receive Time-Out have the same interrupt priority (when enabled by IER[0]), and it is important to serve these interrupts correctly. The receiver issues an interrupt after the number of characters have reached the programmed trigger level. In this case, the SC16IS850L FIFO may hold more characters than the programmed trigger level. Following the removal of a data byte, the user should re-check LSR[0] to see if there are any additional characters. A Receive Time-Out will not occur if the receive FIFO is empty. The time-out counter is reset at the center of each stop bit received or each time the Receive Holding Register (RHR) is read. The actual time-out value is 4 character time, including data information length, start bit, parity bit, and the size of stop bit, that is, 1×, 1.5×, or 2× bit times.

7.8 Programmable baud rate generator

The SC16IS850L UART contains a programmable rational baud rate generator that takes any clock input and divides it by a divisor in the range between 1 and ($2^{16} - 1$). The SC16IS850L offers the capability of dividing the input frequency by rational divisor. The fractional part of the divisor is controlled by the CLKPRES register in the 'first extra feature register set'.

$$baud\ rate = \frac{f_{XTAL1}}{MCR[7] \times \left[16 \times \left(N + \frac{M}{16}\right)\right]} \quad (1)$$

where:

N is the integer part of the divisor in DLL and DLM registers;

M is the fractional part of the divisor in CLKPRES register;

f_{XTAL1} is the clock frequency at XTAL1 pin.

Prescaler = 1 when MCR[7] is set to 0.

Prescaler = 4 when MCR[7] is set to 1.

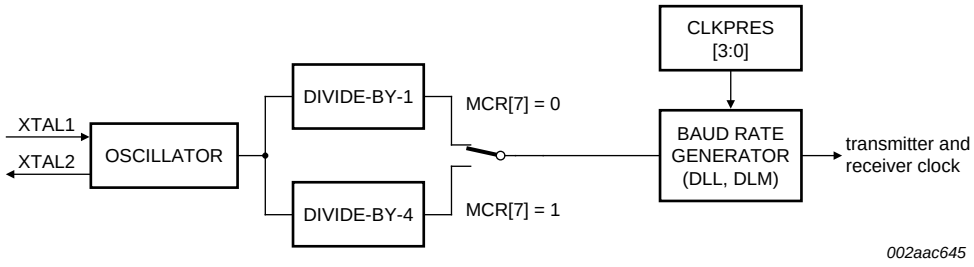


Fig 5. Prescalers and baud rate generator block diagram

A single baud rate generator is provided for the transmitter and receiver. The programmable Baud Rate Generator is capable of operating with a frequency of up to 80 MHz. To obtain maximum data rate, it is necessary to use full rail swing on the clock input. The SC16IS850L can be configured for internal or external clock operation. For internal clock operation, an industry standard crystal is connected externally between the XTAL1 and XTAL2 pins (see Figure 6). Alternatively, an external clock can be connected to the XTAL1 pin (see Figure 7) to clock the internal baud rate generator for standard or custom rates (see Table 5).

The generator divides the input 16× clock by any divisor from 1 to ($2^{16} - 1$). The SC16IS850L divides the basic external clock by 16. The baud rate is configured via the CLKPRES, DLL and DLM internal register functions. Customized baud rates can be achieved by selecting the proper divisor values for the MSB and LSB sections of the baud rate generator.

Programming the baud rate generator registers CLKPRES, DLM (MSB) and DLL (LSB) provides a user capability for selecting the desired final baud rate. The example in Table 5 shows the selectable baud rate table available when using a 1.8432 MHz external clock input when MCR[7] = 0, and CLKPRES = 0x00.

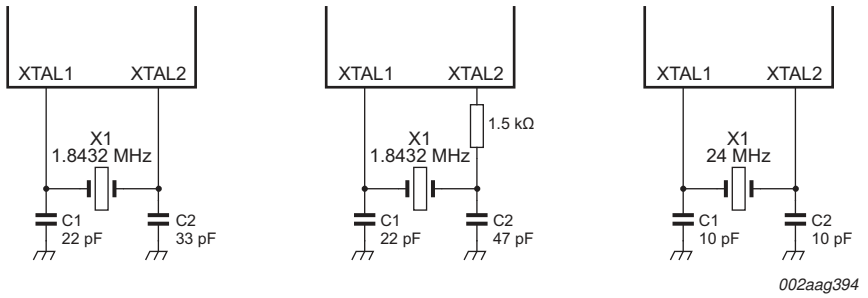
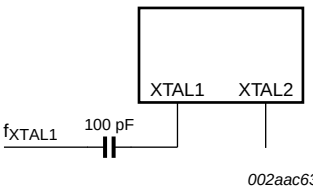


Fig 6. Crystal oscillator connection



If f_{XTAL1} frequency is greater than 50 MHz, then a DC blocking capacitor is required.

XTAL2 pin should be left unconnected when an external clock is used.

Fig 7. External clock connection

Table 5. Baud rate generator programming table using a 1.8432 MHz clock when MCR[7] = 0 and CLKPRES[3:0] = 0

Output baud rate (bit/s)	Output 16× clock divisor (decimal)	Output 16× clock divisor (hexadecimal)	DLM program value (hexadecimal)	DLL program value (hexadecimal)
50	2304	900	09	00
75	1536	600	06	00
110	1047	417	04	17
150	768	300	03	00
300	384	180	01	80
600	192	C0	00	C0
1.2 k	96	60	00	60
2.4 k	48	30	00	30
3.6 k	32	20	00	20
4.8 k	24	18	00	18
7.2 k	16	10	00	10
9.6 k	12	0C	00	0C
19.2 k	6	06	00	06
38.4 k	3	03	00	03
57.6 k	2	02	00	02
115.2 k	1	01	00	01

7.9 Loopback mode

The internal loopback capability allows on-board diagnostics. In the Loopback mode, the normal modem interface pins are disconnected and reconfigured for loopback internally (see Figure 8). MCR[3:0] register bits are used for controlling loopback diagnostic testing. In the Loopback mode, the transmitter output (TX) and the receiver input (RX) are disconnected from their associated interface pins, and instead are connected together internally. The $\overline{CTS}$, $\overline{DSR}$, $\overline{CD}$, and $\overline{RI}$ are disconnected from their normal modem control input pins, and instead are connected internally to $\overline{RTS}$, $\overline{DTR}$, MCR[3] ($\overline{OP2}$) and MCR[2] ($\overline{OP1}$). Loopback test data is entered into the transmit holding register via the user data bus interface, D[7:0]. The transmit UART serializes the data and passes the serial data to the receive UART via the internal loopback connection. The receive UART converts the serial data back into parallel data that is then made available at the user data interface D[7:0]. The user optionally compares the received data to the initial transmitted data for verifying error-free operation of the UART TX/RX circuits.

In this mode, the interrupt pin is 3-stated, therefore, the software must use the polling method (see Section 8.2.2) to send and receive data.

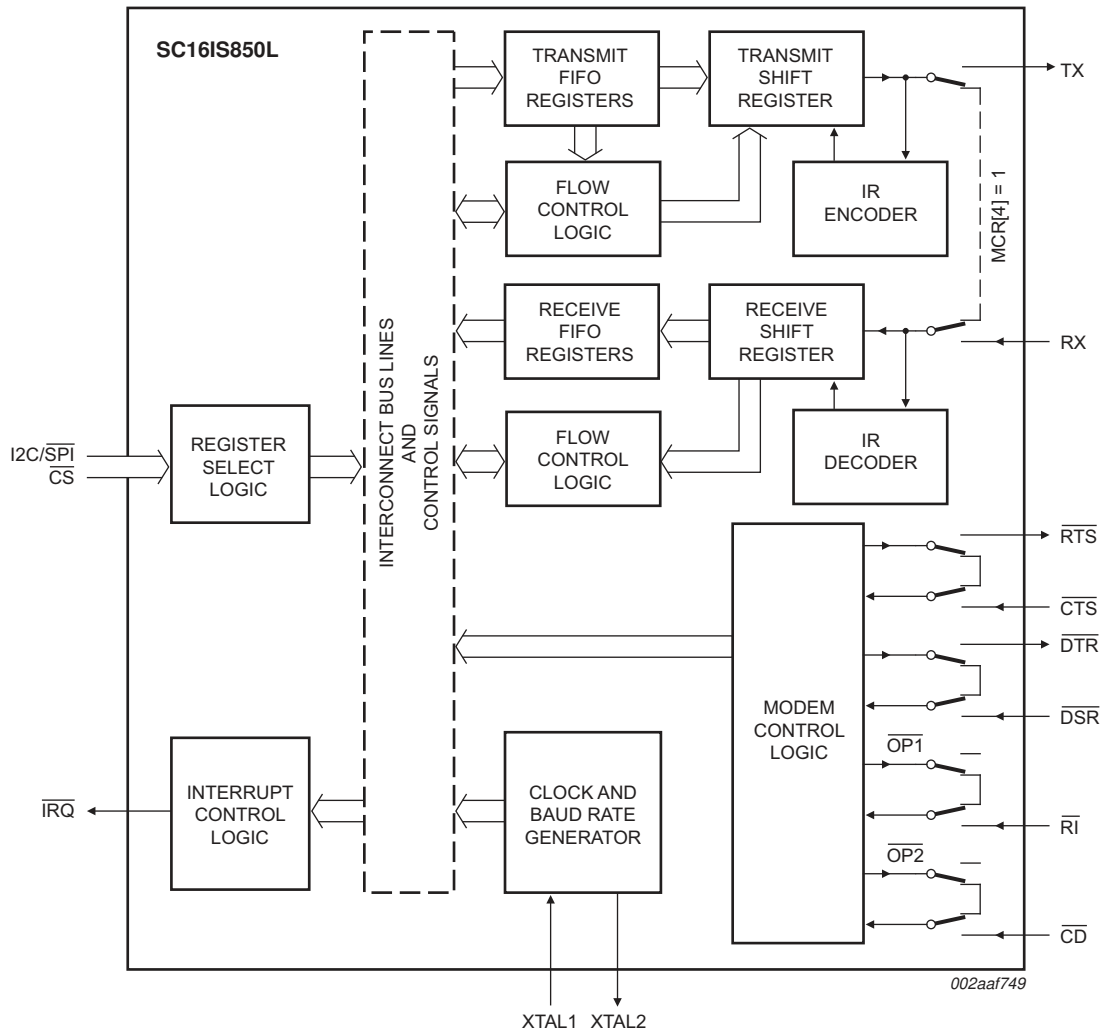


Fig 8. Internal Loopback mode diagram

7.10 Sleep mode

Sleep mode is an enhanced feature of the SC16IS850L UART. It is enabled when EFR[4], the enhanced functions bit, is set **and** when IER[4] bit is set.

7.10.1 Conditions to enter Sleep mode

Sleep mode is entered when:

- Modem input pins are not toggling.
- The serial data input line, RX, is idle for 4 character time (logic HIGH) and AFCR1[4] is logic 0. When AFCR1[4] is logic 1 the device will go to sleep regardless of the state of the RX pin (see Section 8.21 for the description of AFCR1 bit 4).
- The TX FIFO and TX shift register are empty.
- There are no interrupts pending.
- The RX FIFO is empty.

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.

Remark: Writing to the divisor latches, DLL and DLM, 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 DLM.

7.10.2 Conditions to resume normal operation

SC16IS850L resumes normal operation by any of the following:

- Receives a start bit on RX pin.
- Data is loaded into transmit FIFO.
- A change of state on any of the modem input pins

If the device is awakened by one of the conditions described above, it will return to the Sleep mode automatically after all the conditions described in Section 7.10.1 are met. The device will stay in Sleep mode until it is disabled by setting any channel's IER bit 4 to a logic 0.

Wake-up by serial data on RX input pin is supported in UART mode but not in IrDA mode. Refer to application note *AN19064, "How to wake up SC16IS740/750/760 in IrDA mode"* for a software procedure to wake up the device by receiving data in IrDA mode.

When the SC16IS850L is in Sleep mode and the host data bus (D[7:0], A[2:0], $\overline{IOW}$, $\overline{IOR}$, $\overline{CS}$) remains in steady state, either HIGH or LOW, the Sleep mode supply current will be in the μA range as specified in Table 37 "Static characteristics". If any of these signals is toggling or floating then the sleep current will be higher.

7.11 RS-485 features

7.11.1 Auto RS-485 $\overline{\text{RTS}}$ control

Normally the $\overline{\text{RTS}}$ pin is controlled by MCR[1], or if hardware flow control is enabled, the logic state of the $\overline{\text{RTS}}$ pin is controlled by the hardware flow control circuitry. AFCR2[4] will take the precedence over the other two modes; once this bit is set, the transmitter will control the state of the $\overline{\text{RTS}}$ pin. The transmitter automatically asserts the $\overline{\text{RTS}}$ pin (logic 0) once the host writes data to the transmit FIFO, and de-asserts $\overline{\text{RTS}}$ pin (logic 1) once the last bit of the data has been transmitted.

To use the auto RS-485 $\overline{\text{RTS}}$ mode the software would have to disable the hardware flow control function.

7.11.2 RS-485 $\overline{\text{RTS}}$ inversion

AFCR2[5] reverses the polarity of the $\overline{\text{RTS}}$ pin if the UART is in auto RS-485 $\overline{\text{RTS}}$ mode.

When the transmitter has data to be sent it will de-asserts the $\overline{\text{RTS}}$ pin (logic 1), and when the last bit of the data has been sent out the transmitter asserts the $\overline{\text{RTS}}$ pin (logic 0).

7.11.3 Auto 9-bit mode (RS-485)

AFCR2[0] is used to enable the 9-bit mode (Multi-drop or RS-485 mode). In this mode of operation, a ‘master’ station transmits an address character followed by data characters for the addressed ‘slave’ stations. The slave stations examine the received data and interrupt the controller if the received character is an address character (parity bit = 1).

To use the automatic 9-bit mode, the software would have to disable the hardware and software flow control functions.

7.11.3.1 Normal Multi-drop mode

The 9-bit Mode in AFCR2[0] is enabled, but not Special Character Detect (EFR[5]). The receiver is set to Force Parity 0 (LCR[5:3] = 111) in order to detect address bytes.

