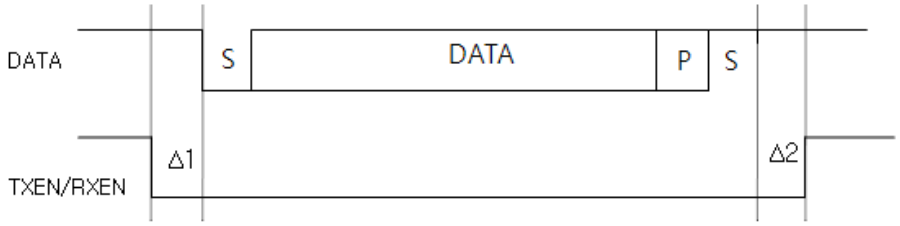
PCIe Endpoint Controller

SB16C1154PCIe

Revision 1.0 Preliminary

SystemBase Co., Ltd.

CONTENTS



1. Description

SB16C1154PCIe is a PCI Express Endpoint Controller with Quad-UART Interface. It offers easy PCI Express Endpoint Card Adapter implementation with a built-in PCIe PHY. SB16C1154PCIe provides high performance serial communication. With a built-in four SB16C1150 Cores that has 256-byte FIFO, SB16C1154PCIe decreases CPU load, is stronger at errors such as Overrun error and works well with simultaneous use of multiple ports. Because SB16C1154PCIe support PCI Express Burst Transfer up to 256-byte and it have built-in four enhanced 8-bit/ 16-bit/ 32-bit accessible UART core. Furthermore, it is capable of waking up PC that is powered off through external Wake-up Requests with PCI Express Power Management implemented. SB16C1154PCIe provides RS422/485 Auto Toggling function and Global Interrupt Function to the built-in UART allowing a more convenient handling of serial communication at a driver level. Finally, with SB16C1154PCIe, it is easy to design and fix the serial cards because it have TQFP 144 package.

2. Features

2.1 PCI Express Interface

- Compliant with PCI Express Base Specification, Revision 1.0a , 1.1 and 2.0
- PCI Express Single-Lane(x1), Compatible with x4, x8 and x16 PCI Express slots
- Built-In x1 PCI Express PHY
- Up to 256-byte PCI Express Burst Operation for serial dump
- One Virtual Channel
- INTx emulation & Message Signaled Interrupt(MSI)
- PCI Express Power Management,
  Supports PCI-PM Rev 1.2 - D0, $D3_{hot}$ and $D3_{cold}$
  Supports ASPM L0s, L1
- Download PCI Express Configuration Data from external serial EEPROM

2.2 Serial Interface

- Built-In four Enhanced UART Core, 16C1150 with 256-byte Tx/Rx FIFO
- Up to 3.9 Mbps Baud Rate (Up to 62.5 MHz Oscillator Input Clock)
- 256-byte Transmit FIFO
- 256-byte Receive FIFO with Error Flags
- Support 9-bit Serial Communication and Auto Address Detection
- Support Enhanced Auto Toggling
- 32-bit/ 16-bit/ 8-bit Accessible UART Core
- Support IrDA v1.0/v1.1 Encoding/Decoding
- Programmable and Selectable Transmit and Receive FIFO Trigger Levels for Interrupt Generation
- Software (Xon/Xoff) / Hardware (nRTS/nCTS) Flow Control
  - Programmable Xon/Xoff Characters
  - Programmable Auto-RTS and Auto-CTS

- Interrupt Poll Control
- 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
- Fast Data Bus Access Time
- Programmable Sleep Mode
- Programmable Serial Interface Characteristics
  - 5, 6, 7, or 8-bit Characters
  - Even, Odd, or No Parity Bit Generation and Detection
  - 1, 1.5, or 2 Stop Bit Generation
- False Start Bit Detection
- Line Break Generation and Detection
- Fully Prioritized Interrupt System Controls
- Modem Control Functions (nRTS, nCTS, nDTR, nDSR, nDCD, and nRI)

2.3 Miscellaneous

- 1.2V and 3.3V Operation
- 5V Tolerant Inputs
- Support 16 GPIO channels
- Industrial Temperature Operating, -40℃ ~ +85℃
- TQFP144 package
- Application
  - 4-port serial multiport PCIe card

