PCI Target Interface Controller

SB16C1058PCI

Revision 1.04

SystemBase Co., Ltd.

CONTENTS

1. Description

SB16C1058PCI is a PCI Target Interface Controller with Octal-UART. It offers easy PCI Target Card Adapter implementation. SB16C1058PCI provides high performance serial communication. With a built-in eight SB16C1050 Cores that have built-in 256-byte FIFO, SB16C1058PCI decreases CPU load, is stronger at errors such as Overrun error and works well with simultaneous use of multiple ports. Furthermore, it is capable of waking up PC that is powered off through interrupts or Wake-up Requests with PCI Power Management implemented. SB16C1058PCI supports up to 8 serial ports extension, provides RS422/485 Auto Toggling function and Global Interrupt Function to the built-in UART allowing a more convenient handling of serial communication at driver level. Finally, with SB16C1058PCI, it is easy to design various modes of Serial Multi-Port with only a pin for Serial Multi-Port.

SB16C1058PCI offers LQFP176 packages.

2. Features

2.1 PCI Interface

- Compliant with PCI Local Bus Specification 2.3
- Supports 32-bit Bus / 33MHz and 66MHz
- Supports data transmission of max. 264MB/sec
- Supports Universal PCI working in both PCI and PCI-X slots
- Supports PCI Power Management 1.2
- Download Configuration Data from external serial EEPROM

- 3.3V Operation
- 5V Tolerant Inputs

2.2 Internal Octal-UART

- 8 Channel High Performance UART with 16C1050 core
- 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
- Fast UART Response Time, 24ns for IOR & IOW operation
- Programmable and Selectable Transmit and Receive FIFO Trigger Levels for Interrupt Generation
- Software (Xon/Xoff) / Hardware (RTS#/CTS#) Flow Control
  - Programmable Xon/Xoff Characters
  - Programmable Auto-RTS and Auto-CTS
- Optional Data Flow Resume by Xon Any Character Control
- Optional Data Flow Additional Halt by Xoff Re-transmit Control
- Control pins for RS-422 Point to Point/Multi-Drop Auto Control
- Control pins for RS-485 Echo/Non Echo Auto Control

- Software Selectable Baud Rate Generator
- Prescaler Provides Additional Divide-by-4 Function
- 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#)

2.3 Development Kit

SystemBase offers SB16C1058PCI Development Kit to minimize effort and cost of development, and maximize application stability.
SB16C1058PCI Development Kit includes hardware schematics, PCB CAD files, software device drivers and source codes and etc.
It will help you develop a new product easily and quickly.

3. Ordering Information

Table 3–1: Ordering Information

Part Number	Package	Operating Temperature Range	Device Status
SB16C1058PCI	176-Pin LQFP	-40 ℃ to +85 ℃	Active

4. Block Diagram

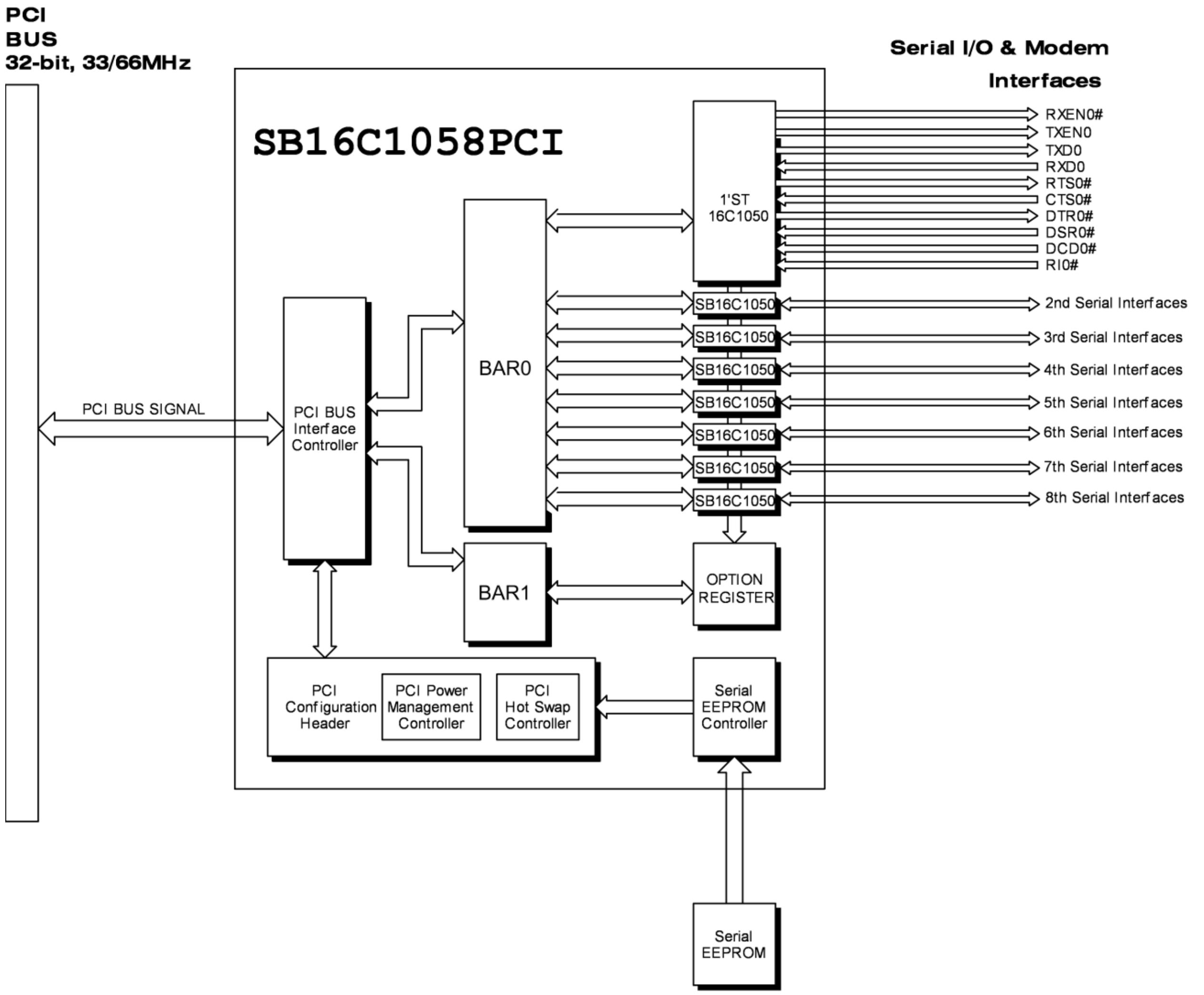


Figure 4–1: Block Diagram

5. Applications

5.1 Serial 8-port

Serial 8-port Card is generally made with SB16C1058PCI. Special logic is not needed to make one serial port since octal-UART are built inside the SB16C1058PCI.

SB16C1058PCI is the best and cost effective one-chip solution for 8-port serial card application. Users can make a serial application with RS232, RS422 or RS485 transceivers easily. Depending on Serial Interface, Transceiver IC of the RS232, RS422 or RS485 needs to be attached for long distance transmission. SB16C1058PCI is used as a PC add-in card type in Figure 5–1

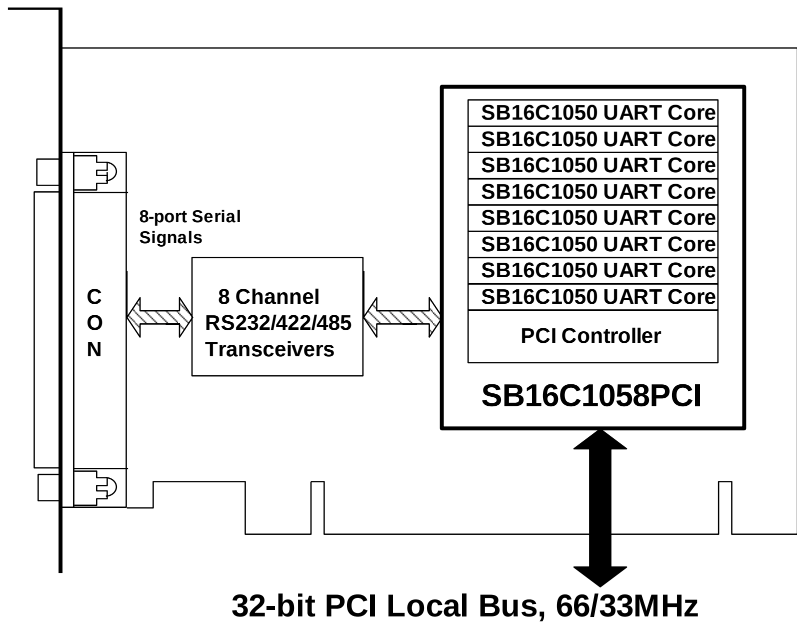


Figure 5–1: Serial 8-port Card Application Block Diagram

6. Pin Configuration

6.1 Pin Configuration for 176-Pin LQFP Package

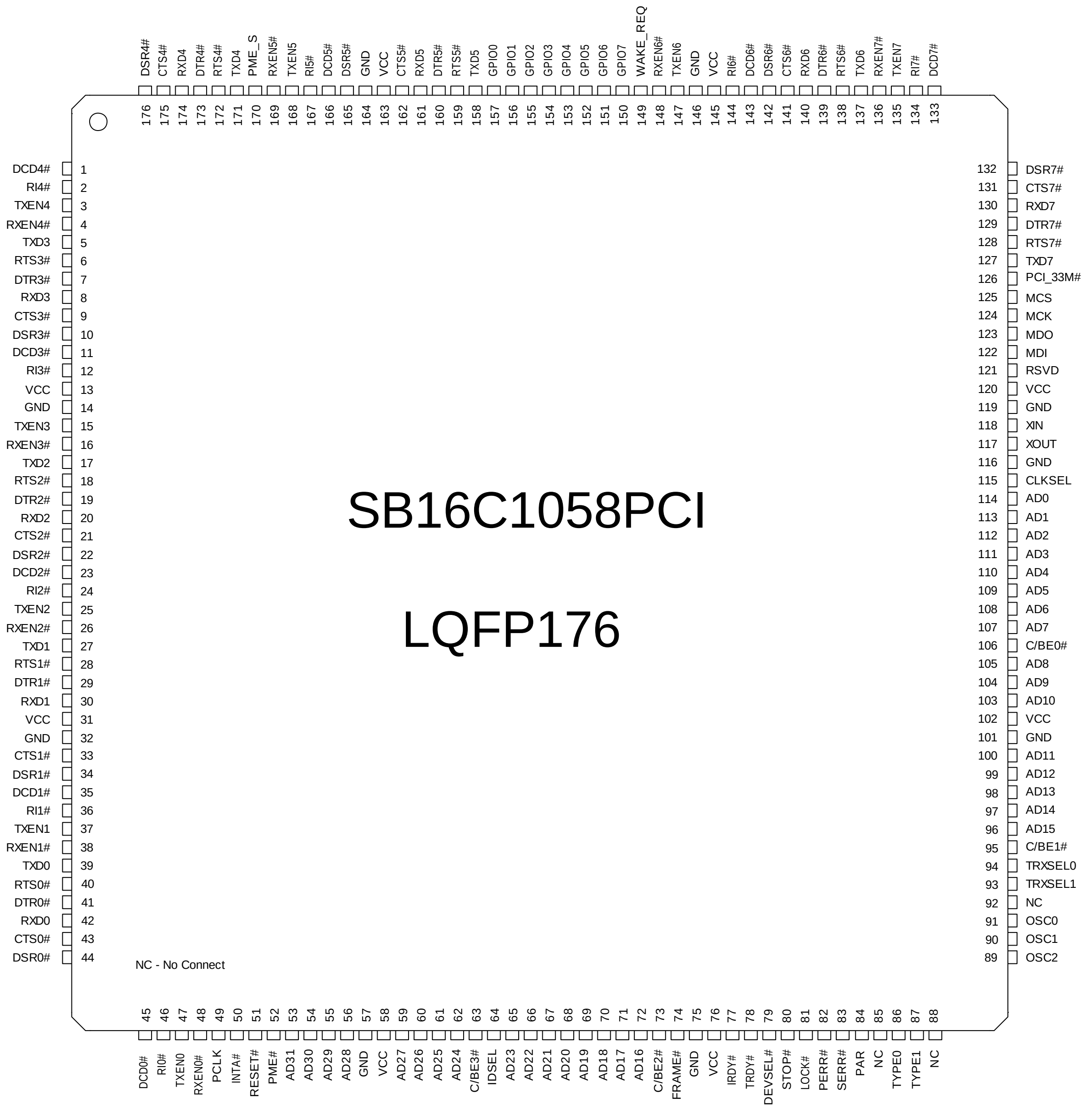


Figure 6–1: 176-Pin LQFP Pin Configuration

Table 6–1: Pin Description

Modem and Serial I/O Interface			
Name	**Pin**	**Type**	**Description**
TXD0 TXD1 TXD2 TXD3 TXD4 TXD5 TXD6 TXD7	39 27 17 5 171 158 137 127	O O O O O O O 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 RXD1 RXD2 RXD3 RXD4 RXD5 RXD6 RXD7	42 30 20 8 174 161 140 130	I I I I I I I 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.
RTS0# RTS1# RTS2# RTS3# RTS4# RTS5# RTS6# RTS7#	40 28 18 6 172 159 138 128	O O O O O O O O	**Request to Send (active low):** These pins indicate that the UART is ready to send data to the modem, and affect transmit and receive operations only when Auto-RTS function is enabled.
CTS0# CTS1# CTS2# CTS3# CTS4# CTS5# CTS6# CTS7#	43 33 21 9 175 162 141 131	I I I I I I I I	**Clear to Send (active low):** These pins indicate the modem is ready to accept transmitted data from the UART, and affect transmit and receive operations only when Auto-CTS function is enabled.
DTR0# DTR1# DTR2# DTR3# DTR4# DTR5# DTR6# DTR7#	41 29 19 7 173 160 139 129	O O O O O O O O	**Data Terminal Ready (active low):** These pins indicate UART is ready to transmit or receive data.