With the receiver initially disabled, it ignores all the data bytes (parity bit = 0) until an address byte is received (parity bit = 1). This address byte will cause the UART to set the parity error. The UART will generate a line status interrupt (IER[2] must be set to ‘1’ at this time), and at the same time puts this address byte in the RX FIFO. After the controller examines the byte it must make a decision whether or not to enable the receiver; it should enable the receiver if the address byte addresses its ID address, and must not enable the receiver if the address byte does not address its ID address.

If the controller enables the receiver, the receiver will receive the subsequent data until being disabled by the controller after the controller has received a complete message from the ‘master’ station. If the controller does not disable the receiver after receiving a message from the ‘master’ station, the receiver will generate a parity error upon receiving another address byte. The controller then determines if the address byte addresses its ID address, if it is not, the controller then can disable the receiver. If the address byte addresses the ‘slave’ ID address, the controller takes no further action, and the receiver will receive the subsequent data.

7.11.3.2 Auto address detection

If Special Character Detect is enabled (EFR[5] is set and the Xoff2 register contains the address byte) the receiver will try to detect an address byte that matches the programmed character in the Xoff2 register. If the received byte is a data byte or an address byte that does not match the programmed character in the Xoff2 register, the receiver will discard these data. Upon receiving an address byte that matches the Xoff2 character, the receiver will be automatically enabled if not already enabled, and the address character is pushed into the RX FIFO along with the parity bit (in place of the parity error bit). The receiver also generates a line status interrupt (IER[2] must be set to '1' at this time). The receiver will then receive the subsequent data from the 'master' station until being disabled by the controller after having received a message from the 'master' station.

If another address byte is received and this address byte does not match the Xoff2 character, the receiver will be automatically disabled and the address byte is ignored. If the address byte matches the Xoff2 character, the receiver will put this byte in the RX FIFO along with the parity bit in the parity error bit (LSR bit 2).

8. Register descriptions

Table 6 details the assigned bit functions for the SC16IS850L internal registers. The assigned bit functions are more fully defined in Section 8.1 through Section 8.23.

Table 6. SC16IS850L internal registers

A2	A1	A0	Register	Default[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R/W
General register set[2]													
0	0	0	RHR	0xXX	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R
0	0	0	THR	0xXX	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	W
0	0	1	IER	0x00	CTS interrupt[3]	RTS interrupt[3]	Xoff interrupt[3]	Sleep mode[3]	modem status interrupt	receive line status interrupt	transmit holding register interrupt	receive holding register interrupt	R/W
0	1	0	FCR	0x00	RCVR trigger (MSB)	RCVR trigger (LSB)	TX trigger (MSB)[3]	TX trigger (LSB)[3]	reserved	XMIT FIFO reset	RCVR FIFO reset	FIFOs enable	W
0	1	0	ISR	0x01	FIFOs enabled	FIFOs enabled	INT priority bit 4	INT priority bit 3	INT priority bit 2	INT priority bit 1	INT priority bit 0	INT status	R
0	1	1	LCR	0x00	divisor latch enable	set break	set parity	even parity	parity enable	stop bits	word length bit 1	word length bit 0	R/W
1	0	0	MCR	0x00	clock select[3]	IrDA enable	reserved	loopback	$\overline{OP2}$	$\overline{OP1}$	$\overline{RTS}$	$\overline{DTR}$	R/W
1	0	1	LSR	0x60	FIFO data error	THR and TSR empty	THR empty	break interrupt	framing error	parity error	overrun error	receive data ready	R
1	0	1	EFCR	0x00	reserved	reserved	reserved	reserved	reserved	Enable extra feature bit 1	Enable extra feature bit 0	Enable TXLVLCNT/ RXLVLCNT	W
1	1	0	MSR	0xX0	CD	RI	DSR	CTS	$\Delta\overline{CD}$	$\Delta\overline{RI}$	$\Delta\overline{DSR}$	$\Delta\overline{CTS}$	R
1	1	1	SPR	0xFF	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
Special register set[4]													
0	0	0	DLL	0xXX	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
0	0	1	DLM	0xXX	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
Second special register set[6]													
0	1	1	TXLVLCNT	0x00	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R
1	0	0	RXLVLCNT	0x00	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R

Table 6. SC16IS850L internal registers *…continued*

A2	A1	A0	Register	Default[1]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	R/W
Enhanced feature register set[5]													
0	1	0	EFR	0x00	Auto CTS	Auto RTS	special character select	Enable IER[7:4], ISR[5:4], FCR[5:4], MCR[7:5]	Cont-3 TX, RX Control	Cont-2 TX, RX Control	Cont-1 TX, RX Control	Cont-0 TX, RX Control	R/W
1	0	0	Xon1	0x00	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
1	0	1	Xon2	0x00	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
1	1	0	Xoff1	0x00	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
1	1	1	Xoff2	0x00	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	R/W
First extra feature register set[7]													
0	1	0	TXINTLVL	0x00	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
1	0	0	RXINTLVL	0x00	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
1	1	0	FLWCNTH	0x00	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
1	1	1	FLWCNTL	0x00	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
Second extra feature register set[8]													
0	1	0	CLKPRES	0x00	reserved	reserved	reserved	reserved	bit 3	bit 2	bit 1	bit 0	R/W
1	0	0	RS485TIME	0x00	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	R/W
1	1	0	AFCR2	0x00	reserved	reserved	RS485 RTS Invert	Auto RS485 RTS	RS485 RTS/DTR	Transmitter Disable	Receiver Disable	9-bit Enable	R/W
1	1	1	AFCR1	0x00	reserved	reserved	reserved	Sleep RXLow	reserved	RTS/CTS mapped to DTR/DSR	Software Reset	TSR Interrupt	R/W

[1] The value shown represents the register's initialized HEX value; X = not applicable.

[2] Accessible only when LCR[7] is logic 0, and EFCR[2:1] are logic 0.

[3] This bit is only accessible when EFR[4] is set.

[4] Baud rate registers accessible only when LCR[7] is logic 1.

[5] Enhanced Feature Register, Xon1/Xon2 and Xoff1/Xoff2 are accessible only when LCR is set to 0xBF, and EFCR[2:1] are logic 0.

[6] Second Special registers are accessible only when EFCR[0] = 1, and EFCR[2:1] are logic 0.

[7] First extra feature register set is only accessible when EFCR[2:0] = 010b.

[8] Second extra feature register set is only accessible when EFCR[2:0] = 100b.

8.1 Transmit (THR) and Receive (RHR) Holding Registers

The serial transmitter section consists of an 8-bit Transmit Hold Register (THR) and Transmit Shift Register (TSR). The status of the THR is provided in the Line Status Register (LSR). Writing to the THR transfers the contents of the data byte [D7:D0] to the transmit FIFO. The THR empty flag in the LSR will be set to a logic 1 when the transmit FIFO is empty or when data is transferred to the TSR.

The serial receive section also contains an 8-bit Receive Holding Register (RHR) and a Receive Serial Shift Register (RSR). Receive data is removed from the SC16IS850L receive FIFO by reading the RHR. The receive section provides a mechanism to prevent false starts. On the falling edge of a start or false start bit, an internal receiver counter starts counting clocks at the 16× clock rate. After $7\frac{1}{2}$ clocks, the start bit time should be shifted to the center of the start bit. At this time the start bit is sampled, and if it is still a logic 0 it is validated. Evaluating the start bit in this manner prevents the receiver from assembling a false character. Receiver status codes will be posted in the LSR.

8.2 Interrupt Enable Register (IER)

The Interrupt Enable Register (IER) masks the interrupts from receiver ready, transmitter empty, line status and modem status registers. These interrupts would normally be seen on the INT output pin.

Table 7. Interrupt Enable Register bits description

Bit	Symbol	Description
7	IER[7]	CTS interrupt. logic 0 = disable the CTS interrupt (normal default condition) logic 1 = enable the CTS interrupt. The SC16IS850L issues an interrupt when the $\overline{\text{CTS}}$ pin transitions from a logic 0 to a logic 1.
6	IER[6]	RTS interrupt. logic 0 = disable the RTS interrupt (normal default condition) logic 1 = enable the RTS interrupt. The SC16IS850L issues an interrupt when the $\overline{\text{RTS}}$ pin transitions from a logic 0 to a logic 1.
5	IER[5]	Xoff interrupt. logic 0 = disable the software flow control, receive Xoff interrupt (normal default condition) logic 1 = enable the receive Xoff interrupt
4	IER[4]	Sleep mode. logic 0 = disable Sleep mode (normal default condition) logic 1 = enable Sleep mode
3	IER[3]	Modem Status Interrupt. This interrupt will be issued whenever there is a modem status change as reflected in MSR[3:0]. logic 0 = disable the modem status register interrupt (normal default condition) logic 1 = enable the modem status register interrupt
2	IER[2]	Receive Line Status interrupt. This interrupt will be issued whenever a receive data error condition exists as reflected in LSR[4:1]. logic 0 = disable the receiver line status interrupt (normal default condition) logic 1 = enable the receiver line status interrupt

Table 7. Interrupt Enable Register bits description *…continued*

Bit	Symbol	Description
1	IER[1]	Transmit Holding Register interrupt. In the non-FIFO mode, this interrupt will be issued whenever the THR is empty, and is associated with LSR[5]. In the FIFO modes, this interrupt will be issued whenever the FIFO is empty. logic 0 = disable the Transmit Holding Register Empty (TXRDY) interrupt (normal default condition) logic 1 = enable the TXRDY (ISR level 3) interrupt
0	IER[0]	Receive Holding Register interrupt. In the non-FIFO mode, this interrupt will be issued when the RHR has data, or is cleared when the RHR is empty. In the FIFO mode, this interrupt will be issued when the FIFO has reached the programmed trigger level or is cleared when the FIFO drops below the trigger level. logic 0 = disable the receiver ready (ISR level 2, RXRDY) interrupt (normal default condition) logic 1 = enable the RXRDY (ISR level 2) interrupt

8.2.1 IER versus Transmit/Receive FIFO interrupt mode operation

When the receive FIFO is enabled (FCR[0] = logic 1), and receive interrupts (IER[0] = logic 1) are enabled, the receive interrupts and register status will reflect the following:

- The receive RXRDY interrupt (Level 2 ISR interrupt) is issued to the external CPU when the receive FIFO has reached the programmed trigger level. It will be cleared when the receive FIFO drops below the programmed trigger level.
- Receive FIFO status will also be reflected in the user accessible ISR register when the receive FIFO trigger level is reached. Both the ISR register receive status bit and the interrupt will be cleared when the FIFO drops below the trigger level.
- The receive data ready bit (LSR[0]) is set as soon as a character is transferred from the shift register (RSR) to the receive FIFO. It is reset when the FIFO is empty.
- When the Transmit FIFO and interrupts are enabled, an interrupt is generated when the transmit FIFO is empty due to the unloading of the data by the TSR and UART for transmission via the transmission media. The interrupt is cleared either by reading the ISR, or by loading the THR with new data characters.

8.2.2 IER versus Receive/Transmit FIFO polled mode operation

When FCR[0] = logic 1, setting IER[3:0] puts the SC16IS850L in the FIFO polled mode of operation. In this mode, interrupts are not generated and the user must poll the LSR register for TX and/or RX data status. Since the receiver and transmitter have separate bits in the LSR either or both can be used in the polled mode by selecting respective transmit or receive control bit(s).

- LSR[0] will be a logic 1 as long as there is one byte in the receive FIFO.
- LSR[4:1] will provide the type of receive errors, or a receive break, if encountered.
- LSR[5] will indicate when the transmit FIFO is empty.
- LSR[6] will indicate when both the transmit FIFO and transmit shift register are empty.
- LSR[7] will show if any FIFO data errors occurred.

8.3 FIFO Control Register (FCR)

This register is used to enable the FIFOs, clear the FIFOs, and set the receive FIFO trigger levels.

8.3.1 FIFO mode

Table 8. FIFO Control Register bits description

Bit	Symbol	Description
7:6	FCR[7:6]	Receive trigger level in 32-byte FIFO mode[1]. These bits are used to set the trigger levels for receive FIFO interrupt and flow control. The SC16IS850L will issue a receive ready interrupt when the number of characters in the receive FIFO reaches the selected trigger level. Refer to Table 9.
5:4	FCR[5:4]	Transmit trigger level in 32-byte FIFO mode[2]. These bits are used to set the trigger level for the transmit FIFO interrupt and flow control. The SC16IS850L will issue a transmit empty interrupt when the number of characters in FIFO drops below the selected trigger level. Refer to Table 10.
3	FCR[3]	reserved
2	FCR[2]	XMIT FIFO reset. logic 0 = no FIFO transmit reset (normal default condition) logic 1 = clears the contents of the transmit FIFO and resets the FIFO counter logic. This bit will return to a logic 0 after clearing the FIFO.
1	FCR[1]	RCVR FIFO reset. logic 0 = no FIFO receive reset (normal default condition) logic 1 = clears the contents of the receive FIFO and resets the FIFO counter logic. 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] For 128-byte FIFO mode, refer to Section 8.16, Section 8.17, Section 8.18.

[2] For 128-byte FIFO mode, refer to Section 8.15, Section 8.17, Section 8.18.

Table 9. RCVR trigger levels

FCR[7]	FCR[6]	RX FIFO trigger level (bytes) in 32-byte FIFO mode[1]
0	0	8
0	1	16
1	0	24
1	1	28

[1] When RXINTLVL, TXINTLVL, FLWCNTL or FLWCNTH contains any value other than 0x00, receive and transmit trigger levels are set by RXINTLVL, TXINTLVL registers (see Section 7.3 “FIFO operation”).

Table 10. TX FIFO trigger levels

FCR[5]	FCR[4]	TX FIFO trigger level (bytes) in 32-byte FIFO mode[1]
0	0	16
0	1	8
1	0	24
1	1	30

[1] When RXINTLVL, TXINTLVL, FLWCNTL or FLWCNTH contains any value other than 0x00, receive and transmit trigger levels are set by RXINTLVL, TXINTLVL registers (see Section 7.3 “FIFO operation”).

