# Quard Uart with 256-Byte FIFO

**IN16C1054**

## 1. Functional Description

IN16C1054 is a quad UART(Universal Asynchronous Receiver/Transmitter) with 256-byte FIFO supporting maximum communication speed of 5.3Mbps. It offers flow control function by hardware or software and signal lines which can open or close the Tx/Rx input/output when communicating by RS-422 or RS-485. It can handle four interrupt signals (INT0, INT1, INT2 and INT3) with one global interrupt signal line (GINT) and offers a new 'Xoff re-transmit' function in addition to Xon any character.
UART can convert 8-bit parallel data to asynchronous serial data and vice versa. It can transmit 5 to 8-bit letters, program I/O interrupt trigger level and has 256-byte I/O data FIFO.
UART can generate any baud rate using clock and programmable divisor, transmit data with even, odd or no parity and 1, 1.5, 2 stop bit, and detect break, idle, framing error, FIFO overflow and parity error in input data.
UART has a software interface for modem controlling.

IN16C1054 offers TQFP80 and PLCC68 packages.

## 2. Features

- 4 Channel UART
- 3.3V Operation
- 5V Tolerant Inputs
- **Pin-to-pin Compatible with Industry Standard SB16C554 with Additional**
- **Enhancements**
- Up to 5.3 Mbps Baud Rate (Up to 85 MHz Oscillator Input Clock)
- 256-byte Transmit FIFO
- 256-byte Receive FIFO with Error Flags
- Industrial Temperature Range (-20 ℃ to +85 ℃)
- Programmable and Selectable Transmit and Receive FIFO Trigger Levels for DMA
- and Interrupt Generation
- Software (Xon/Xoff) / Hardware (RTS#/CTS#) Flow Control
  - Programmable Xon/Xoff Characters
  - Programmable Auto-RTS and Auto-CTS
- Global Interrupt Mask/Poll Control
- Optional Data Flow Resume by Xon Any Character Control
- Optional Data Flow Additional Halt by Xoff Re-transmit Control
- RS-422 Point to Point/Multi-Drop Control
- RS-485 Echo/Non Echo Control
- DMA Signaling Capability for Both Received and Transmitted Data
- Software Selectable Baud Rate Generator
- Prescaler Provides Additional Divide-by-4 Function
- Fast Data Bus Access Time
- Programmable Sleep Mode
- Programmable Serial Interface Characteristics
  - 5, 6, 7, or 8-bit Characters
  - Even, Odd, or No Parity Bit Generation and Detection
  - 1, 1.5, or 2 Stop Bit Generation
- False Start Bit Detection
- Line Break Generation and Detection
- Fully Prioritized Interrupt System Controls
- Modem Control Functions (RTS#, CTS#, DTR#, DSR#, DCD#, and RI#)

## 3. Ordering Information

**Table 1: Ordering Information**

| Part Number | Package | Operating Temperature Range | Device Status |
|---|---|---|---|
| IN16C1054-TQ | 80-Pin TQFP | -20 ℃ to +85 ℃ | Active |
| IN16C1054-PL | 68-Pin PLCC | -20 ℃ to +85 ℃ | Active |

## 4. Block Diagram

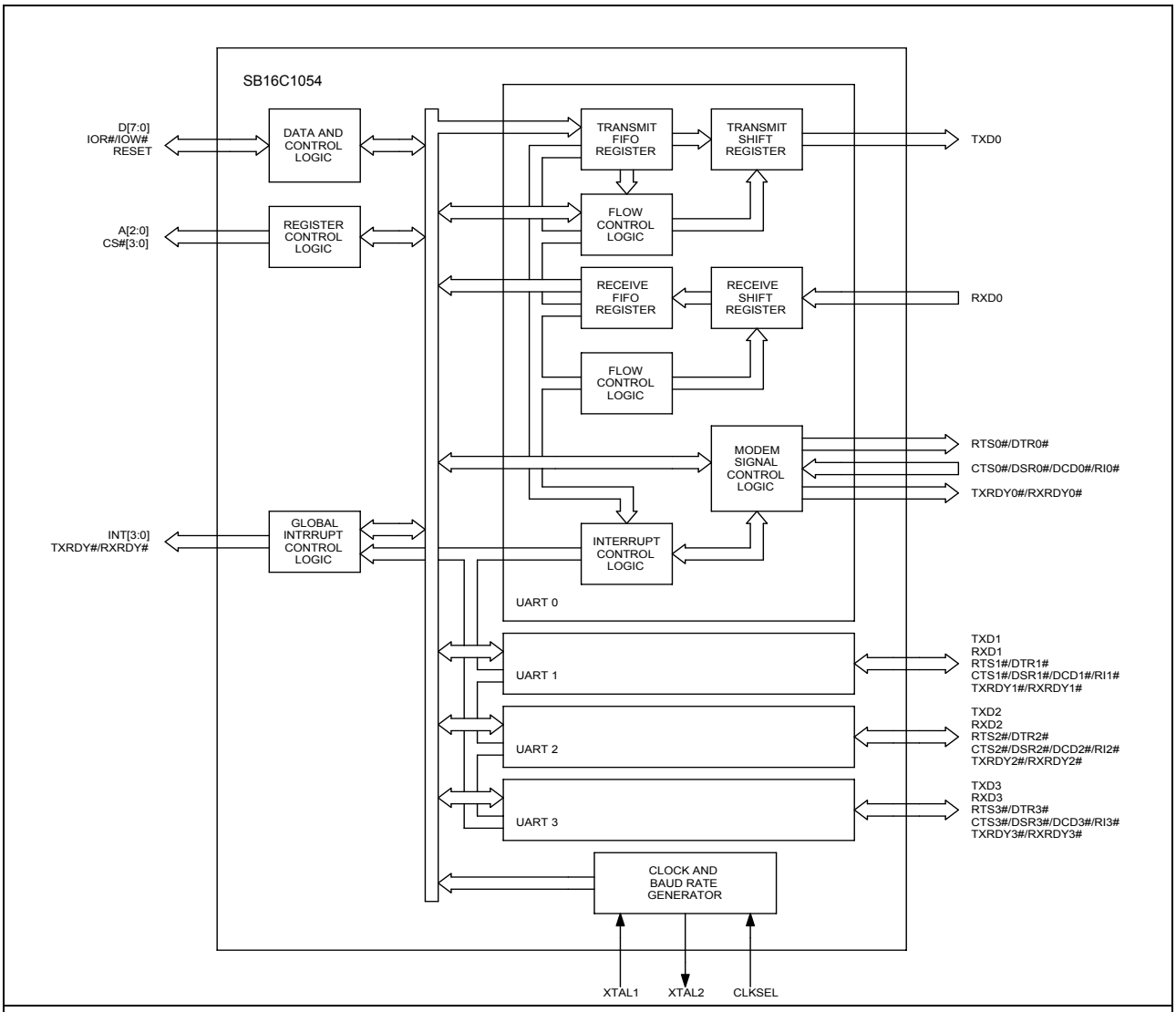



**Figure 1: Block Diagram**

## 5. Pin Configuration

### 5.1 Pin Configuration for 80-Pin TQFP Package

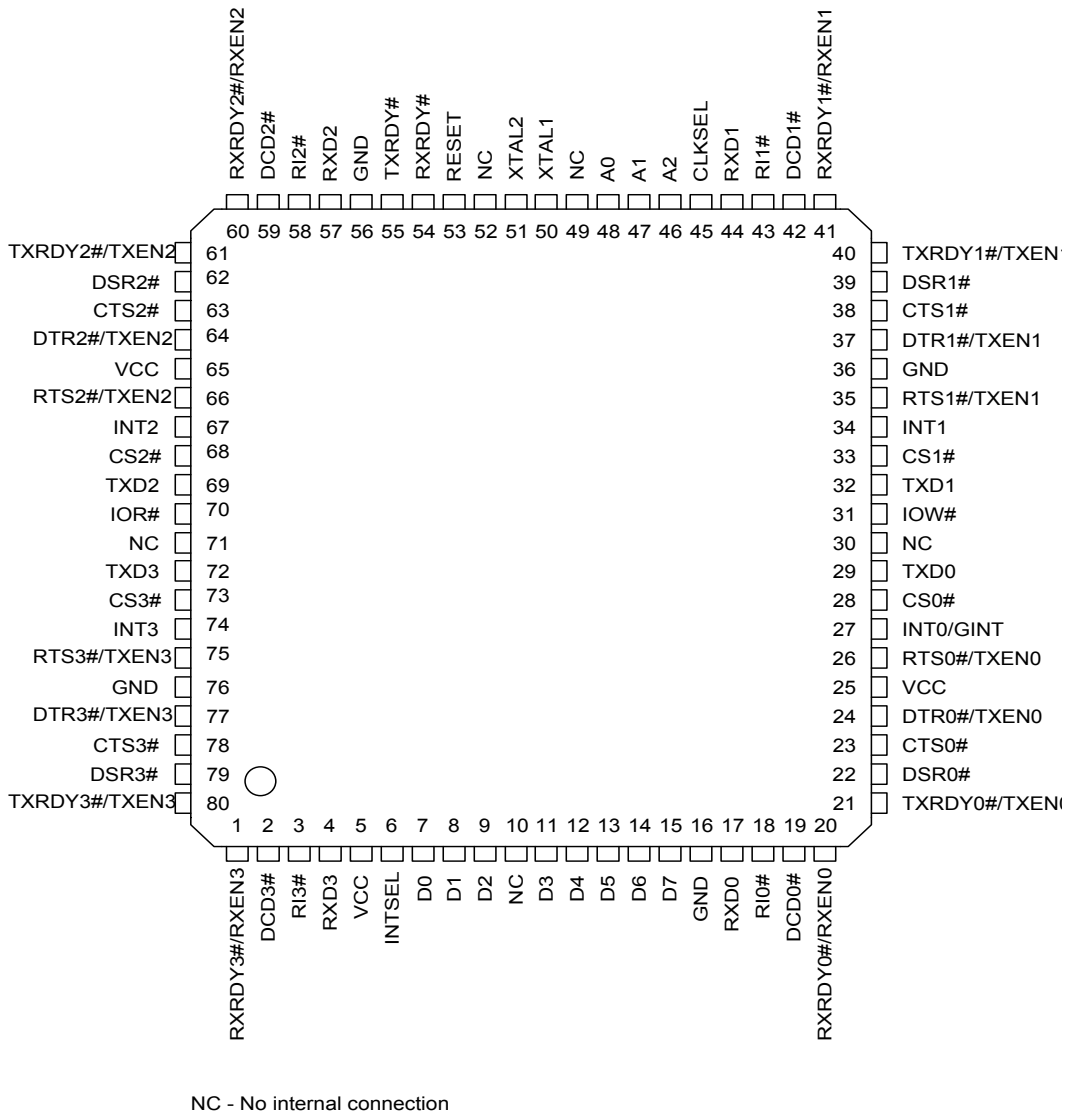



**Figure 2: 80-Pin TQFP Pin Configuration**

### 5.2 Pin Configuration for 68-Pin PLCC Package

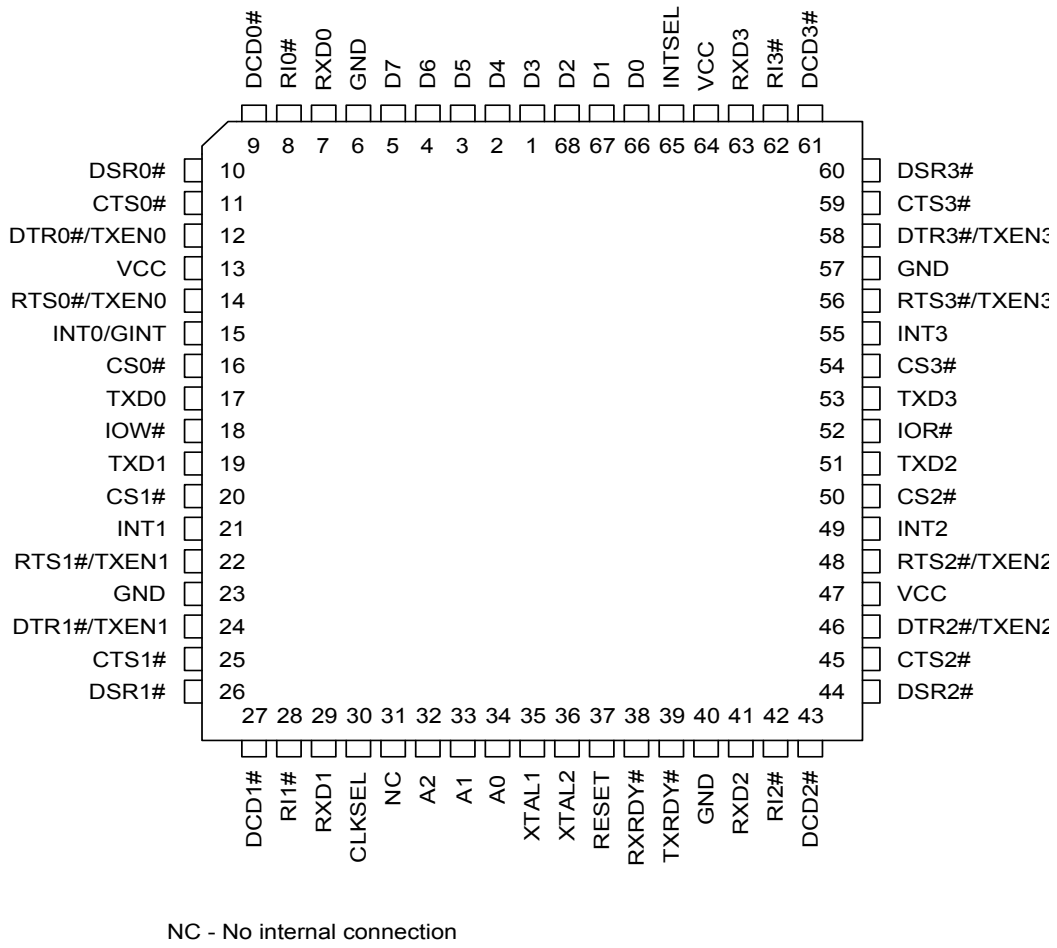



**Figure 3: 68-Pin PLCC Pin Configuration**

## 5.3 Pin Description

**Table 2: Pin Description**

| Data Bus Interface | | | | |
|---|---|---|---|---|
| **Name** | **Pin** | | **Type** | **Description** |
| | **TQFP80** | **PLCC68** | | |
| A0<br>A1<br>A2 | 48<br>47<br>46 | 34<br>33<br>32 | I<br>I<br>I | Address Bus Lines [2:0]. These 3 address lines select one of the internal registers in UART channel 0-3 during a data bus transaction. |
| D0<br>D1<br>D2<br>D3<br>D4<br>D5<br>D6<br>D7 | 7<br>8<br>9<br>11<br>12<br>13<br>14<br>15 | 66<br>67<br>68<br>1<br>2<br>3<br>4<br>5 | I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O<br>I/O | Data Bus Lines [7:0]. These pins are tri-state data bus for data transfer to or from the controlling CPU. |
| IOR# | 70 | 52 | I | Read Data (active low strobe). A valid low level on IOR# will load the data of an internal register defined by address lines A [2:0] onto the UART data bus for access by an external CPU. |
| IOW# | 31 | 18 | I | Write Data (active low strobe). A valid low level on IOW# will transfer the data from external CPU to an internal register that is defined by address lines A [2:0]. |
| CS0#<br>CS1#<br>CS2#<br>CS3# | 28<br>33<br>68<br>73 | 16<br>20<br>50<br>54 | I<br>I<br>I<br>I | Chip Select 0, 1, 2, and 3 (active low). These pins enable data transfers between the external CPU and the UART for the respective channel. |
| INT0/GINT<br>INT1<br>INT2<br>INT3 | 27<br>34<br>67<br>74 | 15<br>21<br>49<br>55 | O<br>O<br>O<br>O | Interrupt 0/Global Interrupt, Interrupt 1, 2, and 3. These pins provide individual channel interrupts or global interrupt. INT0-3 are enabled when MCR[3] is set to '1' and AFR[4] is cleared to '0' (default state). But INT0 operates as GINT and INT1-INT3 are disabled when AFR[4] is set to '1'. INT0-3's asserted state is active high, but GINT's asserted state is determined by AFR[5]. GINT's asserted state is active high when AFR[5] is set to '1', and active low when AFR[5] is cleared to '0'. |
| INTSEL | 6 | 65 | I | Interrupt Select. When INTSEL is left open or low state, the tri-state interrupts available on INT0-3 are enabled by MCR[3]. But, when INTSEL is in high state, INT0-3 are always enabled. |