Table 6–1: Pin Description…*continued*

Modem and Serial I/O Interface			
Name	**Pin**	**Type**	**Description**
DSR0# DSR1# DSR2# DSR3# DSR4# DSR5# DSR6# DSR7#	44 34 22 10 176 165 142 132	I I I I I I I I	**Data Set Ready (active low):** These pins indicate modem is powered-on and is ready for data exchange with UART.
DCD0# DCD1# DCD2# DCD3# DCD4# DCD5# DCD6# DCD7#	45 35 23 11 1 166 143 133	I I I I I I I I	**Carrier Detect (active low):** These pins indicate that a carrier has been detected by modem.
RI0# RI1# RI2# RI3# RI4# RI5# RI6# RI7#	46 36 24 12 2 167 144 134	I I I I I I I I	**Ring Indicator (active low):** These pins indicate the modem has received a ringing signal from telephone line. A low to high transition on these input pins generates a modem status interrupt, if enabled.
TXEN0 TXEN1 TXEN2 TXEN3 TXEN4 TXEN5 TXEN6 TXEN7	47 37 25 15 3 168 147 135	O O O O O O O O	**TX Enable:** These pins are for auto tri-state control of the RS422 or RS485 communication. When serial date is transmitted to TXD, the value set on ATR[5] is transmitted. These pins eliminate additional glue logic outside.
RXEN0# RXEN1# RXEN2# RXEN3# RXEN4# RXEN5# RXEN6# RXEN7#	48 38 26 16 4 169 148 136	O O O O O O O O	**RX Enable:** This pins are for auto tri-state control of the RS422 or RS485 communication. When serial date is transmitted to TXD, the value set on ATR[7] is transmitted. These pins eliminate additional glue logic outside.

Table 6–1: Pin Description…*continued*

Function Configuration Interfaces			
Name	**Pin**	**Type**	**Description**
CLKSEL	115	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#.
OSC[2] OSC[1] OSC[0]	89 90 91	I I I	**Oscillator Select:** These pins are used to select the type of Oscillator used in Serial x-port Mode. The inputted value from these pins are shown in DIR[2:0] of the Option register. OSC[2:0] = 000b : 1.8432MHz OSC[2:0] = 001b : 3.6864MHz OSC[2:0] = 010b : 7.3728MHz OSC[2:0] = 011b : 14.7456MHz OSC[2:0] = 100b : 29.4912MHz OSC[2:0] = 101b : 58.9854MHz
INTF0[1] INTF0[0]	87 86	I I	**Line Interface Type Select:** These pins are used to select the type of Line Transceiver interfaced. The inputted values from these pins are shown in IIR0[5:4] of the Option register. INTF0[1:0] = 00b : RS232 transceiver is selected. INTF0[1:0] = 01b : RS422 transceiver is selected. INTF0[1:0] = 10b : RS485 transceiver is selected. INTF0[1:0] = 11b : Not supported.
TRXSEL[1] TRXSEL[0]	93 94	I I	**Auto Toggling Signal Select:** These pin select the signal for auto toggling in RS422 or RS485 driving. TRXSEL[1:0] = 00b : RTSx# signal is selected. TRXSEL[1:0] = 01b : DTRx# signal is selected. TRXSEL[1:0] = 10b : Dedicated controls(TXENx, RXENx#) are selected.
GPIO[7] GPIO[6] GPIO[5] GPIO[4] GPIO[3] GPIO[2] GPIO[1] GPIO[0]	150 151 152 153 154 155 156 157	I/O I/O I/O I/O I/O I/O I/O I/O	**General Purpose Input and Output:** These pins are controlled by GOER, GOR, GIR of the Option register.

Table 6-1: Pin Description…*continued*

Serial EEPROM Interfaces			
Name	**Pin**	**Type**	**Description**
RCS	125	O	**Serial EEPROM Chip Select:** Connected to CS of serial EEPROM.
RSK	124	O	**Serial EEPROM Data Clk:** Connected to SK of serial EEPROM.
RDI	123	O	**Serial EEPROM Data Input:** Connected to DI of serial EEPROM.
RDO	122	I	**Serial EEPROM Data Output:** Connected to DO of serial EEPROM.

PCI Interfaces			
Name	**Pin**	**Type**	**Description**
CLK	49	I	**PCI Clock:** PCI clock provides timing for all transaction on SB16C1058PCI.
RESET#	51	I	**PCI Reset:** Reset the SB16C1058PCI. The inputted signal indicates when the applied main power is within the specified tolerance and stable. This signal is asynchronous to CLK when asserted or deasserted.
INTA#	50	O/D	**Interrupt A:** Interrupt A is used to request an interrupt. Interrupts on PCI are defined as “level sensitive”, asserted low, using open drain output drivers. The assertion and deassertion of INTA# is asynchronous to CLK.
PME#	52	O/D	**Power Management Event:** This signal can be used by SB16C1058PCI to request a change in the SB16C1058PCI or main system power state. The assertion and deassertion of PME# is asynchronous to CLK. This signal has additional electrical requirements over and above standard open drain signals that allow it to be shared between devices that are powered off and those that are powered on. The use of this pin is specified in the PCI Bus Power Management Interface Specification.
AD[31], AD[30] AD[29], AD[28] AD[27], AD[26] AD[25], AD[24] AD[23], AD[22] AD[21], AD[20] AD[19], AD[18] AD[17], AD[16] AD[15], AD[14] AD[13], AD[12] AD[11], AD[10] AD[09], AD[08] AD[07], AD[06] AD[05], AD[04] AD[03], AD[02] AD[01], AD[00]	53, 54 55, 56 59, 60 61, 62 65, 66 67, 68 69, 70 71, 72 96, 97 98, 99 100,103 104,105 107,108 109,110 111,112 113,114	T/S T/S T/S T/S T/S T/S T/S T/S T/S T/S T/S T/S T/S T/S T/S T/S	**Address and Data:** These signals are multiplexed on the same pins. A Bus transaction consists of an address phase followed by a data phase. SB16C1058PCI does no support both read and write bursts.

Table 6–**1: Pin Description…*continued***

Name	Pin	Type	Description
C/BE[3]# C/BE[2]# C/BE[1]# C/BE[0]#	63 73 95 106	T/S T/S T/S T/S	**Bus Command and Byte Enables:** These signals are multiplexed on same pins. During the address phase of transaction, C/BE[3:0]# define the bus command. During the data phase, C/BE[3:0]# are used as Byte Enables.
PAR	84	T/S	**Parity:** Parity is even parity across AD[31:00] and C/BE[3:0]#.
FRAME#	74	S/T/S	**Cycle Frame:** This signal is driven by the master of main system to indicate the beginning and duration of an access.
IRDY#	77	S/T/S	**Initiator Ready:** This signal indicates the initiating agent's(main system's) ability to complete the current data phase of the transaction.
TRDY#	78	S/T/S	**Target Ready:** This signal indicates the target agent's(SB16C1058PCI's) ability to complete the current data phase of the transaction.
STOP#	80	S/T/S	**Stop:** This signal indicates that the current target(SB16C1058PCI) is requesting the master to stop current transaction.
LOCK#	81	S/T/S	**Lock:** This signal provides for exclusive use of a resource. SB16C1058PCI may be locked by one master at a time. See the PCI Local Bus Specification for the detail operation of lock function.
IDSEL	64	I	**Initialization Device Select:** This is used for chip selection during configuration read and write transaction.
DEVSEL#	79	S/T/S	**Device Select:** This signal indicates that the driving device has decoded its address as the target of the current access. As an input, DEVSEL# indicates whether any device on the bus has been selected.
PERR#	82	S/T/S	**Parity Error:** This signal is only for reporting data parity errors during all PCI transactions except Special Cycle.
SERR#	83	S/T/S	**System Error:** This signal is for reporting address parity errors, data parity errors on the Special Cycle command, or any other system error where the result will be catastrophic.
PCI_33M#	126	I	**PCI Operational Speed:** This input is for selecting the operational speed of PCI Bus. This chip is operated at 33MHz PCI Bus when this pin is cleared to 0b. And it is operated at 66MHz PCI Bus when this pin is setted to 1b.

Other Interfaces

Name	Pin	Type	Description
XIN	118	I	**Crystal or External Clock Input:** This input of up to 85MHz for data rate of 5.3Mbps at 3.3V.
XOUT	117	O	**Crystal or Buffed Clock Output:** This output level is 3.3V.
WAKE_REQ	149	I	**WAKE Request:** PM state of PCI Device goes from D3 state to D0 state with the Wake Up Event. This pin receives the event signal needed for the transition from D3 state to D0 state.
PME_S	170	O	**PME Status:** This signal indicates PM state. If PM state is in D0 state, it is set to 1b and if PM state is in D3 state, it is cleared to 0b.

Table 6-1: Pin Description…*continued*

Power and Ground			
Name	**Pin**	**Type**	**Description**
VCC	13, 31, 58, 76, 102, 120, 145, 163	PWR	**Power Supply:** Connect to +3.3V and to Core Ground through 0.1uF capacitors.
GND	14, 32, 57, 75, 101, 116, 119, 146, 164	GND	**Ground:** Connect to ground.
RSVD	121	I	**Reserved for Test:** Used for SB16C1058PCI internal testing. Must be wired to VCC via the 10K resistance.
NC	85, 88, 92	-	**No Connect.**

7. Configuration Loader

SB16C1058PCI can perform system initialization by reading PCI Configuration header data from external serial EEPROM.

When SB16C1058PCI is reset after power is granted, Configuration Loader inside SB16C1058PCI loads Configuration header data and etc from external serial EEPROM.

7.1 Serial EEPROM Information Table

When power is granted, Configuration Loader downloads Configuration Header data from external serial EEPROM.

Base Address Range that will be used is automatically selected depending on PORT input and no other configuration is needed at serial EEPROM.

Atmel's SPI(Serial Peripheral Interface) type EEPROM AT93C46 is recommended for external serial EEPROM.

Table 7-1: Serial EEPROM Information Table

Address	Description
00h	Vendor ID Low Byte
01h	Vendor ID High Byte
02h	Device ID Low Byte
03h	Device ID High Byte
04h	Revision ID
05h	Class Code Low Byte
06h	Class Code Middle Byte
07h	Class Code High Byte
08h	Sub Vendor ID Low Byte
09h	Sub Vendor ID High Byte
0Ah	Sub System ID Low Byte
0Bh	Sub System ID High Byte
0Ch~	Reserved

Vendor ID : Represents manufacturer of device. It is a unique ID given by PCI SIG and must be downloaded from external serial EEPROM. If you do not own Vendor ID, you can use 14A1h given to SystemBase by PCI SIG with permission.

Device ID : A unique ID of each device and is assigned at manufacturer's discretion and must be downloaded from serial EEPROM. If you do not prepare Device ID, you can use 0008h given to SystemBase.

Revision ID : It is a value representing device revision and must be downloaded from serial EEPROM. You have to use B0h.

Class Code : It provides the description on the function implemented by device. According to PCI specification, serial communication device have 070002h as class code. So you have to use 070002h.

Sub Vendor ID : Shows information about Subsystem manufacturer. Generally, Vendor ID or Device ID is information about Controller chip and Sub Vendor ID or Sub System ID is information about manufacturer who made the product with the chip.
It must be downloaded from serial EEPROM. If you do not prepare Sub Vendor ID, you can use 14A1h given to SystemBase by PCI SIG with permission.

Sub System ID : You can think of this as a Subsystem manufacturer's own Device ID.
It must be downloaded from serial EEPROM. If you do not prepare Sub System ID, you can use 0008h given to SystemBase.