8.4 Interrupt Status Register (ISR)

The SC16IS850L provides six levels of prioritized interrupts to minimize external software interaction. The Interrupt Status Register (ISR) provides the user with six interrupt status bits. Performing a read cycle on the ISR will provide the user with the highest pending interrupt level to be serviced. No other interrupts are acknowledged until the pending interrupt is serviced. A lower level interrupt may be seen after servicing the higher level interrupt and re-reading the interrupt status bits. Table 11 “Interrupt source” shows the data values (bits 5:0) for the six prioritized interrupt levels and the interrupt sources associated with each of these interrupt levels.

Table 11. Interrupt source

Priority level	ISR[5]	ISR[4]	ISR[3]	ISR[2]	ISR[1]	ISR[0]	Source of the interrupt
1	0	0	0	1	1	0	LSR (Receiver Line Status Register)
2	0	0	0	1	0	0	RXRDY (Received Data Ready)
2	0	0	1	1	0	0	RXRDY (Receive Data time-out)
3	0	0	0	0	1	0	TXRDY (Transmitter Holding Register Empty)
4	0	0	0	0	0	0	MSR (Modem Status Register)
5	0	1	0	0	0	0	RXRDY (Received Xoff signal)/ Special character
6	1	0	0	0	0	0	CTS, RTS change of state

Table 12. Interrupt Status Register bits description

Bit	Symbol	Description
7:6	ISR[7:6]	FIFOs enabled. These bits are set to a logic 0 when the FIFOs are not being used in the non-FIFO mode. They are set to a logic 1 when the FIFOs are enabled in the SC16IS850L mode. logic 0 or cleared = default condition
5:4	ISR[5:4]	INT priority bits 4:3. These bits are enabled when EFR[4] is set to a logic 1. ISR[4] indicates that matching Xoff character(s) have been detected. ISR[5] indicates that CTS, RTS have been generated. Note that once set to a logic 1, the ISR[4] bit will stay a logic 1 until Xon character(s) are received. logic 0 or cleared = default condition
3:1	ISR[3:1]	INT priority bits 2:0. These bits indicate the source for a pending interrupt at interrupt priority levels 1, 2, and 3 (see Table 11). logic 0 or cleared = default condition

Table 12. Interrupt Status Register bits description *…continued*

Bit	Symbol	Description
0	ISR[0]	INT status. logic 0 = an interrupt is pending and the ISR contents may be used as a pointer to the appropriate interrupt service routine logic 1 = no interrupt pending (normal default condition)

8.5 Line Control Register (LCR)

The Line Control Register is used to specify the asynchronous data communication format. The word length, the number of stop bits, and the parity are selected by writing the appropriate bits in this register.

Table 13. Line Control Register bits description

Bit	Symbol	Description
7	LCR[7]	Divisor latch enable. The internal baud rate counter latch and Enhanced Feature mode enable. logic 0 = divisor latch disabled (normal default condition) logic 1 = divisor latch enabled
6	LCR[6]	Set break. 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 for alerting the remote receiver to a line break condition
5:3	LCR[5:3]	Programs the parity conditions (see Table 14).
2	LCR[2]	Stop bits. The length of stop bit is specified by this bit in conjunction with the programmed word length (see Table 15). logic 0 or cleared = default condition
1:0	LCR[1:0]	Word length bits 1, 0. These two bits specify the word length to be transmitted or received (see Table 16). logic 0 or cleared = default condition

Table 14. LCR[5:3] 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 15. LCR[2] stop bit length

LCR[2]	Word length (bits)	Stop bit length (bit times)
0	5, 6, 7, 8	1
1	5	1½
1	6, 7, 8	2

Table 16. LCR[1:0] word length

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

8.6 Modem Control Register (MCR)

This register controls the interface with the modem or a peripheral device.

Table 17. Modem Control Register bits description

Bit	Symbol	Description
7	MCR[7]	Clock select logic 0 = divide-by-1 clock input logic 1 = divide-by-4 clock input
6	MCR[6]	IR enable (see Figure 31). logic 0 = enable the standard modem receive and transmit input/output interface (normal default condition) logic 1 = enable infrared IrDA receive and transmit inputs/outputs. While in this mode, the TX/RX output/inputs are routed to the infrared encoder/decoder. The data input and output levels will conform to the IrDA infrared interface requirement. As such, while in this mode, the infrared TX output will be a logic 0 during idle data conditions.
5	MCR[5]	Reserved.
4	MCR[4]	Loopback. Enable the local loopback mode (diagnostics). In this mode the transmitter output (TX) and the receiver input (RX), $\overline{CTS}$, $\overline{DSR}$, $\overline{CD}$, and $\overline{RI}$ are disconnected from the SC16IS850L I/O pins. Internally the modem data and control pins are connected into a loopback data configuration (see Figure 8). In this mode, the receiver and transmitter interrupts remain fully operational. The Modem Control Interrupts are also operational, but the interrupts' sources are switched to the lower four bits of the Modem Control. Interrupts continue to be controlled by the IER register. logic 0 = disable Loopback mode (normal default condition) logic 1 = enable local Loopback mode (diagnostics)
3	MCR[3]	$\overline{OP2}$. This bit is used for internal Loopback mode only. In Loopback mode, this bit is used to write the state of the modem $\overline{CD}$ interface signal.
2	MCR[2]	$\overline{OP1}$. This bit is used for internal Loopback mode only. In Loopback mode, this bit is used to write the state of the modem $\overline{RI}$ interface signal.
1	MCR[1]	$\overline{RTS}$ logic 0 = force $\overline{RTS}$ output to a logic 1 (normal default condition) logic 1 = force $\overline{RTS}$ output to a logic 0
0	MCR[0]	$\overline{DTR}$ logic 0 = force $\overline{DTR}$ output to a logic 1 (normal default condition) logic 1 = force $\overline{DTR}$ output to a logic 0

8.7 Line Status Register (LSR)

This register provides the status of data transfers between the SC16IS850L and the CPU.

Table 18. 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 current FIFO data. This bit is cleared when there are no remaining error flags associated with the remaining data in the FIFO.
6	LSR[6]	THR and TSR empty. This bit is the Transmit Empty indicator. This bit is set to a logic 1 whenever the transmit holding register and the transmit shift register are both empty. It is reset to logic 0 whenever either the THR or TSR contains a data character. In the FIFO mode, this bit is set to '1' whenever the transmit FIFO and transmit shift register are both empty.
5	LSR[5]	THR empty. This bit is the Transmit Holding Register Empty indicator. This bit indicates that the UART is ready to accept a new character for transmission. In addition, this bit causes the UART to issue an interrupt to CPU when the THR interrupt enable is set. The THR bit is set to a logic 1 when a character is transferred from the transmit holding register into the transmitter shift register. The bit is reset to a logic 0 concurrently with the loading of the transmitter holding register by the CPU. In the FIFO mode, this bit is set when the transmit FIFO is empty; it is cleared when at least 1 byte is written to the transmit FIFO.
4	LSR[4]	Break interrupt. logic 0 = no break condition (normal default condition) logic 1 = the receiver received a break signal (RX was a logic 0 for one character frame time). In the FIFO mode, only one break character is loaded into the FIFO.
3	LSR[3]	Framing error. logic 0 = no framing error (normal default condition) logic 1 = framing error. The receive character did not have a valid stop bit(s). In the FIFO mode, this error is associated with the character at the top of the FIFO.
2	LSR[2]	Parity error. logic 0 = no parity error (normal default condition) logic 1 = parity error. The receive character does not have correct parity information and is suspect. In the FIFO mode, this error is associated with the character at the top of the FIFO.
1	LSR[1]	Overrun error. logic 0 = no overrun error (normal default condition) logic 1 = overrun error. A data overrun error occurred in the Receive Shift Register. This happens when additional data arrives while the FIFO is full. In this case, the previous data in the shift register is overwritten. Note that under this condition, the data byte in the Receive Shift Register is not transferred into the FIFO, therefore the data in the FIFO is not corrupted by the error.
0	LSR[0]	Receive data ready. logic 0 = no data in Receive Holding Register or FIFO (normal default condition) logic 1 = data has been received and is saved in the Receive Holding Register or FIFO

8.8 Modem Status Register (MSR)

This register shares the same address as EFCR register. This is a read-only register and it provides the current state of the control interface signals from the modem, or other peripheral device to which the SC16IS850L is connected. Four bits of this register are used to indicate the changed information. These bits are set to a logic 1 whenever a control input from the modem changes state. These bits are set to a logic 0 whenever the CPU reads this register.

When write, the data will be written to EFCR register.

Table 19. Modem Status Register bits description

Bit	Symbol	Description
7	MSR[7]	CD. During normal operation, this bit is the complement of the $\overline{CD}$ input. Reading this bit in the loopback mode produces the state of MCR[3] ($\overline{OP2}$).
6	MSR[6]	RI. During normal operation, this bit is the complement of the $\overline{RI}$ input. Reading this bit in the loopback mode produces the state of MCR[2] ($\overline{OP1}$).
5	MSR[5]	DSR. During normal operation, this bit is the complement of the $\overline{DSR}$ input. During the loopback mode, this bit is equivalent to MCR[0] ($\overline{DTR}$).
4	MSR[4]	CTS. During normal operation, this bit is the complement of the $\overline{CTS}$ input. During the loopback mode, this bit is equivalent to MCR[1] ($\overline{RTS}$).
3	MSR[3]	$\Delta\overline{CD}$ [1] logic 0 = no $\overline{CD}$ change (normal default condition) logic 1 = the $\overline{CD}$ input to the SC16IS850L has changed state since the last time it was read. A modem Status Interrupt will be generated.
2	MSR[2]	$\Delta\overline{RI}$ [1] logic 0 = no $\overline{RI}$ change (normal default condition) logic 1 = the $\overline{RI}$ input to the SC16IS850L has changed from a logic 0 to a logic 1. A modem Status Interrupt will be generated.
1	MSR[1]	$\Delta\overline{DSR}$ [1] logic 0 = no $\overline{DSR}$ change (normal default condition) logic 1 = the $\overline{DSR}$ input to the SC16IS850L has changed state since the last time it was read. A modem Status Interrupt will be generated.
0	MSR[0]	$\Delta\overline{CTS}$ [1] logic 0 = no $\overline{CTS}$ change (normal default condition) logic 1 = the $\overline{CTS}$ input to the SC16IS850L has changed state since the last time it was read. A modem Status Interrupt will be generated.

[1] Whenever any MSR bit 3:0 is set to logic 1, a Modem Status Interrupt will be generated.

8.9 Extra Feature Control Register (EFCR)

This is a write-only register, and it allows the software access to these registers: 'first extra feature register set', 'second extra feature register set', Transmit FIFO Level Counter (TXLVLCNT), and Receive FIFO Level Counter (RXLVLCNT).

Table 20. Extra Feature Control Register bits description

Bit	Symbol	Description
7:3	EFCR[7:3]	reserved
2:1	EFCR[2:1]	Enable Extra Feature Control bits 00 = General register set is accessible 01 = First extra feature register set is accessible 10 = Second extra feature register set is accessible 11 = reserved
0	EFCR[0]	Enable TXLVLCNT and RXLVLCNT access 0 = TXLVLCNT and RXLVLCNT are disabled 1 = TXLVLCNT and RXLVLCNT are enabled and can be read.

Remark: EFCR[2:1] has higher priority than EFCR[0]. TXLVLCNT and RXLVLCNT can only be accessed if EFCR[2:1] are zeroes.

8.10 Scratchpad Register (SPR)

The SC16IS850L provides a temporary data register to store 8 bits of user information.

8.11 Divisor Latch (DLL and DLM)

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

8.12 Transmit FIFO Level Count (TXLVLCNT)

This register is a read-only register. It reports the number of spaces available in the transmit FIFO.

8.13 Receive FIFO Level Count (RXLVLCNT)

This register is a read-only register. It reports the fill level of the receive FIFO (the number of characters in the RX FIFO).

8.14 Enhanced Feature Register (EFR)

Enhanced features are enabled or disabled using this register.

Bits 0 through 4 provide single or dual character software flow control selection. When the Xon1 and Xon2 and/or Xoff1 and Xoff2 modes are selected, the double 8-bit words are concatenated into two sequential numbers.

Table 21. Enhanced Feature Register bits description

Bit	Symbol	Description
7	EFR[7]	Automatic CTS flow control. logic 0 = automatic CTS flow control is disabled (normal default condition) logic 1 = enable automatic CTS flow control. Transmission will stop when $\overline{CTS}$ goes to a logical 1. Transmission will resume when the $\overline{CTS}$ pin returns to a logical 0.
6	EFR[6]	Automatic RTS flow control. Automatic RTS may be used for hardware flow control by enabling EFR[6]. When Auto-RTS is selected, an interrupt will be generated when the receive FIFO is filled to the programmed trigger level and $\overline{RTS}$ will go to a logic 1 at the next trigger level. $\overline{RTS}$ will return to a logic 0 when data is unloaded below the next lower trigger level (programmed trigger level 1). The state of this register bit changes with the status of the hardware flow control. $\overline{RTS}$ functions normally when hardware flow control is disabled. logic 0 = automatic RTS flow control is disabled (normal default condition) logic 1 = enable automatic RTS flow control
5	EFR[5]	Special Character Detect. logic 0 = special character detect disabled (normal default condition) logic 1 = special character detect enabled. The SC16IS850L compares each incoming receive character with Xoff2 data. If a match exists, the received data will be transferred to FIFO and ISR[4] will be set to indicate detection of special character. Bit 0 in the X-registers corresponds with the LSB bit for the receive character. When this feature is enabled, the normal software flow control must be disabled (EFR[3:0] must be set to a logic 0).
4	EFR[4]	Enhanced function control bit. The content of IER[7:4], ISR[5:4], FCR[5:4], and MCR[7:5] can be modified and latched. After modifying any bits in the enhanced registers, EFR[4] can be set to a logic 0 to latch the new values. This feature prevents existing software from altering or overwriting the SC16IS850L enhanced functions. logic 0 = disable/latch enhanced features[1]. (Normal default condition.) logic 1 = enables the enhanced functions[1].
3:0	EFR[3:0]	Cont-3:0 TX, RX control. Logic 0 or cleared is the default condition. Combinations of software flow control can be selected by programming these bits. See Table 22.

[1] Enhanced function control bits IER[7:4], ISR[5:4], FCR[5:4] and MCR[7:5].

Table 22. Software flow control functions[1]

Cont-3	Cont-2	Cont-1	Cont-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 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

[1] When using a software flow control the Xon/Xoff characters cannot be used for data transfer.