**Table 2: Pin Description…*continued***

| Name | Pin | | Type | Description |
|---|---|---|---|---|
| | TQFP80 | PLCC68 | | |
| TXRDY0#/TXEN0<br>TXRDY1#/TXEN1<br>TXRDY2#/TXEN2<br>TXRDY3#/TXEN3 | 21<br>40<br>61<br>80 | -<br>-<br>-<br>- | O<br>O<br>O<br>O | Transmitter Ready 0, 1, 2, and 3/Tx Enable 0, 1, 2, and 3.<br>These pins provide individual channel transmitter ready or transmit enable.<br>TXRDY0-3# are enabled when ATR[1:0] is cleared to '00' (default state). If ATR[1:0] are set to '11', TXRDY0-3# operate as TXEN0-3.<br>TXRDY0-3# (active low) are asserted by TX FIFO/THR status for transmit channels 0-3. TXEN0-3's asserted state is determined by ATR[5:4]. If ATR[4] is cleared to '0', the state holds the same value as ATR[5]. If ATR[4] is set to '1', it is the auto-toggling state based on ATR[5].<br>If these pins are unused, leave them unconnected. |
| RXRDY0#/RXEN0<br>RXRDY1#/RXEN1<br>RXRDY2#/RXEN2<br>RXRDY3#/RXEN3 | 20<br>41<br>60<br>1 | -<br>-<br>-<br>- | O<br>O<br>O<br>O | Receiver Ready 0, 1, 2, and 3/Rx Enable 0, 1, 2, and 3.<br>These pins provide individual channel receiver ready or receive enable.<br>RXRDY0-3# are enabled when ATR[1:0] is cleared to '00' (default state). If ATR[1:0] is set to '11', RXRDY0-3# are changed to RXEN0-3.<br>RXRDY0-3# (active low) are asserted by RX FIFO/RBR status for receive channels 0-3. RXEN0-3's asserted state is determined by ATR[7:6]. If ATR[6] is cleared to '0', the state holds the same value as ATR[7]. If ATR[6] is set to '1', it is the auto-toggling state based on ATR[7].<br>If these pins are unused, leave them unconnected. |
| TXRDY# | 55 | 39 | O | Transmitter Ready (active low). This is asserted by TX FIFO/THR status for transmit channels 0-3. |
| RXRDY# | 54 | 38 | O | Receiver Ready (active low). This is asserted by RX FIFO/RHR status for receive channels 0-3. |
| **Modem and Serial I/O Interface** | | | | |
| **Name** | **Pin** | | **Type** | **Description** |
| | **TQFP80** | **PLCC68** | | |
| TXD0<br>TXD1<br>TXD2<br>TXD3 | 29<br>32<br>69<br>72 | 17<br>19<br>51<br>53 | O<br>O<br>O<br>O | Transmit Data. These pins are individual transmit data output. During the local loop-back mode, the TXD output pin is disabled and TXD data is internally connected to the RXD input. |
| RXD0<br>RXD1<br>RXD2<br>RXD3 | 17<br>44<br>57<br>4 | 7<br>29<br>41<br>63 | I<br>I<br>I<br>I | Receive Data. These pins are individual receive data input. During the local loop-back mode, the RXD input pin is disabled and RXD data is internally connected to the TXD output. |

**Table 2: Pin Description…*continued***

| Name | Pin | | Type | Description |
|---|---|---|---|---|
| | TQFP80 | PLCC68 | | |
| RTS0# | 26 | 14 | O | Request to Send (active low). These pins indicate that the |
| RTS1# | 35 | 22 | O | UART is ready to send data to the modem, and affect |
| RTS2# | 66 | 48 | O | transmit and receive operations only when Auto-RTS |
| RTS3# | 75 | 56 | O | function is enabled. |
| CTS0# | 23 | 11 | I | Clear to Send (active low). These pins indicate the modem |
| CTS1# | 38 | 25 | I | is ready to accept transmitted data from the UART, and |
| CTS2# | 63 | 45 | I | affect transmit and receive operations only when Auto-CTS |
| CTS3# | 78 | 59 | I | function is enabled. |
| DTR0# | 24 | 12 | O | Data Terminal Ready (active low). These pins indicate |
| DTR1# | 37 | 24 | O | UART is ready to transmit or receive data. |
| DTR2# | 64 | 46 | O | |
| DTR3# | 75 | 58 | O | |
| DSR0# | 22 | 10 | I | Data Set Ready (active low). These pins indicate modem is |
| DSR1# | 39 | 26 | I | powered-on and is ready for data exchange with UART. |
| DSR2# | 62 | 44 | I | |
| DSR3# | 79 | 60 | I | |
| DCD0# | 29 | 17 | I | Carrier Detect (active low). These pins indicate that a carrier |
| DCD1# | 32 | 19 | I | has been detected by modem. |
| DCD2# | 69 | 51 | I | |
| DCD3# | 72 | 53 | I | |
| RI0# | 17 | 7 | I | Ring Indicator (active low). These pins indicate the modem |
| RI1# | 44 | 29 | I | has received a ringing signal from telephone line. A low to |
| RI2# | 57 | 41 | I | high transition on these input pins generates a modem |
| RI3# | 4 | 63 | I | status interrupt, if enabled. |
| **Other Interfaces** | | | | |
| **Name** | **Pin** | | **Type** | **Description** |
| | **TQFP80** | **PLCC68** | | |
| XTAL1 | 50 | 35 | I | Crystal or External Clock Input. |
| XTAL2 | 51 | 36 | O | Crystal or Buffered Clock Output. |
| CLKSEL | 45 | 30 | I | Clock Select. This pin selects the divide-by-1 or divide-by-4 prescalable clock. During the reset, The high on CLKSEL selects the divide-by-1 prescaler. The low on CLK selects the divide-by-4 prescaler. The inverting value of CLKSEL is latched into MCR[7] at the trailing edge of RESET. |
| RESET | 53 | 37 | I | Reset (active high). This pin will reset the internal registers and all the outputs. |
| VCC | 5, 25, 65 | 13, 47, 64 | I | Power Supply Input. 3.3V (2.7V ~ 3.6V) |
| GND | 16, 36, 56, 76 | 6, 23, 40, 57 | I | Signal and Power Ground. |
| NC | 10, 30, 47, 52, 71 | 31 | - | No Internal Connection. |

## 6. Functional Description

The IN16C1054 UART is pin-to-pin compatible with the TL16C554 and ST16C654 UARTs.

IN16C1054 offers 16C450 and 16C650 modes. When FIFO is enabled, it has a register configuration compatible with 64-byte FIFO and 16C654, so it becomes compatible with 16C654. If you enable 256-byte FIFO, you use the unique supreme function that IN16C1054 offers. It offers communication speed up to 5.3Mbps and more enhanced functions that other UARTs with 128-byte FIFO do not.

IN16C1054 can select hardware/software flow control. Hardware flow control significantly reduces software overhead and increases system efficiency by automatically controlling serial data flow using the RTS# output and CTS# input signals. Software flow control automatically controls data flow by using programmable Xon/Xoff characters.

### 6.1 FIFO Operation

IN16C1054's FIFO has two modes, 64-byte FIFO mode and 256-byte FIFO mode. Setting FCR[0] to '1' enables FIFO, and if AFR[0] is set to '0', it operates in 64-byte FIFO mode(default). In this mode, Transmit Data FIFO, Receive Data and Receive Status FIFO are 64 bytes. 64-byte FIFO mode allows you to select the Transmit Interrupt Trigger Level from 8, 16, 32, or 56. You can verify this Interrupt Trigger Level by TTR and RTR. In this mode TTR and RTR are Read Only.

And by FCR[5:4], XOFF Trigger Level can be selected to either 8, 16, 56, or 60, and XON Trigger Level to either 0, 8, 16, or 56 by FCR[7:6]. You can verify XON and XOFF Trigger Level by FUR and FLR. In 64-byte FIFO mode TTR and RTR are Read Only.

If you select 256-byte FIFO mode, you can experience more powerful features of IN16C1054. Setting both FCR[0] and AFR[0] to '1' will enable this mode. In this mode, Transmit Data FIFO, Receive Data and Receive Status FIFO are 256 bytes. Interrupt Trigger Level and XON, XOFF Trigger Level are controlled by TTR, RTR, FUR and FLR, not by FCR[7:4]. That is, TTR, RTR, FUR and FLR can both read and write. You can verify free space of Transmit FIFO and the number of characters received in Receive FIFO by TCR, RCR and ISR[7:6].

While TX FIFO is full, the value sent to THR by CPU disappears. And while RX FIFO is full, the data coming from external devices disappear as well, provided that flow control function is not used.

For more information, refer to Register Description.

### 6.2 Hardware Flow Control

Hardware flow control is executed by Auto-RTS and Auto-CTS. Auto-RTS and Auto-CTS can be enabled/disabled independently by programming EFR[7:6]. If Auto-RTS is enabled, it reports that it cannot receive more data by asserting RTS# when the amount of received data in RX FIFO exceeds the written value in FUR. Then after the data stored in RX FIFO is read by CPU, it reports that it can receive new data by deasseting RTS# when the amount of existing data in RX FIFO is less than the written value in FLR.

When Auto-CTS is enabled and CTS# is cleared to '0', transmitting data to TX FIFO has to be suspended because external device has reported that it cannot accept more data. When data transmission has been suspended and CTS# is set to '1', data in TX FIFO is retransmitted because external device has reported that it can accept more data. These operations prevent overrun during communication and if hardware flow control is disabled and transmit data rate exceeds RX FIFO service latency, overrun error occurs.

### 6.2.1 Auto-RTS

To enable Auto-RTS, EFR[6] should be set to ‘1’. Once enabled, RTS# outputs ‘0’. If the number of received data in RX FIFO is larger than the value stored in FUR, RTS# will be changed to ‘1’ and if not, holds ‘0’. This state indicates that RX FIFO can accept more data. After RTS# changed to ‘1’ and reported to the CPU that it cannot accept more data, the CPU reads the data in RX FIFO and then the amount of data in RX FIFO reduces. When the amount of data in RX FIFO equals the value written in FLR, RTS# changes to ‘0’ and reports that it can accept more data. That is, if RTS# is ‘0’ now, RTS# is not changed to ‘1’ until the amount in RX FIFO exceeds the value set in FUR. But if RTS# is ‘1’ now, RTS# is not changed to ‘0’ until the amount in RX FIFO equals the value written in FLR.

The value of FUR and FLR is determined by FIFO mode. If FCR[7:6] holds ‘00’, ’01’, ‘10’, and ‘11’, FUR stores 8, 16, 56, and 60, respectively. And if FCR[5:4] holds ‘00’, ’01’, ‘10’, and ‘11’, FLR stores 0, 8, 16, and 56, respectively in 64-byte FIFO. In 256-byte FIFO mode, users can write FUR and FLR values as they want and use them. But the value of FUR must be larger than that of FLR. While Auto-RTS is enabled, you can verify if RTS# is ‘0’ or ‘1’ by FSR[5]. If FSR[5] is ‘0’, RTS# is ‘0’ and if ‘1’, RTS# is ‘1’, too.

When IER[6] is set to ‘1’ and RTS# is changed from ‘0’ to ‘1’ by Auto-RTS function, interrupt occurs and it is displayed on ISR[5:0]. Interrupts by Auto-RTS function are removed if MSR is read. RTS# is changed from ‘0’ to ‘1’ after the first STOP bit is received. Figure 4 shows the RTS# timing chart while Auto-RTS is enabled.

In Figure 4, Data Byte n-1 is received and RTS# is deasserted when the amount of data in RX FIFO is larger than the value written in FUR. UART completes transmitting new data (DATA BYTE n) which has started being transmitted even though external UART recognizes RTS# has been deasserted. After that, the device stops transmitting more data. If CPU reads data of RX FIFO, the value of RCR decreases and then if that value equals that of FLR, RTS# is asserted for external UART to transmit new data.

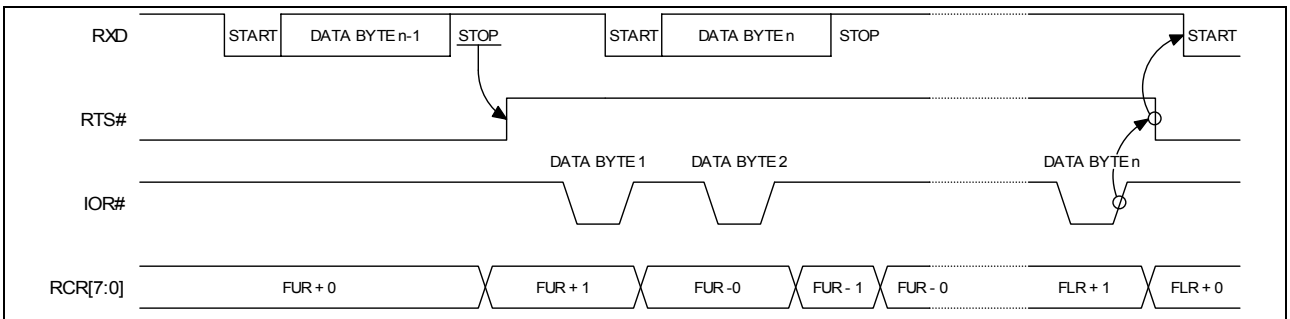



**Figure 4: RTS# Functional Timing**

### 6.2.2 Auto-CTS

Setting EFR[7] to ‘1’ enables Auto-RTS. If enabled, data in TX FIFO are determined to be transmitted or suspended by the value of CTS#. If ‘0’, it means external UART can receive new data and data in TX FIFO are transmitted through TXD pin. If ‘1’, it means external UART can not accept more data and data in TX FIFO are not transmitted. But data being transmitted by then complete transmission. These procedures are performed irrespective of FIFO modes. While Auto-CTS is enabled, you can verify the input value of CTS# by FSR[1]. If ‘0’, CTS# is ‘0’ and it means external UART can accept new data, If ’1’, CTS# is ‘1’ and it means external UART can not accept more data and data in TX FIFO are not being transmitted. If IER[7] is set to ‘1’, interrupt is generated by Auto-CTS when the input of CTS# is changed from ‘0’ to ‘1’, and it is shown on ISR[5:0]. Interrupts generated by Auto-CTS are removed if MSR is read.

### 6.3 Software Flow Control

Software flow control is performed by Xon and Xoff character transmitting/accepting. Software flow control is enabled/disabled independently by programming EFR[3:0] and MCR[6:5, 2]. If TX software flow control is enabled by EFR[3:2], Xoff character is transmitted to report that data can not be accepted when the stored amount of data in RX FIFO exceeds the value in FUR. After the CPU reads the data in RX FIFO and if the read amount is less than the value in FLR, Xon character is transmitted to report that more data can be accepted. If TX software flow control is enabled by EFR[1:0] and Xoff character is inputted through RXD pin, it means no more data can be accepted, and data transmission is suspended even though data are in TX FIFO. If Xon character is received through RXD pin while data transmission is suspended, it means more data can be accepted, and therefore data in TX FIFO are re-transmitted. These procedures prevent overruns during communication. If software flow control is disabled, overrun occurs when the transmit data rate exceeds RX FIFO service latency. Different combinations of software flow control can be enabled by setting different combinations of EFR[3:0] . Table 3 shows software flow control options.

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

| EFR[3] | EFR[2] | EFR[1] | EFR[0] | TX, RX software flow controls |
|---|---|---|---|---|
| 0 | 0 | X | X | No transmit control |
| 1 | 0 | X | X | Transmit Xon1/Xoff1 |
| 0 | 1 | X | X | Transmit Xon2/Xoff2 |
| 1 | 1 | X | X | Transmit Xon1, Xon2/Xoff1, Xoff2 |
| X | X | 0 | 0 | No receive flow control |
| X | X | 1 | 0 | Receiver compares Xon1/Xoff1 |
| X | X | 0 | 1 | Receiver compares Xon2/Xoff2 |
| X | X | 1 | 1 | Receiver compares Xon1, Xon2/Xoff1, Xoff2 |
| 0 | 0 | 0 | 0 | No transmit control, No receive flow control |
| 0 | 0 | 1 | 0 | No transmit control, Receiver compares Xon1/Xoff1 |
| 0 | 0 | 0 | 1 | No transmit control, Receiver compares Xon2/Xoff2 |
| 0 | 0 | 1 | 1 | No transmit control, Receiver compares Xon1, Xon2/Xoff1, Xoff2 |
| 1 | 0 | 0 | 0 | Transmit Xon1/Xoff1, No receive flow control |
| 1 | 0 | 1 | 0 | Transmit Xon1/Xoff1, Receiver compares Xon1/Xoff1 |
| 1 | 0 | 0 | 1 | Transmit Xon1/Xoff1, Receiver compares Xon2/Xoff2 |
| 1 | 0 | 1 | 1 | Transmit Xon1/Xoff1, Receiver compares Xon1, Xon2/Xoff1, Xoff2 |
| 0 | 1 | 0 | 0 | Transmit Xon2/Xoff2, No receive flow control |
| 0 | 1 | 1 | 0 | Transmit Xon2/Xoff2, Receiver compares Xon1/Xoff1 |
| 0 | 1 | 0 | 1 | Transmit Xon2/Xoff2, Receiver compares Xon2/Xoff2 |
| 0 | 1 | 1 | 1 | Transmit Xon2/Xoff2, Receiver compares Xon1, Xon2/Xoff1, Xoff2 |
| 1 | 1 | 0 | 0 | Transmit Xon2/Xoff2, No receive flow control |
| 1 | 1 | 1 | 0 | Transmit Xon2/Xoff2, Xoff2, Receiver compares Xon1/Xoff1 |
| 1 | 1 | 0 | 1 | Transmit Xon1, Xon2/Xoff1, Xoff2, Receiver compares Xon2/Xoff2 |
| 1 | 1 | 1 | 1 | Transmit Xon1, Xon2/Xoff1, Xoff2, Receiver compares Xon1, Xon2/Xoff1, Xoff2 |

### 6.3.1 Transmit Software Flow Control

To make Transmit Software Flow Control enabled, EFR[3:2] must be set to '01', '10' or '11'. Unlike Auto-RTS in which '0' is outputted on RTS# when TX software flow control function is enabled, Xon character is not transmitted at first. If the amount of data in RX FIFO (written in ISR[6] and RCR) is less than the value in FUR, Xon character is not transmitted because Xon is in initial state. But if the amount of data in RX FIFO exceeds the value in FUR, Xoff character is transmitted immediately. Transmitting Xoff character means no more data can be accepted and after CPU reads data in RX FIFO, data in RX FIFO decreases. When the amount of data in RX FIFO is same as the value of FLR, Xon character is transmitted and it means reporting to external UART that it can accept more data. After transmitting Xoff character, Xon character is not transmitted until the amount of data in RX FIFO is same as the value of FLR.