8. PCI Configuration Space

PCI Configuration offers one type of Configuration Space access method.

- PCI Compatible Configuration method

PCI Compatible Configuration method is compatible with PCI version 2.3 and higher and supports 100% binary compatibility to software for operating system agreed bus list and organization.

From 0 byte up to 256 bytes is called PCI Compatible Configuration Space

8.1 Configuration Space Map of SB16C1058PCI

Table 8-1: Configuration Space Map

	Bit[31:24]	Bit[23:16]	Bit[15:8]	Bit[7:0]
Reg00	Device ID		Vendor ID	
Reg04	Status Register		Command Register	
Reg08	Class Code			Revision
Reg0C	BIST	Header Type	Latency Timer	Cache Line Size
Reg10	BAR0 (UART)			
Reg14	BAR1 (OPTION REGISTER)			
Reg18	BAR2 (Reserved)			
Reg1C	BAR3 (Reserved)			
Reg20	BAR4 (Reserved)			
Reg24	BAR5 (Reserved)			
Reg28	CardBus CIS Pointer			
Reg2C	Subsystem ID		Subsystem Vendor ID	
Reg30	Expansion ROM BAR			
Reg34	Reserved			Cap. Pointer
Reg38	Reserved			
Reg3C	Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line
Reg40	Power Management Capability			
Reg44	Power Management Control & Status			
Reg48~FF	Reserved			

Configuration Space of SB16C1058PCI can be divided into 2 following functions.

- PCI Compatible Configuration Registers
- Power Management Registers

PCI Compatible Configuration Registers are from Reg00 to Reg3C and these parts are compatible with existing PCI Configuration Registers. Power Management Registers are from Reg40 to Reg44.

SB16C1058PCI uses Configuration Register of Header Type0 which is used as Endpoint.

Following is a detailed description of PCI Compatible Configuration Register.

8.1.1 Vendor ID

A 16-bit register which represents the manufacturer of the device.
It is a unique ID given by PCI SIG after membership registration and must be downloaded from external serial EEPROM. If you do not own a Vendor ID, it is fine to use 14A1h given to SystemBase by PCI SIG.

8.1.2 Device ID

A 16-bit unique ID of each device given by the Function Manufacturer which can be assigned by the manufacturer freely and must be downloaded from serial EEPROM.
If you do not prepare Device ID, you can use 0008h given to SystemBase. It is related to software driver installation and recognition.

8.1.3 Command Register

Table 8–2: Command Register

Bit	Type	Description
15:11	RO	**Reserved:** Hardwired to 00000b
10	RW	**Interrupt Disable:** This bit controls PCI function's INTx interrupt signal creation ability. When it is 0b, function can assert INTx interrupt signal. When it is 1b, function cannot assert INTx interrupt signal. Default value of this bit is 0b.
9	RO	Hardwired to 0b.
8	RW	**SERR Enable:** If this bit is set and the function detects a non-fatal error and a fatal error, error reporting is executed to Root Complex. You can set the kind of errors to report to Device Control Register. Default value is 0b.
7	RO	Hardwired to 0b.
6	RW	**Parity Error Response:** A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b. Default value is 0b.
5	RO	Hardwired to 0b.
4	RO	Hardwired to 0b.
3	RO	Hardwired to 0b.
2	RW	Hardwired to 0b.
1	RW	**Memory Address Space Decoder Enable:** When this is 0b, Memory Decoder is disabled and Memory Transactions arriving to this device are responded with Completion of Unsupported Request state. When it is 1b, Memory Decoder is enabled and memory transactions arriving to this device are accepted and handled.
0	RW	**IO Address Space Decoder Enable:** When this is 0b, IO decoder is disabled and IO transactions arriving to this device are responded with Completion of Unsupported Request state. When it is 1b, IO decoder is enabled and IO transactions arriving to this device are accepted and handled.

8.1.4 Status Register

Table 8–3: Status Register

Bit	Type	Description
15	RW	**Detected Parity Error:** This bit must be set by the device whenever it detects a parity error, even if parity error handling is disabled. Default value of this bit is 0b.
14	RW	**Signaled System Error:** This bit must be set whenever the device asserts SERR#. Devices that will never assert SERR# do not need to implement this bit.
13	RW	**Received Master Abort:** This bit must be set by a master device whenever its transaction is terminated with Master-Abort. Default value of this bit is 0b.
12	RW	**Received Target Abort:** This bit must be set by a master device whenever its transaction is terminated with Target-Abort. Default value of this bit is 0b.
11	RW	**Signaled Target Abort:** This bit must be set by a target device whenever it terminates a transaction with Target-Abort. Default value of this bit is 0b.
10:9	RO	**DEVSEL Timing:** These bits encode the timing of DEVSEL#. These are encoded as 00b for fast, 01b for medium, and 10b for slow (11b is reserved). Hardwired to 01b.
8	RW	**Master Data Parity Error:** Set when three conditions are met: 1) the bus agent asserted PERR# itself (on a read) or observed PERR# asserted (on a write); 2) the agent setting the bit acted as the bus master for the operation in which the error occurred; and 3) the Parity Error Response bit (Command register) is set. Default value of this bit is 0b.
7	RO	**Fast Back-to-Back Capable:** Hardwired to 0b.
6	RO	**Reserved.** Hardwired to 0b.
5	RO	**66MHz-Capable:** Indicates whether or not this device is capable of running at 66MHz and hardwired to 1b.
4	RO	**Capabilities List:** Hardwired to 1b.
3	RO	**Interrupt Status:** Indicates that the function has an interrupt request that has not been processed yet. (Function is waiting to be serviced after asserting an interrupt signal. In other words, it is 1b when INTx# signal is asserted.)
2:0	RO	**Reserved.** Hardwired to 000b.

8.1.5 Revision

This register shows device revision. Manufacturer can assign it freely and must be downloaded from serial EEPROM. If you do not prepare Revision ID, you can use B0h given to SystemBase. It is also related to software device driver installation.

8.1.6 Class Code

This register contains descriptions on functions the device implements and must be downloaded from serial EEPROM. It is divided as Base Class, Sub Class and Programming Interface in bytes. It must be set to the values provided by PCI Bus Specification.
If you do not prepare Class Code, you can use 070002h given to SystemBase.
This Class Code(070002h) means a serial communication card adaptor.
Base Class Code is 07h(communication controller), Sub Class Code is 00h(serial controller) and Programming Interface is 02h(16C550 compatible).

8.1.7 Cache Line Size

This register assigns size of system's Cache Line. It is implemented as [RW] for compatibility with existing PCI. It is not supported and hardwired to 0000_0000b.

8.1.8 Latency Timer

This register assigns latency clock related to bus master which does burst access. It is not supported and hardwired to 0000_0000b.

8.1.9 Header Type

Configuration Space Header type and [RO]
Bit[7] : Shows whether device is Multi Function or Single Function. This product has default value 0b since it only supports Single Function.
Bit[6:0] : Assign header type after 10h. 00h is target device, 01h is PCI-to-PCI Bridge and 02h is CardBus bridge. This product has default value 00 since it is a target device.

8.1.10 BIST(Built-In Self Test)

Table 8–4: BIST

Bit	Type	Description
7	RO	**BIST Capable:** Hardwired to 0b.
6	RO	**Start BIST:** Hardwired to 0b.
5:4	RO	**Reserved:** Hardwired to 00b.
3:0	RO	**Completion Code:** Hardwired to 0000b.

8.1.11 Base Address Registers

These are spaces for assigning Base address for accessing I/O device or memory on PCI Local Bus. There are 6 spaces from Base Address Register 0 to 5, but spaces from Base Address Register 2 to 5 are set as unused reserved area.

In SB16C1058PCI, Base Address Register 0 is used for UART and Base Address Register 1 is used for Option Registers. Both of these Base Address Register spaces are used as space for I/O. When Base Address Register Bit[0] is 0b, the space is used as Memory space and when 1b, it is used as I/O space. Therefore, Bit[0] of Base Address Register 0 and Base Address Register 1 are all set to value 1b.

8.1.11.1 Base Address Register0

SB16C1058PCI has one Operating Mode which is Serial 8-port Mode. Base Address Register0(BAR0) of PCI Configuration automatically sets the size of the Address Space of eight UARTs. The size of the Address Space is listed on the chart below. See *'9. SB16C1058PCI Register Description'* for more details.

Table 8‒5: I/O Address Space for four Operating Mode

Operating Mode	I/O Address Space
Serial 8-port Mode	00~3Fh

8.1.11.2 Base Address register1

Aside from UART area, SB16C1058PCI contains Option Registers area which controls overall operations of the SB16C1058PCI. SB16C1058PCI sets this area with Base Address Register1(BAR1. I/O Address space size of the Option I/O Register is from 00 to 1Fh. Option Registers indicate information on the two operating modes, the Line Interface through the pin, Oscillator, Interrupt and SB16C1058PCI. See *'9. SB16C1058PCI Register Description'* for more details.

8.2 Power Management Registers of SB16C1058PCI

Sometimes control over power is needed on PCI Bus applied systems. Especially in cases when system uses independent power source like mobile system or when PCI device uses a lot of power, the system must limit power supply to PCI device when it is not in use to make a power efficient system. For this reason, *'PCI specification provides Power Management Interface Specification'* making Power Management more convenient. SB16C1058PCI supports PCI Power Management Interface Specification Revision 1.2. Content of registers that are implemented at Configuration Space Header is shown below. See *'Power Management Spec. Rev 1.2'* for more details.

Table 8–6: Power Management Register Block

<table>
<tr><td>Reg40</td><td colspan="2">Power management Capabilities (PMC)</td><td>Next Item Ptr</td><td>Capability ID</td></tr>
<tr><td>Reg44</td><td>Data</td><td>PMCSR_BSE Bridge Support Extensions</td><td colspan="2">Power Management Control/Status Register (PMCSR)</td></tr>
</table>

8.2.1 Capability ID (40h)

Capability ID regarding Power Management Interface and default value is 01h.

8.2.2 Pointer to Next Capability (41h)

A pointer that stores address of register which has information about next Capability. SB16C1058PCI does not have additional capabilities.
Hardwired to 0000_0000b.

8.2.3 Power Management Capabilities (42~43h)

Table 8–7: Power Management Capabilities

Bit	Type	Description
15:11	RO	**PME_Support:** These bits indicate the power states in which the function may assert PME#. SB16C1058PCI can assert PME# signal in $D3_{hot}$ and $D3_{cold}$ states and has value of 1_1000b.
10	RO	**D2_Support:** Tells if D2 Power Management State is supported. This device does not support D2 state and the bit is set to 0b.
9	RO	**D1_Support:** Tells if D1 Power Management State is supported. This device does not support D1 state and the bit is set to 0b.
8:6	RO	**Aux Current:** Report 3.3Vaux auxiliary current requirements for this device. This device is configured to require 375mA which is the maximum support capacity of an electric current supply and the bits are set to 111b.
5	RO	**Device-Specific Initialization (DSI):** Shows the need of DSI after transition from D3 to D0 uninitialized state. It should be set to 0b since initial value configuration for UART's communication is not needed here.
4	RO	**Reserved**
3	RO	**PME Clock:** Requires PCI clock to generate PME# signal and set to 0b.
2:0	RO	**Version:** Compatible with PCI PM Specification V1.2 and set to 011b.

8.2.4 Power Management Control/Status Register (44~45h)

This 16bit Register manages PCI Function's Power Management state and it is also used to enable and monitor PME.

Table 8–8: Power Management Control/Status Register

Bit	Type	Description
15	RW	**PME_Status:** This bit is set when the function would assert the PME# signal independent of the state of the PME_En bit. Writing 1b to this bit will clear it and cause the function to stop asserting a PME# (if enabled). Writing 0b has no effect. This bit is sticky and must be explicitly cleared by the OS each time the OS is initially loaded since this device supports PME# from $D3_{cold}$,
14:13	RO	**Data Scale:** This device did not implement Data Scale and hardwired to 00b.
12:9	RO	**Data Select:** This device did not implement Data Select and hardwired to 0000b
8	RW	**PME_En:** If PME_En is set to 1b, then the function can assert PME#. If PME_En is cleared to 0b, then the function do not assert PME#. This bit is sticky and must be explicity cleared by the OS each time the device is initially loaded since this device is supported to PME# from $D3_{cold}$.
7:4	RO	**Reserved:** Hardwired to 0b.
3	RO	**No_Soft_Reset:** Device does not execute internal reset when changing from $D3_{hot}$ to D0 through software control of PowerState bits. It's because full Re-Initialization is not needed for device to return to D0. Hardwired to 0b.
2	RO	**Reserved:** Hardwired to 0b.
1:0	RW	**PowerState:** PM S/W can decide Power Management state by configuring this section. PowerState = 00b means D0 state PowerState = 01b means D1 state PowerState = 10b means D2 state PowerState = 11b means $D3_{hot}$ state

9. Power Management

PCI was the most famous and useful bus since it was introduced in 1992. It is used in various computer systems from Laptops to Servers. It supported high performance applications by offering large bandwidth and efficiently supporting multiple masters. Also, it offers efficient power management through Power Management and various types of Form Factor modules and Applications.