8.15 Transmit Interrupt Level Register (TXINTLVL)

This 8-bit register is used to store the transmit FIFO trigger levels used for interrupt generation. Trigger levels from 1 to 128 can be programmed with a granularity of 1. Table 23 shows the TXINTLVL register bit settings.

Table 23. TXINTLVL register bits description

Bit	Symbol	Description
7:0	TXINTLVL[7:0]	This register stores the programmable transmit interrupt trigger levels for 128-byte FIFO mode[1]. 0x00 = trigger level is set to 1 0x01 = trigger level is set to 1 ... 0x80 = trigger level is set to 128

[1] For 32-byte FIFO mode, refer to Section 8.3.

8.16 Receive Interrupt Level Register (RXINTLVL)

This 8-bit register is used store the receive FIFO trigger levels used for interrupt generation. Trigger levels from 1 to 128 can be programmed with a granularity of 1. Table 24 shows the RXINTLVL register bit settings.

Table 24. RXINTLVL register bits description

Bit	Symbol	Description
7:0	RXINTLVL[7:0]	This register stores the programmable receive interrupt trigger levels for 128-byte FIFO mode[1]. 0x00 = trigger level is set to 1 0x01 = trigger level is set to 1 ... 0x80 = trigger level is set to 128

[1] For 32-byte FIFO mode, refer to Section 8.3.

8.17 Flow Control Trigger Level High (FLWCNTH)

This 8-bit register is used to store the receive FIFO high threshold levels to start/stop transmission during hardware/software flow control. Table 25 shows the FLWCNTH register bit settings; see Section 7.4.

Table 25. FLWCNTH register bits description

Bit	Symbol	Description
7:0	FLWCNTH[7:0]	This register stores the programmable HIGH threshold level for hardware and software flow control for 128-byte FIFO mode[1]. 0x00 = trigger level is set to 1 0x01 = trigger level is set to 1 ... 0x80 = trigger level is set to 128

[1] For 32-byte FIFO mode, refer to Section 8.3.

8.18 Flow Control Trigger Level Low (FLWCNTL)

This 8-bit register is used to store the receive FIFO low threshold levels to start/stop transmission during hardware/software flow control. Table 26 shows the FLWCNTL register bit settings; see Section 7.4.

Table 26. FLWCNTL register bits description

Bit	Symbol	Description
7:0	FLWCNTL[7:0]	This register stores the programmable LOW threshold level for hardware and software flow control for 128-byte FIFO mode[1]. 0x00 = trigger level is set to 1 0x01 = trigger level is set to 1 ... 0x80 = trigger level is set to 128

[1] For 32-byte FIFO mode, refer to Section 8.3.

8.19 Clock Prescaler (CLKPRES)

This register hold values for the clock prescaler.

Table 27. Clock Prescaler register bits description

Bit	Symbol	Description
7:4	CLKPRES[7:4]	reserved
3:0	CLKPRES[3:0]	Clock Prescaler value. Reset to 0.

8.20 RS-485 Turn-around time delay (RS485TIME)

The value in this register controls the turn-around time of the external line transceiver in bit time. In automatic 9-bit mode $\overline{RTS}$ or $\overline{DTR}$ pin is used to control the direction of the line driver, after the last bit of data has been shifted out of the transmit shift register the UART will count down the value in this register. When the count value reaches zero, the UART will assert $\overline{RTS}$ or $\overline{DTR}$ pin (logic 0) to turn the external RS-485 transceiver around for receiving.

Table 28. RS-485 programmable turn-around time register bits description

Bit	Symbol	Description
7:0	RS485TIME[7:0]	External RS-485 transceiver turn-around time delay. The value represents the bit time at the programmed baud rate.

8.21 Advanced Feature Control Register 2 (AFCR2)

Table 29. Advanced Feature Control Register 2 register bits description

Bit	Symbol	Description
7:6	AFCR2[7:6]	reserved
5	AFCR2[5]	RTSInvert. Invert RTS or DTR signal in automatic 9-bit mode. logic 0 = RTS or DTR is set to 0 by the UART during transmission, and to 1 during reception logic 1 = RTS or DTR is set to 1 by the UART during transmission, and to 0 during reception
4	AFCR2[4]	RTSCon. Enable the transmitter to control $\overline{RTS}$ or $\overline{DTR}$ pin in automatic 9-bit mode. logic 0 = transmitter does not control $\overline{RTS}$ or DTR pin logic 1 = transmitter controls $\overline{RTS}$ or DTR pin
3	AFCR2[3]	RS485 RTS/DTR. Select $\overline{RTS}$ or DTR pin to control the external transceiver. logic 0 = $\overline{RTS}$ pin is used to control the external transceiver logic 1 = DTR pin is used to control the external transceiver
2	AFCR2[2]	TXDisable. Disable transmitter logic 0 = transmitter is enabled logic 1 = transmitter is disabled
1	AFCR2[1]	RXDisable. Disable receiver logic 0 = receiver is enabled logic 1 = receiver is disabled
0	AFCR2[0]	9-bitMode. Enable 9-bit mode or Multidrop (RS-485) mode logic 0 = normal RS-232 mode logic 1 = enable 9-bit mode

8.22 Advanced Feature Control Register 1 (AFCR1)

Table 30. Advanced Feature Control Register 1 register bits description

Bit	Symbol	Description
7:5	AFCR1[7:5]	reserved
4	AFCR1[4]	Sleep RXLow. Program RX input to be edge-sensitive or level-sensitive. logic 0 = RX input is level-sensitive. If RX pin is LOW, the UART will not go to sleep. Once the UART is in Sleep mode, it will wake up if RX pin goes LOW. logic 1 = RX input is edge-sensitive. UART will go to sleep even if RX pin is LOW, and will wake up when RX pin toggles.
3	AFCR1[3]	reserved
2	AFCR1[2]	RTS/CTS mapped to DTR/DSR. Switch the function of RTS/CTS to DTR/DSR. logic 0 = RTS and CTS signals are used for hardware flow control. logic 1 = DTR and DSR signals are used for hardware flow control. RTS and CTS retain their functionality.
1	AFCR1[1]	SReset. Software Reset. A write to this bit will reset the UART. Once the UART is reset this bit is automatically set to 0.[1]
0	AFCR1[0]	TSR Interrupt. Select TSR interrupt mode logic 0 = transmit empty interrupt occurs when transmit FIFO falls below the trigger level or becomes empty. logic 1 = transmit empty interrupt occurs when transmit FIFO falls below the trigger level, or becomes empty and the last stop bit has been shifted out of the Transmit Shift Register.

[1] It takes 4 XTAL1 clocks to reset the device.

8.23 SC16IS850L external reset condition and software reset

These two reset methods are identical and will reset the internal registers as indicated in Table 31.

Table 31. Reset state for registers

Register	Reset state
IER	IER[7:0] = 0
FCR	FCR[7:0] = 0
ISR	ISR[7:1] = 0; ISR[0] = 1
LCR	LCR[7:0] = 0
MCR	MCR[7:0] = 0
LSR	LSR[7] = 0; LSR[6:5] = 1; LSR[4:0] = 0
MSR	MSR[7:4] = input signals; MSR[3:0] = 0
EFCR	EFCR[7:0] = 0
SPR	SPR[7:0] = 1
DLL	undefined
DLM	undefined
TXLVLCNT	TXLVLCNT[7:0] = 0
RXLVLCNT	RXLVLCNT[7:0] = 0
EFR	EFR[7:0] = 0
Xon1	undefined
Xon2	undefined
Xoff1	undefined
Xoff2	undefined
TXINTLVL	TXINTLVL[7:0] = 0
RXINTLVL	RXINTLVL[7:0] = 0
FLWCNTH	FLWCNTH[7:0] = 0
FLWCNTL	FLWCNTL[7:0] = 0
CLKPRES	CLKPRES[7:0] = 0
RS485TIME	RS485TIME[7:0] = 0
AFCR2	AFCR2[7:0] = 0
AFCR1	AFCR1[7:0] = 0

Table 32. Reset state for outputs

Output	Reset state
TX	logic 1
$\overline{\text{RTS}}$	logic 1
$\overline{\text{DTR}}$	logic 1
INT	logic 0
$\overline{\text{IRQ}}$	open-drain

9. I²C-bus operation

The two lines of the I²C-bus are a serial data line (SDA) and a serial clock line (SCL). Both lines are connected to a positive supply via a pull-up resistor, and remain HIGH when the bus is not busy. Each device is recognized by a unique address whether it is a microcomputer, LCD driver, memory or keyboard interface and can operate as either a transmitter or receiver, depending on the function of the device. A device generating a message or data is a transmitter, and a device receiving the message or data is a receiver. Obviously, a passive function like an LCD driver could only be a receiver, while a microcontroller or a memory can both transmit and receive data.

9.1 Data transfers

One data bit is transferred during each clock pulse (see Figure 9). The data on the SDA line must remain stable during the HIGH period of the clock pulse in order to be valid. Changes in the data line at this time will be interpreted as control signals. A HIGH-to-LOW transition of the data line (SDA) while the clock signal (SCL) is HIGH indicates a START condition, and a LOW-to-HIGH transition of the SDA while SCL is HIGH defines a STOP condition (see Figure 10). The bus is considered to be busy after the START condition and free again at a certain time interval after the STOP condition. The START and STOP conditions are always generated by the master.

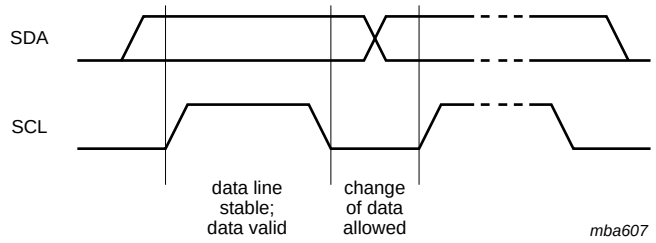


Fig 9. Bit transfer on the I²C-bus

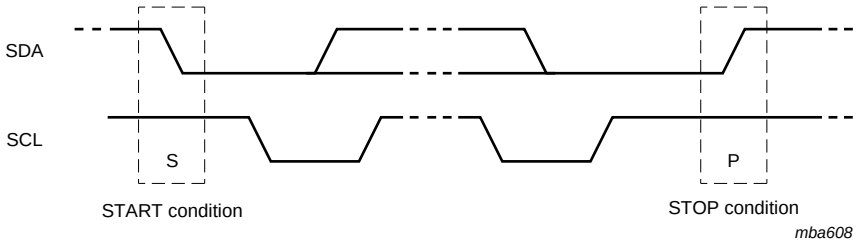


Fig 10. START and STOP conditions

The number of data bytes transferred between the START and STOP condition from transmitter to receiver is not limited. Each byte, which must be eight bits long, is transferred serially with the most significant bit first, and is followed by an acknowledge bit (see Figure 11). The clock pulse related to the acknowledge bit is generated by the master. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, while the transmitting device releases this pulse (see Figure 12).

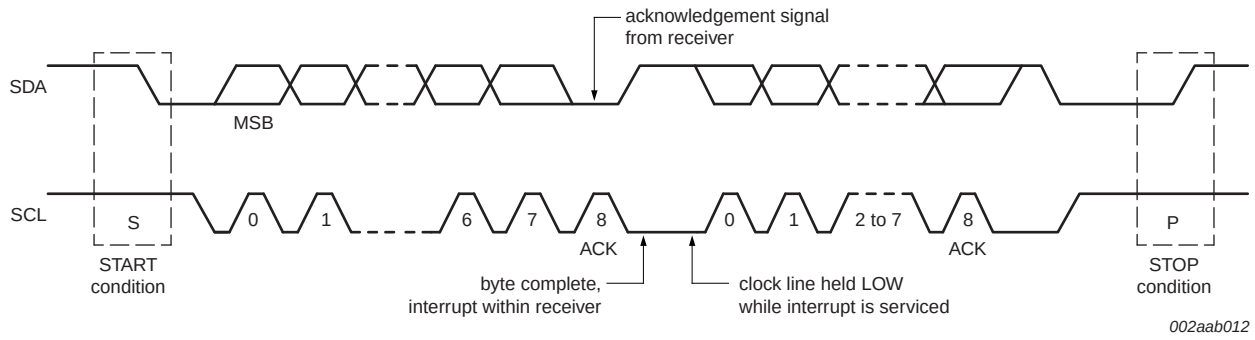


Fig 11. Data transfer on the I²C-bus

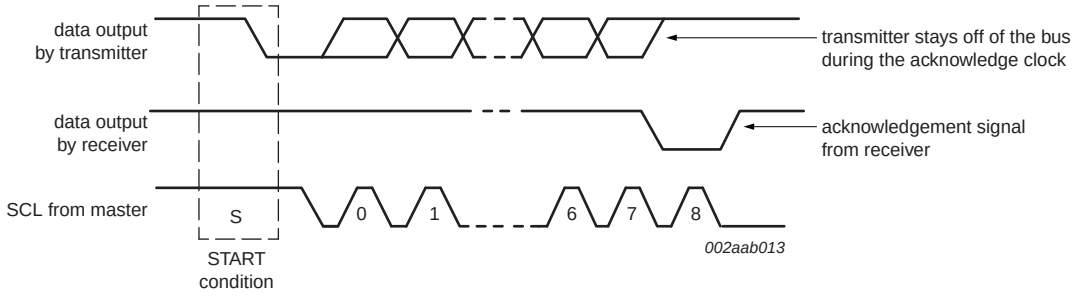


Fig 12. Acknowledge on the I²C-bus

A slave receiver must generate an acknowledge after the reception of each byte, and a master must generate one after the reception of each byte clocked out of the slave transmitter.

There are two exceptions to the ‘acknowledge after every byte’ rule. The first occurs when a master is a receiver: it must signal an end of data to the transmitter by **not** signalling an acknowledge on the last byte that has been clocked out of the slave. The acknowledge related clock, generated by the master should still take place, but the SDA line will not be pulled down. In order to indicate that this is an active and intentional lack of acknowledgement, we shall term this special condition as a ‘negative acknowledge’.

The second exception is that a slave will send a negative acknowledge when it can no longer accept additional data bytes. This occurs after an attempted transfer that cannot be accepted.