The value of FLR is determined by FIFO mode. If FCR[7:6] is '00', '01', '10', and '11', FUR is 8, 16, 56, and 60, respectively. And if FCR[5:4] is '00', '01', '10', and '11', FLR is 0, 8, 16, and 56, respectively in 64-byte FIFO. In 256-byte FIFO mode, users can input values in FUR and FLR as they want and use them. But the value in FUR must be larger than that of FLR. While TX software flow control is active, its status (if Xon or Xoff) can be verified by FSR[4]. If FSR[4] is '0', the status is Xon and if '1', the status is Xoff. It can be verified by FSR[4] only. And for there is no condition to generate interrupt, interrupt doesn't occur. It is different from that interrupt is generated by IER[5] when RX software flow control is enabled.

### 6.3.2 Receive Software Flow Control

To make Receive Software Flow Control enabled, EFR[1:0] must be set to '01', '10' or '11'. When enabled, data in TX FIFO are determined to be transmitted or suspended by incoming Xon/Xoff characters. If Xon character is received, it means external UART can accept new data, and data in TX FIFO are transmitted through TXD pin. If Xoff character is received, it means external UART can not accept more data, and data in TX FIFO are not transmitted. But data being transmitted by that time are completely transmitted. These procedures are performed irrespective of FIFO modes. While Receive Software Flow Control is enabled, you can verify if the RX Software Flow Control status is XON or XOFF by FSR[0]. If it is '0', RX Software Flow Control status is XON and it means external UART can accept new data. If '1', RX Software Flow Control status is XOFF and it means external UART can not accept more data and data in TX FIFO are not being transmitted. If IER[5] is set to '1', interrupt is generated when Xoff character is received and it is shown on ISR[5:0]. Interrupts generated by RX Software Flow Control are removed if ISR is read or Xon character is received.

General problems in using XON/XOFF function and tips for using Xon/Xoff character as one character are as follows.

- When RX Software Flow Control and Auto-CTS are enabled, LSR's Transmit Empty Bit and Transmit Holding Empty Bit are not affected even though RX Flow Control status is XOFF or '1' is inputted on CTS# pin, so data in TX FIFO are suspended. That is, these two bits are set to '1' if there is space available in TX FIFO.
- Xon/Xoff character which generated parity error are treated as normal Xon/Xoff character.
- If Xon and Xoff character are set to same, both characters are treated as Xon character.

Tips for using Xon/Xoff character as two characters are as follows.

- If received characters are Xon1, Xon1 and Xon2, RX flow control status becomes XON and previous Xon1 is ignored.
- If received characters are Xoff1, Xoff1 and Xoff2, RX flow control status becomes XOFF and previous Xoff1 is ignored.
- If received characters are repeated as Xon1 Xoff1, Xon1 and Xoff1, there is no effect in RX flow control status and these characters are not treated as data. But if received characters are Xon1 Xoff1, Xon1, Xoff1, Xon1 and Xon2, RX flow control status becomes XON.
- If received characters are Xon1 Xoff1, Xon1, Xoff1 and Xoff2, RX flow control status becomes XOFF.
- If Xon1 and Xoff1 characters do not precede Xon2 and Xoff2, Xon2 and Xoff2 are treated as data and stored in RX FIFO.
- If Xon1 is not accompanied with Xon2 or Xoff1 character, it is treated as data and stored in RX FIFO.
- If Xoff1 is not accompanied with Xoff2 or Xon1 character, it is treated as data and stored in RX FIFO.

  As seen before, if received characters are Xon1, Xoff2, Xon2 or Xoff1, Xon2, Xoff2, these characters are all treated as data and stored in RX FIFO.

If characters are arrived continuously like Xon1, Xon2 or Xoff1, Xoff2, descriptions are as follows.

- If Xon1, Xon2 characters and Xoff1, Xoff2 characters are same with each other, all characters are treated as normal XON and XOFF characters.
- If Xon1, Xoff1 characters and Xon2, Xoff2 characters are same with each other, these are treated as normal XON characters.
- If Xon1, Xon2, Xoff1 characters are same and Xoff2 is different, these are treated as normal XON, XOFF characters.
- If Xon1, Xon2, Xoff2 characters are same and Xoff1 is different, these are treated as normal XON, XOFF characters.
- If Xon2, Xoff1, Xoff2 characters are same and Xon1 is different, these are treated as normal XON, XOFF characters.
- If Xon1, Xoff1, Xoff2 characters are same and Xon2 is different, these are treated as normal XON, XOFF characters.
- If Xon2, Xoff1 characters are same and Xon1, Xoff2 are different, these are treated as normal XON, XOFF characters.
- If Xon1, Xon2, Xoff1, Xoff2 are all same, these are treated only as normal XON characters.

  In all these cases no XON/XOFF characters are treated as data.
  Refer to Table 4 below.

**Table 4: Xon/Xoff Character Recognition Logic Table**

| Xon1 Char. | Xon2 Char. | Xoff1 Char. | Xoff2 Char. | Recognition of Xon Char. | Recognition of Xoff Char. |
|---|---|---|---|---|---|
| 11h | 11h | 13h | 13h | Yes | Yes |
| 11h | 13h | 11h | 13h | Yes | No |
| 11h | 11h | 11h | 13h | Yes | Yes |
| 11h | 11h | 13h | 11h | Yes | Yes |
| 11h | 13h | 13h | 13h | Yes | Yes |
| 11h | 13h | 11h | 11h | Yes | Yes |
| 11h | 13h | 13h | 14h | Yes | Yes |
| 11h | 11h | 11h | 11h | Yes | No |

In case XON/XOFF software flow control function and Xon Any function is enabled, descriptions are as follows.

If Xon, Xoff characters are used as one character,

- If Xoff character arrives during XON status, status changes to XOFF.
- If Xon character arrives during XOFF status, status changes to XON.
- If Xoff character arrives during XOFF status, status changes to XON but Xoff character is not treated as data.

If Xon, Xoff characters are used as two characters,

- If only Xon1 or Xon1 + Xon2 character arrives during Xoff status, status changes to Xon and all characters are not treated as data.
- If only Xon2 character arrives during Xoff status, status changes to Xon and Xon2 character is treated as data and stored in RX FIFO.
- If Xoff1 + Xoff2 character arrives during XON status, status changes to XON.
- If Xoff1 + Xoff2 character arrives during XOFF status, status is changed to XON by Xoff1 and changed to XOFF again by Xoff2.

In case Software flow control function and Special character function is enabled, descriptions are as follows.

- If Xoff1 character is used as Software flow control character, character in Xoff2 Register is recognized as Special character.
- If Xoff2 character is used as Software flow control character, it is not recognized as Special character but as Xoff character because both are same.
- If Xoff1, Xoff2 character is sequential and Xoff1 + Xoff2 character is used as Software flow control character, it is not recognized as Special character but as Xoff2 character because both are same.
- If Xoff1 + Xoff2 character is used as Software flow control character and Xoff2 character which does not follow after Xoff1 character arrives, it is not recognized as Xoff2 character but as Special character even though both are same.

### 6.3.3 Xon Any Function

While RX Software flow control function is enabled, data in TX FIFO are transmitted when received Xon character and transmission is suspended when Xoff character is received. This status is called 'XOFF status'. Transmission is re-started when status changes to 'XON status' by incoming Xon character or Xon Any function that changes status when any data arrives. Xon Any function is enabled if MCR[5] is set to '1'. While it is enabled, XOFF status changes to XON status though Xoff character arrives.
Details about it are described in 6.3.2 Receive Software Flow Control.

### 6.3.4 Xoff Re-transmit Function

While TX Software flow control function is active, Xoff character is transmitted when the amount of data in RX FIFO exceeds the value of FUR. Though it received Xoff character, external UART may not recognize this character for some reason and continue to transmit data. Under TX Software flow control, because Xoff character had been transmitted once before, it is not transmitted again though more data arrive. In this situation, overflow may occur in RX FIFO. Conventional UARTs can not deal this situation but SB16C1054 does with Xoff Re-transmit function.

Xoff Re-transmit function transmits Xoff character again when more data arrives from external UART though it transmitted Xoff character before. By this function the external UART can recognize Xoff character and stop transmitting data though it didn't recognize the Xoff character before.

There are four Xoff Re-transmitting settings by XRCR[1:0]. Xoff character can be re-transmitted when every 1, 4, 8 or 16 data arrives in XOFF status.

If XRCR[1:0] is '00', Xoff character is re-transmitted whenever 1 more data arrives in XOFF status. If XRCR[1:0] is '01', Xoff character is re-transmitted whenever 4 more data arrives in XOFF status. If '10', 8 more data and if '11', 16 more data. If the value of FUR is approaching the FIFO size, 256-byte, it is good to write XRCR[1:0] '00'. If the 256-FUR value is small, it is good to select '00' of XRCR and if large, it is good to select '11'.

Xoff Re-transmit function is enabled by MCR[6] and MCR[2]. Change MCR[2] from OP1# function to Xoff Re-transmit function by setting MCR[6] to '1' and set MCR[2] to '1' again. Then Xoff Re-transmit function is enabled. When disabling it, first set MCR[6] to '1' and then clear MCR[2] to '0'.

### 6.4 Interrupts

As there are four independent 1-channel UARTs in SB16C1054, so there are four interrupts. Interrupts are assigned INT0, INT1, INT2, and INT3 for each channel. Each interrupt has six prioritized level's interrupt generation capability. The IER enables each of the six types of interrupts and INT signal in response to an interrupt generation. When an interrupt is generated, the ISR indicates that an interrupt is pending and provides the type of interrupt. And SB16C1054 can handle for four interrupts with one global interrupt. Global interrupt treats four of each interrupt as one interrupt, so it is useful when external system has few interrupt resource. Global interrupt line is also used as INT0, and it is determined by AFR[4] that which one is used. If AFR[4] is cleared to '0', INT0/GINT pin is selected as INT0 and if set to '1', GINT. When you treat four interrupts as one interrupt, you should use several additional functions. GICR determines whether global interrupt occurs or not. While GICR[0] is set to '1', an interrupt that is generated in four one-channel UARTs and treated as UNMASK is transmitted to GINT. But if GICR[0] is cleared to '0', an interrupt is not transmitted to GINT though interrupts are generated in four one-channel UARTs and treated as MASK. So this interrupt is not transmitted to external CPU. The status of global interrupt and generation of interrupts in one-channel UART can be verified by GISR. The value set in GICR[0] is reflected in GISR[7], so the status of mask of global interrupt can be verified. GISR[0] shows the status of interrupt of UART that is connected to CS0#. If GISR[0] is cleared to '0', it means that interrupt is not generated in the UART of CS0# and if set to '1', it means that interrupt is generated. The value of GISR[0] shows the status of interrupt generated in the UART of CS0#, irrespective of the value set in GICR[0]. GICR[0] determines whether the interrupts generated in four one-channel UARTs that is connected to CS0#, CS1#, CS2#, and CS3# are transmitted to external devices or not, but does not determine whether the interrupts are generated or not in UARTs. The value of output signal when an interrupt is generated in GINT pin is selected by AFR[5]. That is, GINT can determine the polarity of asserted status. If AFR[5] is cleared to '0', GINT outputs '0' when global interrupt is generated. And if set to '1', outputs '1' when global interrupt is generated.

## 6.5 DMA Operation

Transmitter and Receiver DMA operation is available through TXRDY#, RXRDY#, TXRDY[3:0]#, and RXRDY[3:0]#. There are two modes of DMA operation, DMA mode 0 or DMA mode 1, selected by FCR[3].

In DMA mode 0 or FIFO disable (FCR[3] = 0), DMA occurs in single character transfer. In DMA mode 1, multi-character DMA transfers are managed to relieve the CPU for longer periods of time.

### 6.5.1 Single DMA transfer (DMA Mode 0/FIFO Disable)

Transmitter: There are no character in TX FIFO or THR. And the TXRDY# and TXRDY[3:0]# signals will be in assert state. TXRDY#, TXRDY[3:0]# will switch to deassert state after one character is loaded into TX FIFO or THR.

Receiver: There is at least one character in RX FIFO or RHR. And the RXRDY# and RXRDY[3:0]# signals will be in assert state. Once RXRDY# is asserted, RXRDY[3:0]# signal will switch to deassert state when there are no more characters in RX FIFO or RBR.

Figure 5 shows TXRDY#, TXRDY[3:0]#, RXRDY#, and RXRDY[3:0]# in DMA mode 0/FIFO disable.

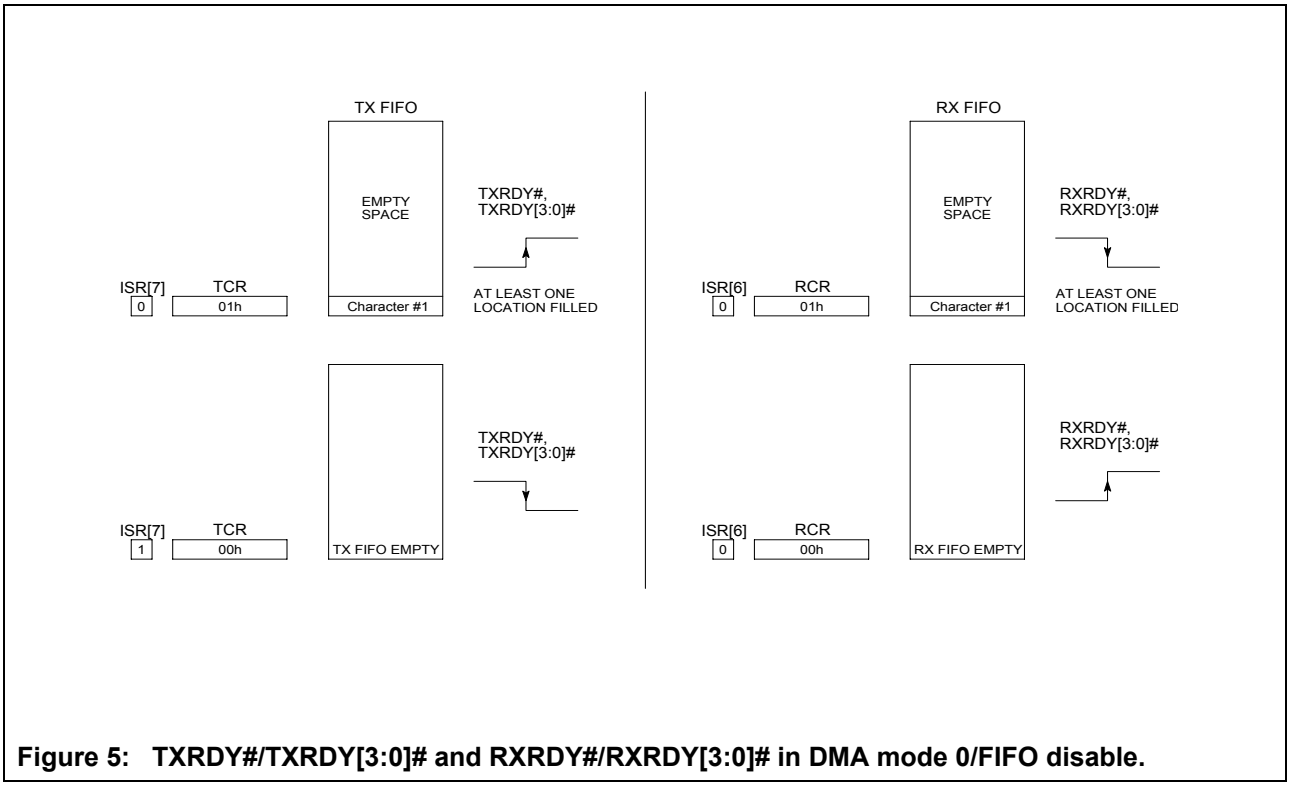



**Figure 5: TXRDY#/TXRDY[3:0]# and RXRDY#/RXRDY[3:0]# in DMA mode 0/FIFO disable.**

### 6.5.2 Block DMA transfer (DMA Mode 1)

Transmitter: When the characters in TX FIFO are less than the trigger level that is set in TTR, TXRDY# or TXRDY[3:0] signal is asserted. When TX FIFO is full, TXRDY# or TXRDY[3:0]# signal is deasserted.

Receiver: When the characters in RX FIFO are more than the trigger level that is set in RTR, RXRDY# or RXRDY[3:0] signal is asserted. When RX FIFO is empty, RXRDY# or RXRDY[3:0]# signal is deasserted.

The figure 6 below shows TXRDY#, TXRDY[3:0]# and RXRDY#, RXRDY[3:0]# in DMA mode 1.