PCI-PM defines four different Power States regarding PCI or PCI Express and interface for controlling these Power States. This device defines two different Power States.

Refer to '*PCI Bus Power Management Interface Specification Revision 1.2'* for more information on Power Management.

9.1 PCI Power Management

9.1.1 PCI Function Power State

4 power states are defined for PCI function. These are D0, D1, D2 and D3; D0 is maximum power consumption state and D3 is minimum power consumption state. D1 and D2 are middle states between D0(Power On) and D3(Power Off) and power consumption decreases as state changes to D3. As device changes from D0 to D3, it consumes lesser power and stores lesser Context information about current state. As a result, waiting time needed for the device to return to D0 increases.

D3 Power State organizes Special Category of Power Management State and Function can change to D3 state by physically removing Power from PCI device. D3 is classified into two states depending on existence or absence of Vcc. Those states are $D3_{hot}$ and $D3_{cold.}$

$D3_{hot}$ is the state where Vcc exist and it goes to maximum power-saving mode when both power and reference clock are supplied. When software writes D0 state on function's PMCSR register to get out of this mode, it can change into D0 state.

$D3_{cold}$ is classified into Power Off and Sleep state depending on existence of Vaux power.

At Power Off state, device's main power and Vaux are cut off and execution of Wake event is not possible. It is $D3_{cold}$ state. To get out of this state, push power button to start system.

At Sleep state, device's main power is cut off and only Vaux is supplied. It is $D3_{hot}$ state and Wake event can be executed. To get out of this state, assert PME# signal to Root Complex. The system wakeup by this can change to D0 state by re-assigning Vcc to this device and assigning RST#.

D0 state is classified into $D0_{uninitialized}$ and $D0_{active}$. $D0_{uninitialized}$ state is before system is initialized after Power has been supplied and $D0_{active}$ state is after system has been initialized.

All PCI function must support D0, $D3_{hot}$ and $D3_{cold.}$ SB16C1058PCI also support D0, $D3_{hot}$ and $D3_{cold}$ and do not support D1 and D2.

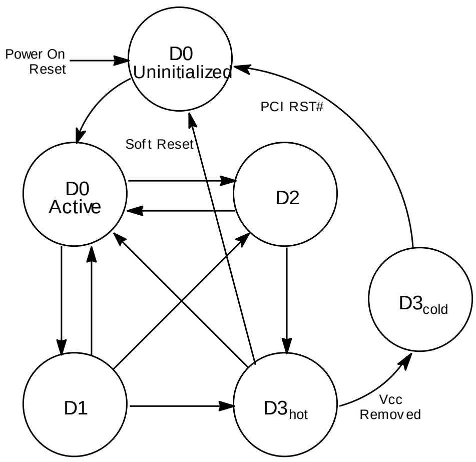


Figure 9-1: PCI Function Power Management State Transition

Cf. Hibernate state is variation of shutdown state. In this state, all states of computer is saved on disk and thus when power comes back, it can be started as current session.

9.2 SB16C1058PCI Power Management Pins and Functions

9.2.1 SB16C1058PCI Pins for Power Management

Table 9-2: SB16C1058PCI Pin Table for PM

Pin Name	Type	Description
WAKEREQ	I	Input of Wake Event Request. For example, it receives wakeup event signal generated by Ring Indicator.
PME#	O	Side band signal that Wakes Root Complex up to restore main power and reference clock that have been removed to implement Wakeup Event.
PME_S	O	Indicates PM state. It can be used for auto-power down function and it will be connected to FORCEOFF# pin on MAX3243.

9.2.2 SB16C1058PCI Power Management Wakeup implementation

Below figure is Logic Diagram of MAX3243. As you can see from this figure, RIN2 input signal (this pin is mainly prepared to be used by Ring Indicator.) is forked to reversed output signal called ROUT2 and output called ROUT2B. Among these, ROUT2B signal is not influenced by FORCEOFF# signal and input/output of buffer is not restricted. RIn input is screened by FORCEOFF# but above ROUT2B logic is Open when FORCEOFF# so RIn signal becomes an input without reversion. This signal is connected to WAKEREQ of SB16C1058PCI and handled as Wake Event. And if it is in $D3_{cold}$ state, this signal is asserted as PME# side band signal.

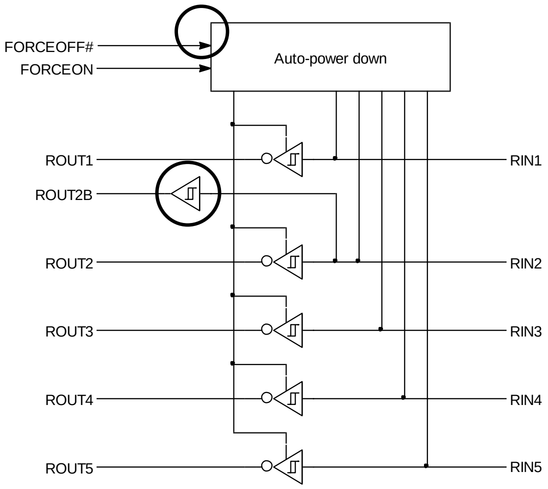
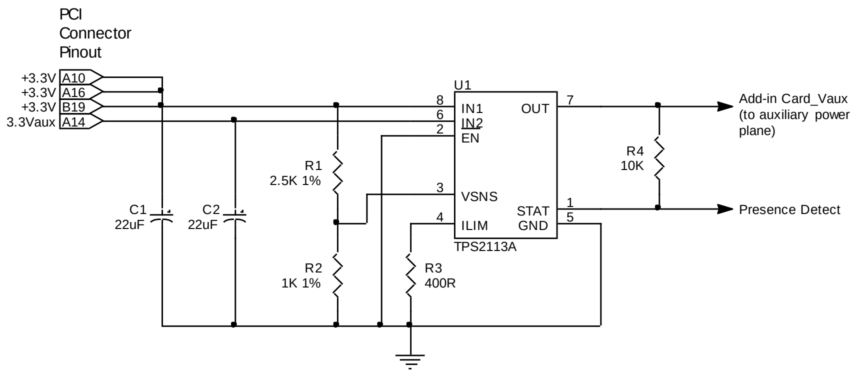


Figure 9‑2: Logic Diagram of MAX3243

9.2.3 3.3Vaux Presence Detection & Power Routing

PCI Add-In Card that implements a function which can generate Power Management Event from D3cold must decide existence of 3.3V on Pin B10(3.3Vaux) of PCI Bus. When weak pull-down attached to Pin B10 is implemented on system board that does not support supply of 3.3Vaux, it should be there to make logic low reference and must be implemented in all Add-In Card. On systems that do not supply 3.3Vaux through Pin B10, PCI Add-In Card must use any voltage source that Add-In Card can supply to provide supply to Aux Power of its own. So depending on existence of 3.3Vaux of Pin B10, design a circuit that supplies Power to its Aux Power as shown above.



VSNS > 0.8V	3.3Vaux (IN2) > +3.3V (IN1)	Add-in Card_Vaux (OUT)
Yes	X	+3.3V
No	No	+3.3V
No	Yes	3.3Vaux

Figure 9‑3: Sample Circuit for Aux Power Supply

10. UART I/O Space

UART I/O Space is determined by Base Address Register 0 (10h ~ 13h) from PCI Configuration Space. This is BAR0 area of Configuration Space and is for accessing actual physical UARTs.

10.1 UART I/O Address Map

8 bytes per Port are assigned since the type of installed UART is 16C550 compatible device. I/O area of BAR0 increased with number of port. For example, if it is 8 ports, I/O area is 64 bytes (8 bytes * 8 ports) size. Space taken by first UART is the least significant bit (LSB) and the space taken by the last UART is the most significant bit (MSB) in continuous UART area. Number for each port is UART number + 1 and next line shows how it is done. UART 0 = Port 1, UART 1 = Port 2, …, UART7 =PORT7.

Table 10–1: UART I/O Address Map

I/O Address	8-serial Mode
00 ~ 07h	UART0
08 ~ 0Fh	UART1
10 ~ 17h	UART2
18 ~ 1Fh	UART3
20 ~ 27h	UART4
28 ~ 2Fh	UART5
30 ~ 37h	UART6
38 ~ 3Fh	UART7

11. Option I/O Space

Option I/O Space is determined by Base Address Register 1 (14h ~ 17h) from PCI Configuration Space.
Contents of I/O register which is installed in this area include basic information about PCI Multi Port hardware. The size of this area is 32(00h ~ 1Fh) bytes total.
This Option Registers are made by SystemBase for users to control and manage Serial Multi-Port more easily and conveniently. Users and software developers can easily control Serial Interface of Multi-Port with them.

Table 11−1: Option I/O Register Map Cf. RO – Read Only, WO – Write Only, RW – Read/Write

I/O Address	Register Name	I/O
00h	GIR0 (General Information Register 0 – Port Number)	RO
01h	GIR1 (General Information Register 1 – Product Version)	RO
02h	GIR2 (General Information Register 2 – Minor Version)	RO
03h	GIR3 (General Information Register 3 – PCI Core Version)	RO
03h	SRR (Software Reset Register)	WO
04h	DIR (Port1 ~ Port8, Device Information Register)	RO
05 ~ 07h	Reserved	-
08h	IIR0 (Port1 ~ Port8, Interface Information Register)	RW
09 ~ 0Bh	Reserved	-
0Ch	IMR (Port1 ~ Port8, Interrupt Mask Register)	RW
0D ~ 0Fhh	Reserved	-
10h	IPR (Port1 ~ Port8, Interrupt Poll Register)	RO
11 ~ 17h	Reserved	-
18h	PPMRR (PM_PME Message Resource Register in D3hot)	RW
19 ~ 1Bh	Reserved	-
1Ch~1Fh	Reserved	-
20h	GOER (GPIO Output Enable Register)	RW
21h	GOR (GPIO Output Register)	RW
22h	GIR (GPIO Input Register)	RO
23 ~ 2Fh	Reserved	-

11.1 General Information Register0 – Port Number (GIR0)

Port Number: Shows how many ports are installed on current Serial Multi-Port and it has 8 ports.
Generally, type of Serial Multi-Port shows number of ports that are installed.

11.2 General Information Register1 – Product Version (GIR1)

General Information Register1 indicates the version of PCI Target Controller. (Currently B0h meaning B.0)

11.3 General Information Register2 – Sub-Product Version (GIR2)

General Information Register2 indicates the sub-product the version of PCI Target Controller.
Sub-Product Version = 03h means Serial 8-port mode

11.4 General Information Register3 – Core Version (GIR3)

General Information Register3 indicates SystemBase's PCI Target Interface Core version (Currently 24h meaning 2.4)

11.5 Software Reset Register

If 52h("R") is written on SRR, Reset is outputted to Serial Multi-Port I/O Bus and this means PCI UART goes to Reset state. If values other than 52h are written on SRR, Reset state is cleared.

11.6 Device Information Register (DIR)

DIR: Device information of Port1 ~ Port8

Table 11–2: Device Information Register Description

Bit	Symbol	Description
7:4	DIR[7:4]	**UART Select**: Content of U[2:0] shows type of UART. 000b: 16C550 compatible UART 001b: 16C1050 010 ~ 111b: Not supported
3:0	DIR[3:0]	**Oscillator Frequency Select:** O[3:0] shows frequency (maximum communication speed) of communication modification sender that is used. 0000b: 1.8432MHz (115.2Kbps) 0001b: 3.6864MHz (230.4Kbps) 0010b: 7.3728MHz (460.8Kbps) 0011b: 14.7456MHz (921.6Kbps) 0100b: 29.4912MHz (1,8432.2Kbps) 0101b: 58,9842MHz (3,6864.4Kbps) 0110 ~ 1111b: Not supported

11.7 Interface Information Register0 ~ 1 (IIR0 ~ 1)

IIR0 indicates interface information for Serial 8-port mode.

Table 11–3: Interface Information Register 0 Description

Bit	Symbol	Description
7:4	IIR0[7:4]	**Interface Type Indicator:** 0000b: RS232 interface is selected by INTF0[1:0]. 0001b: RS422 interface is selected by INTF0[1:0]. 0010b: RS485 interface is selected by INTF0[1:0].
3:2	IIR0[3:2]	**Auto Toggling Signal Indicator:** When IIR0[7:4] is 0000b (RS232 interface selected): Meaningless. When IIR0[7:4] is 0001b or 0010b (RS422 or RS485 interface selected): 00b: RTS signal line selected by TRXSEL[1:0]. 01b: DTR signal line selected by TRXSEL[1:0]. 10b: TXEN, RXEN# signal line selected by TRXSEL[1:0].
1:0	IIR0[1:0]	Not used. Hardwired to 00b

11.8 Interrupt Mask Register (IMR)

IMR enables or disables each interrupt of Serial 8 ports.