9.2 Addressing and transfer formats

Each device on the bus has its own unique address. Before any data is transmitted on the bus, the master transmits on the bus the address of the slave to be accessed for this transaction. A well-behaved slave with a matching address, if it exists on the network, should of course acknowledge the master's addressing. The addressing is done by the first byte transmitted by the master after the START condition.

An address on the network is seven bits long, appearing as the most significant bits of the address byte. The last bit is a direction (R/$\overline{W}$) bit. A '0' indicates that the master is transmitting (write) and a '1' indicates that the master requests data (read). A complete data transfer, comprised of an address byte indicating a 'write' and two data bytes is shown in Figure 13.

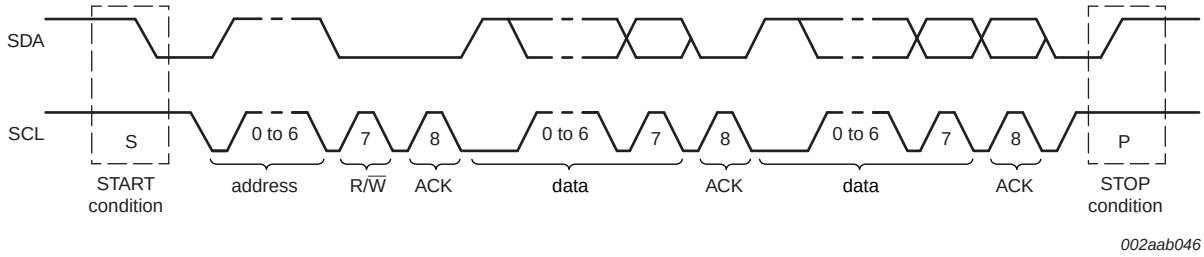


Fig 13. A complete data transfer

When an address is sent, each device in the system compares the first seven bits after the START with its own address. If there is a match, the device will consider itself addressed by the master, and will send an acknowledge. The device could also determine if in this transaction it is assigned the role of a slave receiver or slave transmitter, depending on the R/$\overline{W}$ bit.

Each node of the I²C-bus network has a unique seven-bit address. The address of a microcontroller is of course fully programmable, while peripheral devices usually have fixed and programmable address portions.

When the master is communicating with one device only, data transfers follow the format of Figure 13, where the R/$\overline{W}$ bit could indicate either direction. After completing the transfer and issuing a STOP condition, if a master would like to address some other device on the network, it could start another transaction by issuing a new START.

Another way for a master to communicate with several different devices would be by using a 'repeated START'. After the last byte of the transaction was transferred, including its acknowledge (or negative acknowledge), the master issues another START, followed by address byte and data—without effecting a STOP. The master may communicate with a number of different devices, combining 'reads' and 'writes'. After the last transfer takes place, the master issues a STOP and releases the bus. Possible data formats are demonstrated in Figure 14. Note that the repeated START allows for both change of a slave and a change of direction, without releasing the bus. We shall see later on that the change of direction feature can come in handy even when dealing with a single device.

In a single master system, the repeated START mechanism may be more efficient than terminating each transfer with a STOP and starting again. In a multimaster environment, the determination of which format is more efficient could be more complicated, as when a master is using repeated STARTs it occupies the bus for a long time and thus preventing other devices from initiating transfers.

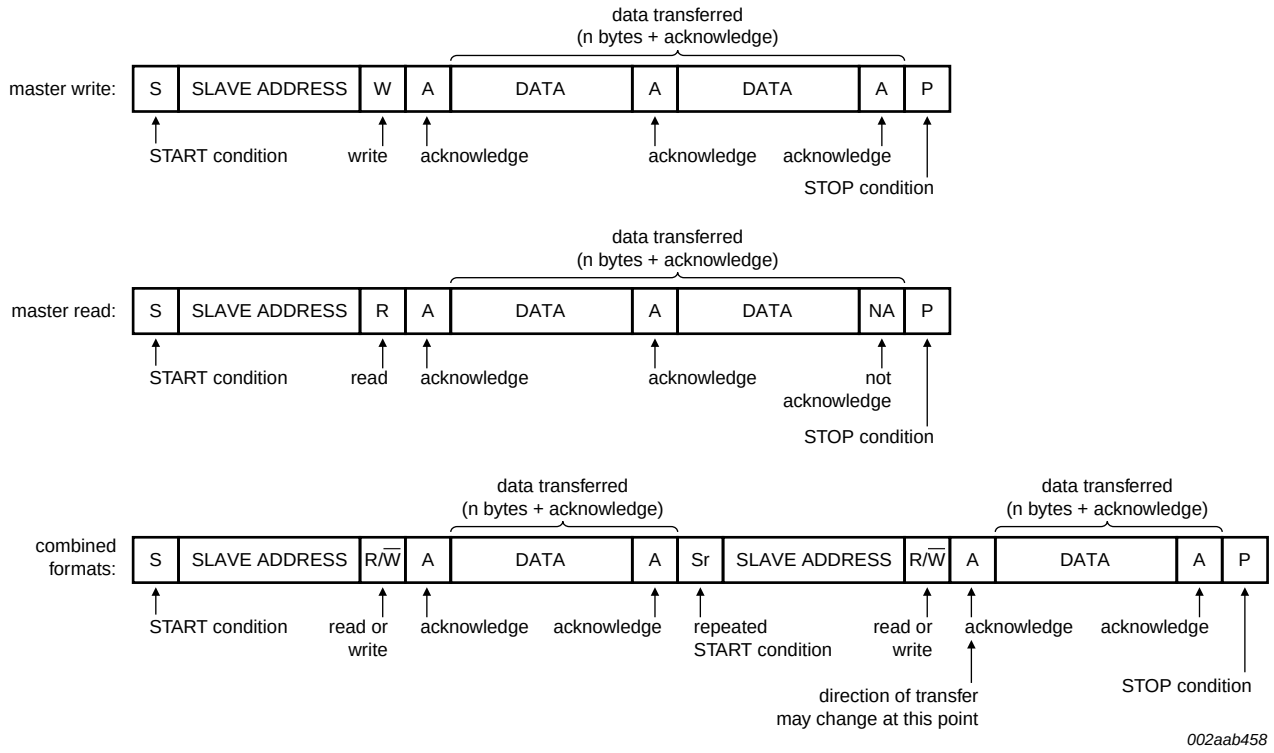


Fig 14. I²C-bus data formats

9.3 Addressing

Before any data is transmitted or received, the master must send the address of the receiver via the SDA line. The first byte after the START condition carries the address of the slave device and the read/write bit. Table 33 shows how the SC16IS850L's address can be selected by using A1 and A0 pins. For example, if these 2 pins are connected to V_{DD}, then the SC16IS850L's address is set to 0x90, and the master communicates with it through this address.

Table 33. SC16IS850L address map

A1	A0	SC16IS750/760 I²C addresses (hex)[1]
V_{DD}	V_{DD}	0x90 (1001 000X)
V_{DD}	V_{SS}	0x92 (1001 001X)
V_{SS}	V_{DD}	0x98 (1001 100X)
V_{SS}	V_{SS}	0x9A (1001 101X)

[1] X = logic 0 for write cycle; X = logic 1 for read cycle.

9.4 Use of subaddresses

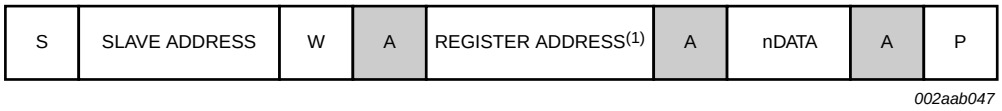
When a master communicates with the SC16IS850L it must send a subaddress in the byte following the slave address byte. This subaddress is the internal address of the word the master wants to access for a single byte transfer, or the beginning of a sequence of

locations for a multi-byte transfer. A subaddress is an 8-bit byte. Unlike the device address, it does not contain a direction (R/$\overline{W}$) bit, and like any byte transferred on the bus it must be followed by an acknowledge.

Table 34 shows the breakdown of the subaddress (register address) byte. Bits [2:0] are not used, bits [5:3] are used to select one of the device's internal registers, and bits [7:6] are not used.

A register write cycle is shown in Figure 15. The START is followed by a slave address byte with the direction bit set to 'write', a subaddress byte, a number of data bytes, and a STOP signal. The subaddress indicates which register the master wants to access, and the data bytes which follow will be written one after the other to the subaddress location.

Table 34 and Table 35 show the bits' presentation at the subaddress byte for I²C-bus and SPI interfaces. Bit 0 is not used, bits 2:1 select the channel, bits 6:3 select one of the UART internal registers. Bit 7 is not used with the I²C-bus interface, but it is used by the SPI interface to indicate a read or a write operation.



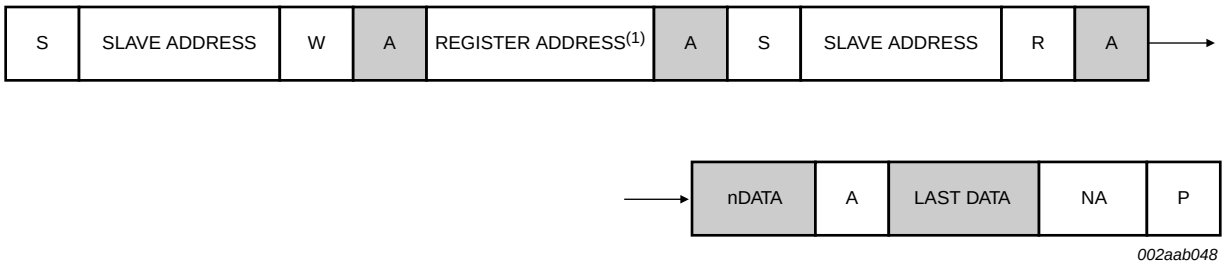
White block: host to SC16IS850L

Grey block: SC16IS850L to host

(1) See Table 34 for additional information.

Fig 15. Master writes to slave

The register read cycle (see Figure 16) commences in a similar manner, with the master sending a slave address with the direction bit set to 'write' with a following subaddress. Then, in order to reverse the direction of the transfer, the master issues a repeated START followed again by the device address, but this time with the direction bit set to 'read'. The data bytes starting at the internal subaddress will be clocked out of the device, each followed by a master-generated acknowledge. The last byte of the read cycle will be followed by a negative acknowledge, signalling the end of transfer. The cycle is terminated by a STOP signal.



White block: host to SC16IS850L

Grey block: SC16IS850L to host

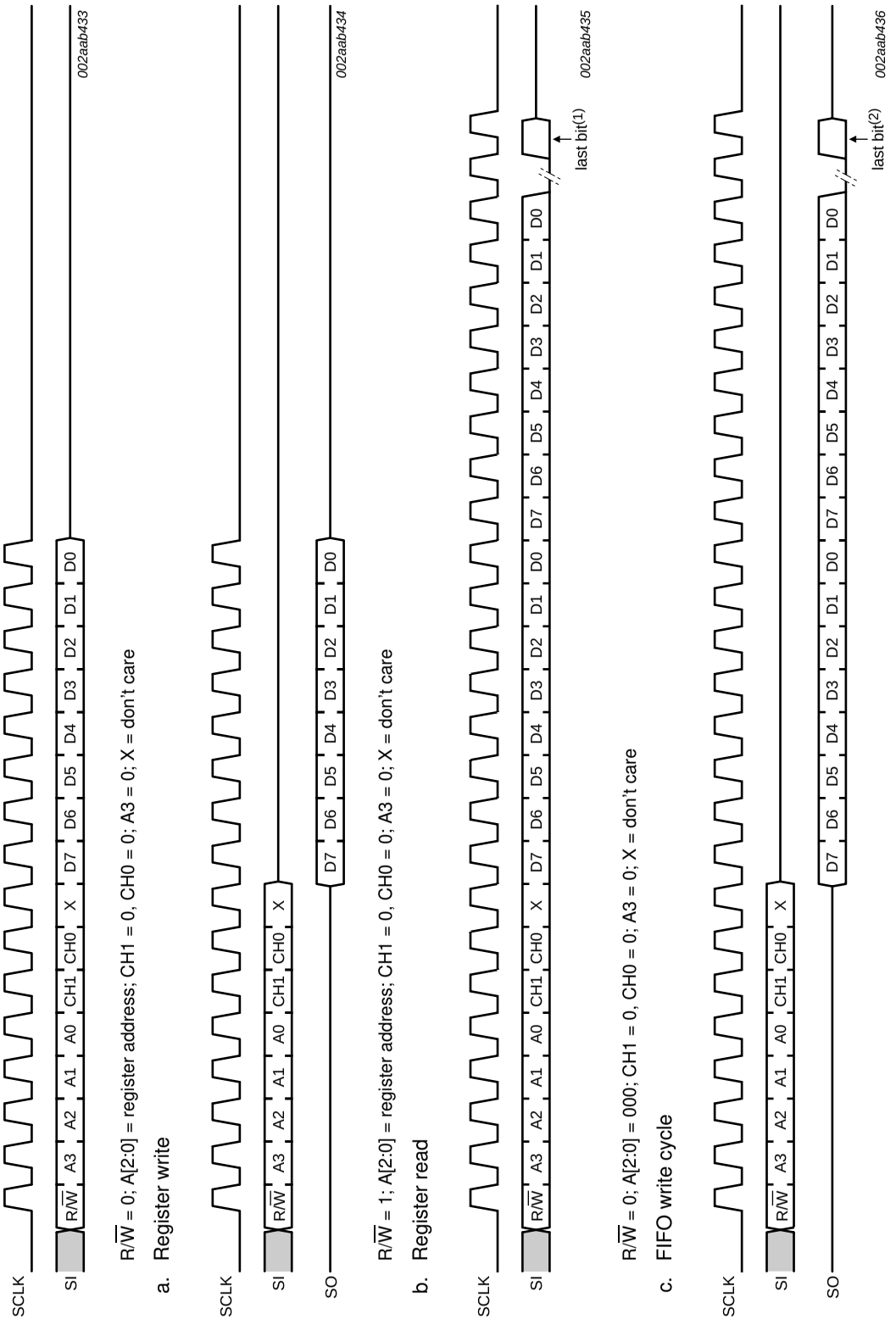
(1) See Table 34 for additional information.