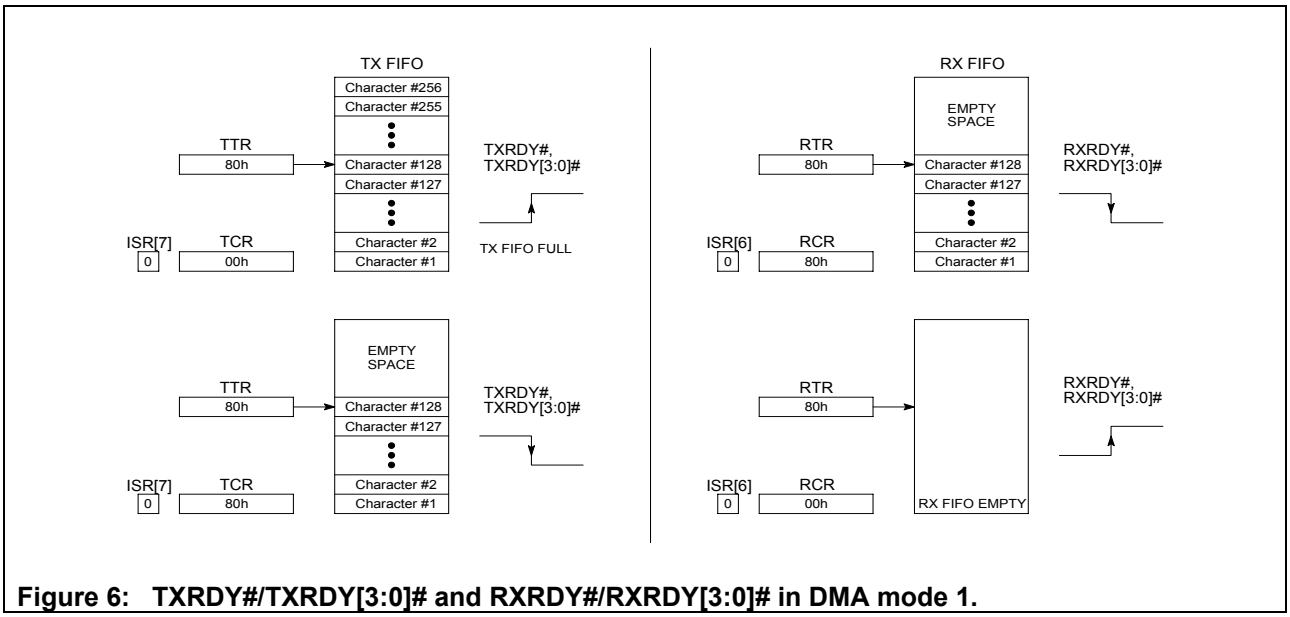



**Figure 6: TXRDY#/TXRDY[3:0]# and RXRDY#/RXRDY[3:0]# in DMA mode 1.**

## 6.6 Sleep Mode with Auto Wake-Up

The SB16C1054 provides sleep mode operation to reduce its power consumption when sleep mode is activated. Sleep mode is enabled when EFR[4] and IER[4] are set to '1'. Sleep mode is activated when:

- RXD input is in idle state.
- CTS#, DSR#, DCD#, and RI# are not toggling.
- The TX FIFO and TSR are in empty state.
- No interrupt is pending except THR and time-out interrupts.

In sleep mode, the SB16C1054 clock and baud rate clock are stopped. Since most registers are clocked using these clocks, the power consumption is greatly reduced. Normal operation is resumed when:

- RXD input receives the data start bit transition.
- Data byte is loaded to the TX FIFO or THR.
- CTS#, DSR#, DCD#, and RI# inputs are changed.

## 6.7 Programmable Baud Rate Generator

The SB16C1054 has a programmable baud rate generator with a prescaler. The prescaler is controlled by MCR[7], as shown in Figure 7. The MCR[7] sets the prescaler to divide the clock frequency by 1 or 4. And the baud rate generator further divides this clock frequency by a programmable divisor (DLL and DLM) between 1 and ($2^{16}$ – 1) to obtain a 16X sampling rate clock of the serial data rate. The sampling rate clock is used by transmitter for data bit shifting and receiver for data sampling.

The divisor of the baud rate generator is:

$$\text{Divisor} = \frac{\left(\frac{\text{XTAL1 Crystal Input Frequency}}{\text{Prescaler}}\right)}{(\text{Desired Baud Rate} \times 16)}$$

MCR[7] is cleared to '0' (prescaler = 1), when CLKSEL input is in low state after reset.

MCR[7] is set to '1' (prescaler = 4), when CLKSEL input is in high state after reset.

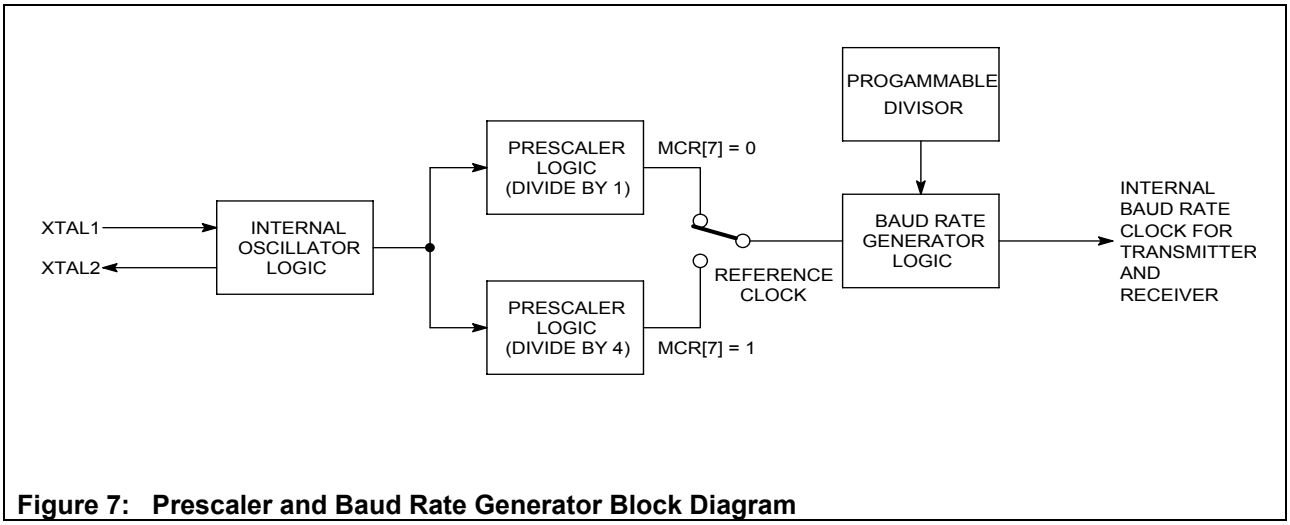



**Figure 7: Prescaler and Baud Rate Generator Block Diagram**

DLL and DLM must be written to in order to program the baud rate. DLL and DLM are the least and most significant byte of the baud rate divisor, respectively. If DLL and DLM are both zero, the SB16C1054 is effectively disabled, as no baud clock will be generated.

Table 5 shows the baud rate and divisor value for prescaler with divide by 1 as well as crystal with frequency 1.8432MHz, 3.6864MHz, 7.3728MHz, and 14.7456MHz, respectively.

Figure 8 shows the crystal clock circuit reference.

**Table 5: Baud Rates**

| Desired Baud Rate | 16X Digit Divisor for Prescaler with Divide by 1 | | | |
|---|---|---|---|---|
| | 1.8432MHz | 3.6864MHz | 7.3728MHz | 14.7456MHz |
| 50 | 0900h | 1200h | 2400h | 4800h |
| 75 | 0600h | 0C00h | 1800h | 3000h |
| 150 | 0300h | 0600h | 0C00h | 1800h |
| 300 | 0180h | 0300h | 0600h | 0C00h |
| 600 | 00C0h | 0180h | 0300h | 0600h |
| 1200 | 0060h | 00C0h | 0180h | 0300h |
| 1800 | 0040h | 0080h | 0100h | 0200h |
| 2000 | 003Ah | 0074h | 00E8h | 01D0h |
| 2400 | 0030h | 0060h | 00C0h | 0180h |
| 3600 | 0020h | 0040h | 0080h | 0100h |
| 4800 | 0018h | 0030h | 0060h | 00C0h |
| 7200 | 0010h | 0020h | 0040h | 0080h |
| 9600 | 000Ch | 0018h | 0030h | 0060h |
| 19.2K | 0006h | 000Ch | 0018h | 0030h |
| 38.4K | 0003h | 0006h | 000Ch | 0018h |
| 57.6K | 0002h | 0004h | 0008h | 0010h |
| 115.2K | 0001h | 0002h | 0004h | 0008h |
| 230.4K | — | 0001h | 0002h | 0004h |
| 460.8K | — | — | 0001h | 0002h |
| 921.6K | — | — | — | 0001h |

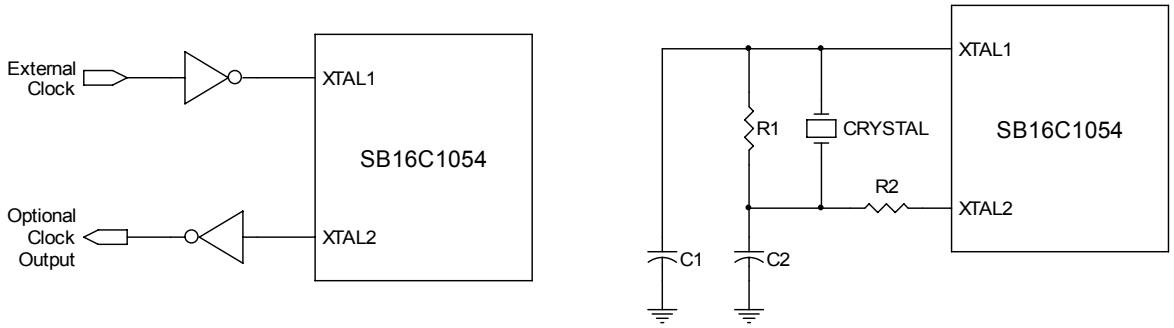



**Figure 8: Prescaler and Baud Rate Generator Block Diagram**

**Table 6: Component Values**

| Frequency Range (MHz) | C1 (pF) | C2 (pF) | R1 (Ω) | R2(Ω) |
|---|---|---|---|---|
| 1.8~8 | 22 | 68 | 220K | 470 ~ 1.5K |
| 8~16 | 33~68 | 33 ~ 68 | 220K ~ 2.2M | 470 ~ 1.5K |

### 6.8 Break and Time-out Conditions

Break Condition:
Break Condition is occurred when TXD signal outputs ‘0’ and sustains for more than one character.
It is occurred if LCR[6] is set to ‘1’ and deleted if ‘0’. If break condition is occurred when normal data are transmitted on TXD, break signal is transmitted and internal serial data are also transmitted, but they are not outputted to external TXD pin. When Break condition is deleted, then they are transmitted to TXD pin.
Figure 9 below shows the Break Condition Block Diagram.

Time-out Condition:
When serial data is received from external UART, characters are stored in RX FIFO. When the number of characters in RX FIFO reaches the trigger level, interrupt is generated for the CPU to treat characters in RX FIFO. But when the number of characters in RX FIFO does not reach the trigger level and no more data arrives from external device, interrupt is not generated and therefore CPU cannot recognize it. SB16C1054 offers time-out function for this situation. Time-out function generates an interrupt and reports to CPU when the number of RX FIFO is less than trigger level and no more data receives for four character time.
Time-out interrupt is enabled when IER[2] is set to ‘1’ and can be verified by ISR.

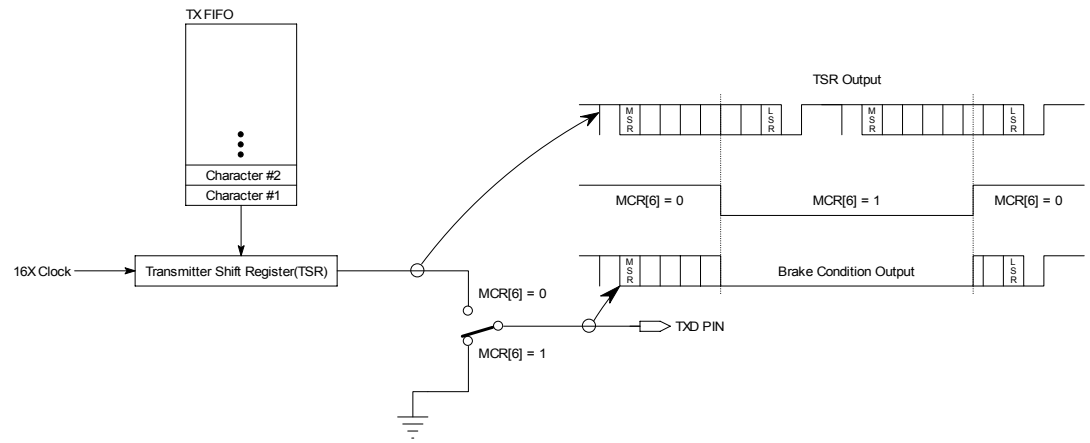



**Figure 9: Break Condition Block Diagram**

## 7. Register Descriptions

Each UART channel in the SB16C1054 has its own set of registers selected by address lines A2, A1, and A0 with a specific channel selected. The complete register set is shown on Table 7 and Table 8.

**Table 7: Internal Registers Map**

| Address A[2:0] | Page 0 LCR[7] = 0 MCR[6] = 0 | Page 1 LCR[7] = 1 LCR[7:0] ≠ BFh | Page 2 LCR[7] = 0 MCR[6] = 1 | Page 3 LCR = BFh PSR[0] = 0 | Page 4 LCR = BFh PSR[0] = 1 |
|---|---|---|---|---|---|
| 0h | THR/RBR | DLL | — | PSR | PSR |
| 1h | IER | DLM | GICR | ATR | AFR |
| 2h | FCR/ISR | | GISR | EFR | XRCR |
| 3h | LCR | | | | |
| 4h | MCR | | | XON1 | TTR |
| 5h | LSR | | TCR | XON2 | RTR |
| 6h | MSR | | RCR | XOFF1 | FUR |
| 7h | SPR | | FSR | XOFF2 | FLR |

**Table 7: Internal Registers Map…*continued***

| Address A[2:0] | Register | Read/Write | Comments |
|---|---|---|---|
| | **Page 0 Registers** | | |
| 0h | THR : Transmit Holding Register<br>RBR : Receive Buffer Register | Write-only<br>Read-only | LCR[7] = 0, MCR[6] = 0 |
| 1h | IER : Interrupt Enable Register | Read/Write | LCR[7] = 0, MCR[6] = 0 |
| 2h | FCR : FIFO Control Register<br>ISR : Interrupt Status Register | Write-only<br>Read-only | LCR[7] = 0, MCR[6] = 0,<br>LCR[7] = 1, LCR ≠ BFh |
| 3h | LCR : Line Control Register | Read/Write | — |
| 4h | MCR : Modem Control Register | Read/Write | LCR[7] = 0, MCR[6] = 0,<br>LCR[7] = 1, LCR ≠ BFh,<br>LCR[7] = 0, MCR[6] = 1 |
| 5h | LSR : Line Status Register | Read-only | LCR[7] = 0, MCR[6] = 0,<br>LCR[7] = 1, LCR ≠ BFh |
| 6h | MSR : Modem Status Register | Read-only | LCR[7] = 0, MCR[6] = 0,<br>LCR[7] = 1, LCR ≠ BFh |
| 7h | SPR : Scratch Pad Register | Read/Write | LCR[7] = 0, MCR[6] = 0,<br>LCR[7] = 1, LCR ≠ BFh |
| | **Page 1 Registers** | | |
| 0h | DLL : Divisor Latch LSB | Read/Write | LCR[7] = 1, LCR ≠ BFh |
| 1h | DLM : Divisor Latch MSB | Read/Write | LCR[7] = 1, LCR ≠ BFh |
| 2h | FCR : FIFO Control Register<br>ISR : Interrupt Status Register | Write-only<br>Read-only | LCR[7] = 0, MCR[6] = 0,<br>LCR[7] = 1, LCR ≠ BFh |
| 3h | LCR : Line Control Register | Read/Write | — |
| 4h | MCR : Modem Control Register | Read/Write | LCR[7] = 0, MCR[6] = 0,<br>LCR[7] = 1, LCR ≠ BFh,<br>LCR[7] = 0, MCR[6] = 1 |
| 5h | LSR : Line Status Register | Read-only | LCR[7] = 0, MCR[6] = 0,<br>LCR[7] = 1, LCR ≠ BFh |
| 6h | MSR : Modem Status Register | Read-only | LCR[7] = 0, MCR[6] = 0,<br>LCR[7] = 1, LCR ≠ BFh |
| 7h | SPR : Scratch Pad Register | Read/Write | LCR[7] = 0, MCR[6] = 0,<br>LCR[7] = 1, LCR ≠ BFh |