Table 11-4: Interrupt Mask Register Description

Bit	Symbol	Description
7	IMR[7]	0b: Disables Port8 interrupt. 1b: Enables Port8 interrupt.
6	IMR[6]	0b: Disables Port7 interrupt. 1b: Enables Port7 interrupt.
5	IMR[5]	0b: Disables Port6 interrupt. 1b: Enables Port6 interrupt.
4	IMR[4]	0b: Disables Port5 interrupt. 1b: Enables Port5 interrupt.
3	IMR[3]	0b: Disables Port4 interrupt. 1b: Enables Port4 interrupt.
2	IMR[2]	0b: Disables Port3 interrupt. 1b: Enables Port3 interrupt.
1	IMR[1]	0b: Disables Port2 interrupt. 1b: Enables Port2 interrupt.
0	IMR[0]	0b: Disables Port1 interrupt. 1b: Enables Port1 interrupt.

11.9 Interrupt Poll Register (IPR)

IPR indicates interrupt generation state of Port 1 ~ Port 8.

Table 11-5: Interrupt Poll Register Description

Bit	Symbol	Description
7	IPR[7]	0b: Port8 interrupt has occurred. 1b: Port8 interrupt has not occurred.
6	IPR[6]	0b: Port7 interrupt has occurred. 1b: Port7 interrupt has not occurred.
5	IPR[5]	0b: Port6 interrupt has occurred. 1b: Port6 interrupt has not occurred.
4	IPR[4]	0b: Port5 interrupt has occurred. 1b: Port5 interrupt has not occurred.
3	IPR[3]	0b: Port4 interrupt has occurred. 1b: Port4 interrupt has not occurred.
2	IPR[2]	0b: Port3 interrupt has occurred. 1b: Port3 interrupt has not occurred.
1	IPR[1]	0b: Port2 interrupt has occurred. 1b: Port2 interrupt has not occurred.
0	IPR[0]	0b: Port1 interrupt has occurred. 1b: Port1 interrupt has not occurred.

11.10 PME# Signal Resource Register (PSRR)

Select event signal to wakeup Root Complex in $D3_{hot}$ state.

Table 11–6: PME# Signal Resource Register Description

Bit	Symbol	Description
1	PSRR[1]	0b: Interrupt is not selected as Wakeup Event for waking up Root Complex (default). 1b: Interrupt is selected as Wakeup Event for waking up Root Complex. Whether interrupt is generated or not is determined by IMR. That is, some port can only generate interrupt or any ports among all ports can generate interrupt. When interrupt occurs, asserts PME# signal to Root Complex.
0	PSRR[0]	0b: WAKEREQ pin is not selected as Wakeup Event for waking up Root Complex (default). 1b: WAKEREQ pin is selected as Wakeup Event for waking up Root Complex. When 1b is received by any logic, asserts PME# signal to Root Complex.

If PSRR[1:0] is set as 11b which means both $D3_{hot}$-Interrupt and $D3_{hot}$-WAKEREQ are set, PME# signal is asserted when only one of both events occurs.

11.11 GPIO Output Enable Register (GOER)

GOER enables or disables GPIO[7:0] to output ports respectively.

Table 11–7: GPIO Output Enable Register Description

Bit	Symbol	Description
7	GOER[7]	0b: GPIO[7] is selected to input port (default). 1b: GPIO[7] is selected to output port.
6	GOER[6]	0b: GPIO[6] is selected to input port (default). 1b: GPIO[6] is selected to output port.
5	GOER[5]	0b: GPIO[5] is selected to input port (default). 1b: GPIO[5] is selected to output port.
4	GOER[4]	0b: GPIO[4] is selected to input port (default). 1b: GPIO[4] is selected to output port.
3	GOER[3]	0b: GPIO[3] is selected to input port (default). 1b: GPIO[3] is selected to output port.
2	GOER[2]	0b: GPIO[2] is selected to input port (default). 1b: GPIO[2] is selected to output port.
1	GOER[1]	0b: GPIO[1] is selected to input port (default). 1b: GPIO[1] is selected to output port.
0	GOER[0]	0b: GPIO[0] is selected to input port (default). 1b: GPIO[0] is selected to output port.

11.12 GPIO Output Register (GOR)

GOR sets output of GPIO[7:0] respectively.

Table 11–8: GPIO Output Register Description

Bit	Symbol	Description
7	GOR[7]	Sets the output of GPIO[7].
6	GOR[6]	Sets the output of GPIO[6].
5	GOR[5]	Sets the output of GPIO[5].
4	GOR[4]	Sets the output of GPIO[4].
3	GOR[3]	Sets the output of GPIO[3].
2	GOR[2]	Sets the output of GPIO[2].
1	GOR[1]	Sets the output of GPIO[1].
0	GOR[0]	Sets the output of GPIO[0].

11.13 GPIO Input Register (GIR)

Reads input of GPIO[7:0] respectively.

Table 11–9: GPIO Input Register Description

Bit	Symbol	Description
7	GIR[7]	Reads the input of GPIO[7].
6	GIR[6]	Reads the input of GPIO[6].
5	GIR[5]	Reads the input of GPIO[5].
4	GIR[4]	Reads the input of GPIO[4].
3	GIR[3]	Reads the input of GPIO[3].
2	GIR[2]	Reads the input of GPIO[2].
1	GIR[1]	Reads the input of GPIO[1].
0	GIR[0]	Reads the input of GPIO[0].

12. UART(SB16C1050) Functional Description

SB16C1050 offers 16C450 and 16C650 modes. When FIFO is enabled, it has a register configuration compatible with 64-byte FIFO and 16C650, so it becomes compatible with 16C650. If you enable 256-byte FIFO, you use the unique supreme function that SB16C1050 offers. It offers communication speed up to 5.3Mbps and more enhanced functions that other UARTs with 128-byte FIFO do not.

SB16C1050 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.

12.1 FIFO Operation

SB16C1050's FIFO has two modes, 64-byte FIFO mode and 256-byte FIFO mode. Setting FCR[0] to 1b enables FIFO, and if AFR[0] is set to 0b, 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 SB16C1050. Setting both FCR[0] and AFR[0] to 1b 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.

12.2 Hardware Flow Control

Hardware flow control is done 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 0b, 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 1b, 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.

12.2.1 Auto-RTS

To enable Auto-RTS, EFR[6] should be set to 1b. Once enabled, RTS# outputs 0b. If the number of received data in RX FIFO is larger than the value stored in FUR, RTS# will be changed to 1b and if not, holds 0b. This state indicates that RX FIFO can accept more data. After RTS# changed to 1b 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 0b and reports that it can accept more data. That is, if RTS# is 0b now, RTS# is not changed to 1b until the amount in RX FIFO exceeds the value set in FUR. But if RTS# is 1b now, RTS# is not changed to 0b 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 00b, '01', '10', and 11b, FUR stores 8, 16, 56, and 60, respectively. And if FCR[5:4] holds 00b, '01', '10', and 11b, 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 0b or 1b by FSR[5]. If FSR[5] is 0b, RTS# is 0b and if 1b, RTS# is 1b, too.

When IER[6] is set to 1b and RTS# is changed from 0b to 1b 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 0b to 1b after the first STOP bit is received. Figure 12–1 shows the RTS# timing chart while Auto-RTS is enabled.

In Figure 12–1, 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.

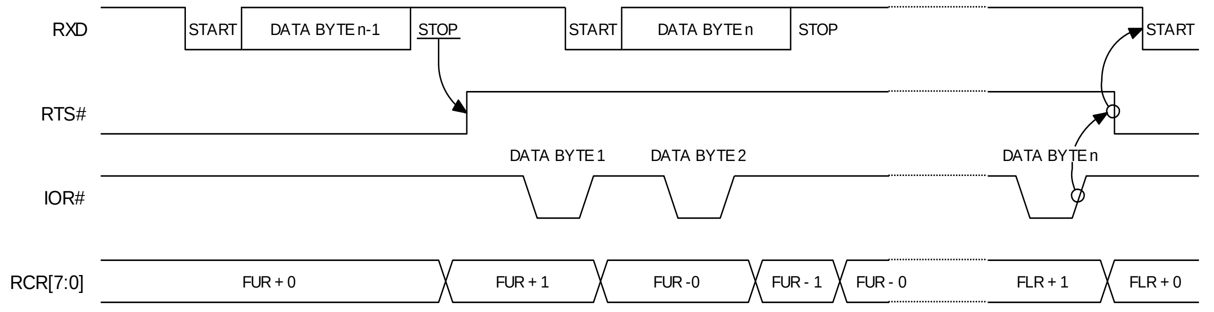


Figure 12–1: RTS# Functional Timing

12.2.2 Auto-CTS

Setting EFR[7] to 1b enables Auto-RTS. If enabled, data in TX FIFO are determined to be transmitted or suspended by the value of CTS#. If 0b, it means external UART can receive new data and data in TX FIFO are transmitted through TXD pin. If 1b, 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 0b, CTS# is 0b and it means external UART can accept new data, If 1b, CTS# is 1b 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 1b, interrupt is generated by Auto-CTS when the input of CTS# is changed from 0b to 1b, and it is shown on ISR[5:0]. Interrupts generated by Auto-CTS are removed if MSR is read.

12.3 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 12–1 shows software flow control options.

12.3.1 Transmit Software Flow Control

To make Transmit Software Flow Control enabled, EFR[3:2] must be set to 01b, 10b or 11b. Unlike Auto-RTS in which 0b 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 00b, 01, 10 , and 11b, FUR is 8, 16, 56, and 60, respectively. And if FCR[5:4] is 00b, 01b, 10b, and 11b, 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 0b, the status is Xon and if 1b, 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.

Table 12-1: 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

12.3.2 Receive Software Flow Control

To make Receive Software Flow Control enabled, EFR[1:0] must be set to 01b, 10b or 11b. 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 0b, RX Software Flow Control status is XON and it means external UART can accept new data. If 1b, 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 1b, 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 1b is inputted on CTS# pin, so data in TX FIFO are suspended. That is, these two bits are set to 1b 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 12–2 below.

Table 12–2: 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

When 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.

When 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.

12.3.3 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 1b. While it is enabled, XOFF status changes to XON status though Xoff character arrives.
Details about it are described in 12.3.2 Receive Software Flow Control.

12.3.4 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 SB16C1050 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 00b, 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 11b, 16 more data. If the value of FUR is approaching the FIFO size, 256-byte, it is good to write XRCR[1:0] 00b. If the 256-FUR value is small, it is good to select 00b of XRCR and if large, it is good to select 11b.

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 1b and set MCR[2] to 1b again. Then Xoff Re-transmit function is enabled. When disabling it, first set MCR[6] to 1b and then clear MCR[2] to 0b.

12.4 Sleep Mode with Auto Wake-Up

The SB16C1050 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 1b.

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 SB16C1050 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.

12.5 Programmable Baud Rate Generator

The SB16C1050 has a programmable baud rate generator with a prescaler. The prescaler is controlled by MCR[7], as shown in Figure 12–2. The MCR[7] sets the prescaler to divide the clock frequency by 1 or 4. 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:

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

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

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

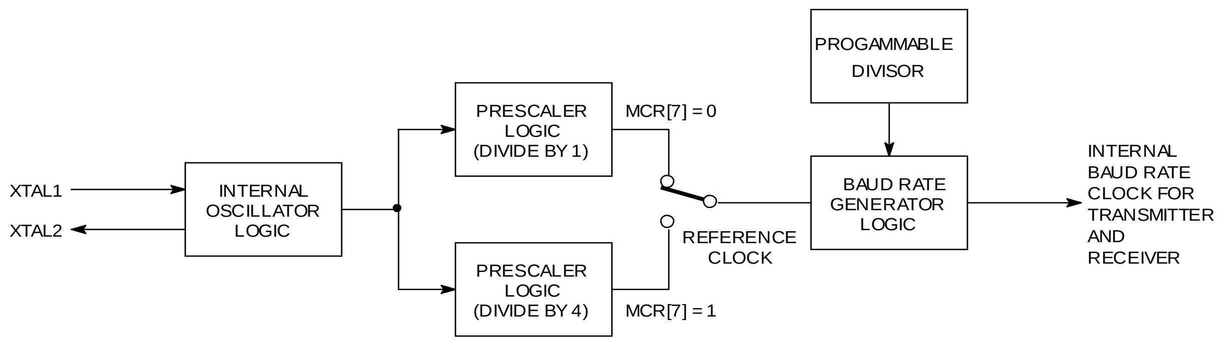


Figure 12–2: Prescaler and Baud Rate Generator Block Diagram

DLL and DLM must be written 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 SB16C1050 is effectively disabled, as no baud clock will be generated.