Fig 16. Master read from slave

Table 34. Register address byte (I²C)

Bit	Name	Function
7:6	-	not used
5:3	A[2:0]	UART's internal register select
2:0	-	not used; set to 0

10. SPI operation



(1) Last bit (D0) of the last byte to be written to the transmit FIFO.

(2) Last bit (D0) of the last byte to be read from the receive FIFO.

Fig 17. SPI operation

Table 35. Register address byte (SPI)

Bit	Name	Function
7	R/$\overline{W}$	1: read from UART 0: write to UART
6	A3	not used; set to 0
5:3	A[2:0]	UART's internal register select
2:0	-	not used; set to 0

11. Limiting values

Table 36. Limiting values

In accordance with the Absolute Maximum Rating System (IEC 60134).

Symbol	Parameter	Conditions		Min	Max	Unit
V_{DD}	supply voltage			-	2.5	V
V_n	voltage on any other pin		[1]	V_{SS} − 0.3	V_{DD} + 0.3	V
T_{amb}	ambient temperature	operating in free air		−40	+85	°C
T_{stg}	storage temperature			−65	+150	°C
P_{tot}/pack	total power dissipation per package			-	500	mW

[1] V_{DD} should not exceed 2.5 V.

12. Static characteristics

Table 37. Static characteristics

T_{amb} = −40 °C to +85 °C; V_{DD} = 1.65 V to 1.95 V; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
Supplies						
V_{DD}	supply voltage		1.65	-	1.95	V
I_{DD}	supply current	operating; no load; f = 4 MHz	-	-	2	mA
Inputs I2C/$\overline{SPI}$, RX, $\overline{CTS}$, $\overline{DSR}$, $\overline{RI}$, $\overline{CD}$						
V_{IH}	HIGH-level input voltage		0.7 × V_{DD}	-	-	V
V_{IL}	LOW-level input voltage		-	-	0.3 × V_{DD}	V
I_L	leakage current		-	-	1	µA
C_i	input capacitance		-	-	3	pF
Outputs TX, $\overline{RTS}$, SO, $\overline{DTR}$						
V_{OH}	HIGH-level output voltage	I_{OH} = −800 µA	1.45	-	-	V
V_{OL}	LOW-level output voltage	I_{OL} = 2 mA	-	-	0.45	V
C_o	output capacitance		-	-	4	pF
Output $\overline{IRQ}$						
V_{OL}	LOW-level output voltage	I_{OL} = 1.6 mA	-		0.45	V
C_o	output capacitance		-		4	pF

Table 37. Static characteristics *...continued*

T_{amb} = −40 °C to +85 °C; V_{DD} = 1.65 V to 1.95 V; unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
I²C-bus input/output SDA						
V_{IH}	HIGH-level input voltage		0.7 × V_{DD}	-	-	V
V_{IL}	LOW-level input voltage		-	-	0.3 × V_{DD}	V
V_{OL}	LOW-level output voltage	I_{OL} = 1 mA	-	-	0.2	V
I_{LIL}	LOW-level input leakage current		-	-	1	μA
I_{LIH}	HIGH-level input leakage current		-	-	1	μA
C_o	output capacitance		-	-	7	pF
I²C-bus inputs SCL, $\overline{CS}$/A0, SI/A1						
V_{IH}	HIGH-level input voltage		0.7 × V_{DD}	-	-	V
V_{IL}	LOW-level input voltage		-	-	0.3 × V_{DD}	V
I_{LIL}	LOW-level input leakage current		-	-	1	μA
I_{LIH}	HIGH-level input leakage current		-	-	1	μA
C_i	input capacitance		-	-	7	pF
Clock input XTAL1[1]						
V_{IH}	HIGH-level input voltage		1.35	-	-	V
V_{IL}	LOW-level input voltage		-	-	0.3	V
I_{LIL}	LOW-level input leakage current		−30	-	+30	μA
I_{LIH}	HIGH-level input leakage current		−30	-	+30	μA
C_i	input capacitance		-		3	pF
Sleep current						
$I_{DD(sleep)}$	sleep mode supply current	inputs are at V_{DD} or ground	-		5	μA

[1] XTAL2 should be left open when XTAL1 is driven by an external clock.

13. Dynamic characteristics

Table 38. I²C-bus timing specifications[1]

All the timing limits are valid within the operating supply voltage, ambient temperature range and output load; T_{amb} = −40 °C to +85 °C; V_{DD} = 1.65 V to 1.95 V, and refer to V_{IL} and V_{IH} with an input voltage of V_{SS} to V_{DD}. All output load = 25 pF, except SDA output load = 400 pF.

Symbol	Parameter	Conditions	Standard mode I²C-bus		Fast mode I²C-bus		Unit
			Min	Max	Min	Max	
f_{SCL}	SCL clock frequency	[2]	0	100	0	400	kHz
t_{BUF}	bus free time between a STOP and START condition		4.7	-	1.3	-	µs
$t_{HD;STA}$	hold time (repeated) START condition		4.0	-	0.6	-	µs
$t_{SU;STA}$	set-up time for a repeated START condition		4.7	-	0.6	-	µs
$t_{SU;STO}$	set-up time for STOP condition		4.7	-	0.6	-	µs
$t_{HD;DAT}$	data hold time		0	-	0	-	ns
$t_{VD;ACK}$	data valid acknowledge time		-	0.6	-	0.6	µs
$t_{VD;DAT}$	data valid time	SCL LOW to data out valid	-	0.6	-	0.6	µs
$t_{SU;DAT}$	data set-up time		250	-	150	-	ns
t_{LOW}	LOW period of the SCL clock		4.7	-	1.3	-	µs
t_{HIGH}	HIGH period of the SCL clock		4.0	-	0.6	-	µs
t_f	fall time of both SDA and SCL signals		-	300	-	300	ns
t_r	rise time of both SDA and SCL signals		-	1000	-	300	ns
t_{SP}	pulse width of spikes that must be suppressed by the input filter		-	10	-	10	ns
t_{d2}	I²C-bus modem input interrupt valid time		0.2	-	0.2	-	µs
t_{d3}	I²C-bus modem input interrupt clear time		0.2	-	0.2	-	µs
t_{d6}	I²C-bus receive interrupt valid time		0.2	-	0.2	-	µs
t_{d7}	I²C-bus receive interrupt clear time		0.2	-	0.2	-	µs
t_{d8}	I²C-bus transmit interrupt clear time		1.0	-	0.5	-	µs
t_{d15}	SCL delay time after reset	[3]	3	-	3	-	µs
$t_{w(rst)}$	reset pulse width		3	-	3	-	µs

[1] A detailed description of the I²C-bus specification, with applications, is given in user manual UM10204: *"I²C-bus specification and user manual"*. This may be found at www.nxp.com/documents/user_manual/UM10204.pdf.

[2] Minimum SCL clock frequency is limited by the bus time-out feature, which resets the serial bus interface if SDA is held LOW for a minimum of 25 ms.

[3] 2 XTAL1 clocks or 3 µs, whichever is less.

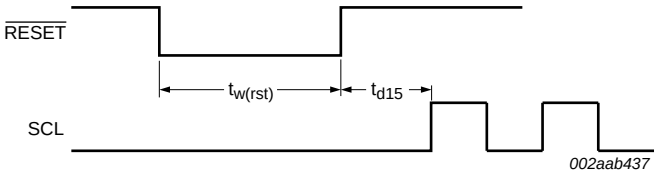
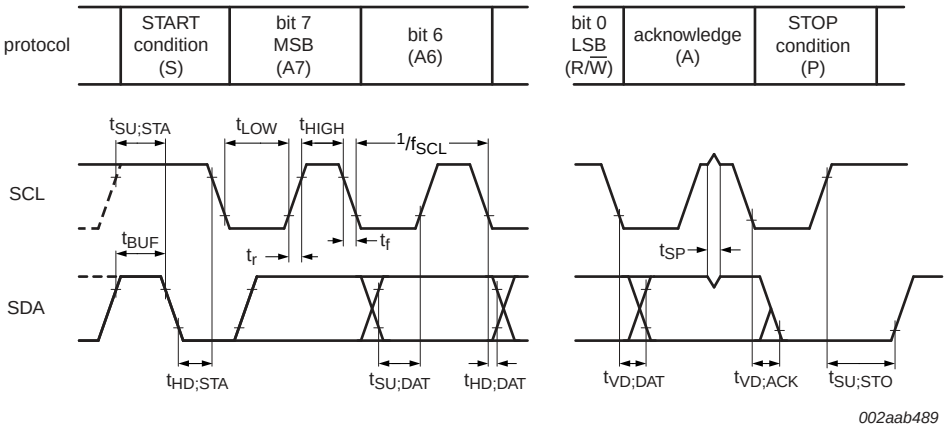


Fig 18. SCL delay after reset



Rise and fall times refer to V_{IL} and V_{IH}.

Fig 19. I²C-bus timing diagram

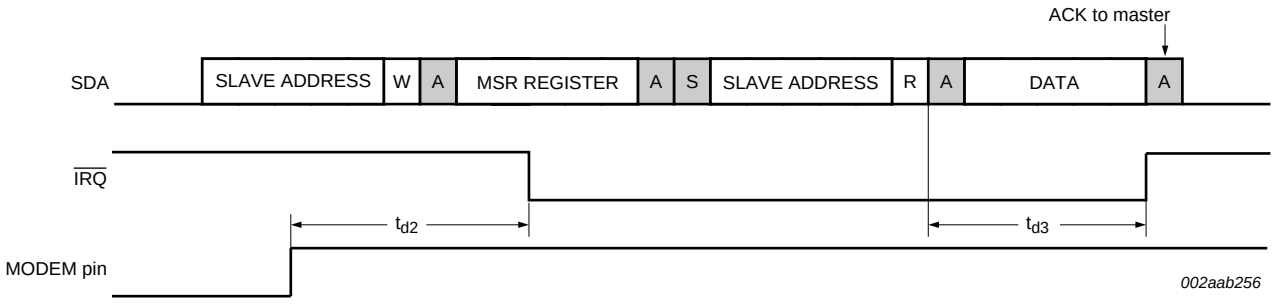


Fig 20. Modem input pin interrupt

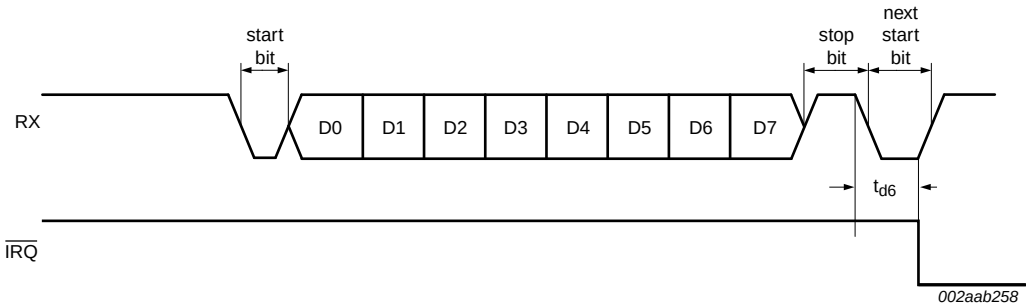


Fig 21. Receive interrupt

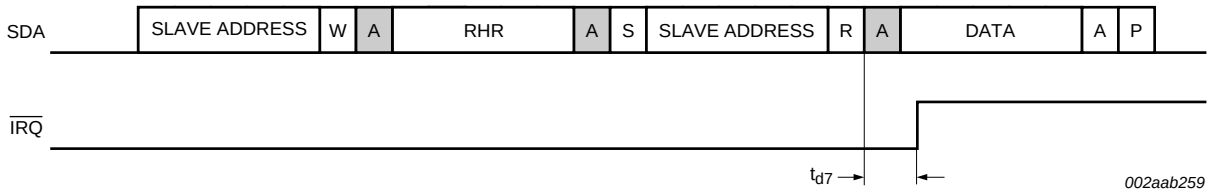


Fig 22. Receive interrupt clear

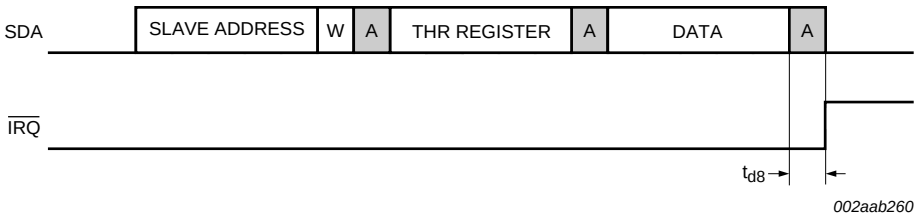


Fig 23. Transmit interrupt clear

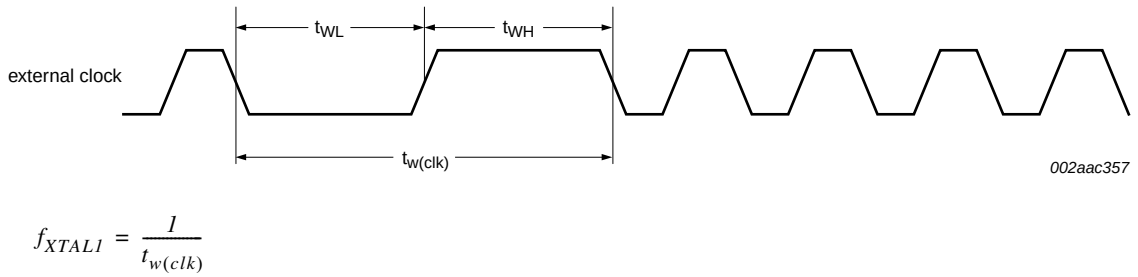
Table 39. f_{XTAL1} dynamic characteristics

T_{amb} = −40 °C to +85 °C; V_{DD} = 1.65 V to 1.95 V.

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
t_{WH}	pulse width HIGH			6	-	-	ns
t_{WL}	pulse width LOW			6	-	-	ns
f_{XTAL1}	frequency on pin XTAL1		[1][2]	-	-	80	MHz

[1] Applies to external clock, crystal oscillator max. 24 MHz.