**Table 7: Internal Registers Map…*continued***

| Address A[2:0] | Register | Read/Write | Comments |
|---|---|---|---|
| **Page 2 Registers** | | | |
| 0h | None | — | — |
| 1h | GICR : Global Interrupt Control Register | Write-only | LCR[7] = 0, MCR[6] = 1 |
| 2h | GISR : Global Interrupt Status Register | Read-only | LCR[7] = 0, MCR[6] = 1 |
| 3h | LCR : Line Control Register | Read/Write | — |
| 4h | MCR : Modem Control Register | Read/Write | LCR[7] = 0, MCR[6] = 0,<br>LCR[7] = 1, LCR ≠ BFh,<br>LCR[7] = 0, MCR[6] = 1 |
| 5h | TCR : Transmit FIFO Count Register | Read-only | LCR[7] = 0, MCR[6] = 1 |
| 6h | RCR : Receive FIFO Count Register | Read-only | LCR[7] = 0, MCR[6] = 1 |
| 7h | FSR : Flow Control Status Register | Read-only | LCR[7] = 0, MCR[6] = 1 |
| **Page 3 Registers** | | | |
| 0h | PSR : Page Select Register | Read/Write | LCR = BFh, PSR[0] = 0,<br>LCR = BFh, PSR[0] = 1 |
| 1h | ATR : Auto Toggle Control Register | Read/Write | LCR = BFh, PSR[0] = 0 |
| 2h | EFR : Enhanced Feature Register | Read/Write | LCR = BFh, PSR[0] = 0 |
| 3h | LCR : Line Control Register | Read/Write | — |
| 4h | XON1 : Xon1 Character Register | Read/Write | LCR = BFh, PSR[0] = 0 |
| 5h | XON2 : Xon2 Character Register | Read/Write | LCR = BFh, PSR[0] = 0 |
| 6h | XOFF1 : Xoff1 Character Register | Read/Write | LCR = BFh, PSR[0] = 0 |
| 7h | XOFF2 : Xoff2 Character Register | Read/Write | LCR = BFh, PSR[0] = 0 |
| **Page 4 Registers** | | | |
| 0h | PSR : Page Select Register | Read/Write | LCR = BFh, PSR[0] = 0,<br>LCR = BFh, PSR[0] = 1 |
| 1h | AFR : Additional Feature Register | Read/Write | LCR = BFh, PSR[0] = 1 |
| 2h | XRCR : Xoff Re-transmit Count Register | Read/Write | LCR = BFh, PSR[0] = 1 |
| 3h | LCR : Line Control Register | Read/Write | — |
| 4h | TTR : Transmit FIFO Trigger Level Register | Read/Write | LCR = BFh, PSR[0] = 1 |
| 5h | RTR : Receive FIFO Trigger Level Register | Read/Write | LCR = BFh, PSR[0] = 1 |
| 6h | FUR : Flow Control Upper Threshold Register | Read/Write | LCR = BFh, PSR[0] = 1 |
| 7h | FLR : Flow Control Lower Threshold Register | Read/Write | LCR = BFh, PSR[0] = 1 |

**Table 8: Internal Registers Description**

| Addr. A[2:0] | Reg. | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| **Page 0 Registers** | | | | | | | | | |
| 0h | THR | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 0h | RBR | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 1h | IER | 0/CTS# Interrupt Enable | 0/RTS# Interrupt Enable | 0/Xoff Interrupt Enable | 0/Sleep Mode Enable | Modem Status Interrupt Enable | Receive Line Status Interrupt Enable | THR Empty Interrupt Enable | Receive Data Available Interrupt Enable |
| 2h | ISR | FCR[0]/ 256-TX FIFO Empty | FCR[0]/ 256-RX FIFO Full | Interrupt Priority Bit 5 | Interrupt Priority Bit 4 | Interrupt Priority Bit 3 | Interrupt Priority Bit 2 | Interrupt Priority Bit 1 | Interrupt Priority Bit 0 |
| 2h | FCR | RX Trigger Level (MSB) | RX Trigger Level (LSB) | 0/TX Trigger Level (MSB) | 0/TX Trigger Level (LSB) | DMA Mode Select | TX FIFO Reset | RX FIFO Reset | FIFO Enable |
| 3h | LCR | Divisor Enable | Set TX Brake | Set Parity | Parity Type Select | Parity Enable | Stop Bits | Word Length Bit 1 | Word Length Bit 0 |
| 4h | MCR | Clock Select | Page 2 Select/Xoff Re-Transmit Access Enable | 0/Xon Any | 0/Loop Back | OUT2/ INTx Enable | OUT1/ Xoff Re-Transmit Enable | RTS# | DTR# |
| 5h | LSR | RX FIFO Data Error | THR & TSR Empty | THR Empty | Receive Break | Framing Error | Parity Error | Overrun Error | Receive Data Ready |
| 6h | MSR | DCD# | RI# | DSR# | CTS# | ΔDCD# | ΔRI# | ΔDSR# | ΔCTS# |
| 7h | SCR | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| **Page 1 Registers** | | | | | | | | | |
| 0h | DLL | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 1h | DLM | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 |
| **Page 2 Registers** | | | | | | | | | |
| 1h | GICR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Global Interrupt Mask |
| 2h | GISR | Global Interrupt Mask Status | 0 | 0 | 0 | CH 3 Interrupt Status | CH 2 Interrupt Status | CH 1 Interrupt Status | CH 0 Interrupt Status |
| 5h | TCR | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 6h | RCR | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 7h | FSR | 0 | 0 | TX HW Flow Control Status | TX SW Flow Control Status | 0 | 0 | RX HW Flow Control Status | RX SW Flow Control Status |

**Table 8: Internal Registers Description…*continued***

| Addr. A[2:0] | Reg. | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
|---|---|---|---|---|---|---|---|---|---|
| **Page 3 Registers** | | | | | | | | | |
| 0h | PSR | 1 | 0 | 1 | 0 | 0 | 1 | 0 | Page Select |
| 1h | ATR | RXEN Polarity Select | RXEN Enable | TXEN Polarity Select | TXEN Enable | 0 | 0 | Auto Toggle Mode Bit 1 | Auto Toggle Mode Bit 0 |
| 2h | EFR | Auto-CTS# Enable | Auto-RTS# Enable | Special Character Detect Enable | Enhanced Feature Enable | Software Flow Control Bit 3 | Software Flow Control Bit 2 | Software Flow Control Bit 1 | Software Flow Control Bit 0 |
| 4h | XON1 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 5h | XON2 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 6h | XOFF1 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 7h | XOFF2 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| **Page 4 Registers** | | | | | | | | | |
| 1h | AFR | 0 | 0 | Global Interrupt Polarity Select | Global Interrupt Enable | 0 | 0 | 0 | 256-FIFO Enable |
| 2h | XRCR | 0 | 0 | 0 | 0 | 0 | 0 | Bit 1 | Bit 0 |
| 4h | TTR | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 5h | RTR | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 6h | FUR | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |
| 7h | FLR | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 |

### 7.1 Transmit Holding Register (THR, Page 0)

The transmitter section consists of the Transmit Holding Register (THR) and Transmit Shift Register (TSR). The THR is actually a 64-byte FIFO or a 256-byte FIFO. The THR receives data and shifts it into the TSR, where it is converted to serial data and moved out on the TX terminal. If the FIFO is disabled, location zero of the FIFO is used to store the byte. Characters are lost if overflow occurs.

### 7.2 Receive Buffer Register (RBR, Page 0)

The receiver section consists of the Receive Buffer Register (RBR) and Receive Shift Register (RSR). The RBR is actually a 64-byte FIFO or a 256-byte FIFO. The RSR receives serial data from external terminal. The serial data is converted to parallel data and is transferred to the RBR. This receiver section is controlled by the line control register. If the FIFO is disabled, location zero of the FIFO is used to store the characters. If overflow occurs, characters are lost. The RBR also stores the error status bits associated with each character.

### 7.3 Interrupt Enable Register (IER, Page 0)

IER enables each of the seven types of Interrupt, namely receive data ready, transmit

empty, line status, modem status, Xoff received, RTS# state transition from low to high, and CTS# state transition from low to high. All interrupts are disabled if bit[7:0] are cleared. Interrupt is enabled by setting appropriate bits. Table 9 shows IER bit settings.

**Table 9: Interrupt Enable Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7 | IER[7] | CTS# Interrupt Enable (Requires EFR[4] = 1).<br>0 : Disable the CTS# interrupt (default).<br>1 : Enable the CTS# interrupt. |
| 6 | IER[6] | RTS# Interrupt Enable (Requires EFR[4] = 1).<br>0 : Disable the RTS# interrupt (default).<br>1 : Enable the RTS# interrupt. |
| 5 | IER[5] | Xoff Interrupt Enable (Requires EFR[4] = 1).<br>0 : Disable the Xoff interrupt (default).<br>1 : Enable the Xoff interrupt. |
| 4 | IER[4] | Sleep Mode Enable (Requires EFR[4] = 1).<br>0 : Disable sleep mode (default).<br>1 : Enable sleep mode. |
| 3 | IER[3] | Modem Status Interrupt Enable<br>0 : Disable the modem status register interrupt (default).<br>1: Enable the modem status register interrupt. |
| 2 | IER[2] | Receive Line Status Interrupt Enable<br>0 : Disable the receive line status interrupt (default).<br>1: Enable the receive line status interrupt. |
| 1 | IER[1] | Transmit Holding Register Interrupt Enable<br>0 : Disable the THR interrupt (default).<br>1 : Enable the THR interrupt. |
| 0 | IER[0] | Receive Buffer Register Interrupt Enable<br>0 : Disable the RBR interrupt (default).<br>1 : Enable the RBR interrupt. |

### 7.4 Interrupt Status Register (ISR, Page 0)

The UART provides multiple levels of prioritized interrupts to minimize software work load. ISR provides the source of interrupt in a prioritized manner.

Table 10 shows ISR[7:0] bit settings.

**Table 10: Interrupt Status Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7 | ISR[7] | FCR[0]/256 TX FIFO Empty.<br>When 256-byte FIFO mode is disabled (default).<br>Mirror the content of FCR[0].<br>When 256-byte FIFO mode is enabled.<br>0 : 256-byte TX FIFO is full.<br>1 : 256-byte TX FIFO is not full.<br>When TCR is '00h', there are two situations of TX FIFO full and TX FIFO empty. If 256 TX empty bit is '1', it means TX FIFO is empty and if '0', it means 256 bytes character is fully stored in TX FIFO. |
| 6 | ISR[6] | FCR[0]/256 RX FIFO Full.<br>When 256-byte FIFO mode is disabled (default).<br>Mirror the content of FCR[0].<br>When 256-byte FIFO mode is enabled.<br>0 : 256-byte RX FIFO is not full.<br>1 : 256-byte RX FIFO is full.<br>When RCR is '00h', there are two situations of RX FIFO full and RX FIFO empty. If 256 RX empty bit is '1', it means 256 bytes character is fully stored in RX FIFO and if '0', it means RX FIFO is empty. |

**Table 10: Interrupt Status Register Description…*continued***

| Bit | Interrupt Priority List and Reset Functions | | | |
|---|---|---|---|---|
| **5:0** | **Priority** | **Interrupt Type** | **Interrupt Source** | **Interrupt Reset Control** |
| 00_0001 | — | None | None | — |
| 00_0110 | 1 | Receiver Line Status | OE, PE, FE, BI | Reading the LSR. |
| 00_1100 | 2 | Receive Data Available | Receiver data available, reaches trigger level. | Reading the RBR or FIFO falls below trigger level. |
| 00_0100 | 2 | Character Timeout Indication | At least one data is in RX FIFO and there are no more data in FIFO during four character time. | Reading the RBR. |
| 00_0010 | 3 | Transmit Holding Register Empty | When THR is empty or TX FIFO passes above trigger level (FIFO enable). | Reading the ISR or write data on THR. |
| 00_0000 | 4 | Modem Status | CTS#, DSR#, DCD#, RI# | Reading the MSR. |
| 01_0000 | 5 | Receive Xoff or Special Character | Detection of a Xoff or special character. | Reading the ISR. |
| 10_0000 | 6 | RTS#, CTS# Status during Auto RTS/CTS flow control | RTS# pin or CTS# pin change state from '0' to '1'. | Reading the ISR. |

## 7.5 FIFO Control Register (FCR, Page 0)

FCR is used for enabling the FIFOs, clearing the FIFOs, setting transmit/receive FIFO trigger level, and selecting the DMA modes. Table 11 shows FCR bit settings.

**Table 11: FIFO Control Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7:6 | FCR[7:6] | RX FIFO Trigger Level Select<br>00 : 8 characters (default)<br>01 : 16 characters<br>10 : 56 characters<br>11 : 60 characters |
| 5:4 | FCR[5:4] | TX FIFO Trigger Level Select<br>00 : 8 characters (default)<br>01 : 16 characters<br>10 : 32 characters<br>11 : 56 characters<br>FCR[5:4] can only be modified and enabled when EFR[4] is set. |
| 3 | FCR[3] | DMA Mode Select<br>0 : Set DMA mode 0 (default)<br>1 : Set DMA mode 1 |
| 2 | FCR[2] | TX FIFO Reset<br>0 : No TX FIFO reset (default)<br>1 : Reset TX FIFO pointers and TX FIFO level counter logic.<br>This bit will return to '0' after resetting FIFO. |
| 1 | FCR[1] | RX FIFO Reset<br>0 : No RX FIFO reset (default)<br>1 : Reset RX FIFO pointers and RX FIFO level counter logic.<br>This bit will return to '0' after resetting FIFO. |
| 0 | FCR[0] | FIFO enable<br>0 : Disable the TX and RX FIFO (default).<br>1 : Enable the TX and RX FIFO |

### 7.6 Line Control Register (LCR, Page 0)

LCR controls the asynchronous data communication format. The word length, the number of stop bits, and the parity type are selected by writing the appropriate bits to the LCR. Table 12 shows LCR bit settings.

**Table 12: Line Control Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7 | LCR[7] | Divisor Latch Enable.<br>0 : Disable the divisor latch (default).<br>1 : Enable the divisor latch. |
| 6 | LCR[6] | Break Enable.<br>0 : No TX break condition output (default).<br>1 : Forces TXD output to '0', for alerting the communication terminal to a line break condition. |
| 5 | LCR[5] | Set Stick Parity.<br>LCR[5:3] = xx0 : No parity is selected.<br>LCR[5:3] = 0x1 : Stick parity disabled. (default)<br>LCR[5:3] = 101 : Stick parity is forced to '1'.<br>LCR[5:3] = 111 : Stick parity is forced to '0'. |
| 4 | LCR[4] | Parity Type Select.<br>LCR[5:3] =001 : Odd parity is selected.<br>LCR[5:3] =011 : Even parity is selected. |
| 3 | LCR[3] | Parity Enabled.<br>0 : No parity (default).<br>1 : A parity bit is generated during the transmission and the receiver checks for receive parity. |
| 2 | LCR[2] | Number of Stop Bits.<br>LCR[2:0] = 0xx : 1 stop bit (word length = 5, 6, 7, 8).<br>LCR[2:0] = 100 : 1.5 stop bits (word length = 5).<br>LCR[2:0] = 11x or 1x1 : 2 stop bits (word length = 6, 7. 8). |
| 1:0 | LCR[1:0] | Word Length Bits.<br>00 : 5 bits (default).<br>01 : 6 bits.<br>10 : 7 bits.<br>11 : 8 bits. |

### 7.7 Modem Control Register (MCR, Page 0)

MCR controls the interface with the modem, data set, or peripheral device that is emulating the modem. Table 13 shows MCR bit settings.

**Table 13: Modem Control Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7 | MCR[7] | Clock Prescaler Select.<br>0 : Divide by 1 clock input (default).<br>1 : Divide by 4 clock input. |
| 6 | MCR[6] | Page 2 Select/Xoff Re-Transmit Access Enable<br>0 : Enable access to page 0 register when LCR[7] is '0' (default).<br>1 : Enable access to page 2 register and Xoff re-transmit bit when LCR[7] is '0'. |
| 5 | MCR[5] | Xon Any Enable.<br>0 : Disable Xon any (default).<br>1 : Enable Xon any. |
| 4 | MCR[4] | Internal Loop Back Enable.<br>0 : Disable loop back mode (default).<br>1 : Enable internal loop back mode. In this mode the MCR[3:0] signals are looped back into MSR[7:4] and TXD output is looped back to RXD input internally. |
| 3 | MCR[3] | OUT2/Interrupt Output Enable.<br>0 : INTx outputs disabled (default). During loop back mode, OUT2 output '0' and it controls MSR[7] to '1'.<br>1 : INTx outputs enabled. During loop back mode, OUT2 output '1' and it controls MSR[7] to '0'.<br>OUT2 is not available as an output pin on the SB16C1054. |
| 2 | MCR[2] | OUT1/Xoff Re-transmit Enable.<br>0 : Xoff re-transmit disable when MCR[6] is '0'. During loop back mode, OUT1 output to '0' and it controls MSR[6] to '1'.<br>1 : Xoff re-transmit enable when MCR[6] is '1'. During loop back mode, OUT1 output to '1' and it controls MSR[6] to '0'.<br>OUT1 is not available as an output pin on the SB16C1054.<br>Xoff re-transmit is operated with XRCR, refer to XRCR. |
| 1 | MCR[1] | RTS# Output.<br>0 : Force RTS# output to '1'. During loop back mode, controls MSR[4] to '1'.<br>1 : Force RTS# output to '0'. During loop back mode, controls MSR[4] to '0'. |
| 0 | MCR[0] | DTR# Output.<br>0 : Force DTR# output to '1'. During loop back mode, controls MSR[5] to '1'.<br>1 : Force DTR# output to '0'. During loop back mode, controls MSR[5] to '0'. |

### 7.8 Line Status Register (LSR, Page 0)

LSR provides the status of data transfers between the UART and the CPU. When LSR is read, LSR[4:2] reflect the error bits (BI, FE, PE) of the character at the top of the RX FIFO. The errors in a character are identified by reading LSR and then reading RBR. Reading LSR does not cause an increment of the RX FIFO read pointer. The RX FIFO read pointer is incremented by reading the RBR. Table 14 shows LSR bit settings.