Table 12–3 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 12–3 shows the crystal clock circuit reference.

Table 12–3: 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

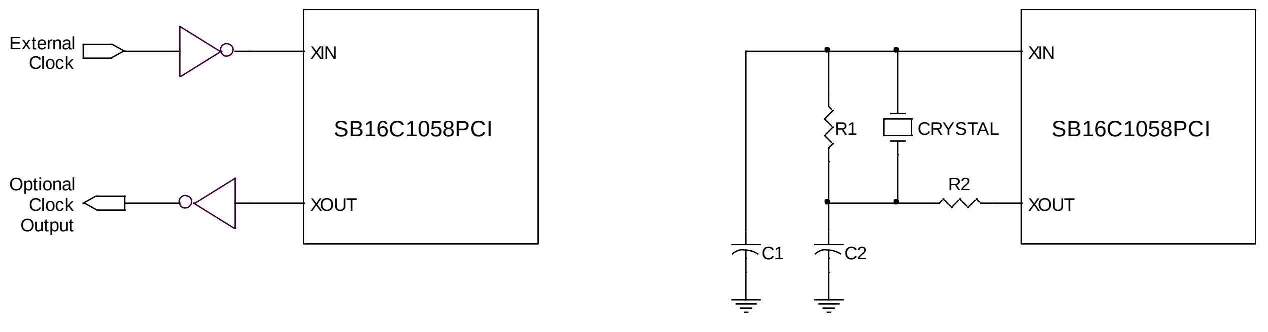


Figure 12–3: Crystal Clock Circuit Reference Diagram

Table 12–4: Component Values

Frequency Range (MHz)	C1 (pF)	C2 (pF)	R1 (Ω)	R2(Ω)
3.6~8	22	68	220K	470 ~ 1.5K
8~16	33~68	33 ~ 68	220K ~ 2.2M	470 ~ 1.5K

12.6 Break and Time-out Conditions

Break Condition:

Break Condition occurs when TXD signal outputs 0b and sustains for more than one character.

It occurs if LCR[6] is set to 1b and deleted if 0b. If break condition occurs 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 12–4 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. SB16C1050 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 1b and can be verified by ISR.

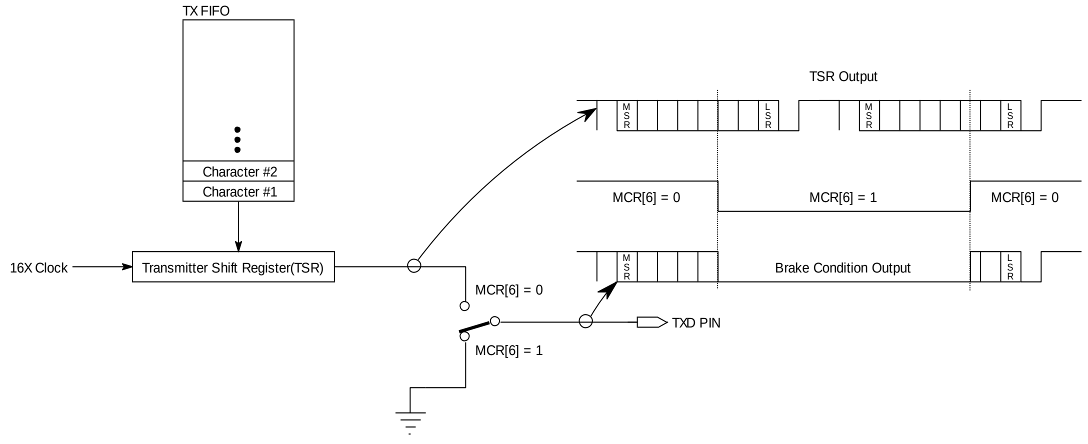


Figure 12–4: Break Condition Block Diagram

13. Register Descriptions

Each UART channel in the SB16C1050 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 13–1 and Table 13–2.

Table 13–1: Internal Registers Map

<table>
<tr><th rowspan="2">Address A[2:0]</th><th>Page 0</th><th>Page 1</th><th>Page 2</th><th>Page 3</th><th>Page 4</th></tr>
<tr><th>LCR[7] = 0
MCR[6] = 0</th><th>LCR[7] = 1
LCR[7:0] ≠ BFh</th><th>LCR[7] = 0
MCR[6] = 1</th><th>LCR = BFh
PSR[0] = 0</th><th>LCR = BFh
PSR[0] = 1</th></tr>
<tr><td>0h</td><td>THR/RBR</td><td>DLL</td><td>—</td><td>PSR</td><td>PSR</td></tr>
<tr><td>1h</td><td>IER</td><td>DLM</td><td>—</td><td>ATR</td><td>AFR</td></tr>
<tr><td>2h</td><td colspan="2">FCR/ISR</td><td>—</td><td>EFR</td><td>XRCR</td></tr>
<tr><td>3h</td><td colspan="5">LCR</td></tr>
<tr><td>4h</td><td colspan="3">MCR</td><td>XON1</td><td>TTR</td></tr>
<tr><td>5h</td><td colspan="2">LSR</td><td>TCR</td><td>XON2</td><td>RTR</td></tr>
<tr><td>6h</td><td colspan="2">MSR</td><td>RCR</td><td>XOFF1</td><td>FUR</td></tr>
<tr><td>7h</td><td colspan="2">SPR</td><td>FSR</td><td>XOFF2</td><td>FLR</td></tr>
</table>

Table 13–1: Internal Registers Map…*continued*

Address A[2:0]	Register	Read/Write	Comments
	Page 0 Registers		
0h	THR : Transmit Holding Register RBR : Receive Buffer Register	Write-only 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 ISR : Interrupt Status Register	Write-only Read-only	LCR[7] = 0, MCR[6] = 0, 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, LCR[7] = 1, LCR ≠ BFh, LCR[7] = 0, MCR[6] = 1
5h	LSR : Line Status Register	Read-only	LCR[7] = 0, MCR[6] = 0, LCR[7] = 1, LCR ≠ BFh
6h	MSR : Modem Status Register	Read-only	LCR[7] = 0, MCR[6] = 0, LCR[7] = 1, LCR ≠ BFh
7h	SPR : Scratch Pad Register	Read/Write	LCR[7] = 0, MCR[6] = 0, 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 ISR : Interrupt Status Register	Write-only Read-only	LCR[7] = 0, MCR[6] = 0, 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, LCR[7] = 1, LCR ≠ BFh, LCR[7] = 0, MCR[6] = 1
5h	LSR : Line Status Register	Read-only	LCR[7] = 0, MCR[6] = 0, LCR[7] = 1, LCR ≠ BFh
6h	MSR : Modem Status Register	Read-only	LCR[7] = 0, MCR[6] = 0, LCR[7] = 1, LCR ≠ BFh
7h	SPR : Scratch Pad Register	Read/Write	LCR[7] = 0, MCR[6] = 0, LCR[7] = 1, LCR ≠ BFh

Table 13–1: Internal Registers Map…*continued*

Address A[2:0]	Register	Read/Write	Comments
	Page 2 Registers		
0h	None	—	—
3h	LCR : Line Control Register	Read/Write	—
4h	MCR : Modem Control Register	Read/Write	LCR[7] = 0, MCR[6] = 0, LCR[7] = 1, LCR ≠ BFh, 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, 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	XON1b: Xon1 Character Register	Read/Write	LCR = BFh, PSR[0] = 0
5h	XON2 : Xon2 Character Register	Read/Write	LCR = BFh, PSR[0] = 0
6h	XOFF1b: 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, 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 13–2: 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				
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 13–2: 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	0	0	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

13.1 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.

13.2 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.

13.3 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 13-3 shows IER bit settings.

Table 13–3: Interrupt Enable Register Description

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

13.4 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 13-4 shows ISR[7:0] bit settings.

Table 13–4: Interrupt Status Register Description

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

Table 13–4: 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_0100	2	Receive Data Available	Receiver data available, reaches trigger level.	Reading the RBR or RCR falls below trigger level.
00_1100	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 TCR 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 0b to 1b.	Reading the ISR.

13.5 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 13-5 shows FCR bit settings.

Table 13–5: FIFO Control Register Description

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

13.6 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 13–6 shows LCR bit settings.

Table 13–6: Line Control Register Description

Bit	Symbol	Description
7	LCR[7]	**Divisor Latch Enable:** 0b: Disable the divisor latch (default). 1b: Enable the divisor latch.
6	LCR[6]	**Break Enable:** 0b: No TX break condition output (default). 1b: Forces TXD output to 0b, for alerting the communication terminal to a line break condition.
5	LCR[5]	**Set Stick Parity:** LCR[5:3] = xx0b: No parity is selected. LCR[5:3] = 0x1b: Stick parity disabled. (default) LCR[5:3] = 101b: Stick parity is forced to 1b. LCR[5:3] = 111b: Stick parity is forced to 0b.
4	LCR[4]	**Parity Type Select:** LCR[5:3] =001b: Odd parity is selected. LCR[5:3] =011b: Even parity is selected.
3	LCR[3]	**Parity Enabled:** 0b: No parity (default). 1b: A parity bit is generated during the transmission and the receiver checks for receive parity.
2	LCR[2]	**Number of Stop Bits:** LCR[2:0] = 0xxb: 1 stop bit (word length = 5, 6, 7, 8). LCR[2:0] = 100b: 1.5 stop bits (word length = 5). LCR[2:0] = 11xb or 1x1b: 2 stop bits (word length = 6, 7. 8).
1:0	LCR[1:0]	**Word Length Bits:** 00b: 5 bits (default). 01b: 6 bits. 10b: 7 bits. 11b: 8 bits.

13.7 Modem Control Register (MCR, Page 0)

MCR controls the interface with the modem, data set, or peripheral device that is emulating the modem. Table 13–7 shows MCR bit settings.

Table 13–7: Modem Control Register Description

Bit	Symbol	Description
7	MCR[7]	**Clock Prescaler Select:** 0b: Divide by 1 clock input (default). 1b: Divide by 4 clock input.
6	MCR[6]	**Page 2 Select/Xoff Re-Transmit Access Enable:** 0b: Enable access to page 0 register when LCR[7] is 0b (default). 1b: Enable access to page 2 register and Xoff re-transmit bit when LCR[7] is 0b.
5	MCR[5]	**Xon Any Enable:** 0b: Disable Xon any (default). 1b: Enable Xon any.
4	MCR[4]	**Internal Loop Back Enable:** 0b: Disable loop back mode (default). 1b: 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:** 0b: INTx outputs disabled (default). During loop back mode, OUT2 output 0b and it controls MSR[7] to 1b. 1b: INTx outputs enabled. During loop back mode, OUT2 output 1b and it controls MSR[7] to 0b. OUT2 is not available as an output pin on the SB16C1050.
2	MCR[2]	**OUT1/Xoff Re-transmit Enable:** 0b: Xoff re-transmit disable when MCR[6] is 0b. During loop back mode, OUT1 output to 0b and it controls MSR[6] to 1b. 1b: Xoff re-transmit enable when MCR[6] is 1b. During loop back mode, OUT1 output to 1b and it controls MSR[6] to 0b. OUT1 is not available as an output pin on the SB16C1050. Xoff re-transmit is operated with XRCR, refer to XRCR.
1	MCR[1]	**RTS# Output:** 0b: Force RTS# output to 1b. During loop back mode, controls MSR[4] to 1b. 1b: Force RTS# output to 0b. During loop back mode, controls MSR[4] to 0b.
0	MCR[0]	**DTR# Output:** 0b: Force DTR# output to 1b. During loop back mode, controls MSR[5] to 1b. 1b: Force DTR# output to 0b. During loop back mode, controls MSR[5] to 0b.

13.8 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 13–8 shows LSR bit settings.

Table 13–8: Line Status Register Description

Bit	Symbol	Description
7	LSR[7]	**RX FIFO data error Indicator:** 0b: No RX FIFO error (default). 1b: 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:** 0b: THR or TSR is not empty. 1b: THR and TSR are empty.
5	LSR[5]	**THR Empty Indicator:** 0b: THR is not empty. 1b: 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 1b.
4	LSR[4]	**Break Interrupt Indicator:** 0b: No break condition (default). 1b: The receiver received a break signal (RXD was 0b 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:** 0b: No framing error (default). 1b: Framing error. It indicates that the received character did not have a valid stop bit.
2	LSR[2]	**Parity Error Indicator:** 0b: No parity error (default). 1b: 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:** 0b: No overrun error (default). 1b: 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:** 0b: No character in the RBR or RX FIFO. 1b: At least one character in the RBR or RX FIFO.

13.9 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 1b when input from modem changes and cleared to 0b as soon as CPU reads MSR. Table 13–9 shows MSR bit settings.

Table 13–9: Modem Status Register Description

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

13.10 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.