[2] $$f_{XTAL1} = \frac{1}{t_{w(clk)}}$$



$$f_{XTAL1} = \frac{1}{t_{w(clk)}}$$

Fig 24. External clock timing

Table 40. SC16IS850L SPI-bus timing specifications

All the timing limits are valid within the operating supply voltage, ambient temperature range and output load; T_{amb} = −40 °C to +85 °C; V_{DD} = 1.65 V to 1.95 V, and refer to V_{IL} and V_{IH} with an input voltage of V_{SS} to V_{DD}. All output load = 25 pF, unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
t_{TR}	$\overline{CS}$ HIGH to SO 3-state delay time	C_L = 25 pF	-	-	50	ns
t_{CSS}	$\overline{CS}$ to SCLK setup time		10	-	-	ns
t_{CSH}	$\overline{CS}$ to SCLK hold time		10	-	-	ns
t_{DO}	SCLK fall to SO valid delay time	C_L = 25 pF	-	-	35	ns
t_{DS}	SI to SCLK setup time		10	-	-	ns
t_{DH}	SI to SCLK hold time		10	-	-	ns
t_{CP}	SCLK period	$t_{CL} + t_{CH}$	60	-	-	ns
t_{CH}	SCLK HIGH time		30	-	-	ns
t_{CL}	SCLK LOW time		30	-	-	ns
t_{CSW}	$\overline{CS}$ HIGH pulse width		80	-	-	ns
t_{d11}	SPI transmit interrupt clear time		200	-	-	ns
t_{d12}	SPI modem input interrupt clear time		200	-	-	ns
t_{d14}	SPI receive interrupt clear time		200	-	-	ns
$t_{w(rst)}$	reset pulse width		3	-	-	µs

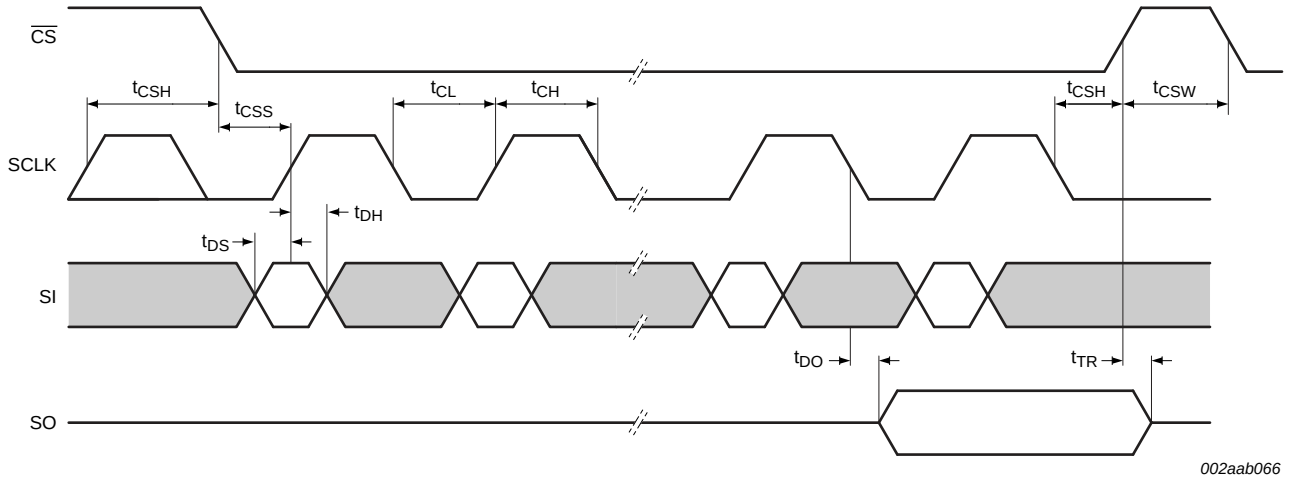
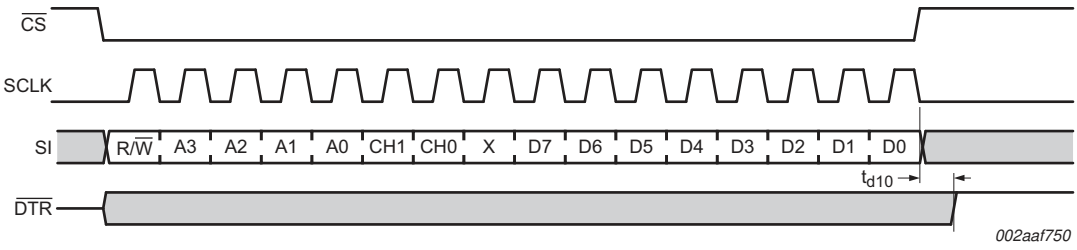
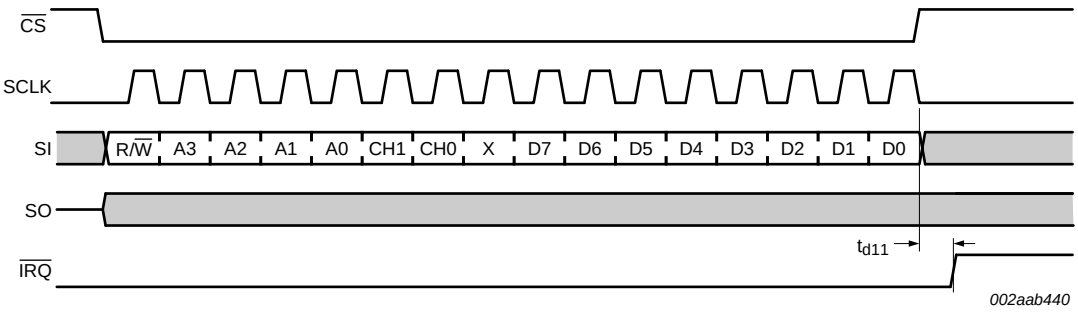


Fig 25. Detailed SPI-bus timing



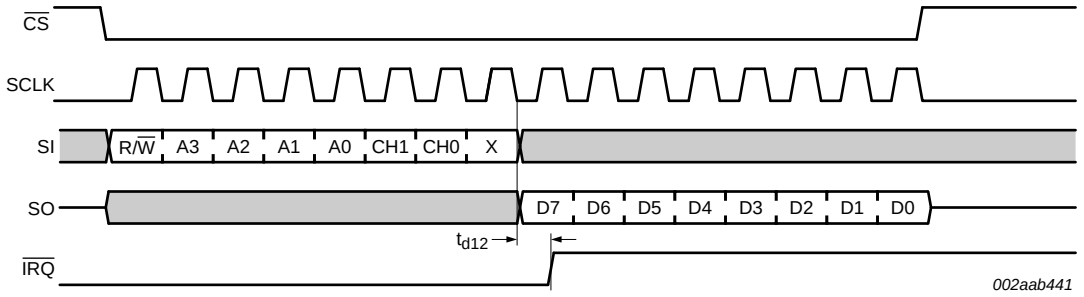
R/$\overline{W}$ = 0; A[3:0] = MCR (0x04); CH1 = 0; CH0 = 0

Fig 26. SPI write MCR to DTR output switch



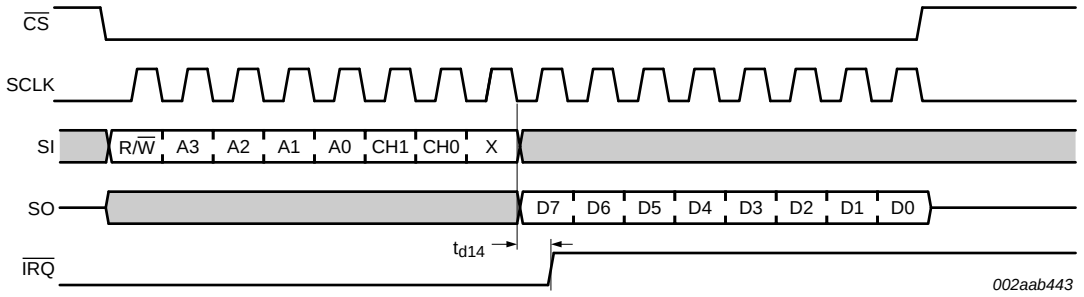
R/$\overline{W}$ = 0; A[3:0] = THR (0x00); CH1 = 0; CH0 = 0

Fig 27. SPI write THR to clear TX INT



R/W = 1; A[3:0] = MSR (0x06); CH1 = 0; CH0 = 0

Fig 28. Read MSR to clear modem INT



R/W = 1; A[3:0] = RHR (0x00); CH1 = 0; CH0 = 0

Fig 29. Read RHR to clear RX INT

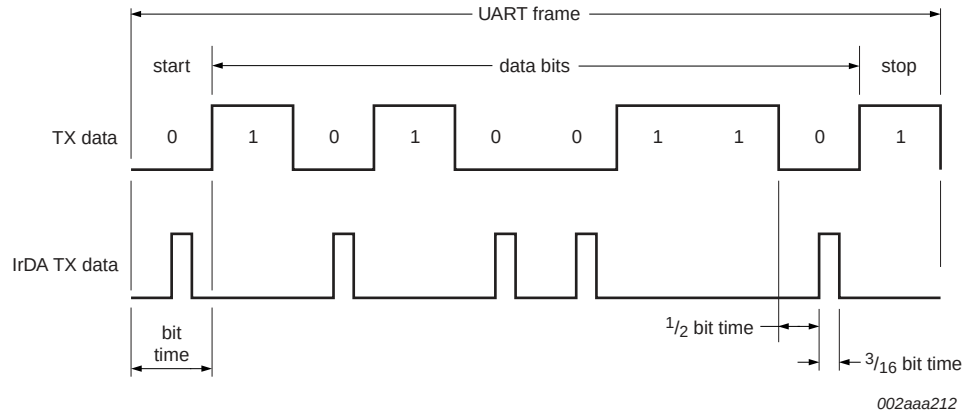


Fig 30. Infrared transmit timing

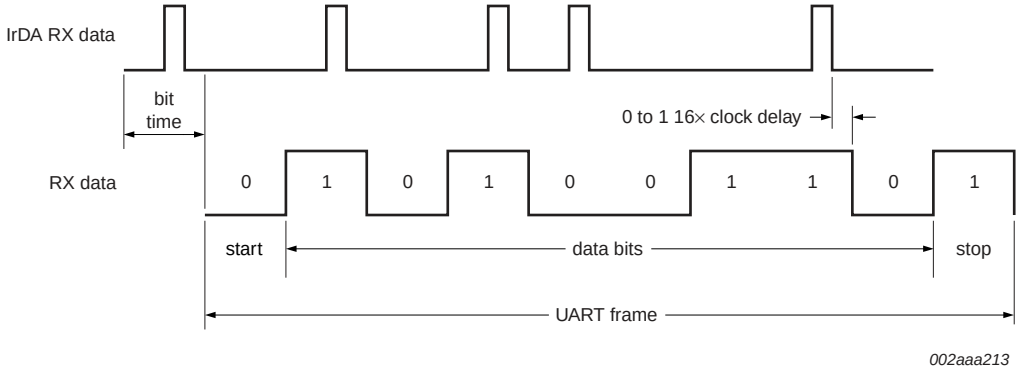
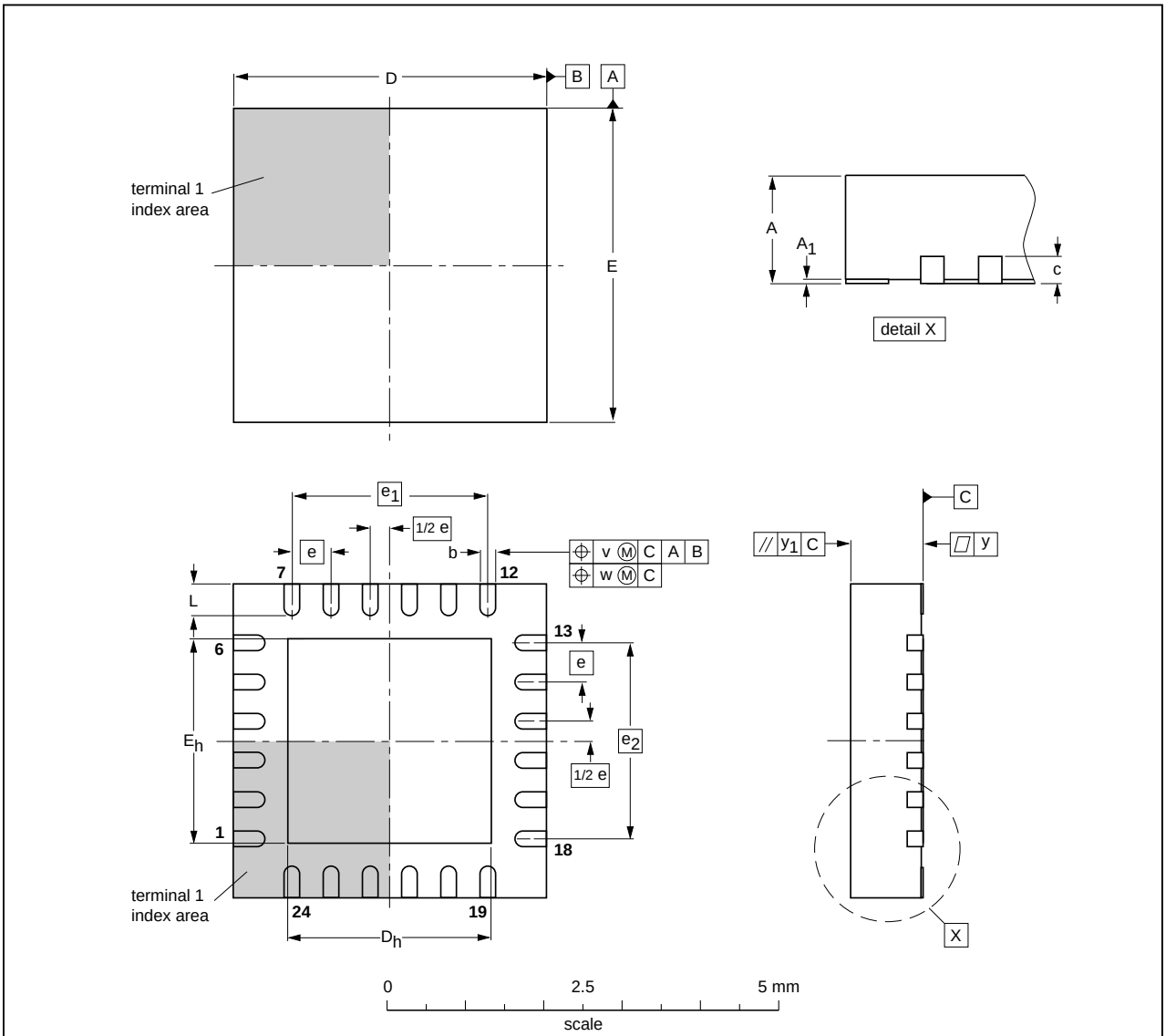