**Table 14: Line Status Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7 | LSR[7] | RX FIFO data error Indicator.<br>0 : No RX FIFO error (default).<br>1 : At least one parity error, framing error, or break indication is in the RX FIFO. This bit is cleared when there is no more error in any of characters in the RX FIFO. |
| 6 | LSR[6] | THR and TSR Empty Indicator.<br>0 : THR or TSR is not empty.<br>1 : THR and TSR are empty. |
| 5 | LSR[5] | THR Empty Indicator.<br>0 : THR is not empty.<br>1 : THR is empty. It indicates that the UART is ready to accept a new character for transmission. In addition, it uses the UART to generate an interrupt to the CPU when the THR empty interrupt enable is set to '1'. |
| 4 | LSR[4] | Break Interrupt Indicator.<br>0 : No break condition (default).<br>1 : The receiver received a break signal (RXD was '0' for at least one character frame time). In FIFO mode, only one character is loaded into the RX FIFO. |
| 3 | LSR[3] | Framing Error Indicator.<br>0 : No framing error (default).<br>1 : Framing error. It indicates that the received character did not have a valid stop bit. |
| 2 | LSR[2] | Parity Error Indicator.<br>0 : No parity error (default).<br>1 : Parity error. It indicates that the receive character did not have the correct even or odd parity, as selected by the LCR[4] |
| 1 | LSR[1] | Overrun Error Indicator.<br>0 : No overrun error (default).<br>1 : Overrun error. It indicates that the character in the RBR or RX FIFO was not read by the CPU, thereby ignored the receiving character. |
| 0 | LSR[0] | Receive Data Ready Indicator.<br>0 : No character in the RBR or RX FIFO.<br>1 : At least one character in the RBR or RX FIFO. |

### 7.9 Modem Status Register (MSR, Page 0)

MSR provides the current status of control signals from modem or auxiliary devices. MSR[3:0] are set to '1' when input from modem changes and cleared to '0' as soon as CPU reads MSR. Table 15 shows MSR bit settings.

**Table 15: Modem Status Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7 | MSR[7] | DCD Input Status.<br>Complement of Data Carrier Detect (DCD#) input.<br>In loop back mode this bit is equivalent to OUT2 in the MCR. |
| 6 | MSR[6] | RI Input Status.<br>Complement of Ring Indicator (RI#) input.<br>In loop back mode this bit is equivalent to OUT1 in the MCR. |
| 5 | MSR[5] | DSR Input Status.<br>Complement of Data Set Ready (DSR#) input.<br>In loop back mode this bit is equivalent to DTR in the MCR. |
| 4 | MSR[4] | CTS Input Status.<br>Complement of Clear To Send (CTS#) input.<br>In loop back mode this bit is equivalent to RTS in the MCR. |
| 3 | MSR[3] | ΔDCD Input Status.<br>0 : No change on CD# input (default).<br>1 : Indicates that the DCD# input has changed state. |
| 2 | MSR[2] | ΔRI Input Status.<br>0 : No change on RI# input (default).<br>1 : Indicates that the RI# input has changed state from '0' to '1'. |
| 1 | MSR[1] | ΔDSR Input Status.<br>0 : No change on DSR# input (deault).<br>1 : Indicates that the DSR# input has changed state. |
| 0 | MSR[0] | ΔCTS Input Status.<br>0 : No change on CTS# input (deault).<br>1 : Indicates that the CTS# input has changed state. |

### 7.10 Scratch Pad Register (SPR, Page 0)

This 8-bit Read/Write Register does not control the UART in anyway. It is intended as a scratch pad register to be used by the programmer to hold data temporarily.

### 7.11 Divisor Latches (DLL, DLH, Page 1)

Two 8-bit registers which store the 16-bit divisor for generation of the clock in baud rate generator. DLH stores the most significant part of the divisor, and DLL stores the least significant part of the divisor. Divisor of zero is not recommended.

Note that DLL and DLH can only be written to before sleep mode is enabled, i.e., before IER[4] is set. Chapter 6.7 describes the details of divisor latches.

### 7.12 Global Interrupt Control Register (GICR, Page 2)

GICR is a register that internal four 16C1050 UARTs share to use. It is used when determining whether each interrupt generated at four 16C1050 UARTs are transmitted to global interrupts or not. Table 16 shows the GICR bit settings.

**Table 16: Global Interrupt Control Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7:1 | GICR[7:1] | Not used, always '000_0000' |
| 0 | GICR[0] | Global Interrupt Mask.<br>0 : Deasserted, irrespective of occurring interrupts of four 16C1050 UARTs displayed on GISR[3:0]. '1' is outputted if global interrupt polarity, AFR[5] is '0' and outputted '0' if '1'.<br>1 : Generates interrupt when all the values of GISR[3:0] are not '0'. If all the values of GISR[3:0] are '0', interrupt is not outputted on GINT pin. '1' is outputted if AFR[5] is '0' and '0' is outputted if '1'. |

### 7.13 Global Interrupt Status Register (GISR, Page 2)

GISR is a register that internal four 16C1050 UARTs share to use. It is used to verify the generation status of each interrupt of four 16C1050 UARTs when global interrupt function is enabled. Table 17 shows GISR bit settings.

**Table 17: Global Interrupt Status Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7 | GISR[7] | Mirror the content of GICR[0]. |
| 6:4 | GISR[6:4] | Not used, always '00'. |
| 3 | GISR[3] | UART of CS3# Interrupt Status.<br>0 : UART of CS3# interrupt was not generated.<br>1 : UART of CS3# interrupt was generated. |
| 2 | GISR[2] | UART of CS2# Interrupt Status.<br>0 : UART of CS2# interrupt was not generated.<br>1 : UART of CS2# interrupt was generated. |
| 1 | GISR[1] | UART of CS1# Interrupt Status.<br>0 : UART of CS1# interrupt was not generated.<br>1 : UART of CS1# interrupt was generated. |
| 0 | GISR[0] | UART of CS0# Interrupt Status.<br>0 : UART of CS0# interrupt was not generated.<br>1 : UART of CS0# interrupt was generated. |

### 7.14 Transmit FIFO Count Register (TCR, Page 2)

TCR shows the number of characters that can be stored in TX FIFO. In 64-byte FIFO mode, it consists of only TCR[6:0]. If the number of characters that can be stored in TX FiFO is 0, it is shown as '0000_0000' and if 64, it is shown as '0100_0000'. In 256-byte FIFO mode, it consists of ISR[7] + TCR[7:0]. If the number of characters that can be stored in TX FiFO is 0, it is shown as '0_0000_0000' and if 255, it is shown as '0_1111_1111'. And in case of the maximum number 256, it is shown as '1_0000_0000'.

### 7.15 Receive FIFO Count Register (RCR, Page 2)

RCR shows the number of characters that is stored in RX FIFO. In 64-byte FIFO mode, it consists of only RCR[6:0]. If the number of characters that is stored in RX FiFO is 0, it is shown as '0000_0000' and if 64, it is shown as '0100_0000'. In 256-byte FIFO mode, it consists of ISR[6] + RCR[7:0]. If the number of characters that is stored in RX FiFO is 0, it is shown as '0_0000_0000' and if 255, it is shown as '0_1111_1111'. And in case of the maximum number 256, it is shown as '1_0000_0000'.

### 7.16 Flow Control Status Register (FSR, Page 2)

FSR show the status of operation of TX Hardware Flow Control, RX Hardware Flow Control, TX Software Flow Control, and RX Software Flow Control.

**Table 18: Flow Control Status Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7:6 | FSR[7:6] | Not used, always '00'. |
| 5 | FSR[5] | TX Hardware Flow Control Status.<br>0 : When FIFO or Auto-RTS flow control is disabled.<br>If FIFO and Auto-RTS flow control is enabled, it means the number of data received in RX FIFO at the first time is less than the value of FUR, or it means the number of data in RX FIFO was more than the value of FUR and after the CPU read them, the number of data that remains unread is less than or equal to the value of FLR. That is, UART reports external device that it can receive more characters.<br>1 : It shows that the number of data received in RX FIFO exceeds the value of FUR and UART reports external device that it cannot receive more data. If RX FIFO has space to store more data, new data are stored in RX FIFO but after it gets full, they are lost.<br>For more details, refer to '6.2 Hardware Flow Control'. |
| 4 | FSR[4] | TX Software Flow Control Status.<br>0 : When FIFO or Software flow control is disabled.<br>If FIFO and Software flow control is enabled, it means the number of data received in RX FIFO at the first time is less than the value of FUR, or it means the number of data in RX FIFO was more than the value of FUR and after the CPU read them, the number of data that remains unread after the CPU read the data received in RX FIFO is less than or equal to the value of FLR. That |

| | | |
|---|---|---|
| | | is, UART transmits Xon character to report external device that it can receive more data.<br>1 : It shows that the number of data received in RX FIFO exceeds the value of FUR and transmitting Xoff character to report external device that it cannot receive more data. If RX FIFO has space to store more data, new data are stored in RX FIFO but after it gets full, they are lost.<br>For more details, refer to '6.3 Software Flow Control'. |
| 3:2 | FSR[3:2] | Not used, always '00'. |
| 1 | FSR[1] | RX Hardware Flow Control Status.<br>0 : When FIFO or Auto-CTS flow control is disabled. If FIFO and Auto-CTS flow control is enabled, '0' is inputted in CTS# pin and it means external device can receive more data. This time data in TX FIFO are transmitted.<br>1 : If FIFO and Auto-CTS flow control is enabled, '1' is inputted in CTS# pin and it means external device can not receive more data. This time data in TX FIFO are not transmitted.<br>For more details, refer to '6.2 Hardware Flow Control'. |
| 0 | FSR[0] | RX Software Flow Control Status.<br>0 : When FIFO or RX Software flow control is disabled. If FIFO and RX Software flow control is enabled, it means Xoff character has never arrived or Xon character arrived after Xoff character had arrived(it means external device can receive more data). This time data in TX FIFO are transmitted.<br>1 : If FIFO and RX Software flow control is enabled, it means Xoff character has arrived and external device can not receive data any more. This time characters in TX FIFO are not transmitted.<br>For more details, refer to '6.3 Software Flow Control'. |

### 7.17 Page Select Register (PSR, Page 3)

If BFh is written in LCR, registers in Page3 and Page4 can be accessed. PSR is used to determine which page to use. Table 19 shows PSR bit settings.

**Table 19: Page Select Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7:1 | PSR[7:1] | Access Key.<br>When writing data on PSR to change page, Access Key must be correspondent. If the value of PSR[7:1] is '1010_010', data is written on PSR[0] and page can be selected. If PSR[7:1] is read, it reads '0000_000' which is irrespective of Access Key. |
| 0 | PSR[0] | Page Select.<br>0 : Page 3 is selected (default).<br>1 : Page 4 is selected. |

## 7.18 Auto Toggle Control Register (ATR, Page 3)

ATR controls the signals for controlling input/output signals when using Line Interface as RS422 or RS485, so eliminates additional glue logic outside. Table 20 shows ATR bit settings.

**Table 20: Auto Toggle Control Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7 | ATR[7] | RXEN Polarity Select.<br>0 : Asserted output of RXEN is '0'. (default)<br>1 : Asserted output of RXEN is '1'. |
| 6 | ATR[6] | RXEN Control Mode Select.<br>0 : RXEN is outputted as same as ATR[7], irrespective of TXD signal. (default)<br>1 : RXEN is outputted after making complement of ATR[7] when TXD signal is transmitting. And outputted as same as ATR[7] when TXD is not transmitting. |
| 5 | ATR[5] | TXEN Polarity Select.<br>0 : Asserted output of TXEN is '0'. (default)<br>1 : Asserted output of TXEN is '1'. |
| 4 | ATR[4] | TXEN Control Mode Select.<br>Only when ATR[1:0] is '11';<br>0 : TXEN is outputted as same as ATR[5], irrespective of TXD signal. (default)<br>1 : RXEN is outputted after making complement of ATR[5] when TXD signal is transmitting, and outputted after making complement of ATR[7] when TXD is not transmitting.. |
| 3:2 | ATR[3:2] | Not used, always '00'. |
| 1:0 | ATR[1:0] | Auto Toggle Enable.<br>00 : Auto toggle is disabled (default).<br>RTS#/TXEN, DTR#TXEN pin operate as RTS#, DTR#. If 80-pin, each of TXRDY/TXEN, RXRDY/RXEN operates as TXRDY, RXRDY.<br>01 : RTS#/TXEN pin operates as TXEN. DTR#/TXEN pin operates as DTR#. If 80-pin, each of TXRDY/TXEN, RXRDY/RXEN operates as TXRDY, RXRDY.<br>10 : DTR#/TXEN pin operates as TXEN. RTS#/TXEN operates as RTS#. If 80-pin, each of TXRDY/TXEN, RXRDY/RXEN operates as TXRDY, RXRDY.<br>11 : Only in 80-pin. TXRDY/TXEN, RXRDY/RXEN pin operates as TXEN, RXEN. RTS#/TXEN, DTR#/TXEN operates as RTS#, DTR#. |

## 7.19 Enhanced Feature Register (EFR, Page 3)

EFR enables or disables the enhanced features of the UART. Table 21 shows EFR bit settings.

**Table 21: Enhanced Feature Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7 | EFR[7] | Auto-CTS Flow Control Enable.<br>0 : Auto-CTS flow control is disabled (default).<br>1 : Auto-CTS flow control is enabled. Transmission stops when CTS# pin is inputted '1'. Transmission resumes when CTS# pin is inputted '0'. |
| 6 | EFR[6] | Auto-RTS Flow Control Enable.<br>0 : Auto-RTS flow control is disabled (default).<br>1 : Auto-RTS flow control is enabled. The RTS# pin outputs '1' when data in RX FIFO fill above the FUR. RTS# pin outputs '0' when data in RX FIFO fall below the FLR. |
| 5 | EFR[5] | Special Character Detect.<br>0 : Special character detect disabled (default).<br>1 : Special character detect enabled. The UART compares each incoming character with data in Xoff2 register. If a match occurs, the received data is transferred to RX FIFO and ISR[4] is set to '1' to indicate that a special character has been detected. |
| 4 | EFR[4] | Enhanced Function Bits Enable.<br>0 : Disables enhanced functions and writing to IER[7:4], FCR[5:4], MCR[7:5].<br>1 : Enables enhanced function IER[7:4], FCR[5:4], and MCR[7:5] can be modified, i.e., this bit is therefore a write enable. |
| 3:0 | EFR[3:0] | Software Flow Control Select.<br>Single character and dual sequential characters software flow control is supported. Combinations of software flow control can be selected by programming these bits. See Table 3 "Software flow control options (EFR[3:0])" on page 11. |

### 7.23 Additional Feature Register (AFR, Page 4)

AFR enables or disables the 256-byte FIFO mode and controls the global interrupt. Table 22 shows AFR bit settings.

**Table 22: Additional Feature Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7:6 | AFR[7:6] | Not used, always '00'. |
| 5 | AFR[5] | Global Interrupt Polarity Select<br>0 : GINT pin outputs '0' when interrupt is generated (default).<br>1 : GINT pin outputs '1' when interrupt is generated. |
| 4 | AFR[4] | Global Interrupt Enable<br>0 : INT0/GINT pin is selected to INT0 (default).<br>1 : INT0/GINT pin is selected to GINT. |
| 3:1 | AFR[3:1] | Not used, always '000'. |
| 0 | AFR[0] | 256-byte FIFO Enable.<br>0 : 256-byte FIFO mode is disabled and this means SB16C1054 operates as Non FIFO mode or 64-byte FIFO mode (default).<br>1 : 256-byte FIFO mode is enabled and ISR[7:6] operates as 256-TX FIFO Empty and 256-RX FIFO Full. |

### 7.24 Xoff Re-transmit Count Register (XRCR, Page 4)

XRCR operates only when Software flow control is enabled by EFR[3:0] and Xoff Re-transmit function of MCR[2] is also enabled. And it determines the period of retransmission of Xoff character. Table 23 shows XRCR bit settings.

**Table 23: Xoff Re-transmit Count Register Description**

| Bit | Symbol | Description |
|---|---|---|
| 7:2 | XRCR[7:2] | Not used, always '0000_00'. |
| 1:0 | XRCR[1:0] | Xoff Re-transmit Count Select<br>00 : Transmits Xoff character whenever the number of received data is 1 during XOFF status. (default)<br>01 : Transmits Xoff character whenever the number of received data is 4 during XOFF status.<br>10 : Transmits Xoff character whenever the number of received data is 8 during XOFF status.<br>11 : Transmits Xoff character whenever the number of received data is 16 during XOFF status. |

### 7.25 Transmit FIFO Trigger Level Register (TTR, Page 4)

Operates only when 256-byte FIFO mode is enabled. It sets the trigger level of 256-byte TX FIFO for generating transmit interrupt. Interrupt is generated when the number of data remained in TX FIFO after transmitting through TXD pin is less than the value of TTR. Initial value is 128h, '1000_0000' and '0000_0000' must not be written. If written, unexpected operation may occur.

### 7.26 Receive FIFO Trigger Level Register (RTR, Page 4)

Operates only when 256-byte FIFO mode is enabled. It sets the trigger level of 256-byte RX FIFO for generating receive interrupt. Interrupt is generated when the number of data remained in RX FIFO exceeds the value of RTR(this time, timeout or interrupt is valid). Initial value is 128h, '1000_0000' and '0000_0000' must not be written. If written, unexpected operation may occur.

### 7.27 Flow Control Upper Threshold Register (FUR, Page 4)

It can be written only when 256-byte FIFO mode is enabled and one of TX software flow control or Auto-RTS is enabled (In 64-byte mode, it cannot be written but can be read only, and follows the value of trigger level set in FCR[5:4]). While TX software flow control is enabled, Xoff character is transmitted when the number of data in RX FIFO exceeds the value of FUR. If Auto-RTS is enabled, '1' is outputted on RTS# pin to report that it cannot receive data any more. If both TX software flow control and Auto-RTS is enabled, Xoff character is transmitted and '1' is outputted on RTS# pin. The value of FUR must be larger than that of FLR.