13.11 Divisor Latches (DLL, DLM, Page 1)

Two 8-bit registers which store the 16-bit divisor for generation of the clock in baud rate generator. DLM 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 DLM can only be written to before sleep mode is enabled, i.e., before IER[4] is set. Chapter 12.7 describes the details of divisor latches.

13.12 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_0000b and if 64, it is shown as 0100_0000b. 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_0000b and if 255, it is shown as 0_1111_1111b. And in case of the maximum number 256, it is shown as 1_0000_0000b.

13.13 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_0000b and if 64, it is shown as 0100_0000b. 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_0000b and if 255, it is shown as 0_1111_1111b. And in case of the maximum number 256, it is shown as 1_0000_0000b.

13.14 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 13–10: Flow Control Status Register Description

Bit	Symbol	Description
7:6	FSR[7:6]	Not used, always 00b.
5	FSR[5]	**TX Hardware Flow Control Status:** 0b: When FIFO or Auto-RTS flow control is disabled. 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. 1b: 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. For more details, refer to '12.2 Hardware Flow Control'.
4	FSR[4]	**TX Software Flow Control Status:** 0b: When FIFO or Software flow control is disabled. 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. 1b: 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. For more details, refer to '12.3 Software Flow Control'.
3:2	FSR[3:2]	Not used, always 00b.
1	FSR[1]	**RX Hardware Flow Control Status:** 0b: When FIFO or Auto-CTS flow control is disabled. If FIFO and Auto-CTS flow control is enabled, 0b is inputted in CTS# pin and it means external device can receive more data. This time data in TX FIFO are transmitted. 1b: If FIFO and Auto-CTS flow control is enabled, 1b is inputted in CTS# pin and it means external device can not receive more data. This time data in TX FIFO are not transmitted. For more details, refer to '12.2 Hardware Flow Control'.
0	FSR[0]	**RX Software Flow Control Status:** 0b: 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. 1b: 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. For more details, refer to '12.3 Software Flow Control'.

13.15 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 13–11 shows PSR bit settings.

Table 13–11: Page Select Register Description

Bit	Symbol	Description
7:1	PSR[7:1]	**Access Key:** When writing data on PSR to change page, Access Key must be correspondent. If the value of PSR[7:1] is 1010_010b, data is written on PSR[0] and page can be selected. If PSR[7:1] is read, it reads 0000_000b which is irrespective of Access Key.
0	PSR[0]	**Page Select:** 0b: Page 3 is selected (default). 1b: Page 4 is selected.

13.16 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 13–12 shows ATR bit settings.

Table 13–12: Auto Toggle Control Register Description

Bit	Symbol	Description
7	ATR[7]	**RXEN# Polarity Select:** 0b: Asserted output of RXEN# is 0b. 1b: Asserted output of RXEN# is 1b. (default)
6	ATR[6]	**RXEN# Control Mode Select:** Only when ATR[1:0] is 11b; 0b: RXEN# is outputted as same as ATR[7], irrespective of TXD signal. (default) 1b: RXEN# is outputted as same as ATR[7] when TXD signal is not transmitting. And outputted as complement of ATR[7] when TXD signal is transmitting.
5	ATR[5]	**TXEN Polarity Select:** 0b: Asserted output of TXEN is 0b. 1b: Asserted output of TXEN is 1b. (default)
4	ATR[4]	**TXEN Control Mode Select:** 0b: TXEN is outputted as same as ATR[5], irrespective of TXD signal. (default) 1b: TXEN is outputted as complement of ATR[5] when TXD signal is not transmitting, and outputted as same as ATR[5] when TXD signal is transmitting..
3:2	ATR[3:2]	Not used, always 00b.
1:0	ATR[1:0]	**Auto Toggle Enable:** 00b: Auto toggle disable (default). 01b: Not used. 10b: Not used. 11b: Auto toggle enable.

Cf. After reset, TXEN and RXEN# output '0b'.

13.17 Enhanced Feature Register (EFR, Page 3)

EFR enables or disables the enhanced features of the UART. Table 13–13 shows EFR bit settings.

Table 13–13: Enhanced Feature Register Description

Bit	Symbol	Description
7	EFR[7]	**Auto-CTS Flow Control Enable:** 0b: Auto-CTS flow control is disabled (default). 1b: Auto-CTS flow control is enabled. Transmission stops when CTS# pin is inputted 1b. Transmission resumes when CTS# pin is inputted 0b.
6	EFR[6]	**Auto-RTS Flow Control Enable:** 0b: Auto-RTS flow control is disabled (default). 1b: Auto-RTS flow control is enabled. The RTS# pin outputs 1b when data in RX FIFO fill above the FUR. RTS# pin outputs 0b when data in RX FIFO fall below the FLR.
5	EFR[5]	**Special Character Detect:** 0b: Special character detect disabled (default). 1b: 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 1b to indicate that a special character has been detected.
4	EFR[4]	**Enhanced Function Bits Enable:** 0b: Disables enhanced functions and writing to IER[7:4], FCR[5:4], MCR[7:5]. 1b: 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:** 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 12–1 “Software flow control options (EFR[3:0])” on page 45.

13.18 Additional Feature Register (AFR, Page 4)

AFR enables or disables the 256-byte FIFO mode and controls the global interrupt. Table 13–14 shows AFR bit settings.

Table 13–14: Additional Feature Register Description

Bit	Symbol	Description
7:6	AFR[7:1]	Not used, always 000_0000b.
0	AFR[0]	**256-byte FIFO Enable:** 0b: 256-byte FIFO mode is disabled and this means SB16C1050 operates as Non FIFO mode or 64-byte FIFO mode (default). 1b: 256-byte FIFO mode is enabled and ISR[7:6] operates as 256-TX FIFO Empty and 256-RX FIFO Full.

13.19 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 13–15 shows XRCR bit settings.

Table 13–15: Xoff Re-transmit Count Register Description

Bit	Symbol	Description
7:2	XRCR[7:2]	Not used, always 0000_00b.
1:0	XRCR[1:0]	**Xoff Re-transmit Count Select:** 00b: Transmits Xoff character whenever the number of received data is 1 during XOFF status. (default) 01b: Transmits Xoff character whenever the number of received data is 4 during XOFF status. 10b: Transmits Xoff character whenever the number of received data is 8 during XOFF status. 11b: Transmits Xoff character whenever the number of received data is 16 during XOFF status.

13.20 Transmit FIFO Trigger Level Register (TTR, Page 4)

TTR 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 80h, 1000_0000b. 0000_0000b should never be written. If written, unexpected operation may occur.

13.21 Receive FIFO Trigger Level Register (RTR, Page 4)

RTR 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 80h, 1000_0000b. 0000_0000b should never be written. If written, unexpected operation may occur.

13.22 Flow Control Upper Threshold Register (FUR, Page 4)

FUR 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, 1b 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 1b is outputted on RTS# pin. The value of FUR must be larger than that of FLR.

13.23 Flow Control Lower Threshold Register (FLR, Page 4)

FLR 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, 0b 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 0b is outputted on RTS# pin. The value of FLR must be less than that of FUR.

Table 13–16: SB16C1050 Reset Conditions

Registers	Reset State
	Page 0
RBR	[7:0] = XXXX_XXXXb
IER	[7:0] = 0000_0000b
FCR	[7:0] = 0000_0000b
ISR	[7:0] = 0000_0001b
LCR	[7:0] = 0000_0000b
MCR	[7:0] = 0000_0000b
LSR	[7:0] = 0110_0000b
MSR	[7:4] = 0000b [3:0] = Logic levels of the inputs inverted
SPR	[7:0] = 0000_0000b
	Page 1
DLL	[7:0] = 1111_1111b
DLM	[7:0] = 1111_1111b
	Page 2
TCR	[7:0] = 0000_0000b
RCR	[7:0] = 0000_0000b
FSR	[7:0] = 0000_0000b
	Page 3
PSR	[7:0] = 0000_0000b
ATR	[7:0] = 1010_0000b
EFR	[7:0] = 0000_0000b
XON1	[7:0] = 0000_0000b
XON2	[7:0] = 0000_0000b
XOFF1	[7:0] = 0000_0000b
XOFF2	[7:0] = 0000_0000b
	Page 4
AFR	[7:0] = 0000_0000b
XRCR	[7:0] = 0000_0000b
TTR	[7:0] = 1000_0000b
RTR	[7:0] = 1000_0000b
FUR	[7:0] = 0000_0000b
FLR	[7:0] = 0000_0000b
Output Signals	**Reset State**
TXD, RTS#, DTR#	Logic 1
TXEN, RXEN#	Logic 0
INT	Tri-State Condition = INTSEL is open or low state Logic 0 = INTSEL is high state

14. 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 14–1: Register Programming Guide

Command	Action
Set Baud Rate to VALUE1, VALUE2	Read LCR, then save in temp Set LCR to 80h Set DLL to VALUE1 Set DLM to VALUE2 Set LCR to temp
Set Xon1, Xoff1 to VALUE1, VALUE2	Read LCR, then save in temp Set LCR to BFh Set Xon1 to VALUE1 Set Xoff1 to VALUE2 Set LCR to temp
Set Xon2, Xoff2 to VALUE1, VALUE2	Read LCR, then save in temp Set LCR to BFh Set Xon2 to VALUE1 Set Xoff2 to VALUE2 Set LCR to temp
Set Software Flow Control Mode to VALUE	Read LCR, then save in temp Set LCR to BFh Set EFR to VALUE Set LCR to temp
Set flow control threshold for 64-byte FIFO Mode	1) Set FCR to 0000_xxx1b → Set FUR to 8, set FLR to 0 2) Set FCR to 0101_xxx1b → Set FUR to 16, set FLR to 8 3) Set FCR to 1010_xxx1b → Set FUR to 56, set FLR to 16 4) Set FCR to 1111_xxx1b → Set FUR to 60, set FLR to 56
Set flow control threshold for 256-byte FIFO Mode	Set FCR to xxxx_xxx1b Read LCR, then save in temp Set LCR to BFh Set PSR to A5h Set AFR to 01h

Table 14–1: Register Programming Guide…*continued*

Command	Action
	Set FUR to Upper Threshold Value Set FLR to Lower Threshold Value Set PSR to A4h Set LCR to temp
Set TX FIFO / RX FIFO Interrupt Trigger Level for 64-byte FIFO Mode	1) Set FCR to 0000_xxx1b → Set RTR to 8, set TTR to 8 2) Set FCR to 0101_xxx1b → Set RTR to 16, set TTR to 16 3) Set FCR to 1010_xxx1b → Set RTR to 56, set TTR to 32 4) Set FCR to 1111_xxx1b → 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_xxx1b Read LCR, then save in temp Set LCR to BFh Set PSR to A5h Set AFR to 01h Set TTR to TX FIFO Trigger Level Value Set RTR to RX FIFO Trigger Level Value Set PSR to A4h Set LCR to temp
Read Flow Control Status	Read LCR, then save in temp1 Read MCR, then save in temp2 Set LCR to (0111_1111b **AND** temp1) Set MCR to (0100_0000b **OR** temp2) Read FSR, then save in temp3 Pass temp3 back to host Set MCR to temp2 Set LCR to temp1
Read TX FIFO / RX FIFO Count Value	Read LCR, then save in temp1 Read MCR, then save in temp2 Set LCR to (0111_1111b **AND** temp1) Set MCR to (0100_0000b **OR** temp2) Read TCR, then save in temp3 Read RCR, then save in temp4 Pass temp3 back to host Pass temp4 back to host Set MCR to temp2 Set LCR to temp1

Table 14–1: Register Programming Guide…*continued*

Command	Action
Read 256-byte TX FIFO Empty Status / RX FIFO Full Status	Set FCR to xxxx_xxx1b Read LCR, then save in temp1 Set LCR to BFh Set PSR to A5h Set AFR to 01h Set PSR to A4h Set LCR to temp1 Read ISR, then save in temp2 Pass temp2 back to host
Enable Xoff Re-transmit	Read LCR, then save in temp1 Set LCR to not BFh Read MCR, then save in temp2 Set MCR to (0100_0000b **OR** temp2) Set MCR to (0100_0100b **OR** temp2) Set MCR to (1011_1111b **AND** temp2) Set MCR to temp2 Set LCR to temp1
Disable Xoff Re-transmit	Read LCR, then save in temp1 Set LCR to not BFh Read MCR, then save in temp2 Set MCR to (0100_0000b **OR** temp2) Set MCR to (1011_1011b **AND** temp2) Set MCR to temp2 Set LCR to temp1

Set Prescaler Value to Divide-by-1 or 4	Read LCR, then save in temp1
	Set LCR to BFh
	Read EFR, then save in temp2
	Set EFR to (0001_0000b **OR** temp2)
	Set LCR to 00h
	Read MCR, then save in temp3
	
	if Divide-by-1 = OK then
	    Set MCR to (0111_1111b **AND** temp3)
	else
	    Set MCR to (1000_0000b **OR** temp3)
	