Fig 31. Infrared receive timing

14. Package outline

HVQFN24: plastic thermal enhanced very thin quad flat package; no leads; 24 terminals; body 4 x 4 x 0.85 mm **SOT616-3**



DIMENSIONS (mm are the original dimensions)

UNIT	A(1) max.	A_1	b	c	D(1)	D_h	E(1)	E_h	e	e_1	e_2	L	v	w	y	y_1
mm	1	0.05 0.00	0.30 0.18	0.2	4.1 3.9	2.75 2.45	4.1 3.9	2.75 2.45	0.5	2.5	2.5	0.5 0.3	0.1	0.05	0.05	0.1

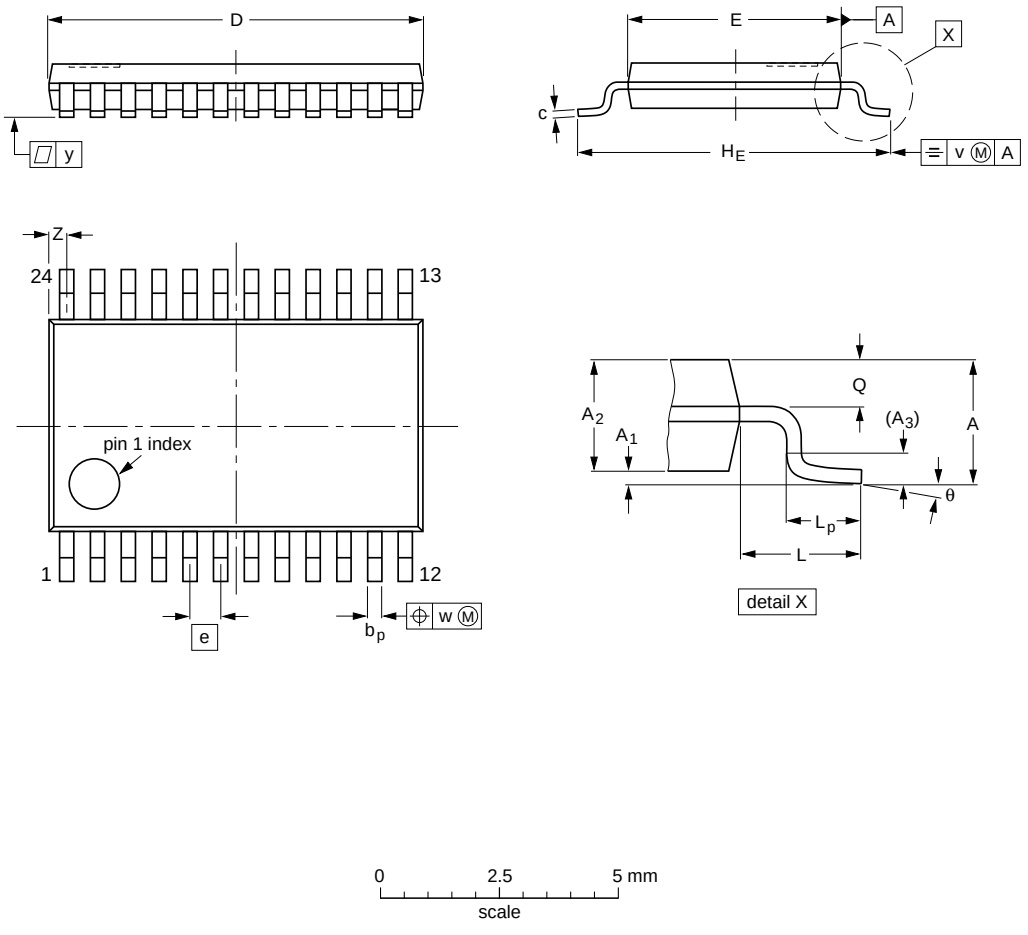
Note

1. Plastic or metal protrusions of 0.075 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT616-3	- - -	MO-220	- - -			~~04-11-19~~ 05-03-10

Fig 32. Package outline SOT616-3 (HVQFN24)

TSSOP24: plastic thin shrink small outline package; 24 leads; body width 4.4 mm **SOT355-1**



DIMENSIONS (mm are the original dimensions)

UNIT	A max.	A_1	A_2	A_3	b_p	c	$D^{(1)}$	$E^{(2)}$	e	H_E	L	L_p	Q	v	w	y	$Z^{(1)}$	θ
mm	1.1	0.15 0.05	0.95 0.80	0.25	0.30 0.19	0.2 0.1	7.9 7.7	4.5 4.3	0.65	6.6 6.2	1	0.75 0.50	0.4 0.3	0.2	0.13	0.1	0.5 0.2	8^o 0^o

Notes

1. Plastic or metal protrusions of 0.15 mm maximum per side are not included.
2. Plastic interlead protrusions of 0.25 mm maximum per side are not included.

OUTLINE VERSION	REFERENCES				EUROPEAN PROJECTION	ISSUE DATE
	IEC	JEDEC	JEITA			
SOT355-1		MO-153				~~99-12-27~~ 03-02-19

Fig 33. Package outline SOT355-1 (TSSOP24)

15. Soldering of SMD packages

This text provides a very brief insight into a complex technology. A more in-depth account of soldering ICs can be found in Application Note *AN10365 "Surface mount reflow soldering description"*.

15.1 Introduction to soldering

Soldering is one of the most common methods through which packages are attached to Printed Circuit Boards (PCBs), to form electrical circuits. The soldered joint provides both the mechanical and the electrical connection. There is no single soldering method that is ideal for all IC packages. Wave soldering is often preferred when through-hole and Surface Mount Devices (SMDs) are mixed on one printed wiring board; however, it is not suitable for fine pitch SMDs. Reflow soldering is ideal for the small pitches and high densities that come with increased miniaturization.

15.2 Wave and reflow soldering

Wave soldering is a joining technology in which the joints are made by solder coming from a standing wave of liquid solder. The wave soldering process is suitable for the following:

- Through-hole components
- Leaded or leadless SMDs, which are glued to the surface of the printed circuit board

Not all SMDs can be wave soldered. Packages with solder balls, and some leadless packages which have solder lands underneath the body, cannot be wave soldered. Also, leaded SMDs with leads having a pitch smaller than ~0.6 mm cannot be wave soldered, due to an increased probability of bridging.

The reflow soldering process involves applying solder paste to a board, followed by component placement and exposure to a temperature profile. Leaded packages, packages with solder balls, and leadless packages are all reflow solderable.

Key characteristics in both wave and reflow soldering are:

- Board specifications, including the board finish, solder masks and vias
- Package footprints, including solder thieves and orientation
- The moisture sensitivity level of the packages
- Package placement
- Inspection and repair
- Lead-free soldering versus SnPb soldering

15.3 Wave soldering

Key characteristics in wave soldering are:

- Process issues, such as application of adhesive and flux, clinching of leads, board transport, the solder wave parameters, and the time during which components are exposed to the wave
- Solder bath specifications, including temperature and impurities

15.4 Reflow soldering

Key characteristics in reflow soldering are:

- Lead-free versus SnPb soldering; note that a lead-free reflow process usually leads to higher minimum peak temperatures (see Figure 34) than a SnPb process, thus reducing the process window
- Solder paste printing issues including smearing, release, and adjusting the process window for a mix of large and small components on one board
- Reflow temperature profile; this profile includes preheat, reflow (in which the board is heated to the peak temperature) and cooling down. It is imperative that the peak temperature is high enough for the solder to make reliable solder joints (a solder paste characteristic). In addition, the peak temperature must be low enough that the packages and/or boards are not damaged. The peak temperature of the package depends on package thickness and volume and is classified in accordance with Table 41 and 42

Table 41. SnPb eutectic process (from J-STD-020C)

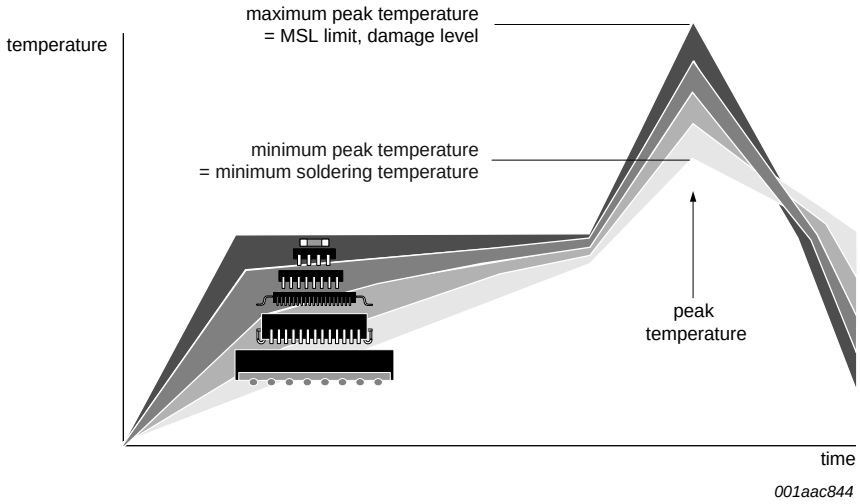
Package thickness (mm)	Package reflow temperature (°C)	
	Volume (mm³)	
	< 350	≥ 350
< 2.5	235	220
≥ 2.5	220	220

Table 42. Lead-free process (from J-STD-020C)

Package thickness (mm)	Package reflow temperature (°C)		
	Volume (mm³)		
	< 350	350 to 2000	> 2000
< 1.6	260	260	260
1.6 to 2.5	260	250	245
> 2.5	250	245	245

Moisture sensitivity precautions, as indicated on the packing, must be respected at all times.

Studies have shown that small packages reach higher temperatures during reflow soldering, see Figure 34.



MSL: Moisture Sensitivity Level

Fig 34. Temperature profiles for large and small components

For further information on temperature profiles, refer to Application Note *AN10365 "Surface mount reflow soldering description"*.

16. Abbreviations

Table 43. Abbreviations

Acronym	Description
CMOS	Complementary Metal-Oxide Semiconductor
CPU	Central Processing Unit
FIFO	First In, First Out
I²C-bus	Inter-Integrated Circuit-bus
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
SIR	Serial InfraRed
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter

17. Revision history

Table 44. Revision history

Document ID	Release date	Data sheet status	Change notice	Supersedes
SC16IS850L v.2	20120718	Product data sheet	-	SC16IS850L v.1
Modifications:	• Footnote removed from page 1.			
SC16IS850L v.1	20110722	Product data sheet	-	-

18. Legal information

18.1 Data sheet status

Document status[1][2]	Product status[3]	Definition
Objective [short] data sheet	Development	This document contains data from the objective specification for product development.
Preliminary [short] data sheet	Qualification	This document contains data from the preliminary specification.
Product [short] data sheet	Production	This document contains the product specification.

[1] Please consult the most recently issued document before initiating or completing a design.

[2] The term 'short data sheet' is explained in section "Definitions".

[3] The product status of device(s) described in this document may have changed since this document was published and may differ in case of multiple devices. The latest product status information is available on the Internet at URL http://www.nxp.com.

18.2 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

Short data sheet — A short data sheet is an extract from a full data sheet with the same product type number(s) and title. A short data sheet is intended for quick reference only and should not be relied upon to contain detailed and full information. For detailed and full information see the relevant full data sheet, which is available on request via the local NXP Semiconductors sales office. In case of any inconsistency or conflict with the short data sheet, the full data sheet shall prevail.

Product specification — The information and data provided in a Product data sheet shall define the specification of the product as agreed between NXP Semiconductors and its customer, unless NXP Semiconductors and customer have explicitly agreed otherwise in writing. In no event however, shall an agreement be valid in which the NXP Semiconductors product is deemed to offer functions and qualities beyond those described in the Product data sheet.

18.3 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information. NXP Semiconductors takes no responsibility for the content in this document if provided by an information source outside of NXP Semiconductors.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors and its suppliers accept no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Limiting values — Stress above one or more limiting values (as defined in the Absolute Maximum Ratings System of IEC 60134) will cause permanent damage to the device. Limiting values are stress ratings only and (proper) operation of the device at these or any other conditions above those given in the Recommended operating conditions section (if present) or the Characteristics sections of this document is not warranted. Constant or repeated exposure to limiting values will permanently and irreversibly affect the quality and reliability of the device.

Terms and conditions of commercial sale — NXP Semiconductors products are sold subject to the general terms and conditions of commercial sale, as published at http://www.nxp.com/profile/terms, unless otherwise agreed in a valid written individual agreement. In case an individual agreement is concluded only the terms and conditions of the respective agreement shall apply. NXP Semiconductors hereby expressly objects to applying the customer's general terms and conditions with regard to the purchase of NXP Semiconductors products by customer.

No offer to sell or license — Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from competent authorities.

Non-automotive qualified products — Unless this data sheet expressly states that this specific NXP Semiconductors product is automotive qualified, the product is not suitable for automotive use. It is neither qualified nor tested in accordance with automotive testing or application requirements. NXP Semiconductors accepts no liability for inclusion and/or use of non-automotive qualified products in automotive equipment or applications.

In the event that customer uses the product for design-in and use in automotive applications to automotive specifications and standards, customer (a) shall use the product without NXP Semiconductors' warranty of the product for such automotive applications, use and specifications, and (b) whenever customer uses the product for automotive applications beyond NXP Semiconductors' specifications such use shall be solely at customer's own risk, and (c) customer fully indemnifies NXP Semiconductors for any liability, damages or failed product claims resulting from customer design and use of the product for automotive applications beyond NXP Semiconductors' standard warranty and NXP Semiconductors' product specifications.

18.4 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

I²C-bus — logo is a trademark of NXP B.V.

19. Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

20. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 18 July 2012
Document identifier: SC16IS850L