### 7.28 Flow Control Lower Threshold Register (FLR, Page 4)

It can be written only when 256-byte FIFO mode is enabled and one of TX software flow control, or Auto-RTS is enabled (In 64-byte mode, it cannot be written but can be read only, and follows the value of trigger level set in FCR[7:6]). While TX software flow control is enabled, Xon character is transmitted when the number of data in RX FIFO is less than the value of FUR only if Xoff character is transmitted before. If Auto-RTS is enabled, '0' is outputted on RTS# pin to report that it can receive more data. If both TX software flow control and Auto-RTS is enabled, Xon character is transmitted only if Xoff character is transmitted before and '0' is outputted on RTS# pin. The value of FLR must be less than that of FUR.

**Table 24: SB16C1054 Reset Conditions**

| Registers | Reset State |
|---|---|
| **Page 0** | |
| RBR | [7:0] = 'XXXX_XXXX' |
| IER | [7:0] = '0000_0000' |
| FCR | [7:0] = '0000_0000' |
| ISR | [7:0] = '0000_0001' |

| | |
|---|---|
| LCR | [7:0] = '0000_0000' |
| MCR | [7:0] = '0000_0000' |
| LSR | [7:0] = '0110_0000' |
| MSR | [7:4] = '0000'<br>[3:0] = Logic levels of the inputs inverted |
| SPR | [7:0] = '0000_0000' |
| **Page 1** | |
| DLL | [7:0] = '1111_1111' |
| DLM | [7:0] = '1111_1111' |
| **Page 2** | |
| GICR | [7:0] = '0000_0000' |
| GISR | [7:0] = '0000_0000' |
| TCR | [7:0] = '0000_0000' |
| RCR | [7:0] = '0000_0000' |
| FSR | [7:0] = '0000_0000' |
| **Page 3** | |
| PSR | [7:0] = '0000_0000' |
| ATR | [7:0] = '0000_0000' |
| EFR | [7:0] = '0000_0000' |
| XON1 | [7:0] = '0000_0000' |
| XON2 | [7:0] = '0000_0000' |
| XOFF1 | [7:0] = '0000_0000' |
| XOFF2 | [7:0] = '0000_0000' |
| **Page 4** | |
| AFR | [7:0] = '0000_0000' |
| XRCR | [7:0] = '0000_0000' |
| TTR | [7:0] = '1000_0000' |
| RTR | [7:0] = '1000_0000' |
| FUR | [7:0] = '0000_0000' |
| FLR | [7:0] = '0000_0000' |
| **Output Signals** | **Reset State** |
| TXD, RTS#, DTR# | Logic 1 |
| TXRDY# | Logic 0 |
| RXRDY# | Logic 1 |
| INT | Tri-State Condition = INTSEL is open or low state<br>Logic 0 = INTSEL is high state |

## 8. Programmer's Guide

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

**Table 25: Register Programming Guide**

| Command | Action |
|---|---|
| Set Baud Rate to VALUE1, VALUE2 | Read LCR, save in temp<br>Set LCR to 80h<br>Set DLL to VALUE1<br>Set DLM to VALUE2<br>Set LCR to temp |
| Set Xon1, Xoff1 to VALUE1, VALUE2 | Read LCR, save in temp<br>Set LCR to BFh<br>Set Xon1 to VALUE1<br>Set Xoff1 to VALUE2<br>Set LCR to temp |
| Set Xon2, Xoff2 to VALUE1, VALUE2 | Read LCR, save in temp<br>Set LCR to BFh<br>Set Xon2 to VALUE1<br>Set Xoff2 to VALUE2<br>Set LCR to temp |
| Set Software Flow Control Mode to VALUE | Read LCR, save in temp<br>Set LCR to BFh<br>Set EFR to VALUE<br>Set LCR to temp |
| Set flow control threshold for 64-byte FIFO Mode | 1) Set FCR to '0000_xxx1'<br>→ Set FUR to 8, set FLR to 0<br>2) Set FCR to '0101_xxx1'<br>→ Set FUR to 16, set FLR to 8<br>3) Set FCR to '1010_xxx1'<br>→ Set FUR to 56, set FLR to 16<br>4) Set FCR to '1111_xxx1'<br>→ Set FUR to 60, set FLR to 56 |
| Set flow control threshold for 256-byte FIFO Mode | Set FCR to 'xxxx_xxx1'<br>Read LCR, save in temp<br>Set LCR to BFh<br>Set PSR to A5h<br>Set AFR to 01h |

**Table 25: Register Programming Guide…*continued***

| Command | Action |
|---|---|
| | Set FUR to Upper Threshold Value<br>Set FLR to Lower Threshold Value<br>Set PSR to A4h<br>Set LCR to temp |
| Set TX FIFO / RX FIFO Interrupt Trigger Level for 64-byte FIFO Mode | 1) Set FCR to '0000_xxx1'<br>→ Set RTR to 8, set TTR to 8<br>2) Set FCR to '0101_xxx1' |

| | |
|---|---|
| | → Set RTR to 16, set TTR to 16<br>3) Set FCR to '1010_xxx1'<br>→ Set RTR to 56, set TTR to 32<br>4) Set FCR to '1111_xxx1'<br>→ Set RTR to 60, set TTR to 56 |
| Set TX FIFO / RX FIFO Interrupt Trigger Level for 256-byte FIFO Mode | Set FCR to 'xxxx_xxx1'<br>Read LCR, save in temp<br>Set LCR to BFh<br>Set PSR to A5h<br>Set AFR to 01h<br>Set TTR to TX FIFO Trigger Level Value<br>Set RTR to RX FIFO Trigger Level Value<br>Set PSR to A4h<br>Set LCR to temp |
| Read Flow Control Status | Read LCR, save in temp1<br>Read MCR, save in temp2<br>Set LCR to ('0111_1111' **AND** temp1)<br>Set MCR to ('0100_0000' **OR** temp2)<br>Read FSR, save in temp3<br>Pass temp3 back to host<br>Set MCR to temp2<br>Set LCR to temp1 |
| Read TX FIFO / RX FIFO Count Value | Read LCR, save in temp1<br>Read MCR, save in temp2<br>Set LCR to ('0111_1111' **AND** temp1)<br>Set MCR to ('0100_0000' **OR** temp2)<br>Read TCR, save in temp3<br>Read RCR, save in temp4<br>Pass temp3 back to host<br>Pass temp4 back to host<br>Set MCR to temp2<br>Set LCR to temp1 |

**Table 25: Register Programming Guide…*continued***

| Command | Action |
|---|---|
| Read 256-byte TX FIFO Empty Status / RX FIFO Full Status | Set FCR to 'xxxx_xxx1'<br>Read LCR, save in temp1<br>Set LCR to BFh<br>Set PSR to A5h<br>Set AFR to 01h<br>Set PSR to A4h<br>Set LCR to temp1<br>Read ISR, save in temp2 |

| | |
|---|---|
| | Pass temp2 back to host |
| Enable Xoff Re-transmit | Read LCR, save in temp1<br>Set LCR to not BFh<br>Read MCR, save in temp2<br>Set MCR to ('0100_0000' **OR** temp2)<br>Set MCR to ('0100_0100' **OR** temp2)<br>Set MCR to ('1011_1111' **AND** temp2)<br>Set MCR to temp2<br>Set LCR to temp1 |
| Disable Xoff Re-transmit | Read LCR, save in temp1<br>Set LCR to not BFh<br>Read MCR, save in temp2<br>Set MCR to ('0100_0000' **OR** temp2)<br>Set MCR to ('1011_1011' **AND** temp2)<br>Set MCR to temp2<br>Set LCR to temp1 |
| Set Prescaler Value to Divide-by-1 or 4 | Read LCR, save in temp1<br>Set LCR to BFh<br>Read EFR, save in temp2<br>Set EFR to ('0001_0000' **OR** temp2)<br>Set LCR to 00h<br>Read MCR, save in temp3<br><br>if Divide-by-1 = OK then<br>Set MCR to ('0111_1111' **AND** temp3)<br>else<br>Set MCR to ('1000_0000' **OR** temp3)<br><br>Set LCR to BFh<br>Set EFR to temp2<br>Set LCR to temp1 |

## 9. Electrical Characteristics

### Absolute Maximum Ratings

| Symbol | Parameter | Conditions | Min | Max | Unit |
|---|---|---|---|---|---|
| $V_{CC}$ | Supply voltage | | − | 3.6 | V |
| $V_I$ | Input voltage | | − 0.5 | 5.5 | V |
| $V_O$ | Output voltage | | GND + 0.1 | $V_{CC}$ – 0.1 | V |
| $T_{amb}$ | Operating ambient temperature | In free-air | − 40 | +85 | ℃ |
| $T_{stg}$ | Storage temperature | | − 60 | +150 | ℃ |

## DC Electrical Characteristics

| Symbol | Parameter | Conditions | 3.3V | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min | Nom | Max | |
| $V_{CC}$ | Supply voltage | | 2.7 | 3.3 | 3.6 | V |
| $V_I$ | Input voltage | | 0 | – | $V_{CC}$ | V |
| $V_{IH}$ | High-level input voltage | | $V_{CC}$ x 0.7 | – | 5.5 | V |
| $V_{IL}$ | Low-level input voltage | | 0 | – | $V_{CC}$ x 0.3 | V |
| $V_O$ | Output voltage | | 0 | – | $V_{CC}$ | V |
| $V_{OH}$ | High-level output voltage | IOH = – 8mA | 2.4 | – | – | V |
| $V_{OL}$ | Low-level output voltage | IOL = 8mA | – | – | 0.4 | V |
| $C_I$ | Input capacitance | | – | – | 9 | pF |
| | Oscillator/Clock speed | | – | – | 85 | MHz |
| | Clock duty cycle | | – | 50 | – | % |
| $I_{CC}$ | Supply current | | – | – | | mA |
| $I_{CCsleep}$ | Sleep current | | – | – | | mA |

| Symbol | Parameter | Min | Max | Unit |
|---|---|---|---|---|
| $t_{rd}$ | Pulse duration, IOR# low | 24 | | ns |
| $t_{csr}$ | Set up time, CSx# valid before IOR# low † | 10 | | ns |
| $t_{ar}$ | Set up time, A2~A0 valid before IOR# low † | 10 | | ns |
| $t_{ra}$ | Hold time, A2~A0 valid after IOR# high † | 2 | | ns |
| $t_{rcs}$ | Hold time, CSx# valid after IOR# high † | 0 | | ns |
| $t_{frc}$ | Delay time, $t_{ar}+t_{rd}+t_{rc}$ ‡ | 54 | | ns |
| $t_{rc}$ | Delay time, IOR# high to IOR# or IOW# low | 20 | | ns |
| $t_{wr}$ | Pulse duration, IOW#↓ | 24 | | ns |
| $t_{csw}$ | Setup time, CSx# valid before IOW#↓ | 10 | | ns |
| $t_{aw}$ | Setup time, A2~A0 valid before IOW#↓ | 10 | | ns |

| | | | | |
|---|---|---|---|---|
| $t_{ds}$ | Setup time, D7~D0 valid before IOW#↑ | 15 | | ns |
| $t_{wa}$ | Hold time, A2~A0 valid after IOW#↑ | 2 | | ns |
| $t_{wcs}$ | Hold time, CSx# valid after IOW#↑ | 2 | | ns |
| $t_{dh}$ | Hold time, D7~D0 valid after IOW#↑ | 5 | | ns |
| $t_{fwc}$ | Delay time, $t_{aw}+t_{wr}+t_{wc}$ | 54 | | ns |
| $t_{wc}$ | Delay time, IOW#↑ to IOW# or IOR#↓ | 20 | | ns |
| $t_{rvd}$ | Enable time, IOR#↓ to D7~D0 valid | | 24 | ns |
| $t_{hz}$ | Disable time, IOR# to D7~D0 released | 4 | | ns |
| $t_{irs}$ | Delay time, INTx↓ to TXDx↓ at start | 8 | 24 | RCLK |
| $t_{sti}$ | Delay time, TXDx↓ at start to INTx↑ | 8 | 8 | RCLK |
| $t_{si}$ | Delay time, IOW# high or low (WR THR) to INTx↑ | 16 | 32 | RCLK |
| $t_{sxa}$ | Delay time, TXDx↓ at start to TXRDY#↓ | | 8 | RCLK |
| $t_{hr}$ | Propagation delay time, IOW#(WR THR)↓ to INTx↓ | | 12 | ns |
| $t_{ir}$ | Propagation delay time, IOR#(RD IIR)↑ to INTx↓ | | 12 | ns |
| $t_{wxi}$ | Propagation delay time, IOW#(WR THR) ↓ to TXRDY#↑ | | 10 | ns |
| $t_{sint}$ | Delay time, stop bit to INTx↑ or stop bit to RXRDY# or read RBR to set interrupt | 1 | | RCLK |
| $t_{rint}$ | Propagation delay time, Read RBR/LSR to INTx↓/LSR interrupt↓ | | 12 | ns |
| $t_{rint}$ | Propagation delay time, IOR# RCLK↓ to RXRDY#↑ | | 12 | ns |
| $t_{mdo}$ | Propagation delay time, IOW#(WR MCR)↑ to RTSx#, DTRx#↑ | | 12 | ns |
| $t_{sim}$ | Propagation delay time, modem input CTSx#, DSRx#, and DCDx#↓↑ to INTx↑ | | 12 | ns |
| $t_{rim}$ | Propagation delay time, IOR#(RD MSR)↑ to interrupt↓ | | 3 | ns |
| $t_{sim}$ | Propagation delay time, Rix#↑ to INTx#↓ | | 12 | ns |

† The internal address strobe is always in active state.

‡ In the FIFO mode, td1= xxns (min) between reads of the FIFO and the status register.