	Set LCR to BFh
	Set EFR to temp2
	Set LCR to temp1

Table 14−2: SB16C1058PCI Programming Guide

Command	Action
Initialize Process	
1. Set Baud Rate to 0001h	Read LCR, then save in temp Set LCR to 80h Set DLL to 01h Set DLM to 00h Set LCR to temp
2. Set TTR to 20h	Set LCR to BFh Set PSR to A5h Set TTR to 20h
3. Set RTR to 80h	Set RTR to 80h
4. Enable 256-byte FIFO	Set AFR to 01h
5. Set Line Control Register to 8-data but, no parity, 1 stop bit	Set PSR to A4h Set LCR to 03h
6. Enable TX, RX interrupts	Set IER to 03h
Serial Output Process 1. TX Interrupt is generated and Jumped to Interrupt Service Routine	
2. Read ISR	Read ISR, then save in temp1

3. Check TX Interrupt Status	If temp1 = xx00_0100b then
	Goto RX Interrupt Service Routine
	Else if temp1 = xx00_0010b then
	Goto TX Interrupt Service Routine
	Else
	Return from Interrupt Service Routine
	
	RX Interrupt Service Routine:
	…..
	TX Interrupt Service Routine:
	Read MCR, then save in temp2
	Set MCR to (temp2 OR 40h)

4. Read TX FIFO Count	Read TCR, then save in temp3
	Set MCR to temp2
	If temp1[7] = 1b then
	For (Cnt = 0; Cnt <= 127; Cnt++)

5. Read Data	Read TX_Data from TX_User_Buffer
6. Output TX	Set THR to TX_Data Else if temp3 > 128 then For (Cnt = 0; Cnt <= 127; Cnt++)
5. Read Data	Read TX_Data from TX_User_Buffer
6. Output TX	Set THR to TX_Data

Table 14−2: SB16C1058PCI Programming Guide…*continued*

Command	Action
	Else
5. Read Data	For (Cnt = 0; Cnt < temp3; Cnt++)
6. Output TX	Read TX_Data from TX_User_Buffer
	Set THR to TX_Data
	Return from Interrupt Service Routine
Serial Input Process	
1. RX Interrupt is generated and Jumped to Interrupt Service Routine	
2. Read ISR	Read ISR, then save in temp1
3. Check TX Interrupt Status	If temp1 = xx00_0100b then
	Goto RX Interrupt Service Routine
	Else if temp1 = xx00_0010b then
	Goto TX Interrupt Service Routine
	Else
	Return from Interrupt Service Routine
	TX Interrupt Service Routine:
	…..
	RX Interrupt Service Routine:
	Read MCR, then save in temp2
	Set MCR to (temp2 OR 40h)
4. Read RX FIFO Count	Read RCR, then save in temp3
	Set MCR to temp2
	If temp1[6] = 1b then
	For (Cnt = 0; Cnt <= 255; Cnt++)
5. Read RX Data	Read RBR, save in RX_User_Buffer
	Else
	For (Cnt = 0; Cnt < temp3; Cnt++)
5. Read Data	Read RBR, save in RX_User_Buffer
	Return from Interrupt Service Routine

15. Electrical Information

15.1 Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
V_{DD}	DC Supply Voltage	-0.5	7.0	V
V_{IN}	Input Voltage	-0.5	V_{DD}+0.5	V
V_{OUT}	Output Voltage Range	0	V_{DD}+0.5	V
T_{STG}	Storage Temperature	-40	150	℃
T_{OP}	Operational Temperature	-40	85	℃

Absolute maximum ratings are the values beyond which damage to the device may occur. Exposure to these conditions or beyond may adversely affect device reliability. Functional operation under absolute maximum ratings is not implied.

15.2 Power Consumption

Power Consumption	Minimum	Typical	Maximum	Unit
SB16C1058PCI-TQ	-	62.7	100	mW

When the clock source is 14.7456MHz and serial comm. speed is 921.6Kbps, the power consumption of SB16C1058PCI is 62.7mW (3.3V x 19mA). When the chip is on idle status, it's power consumption is 42.9mW (3.3V x 13mA).
If this chip is on maximum rate and best communication speed with 5.3Mbps, the maximum power consumption will be 500mW. If you use the chip in the normal serial communication environment, you can use the upper table values for your reference.

15.3 DC Characteristics

15.3.1 PCI PAD 3.3V 66MHz DC Signaling Characteristics

Symbol	Parameter	Condition	Min	Max	Unit
V_{DD}	Supply Voltage		3.0	3.6	V
V_{IH}	Input High Voltage		0.5V_{DD}	V_{DD}+0.5	V
V_{IL}	Input Low Voltage		-0.5	0.3V_{DD}	V
I_{IL}	Input Leakage Current	$0 < V_{IN} < V_{DD}$		+/-10	uA
V_{OH}	Output High Voltage	I_{out} = -500uA	0.9V_{DD}		V
V_{OL}	Output Low Voltage	I_{out} = 1500uA		0.1V_{DD}	V

15.3.1 PCI PAD 3.3V 66MHz DC Signaling Characteristics

Symbol	Parameter	0 ℃	100 ℃	Conditions	
		Min.	Max.	VDD	
V_{IL}	Low Level Input Voltage	-0.5V	$0.3V_{DD}$	2.7V~3.6V	Guaranteed Input Low Voltage
V_{IH}	High Level Input Voltage	$0.7V_{DD}$	V_{DD}+0.5V	2.7V~3.6V	Guaranteed Input High Voltage
V_{OL}	Low Level Output Voltage		V_{SS}+0.1V	2.7V	I_{OL}=0.8mA
V_{OH}	High Level Output Voltage	V_{DD}-0.1V		2.7V	I_{OH}=0.8mA
I_I	Input Current at Max Voltage		1mA	2.7V~3.6V	Input=5.5V

16. Timing Specification

16.1 PCI BUS Timing Specifications

Symbol	Parameter	66MHz		33MHz		Units
		Min	Max	Min	Max	
T_{val}	CLK to signal valid delay - bused signals	2	6	2	11	ns
T_{val}(ptp)	CLK to signal valid delay - point to point signals	2	6	2	12	ns
T_{on}	float to active delay	2		2		ns
T_{off}	active to float delay		14		28	ns
T_{su}	input setup time to CLK – bused signals	3		7		ns
T_{su}(ptp)	input setup time to CLK - point to point signals	5		10, 12		ns
T_h	Input hold time from CLK	0		0		ns
T_{rst}	Reset active time after power stable	1		1		ms
$T_{rst\text{-}clk}$	Reset active time after CLK stable	100		100		us
$T_{rst\text{-}off}$	Reset active to output float delay		40		40	ns
T_{rhfa}	RST# high to first configuration access	2		2		clocks
T_{rhff}	RST# high to first FRAME# assertion	5		5		clocks

Table 16–1: PCI Bus Timing Specifications

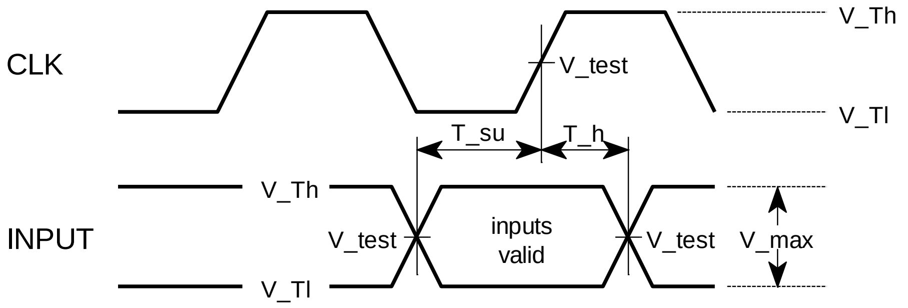


Figure 16−1: Input Timing Measurement Conditions

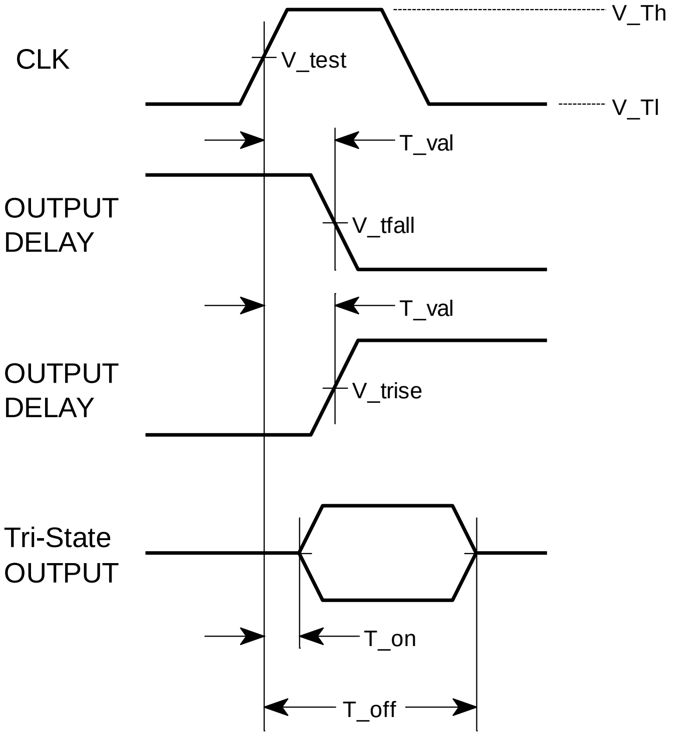


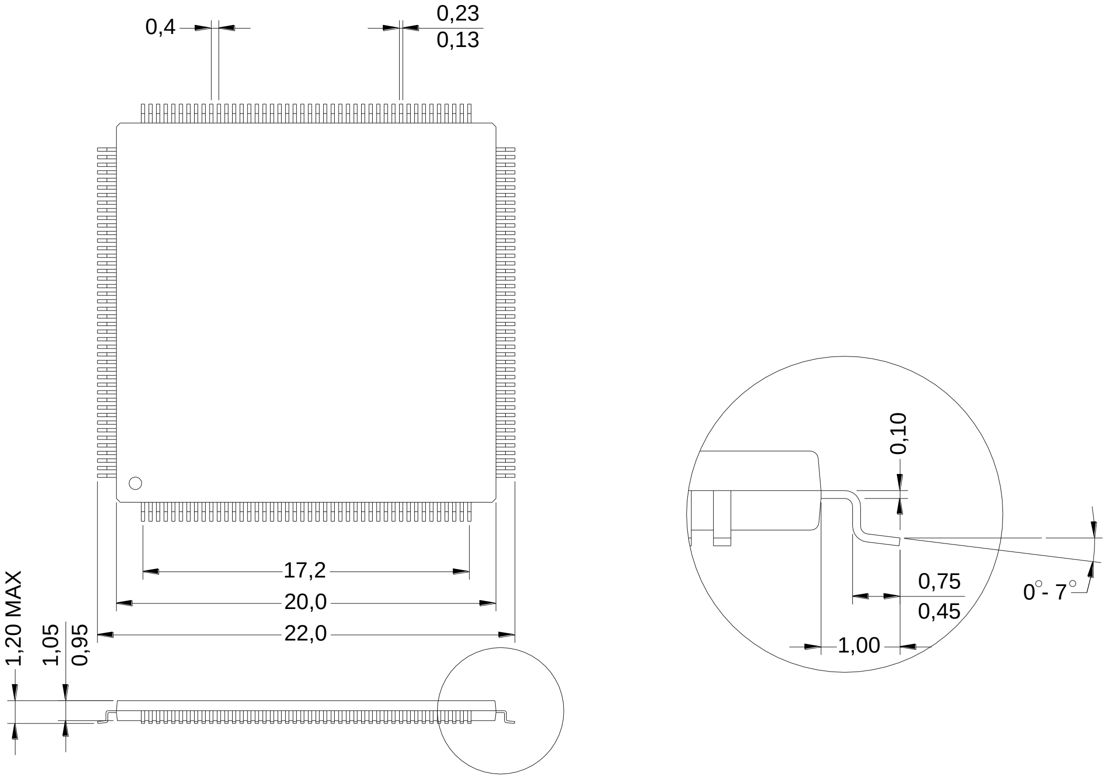
Figure 16−2: Output Timing Measurement Conditions

Symbol	3.3V Signaling	Units
V_{th}	$0.6V_{cc}$	V
V_{tl}	$0.2V_{cc}$	V
V_{test}	$0.4V_{cc}$	V
V_{trise}	$0.285V_{cc}$	V
V_{tfall}	$0.615V_{cc}$	V
V_{max}	$0.4V_{cc}$	V
Input Signal Slew Rate	1.5	V/ns

Table 16−2: PCI Bus Timing Measurement Condition Parameters

17. Package Outline

176-Pin LQFP: Low-profile Quad Flat Package; Body 20 x 20 x 1.4 mm



Note :

All dimensions are in millimeters.