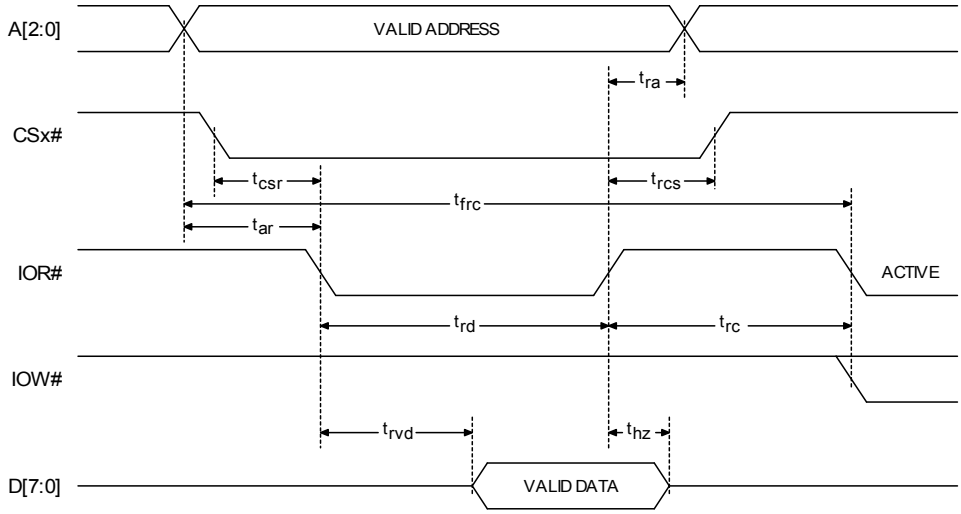



**Figure 10: Read Cycle Timing**

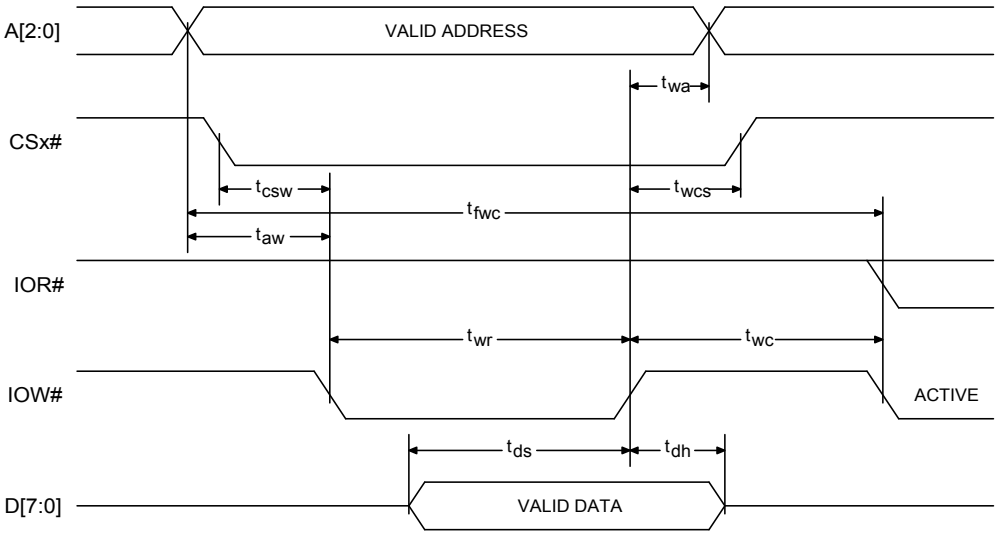



**Figure 11: Write Cycle Timing**

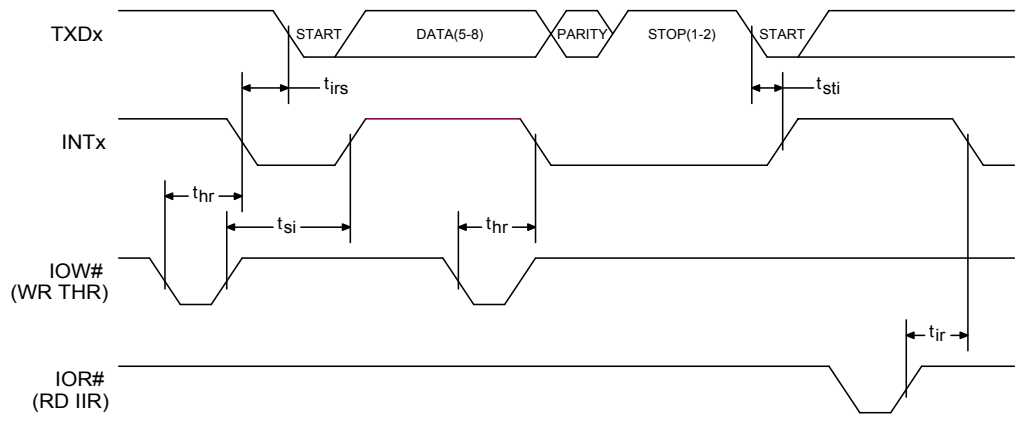



**Figure 12: Transmitter Timing**

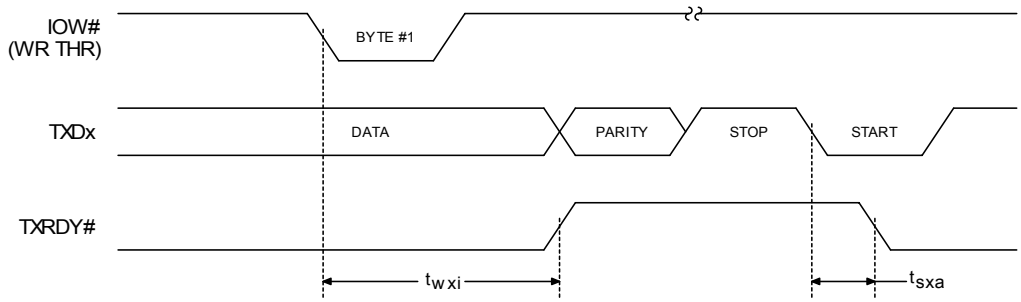



**Figure 13: Transmitter Ready Mode 0 Timing**

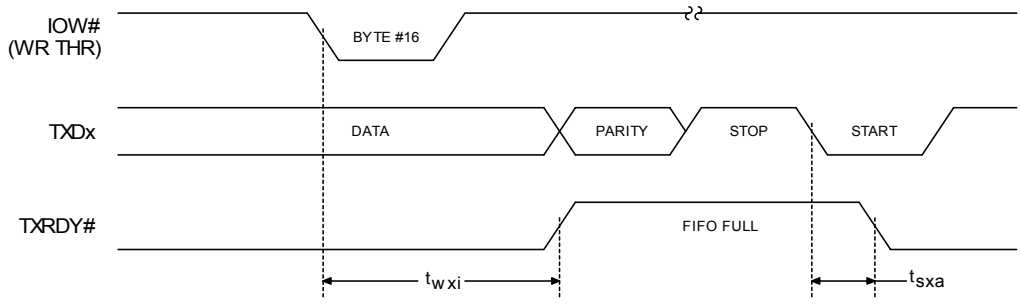



**Figure 14: Transmitter Ready Mode 1 Timing**

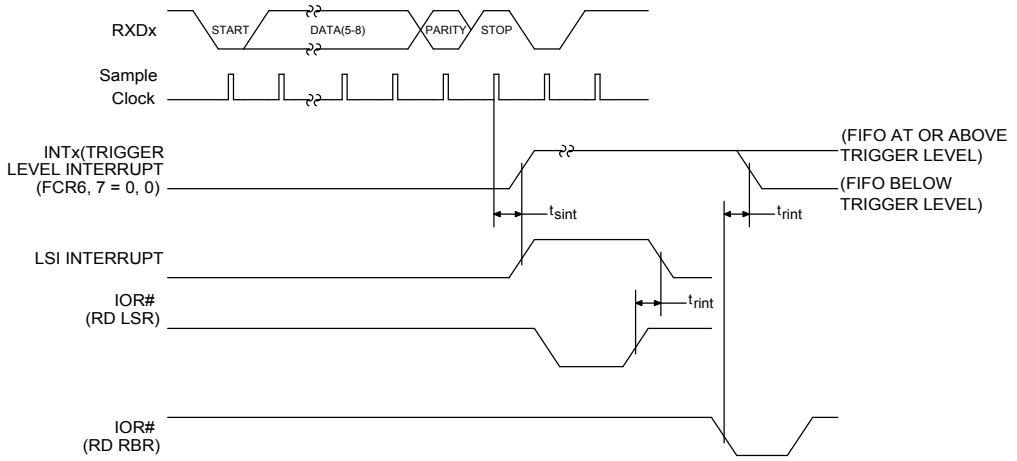



**Figure 15: Receiver FIFO First Byte (Sets RBR) Timing**

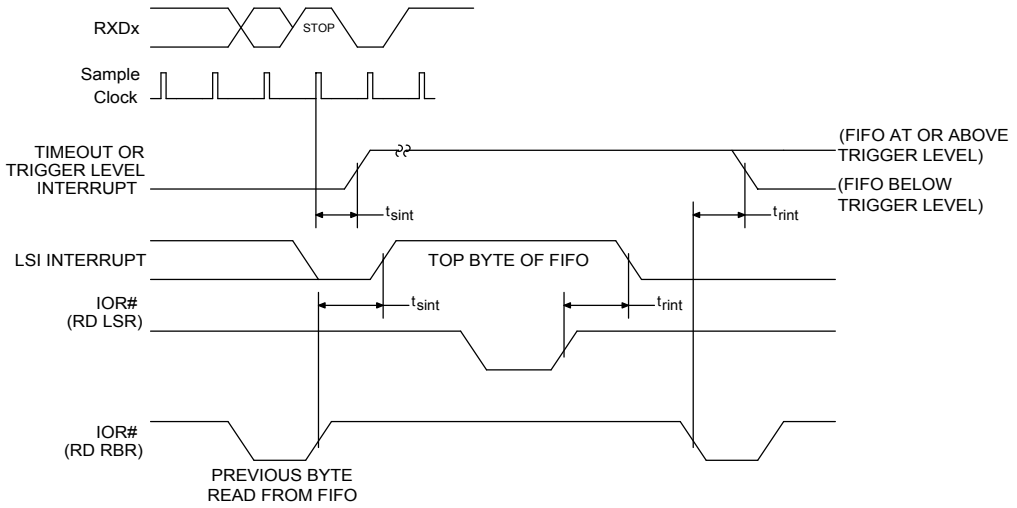



**Figure 16: Receiver FIFO After First Byte (After RBR Set) Timing**

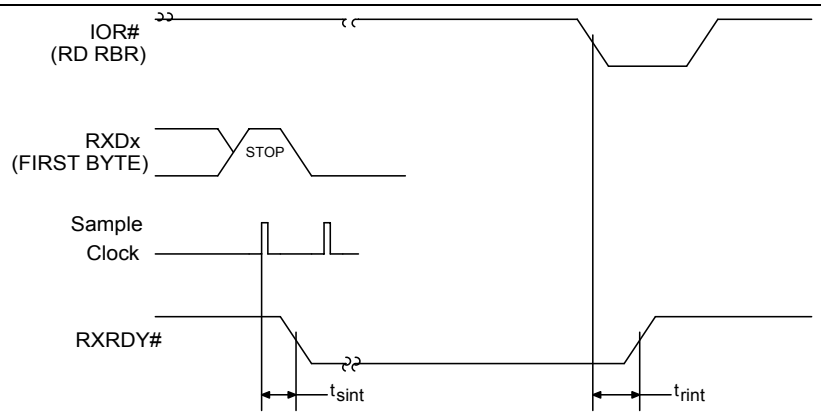



**Figure 17: Receiver Ready Mode 0 Timing**

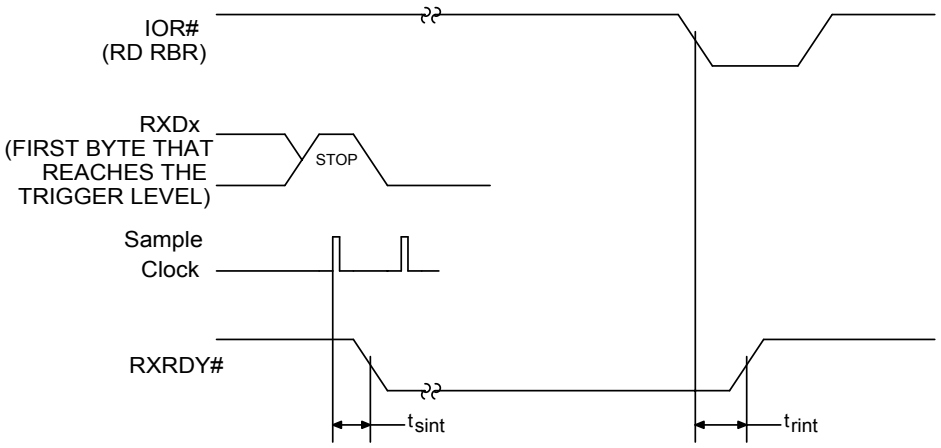



**Figure 18: Receiver Ready Mode 1 Timing**

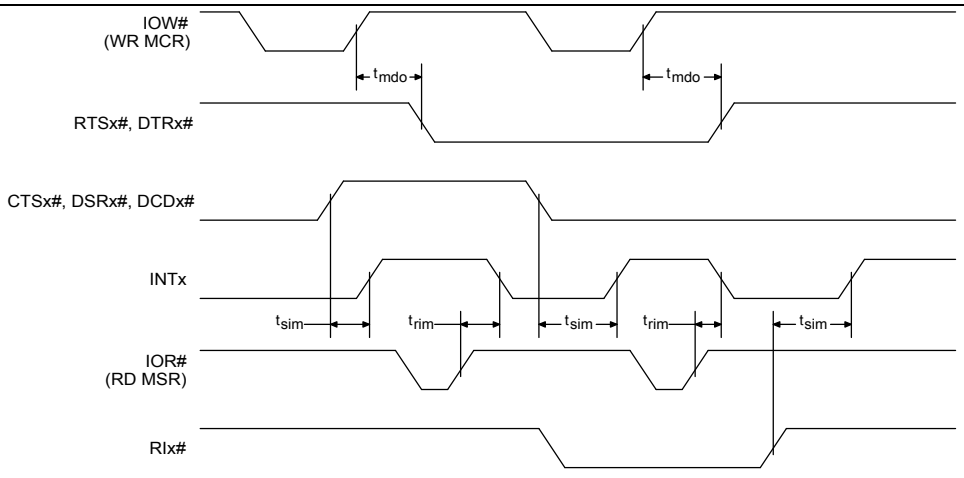
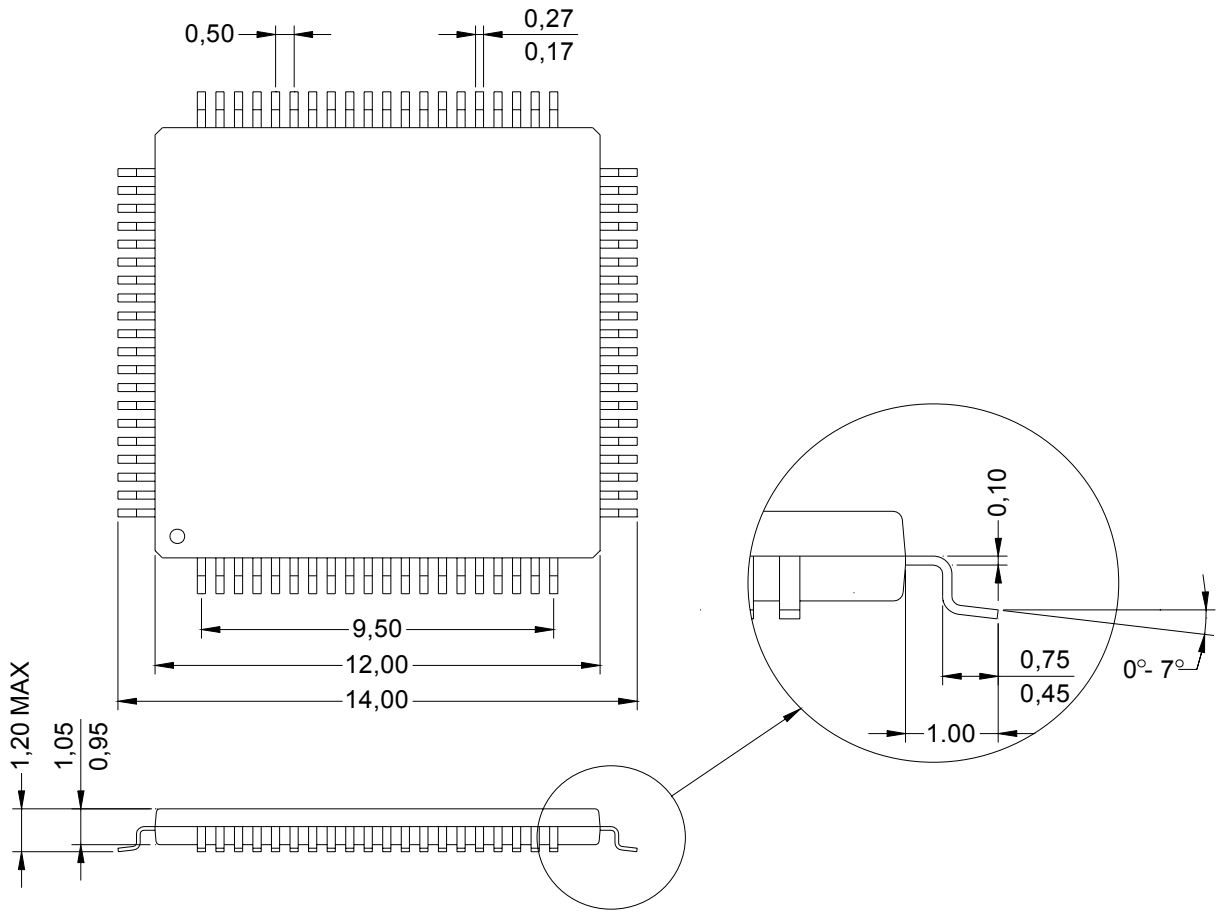



**Figure 19: Modem Control Timing**

## 10. Package Outline

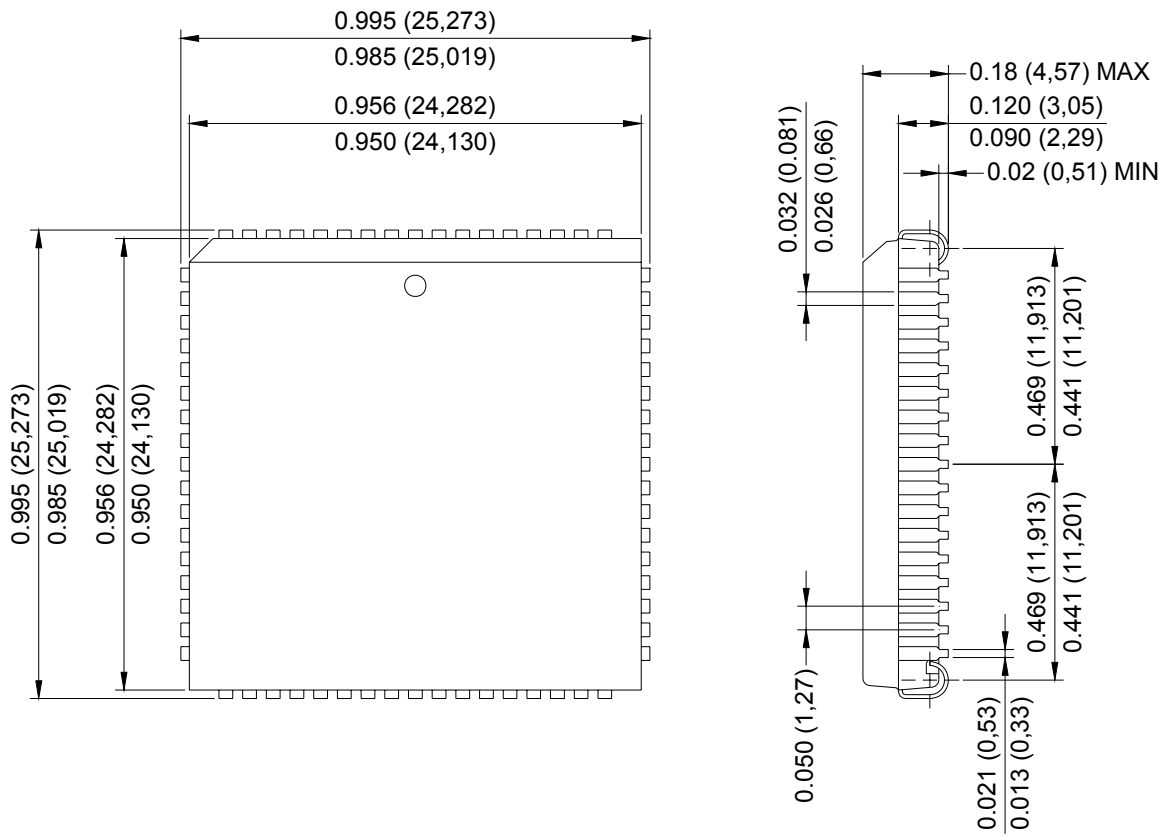
**80-Pin TQFP: Thin Plastic Quad Flat Package; Body 12 x 12 x 1.0 mm**




**Note :**

1. All dimensions are in millimeters.
2. Falls within ANSI Y14.5-1982

**68-Pin PLCC: Plastic Leaded Chip Carrier**




**Note :**

1. All dimensions are in inches (millimeters).
2. Falls within ANSI Y14.5-1982