2.4 Development Kit → Manufacturing Kit ??

SystemBase offers the SB16C1154PCIe Development Kit to minimize efforts and costs, and to maximize application stability.
SB16C1154PCIe Development Kit includes Hardware schematics, PCB CAD files, software device driver 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
SB16C1154PCIe-TQ	144-Pin TQFP	-40 ℃ to +85 ℃	Active

4. Block Diagram

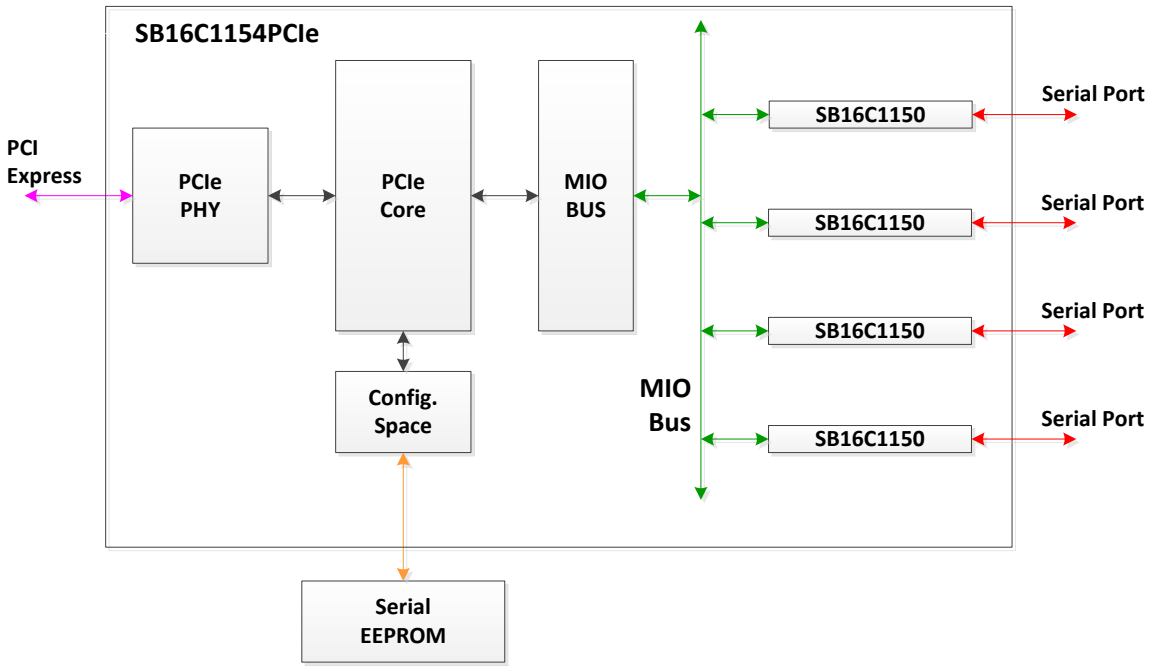


Figure 4–1: Block Diagram

5. Applications

5.1 Serial 4-port

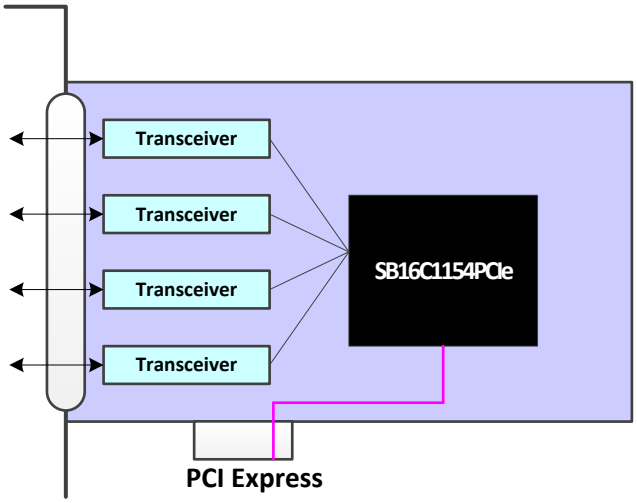


Figure 5–1: Serial 4-port Mode Application Block Diagram

4-port Serial card is generally made with SB16C1154PCIe (called by Serial 4-port Mode). Special logic is not needed to make 4-port serial Multi-Port since Quad-UART is built inside the SB16C1154PCIe. Depending on Serial Interface, Transceiver IC of the RS232, RS422 or RS485 needs to be attached for long distance transmission.

6. Pin Configuration

6.1 Pin Configuration for 144-Pin TQFP Package

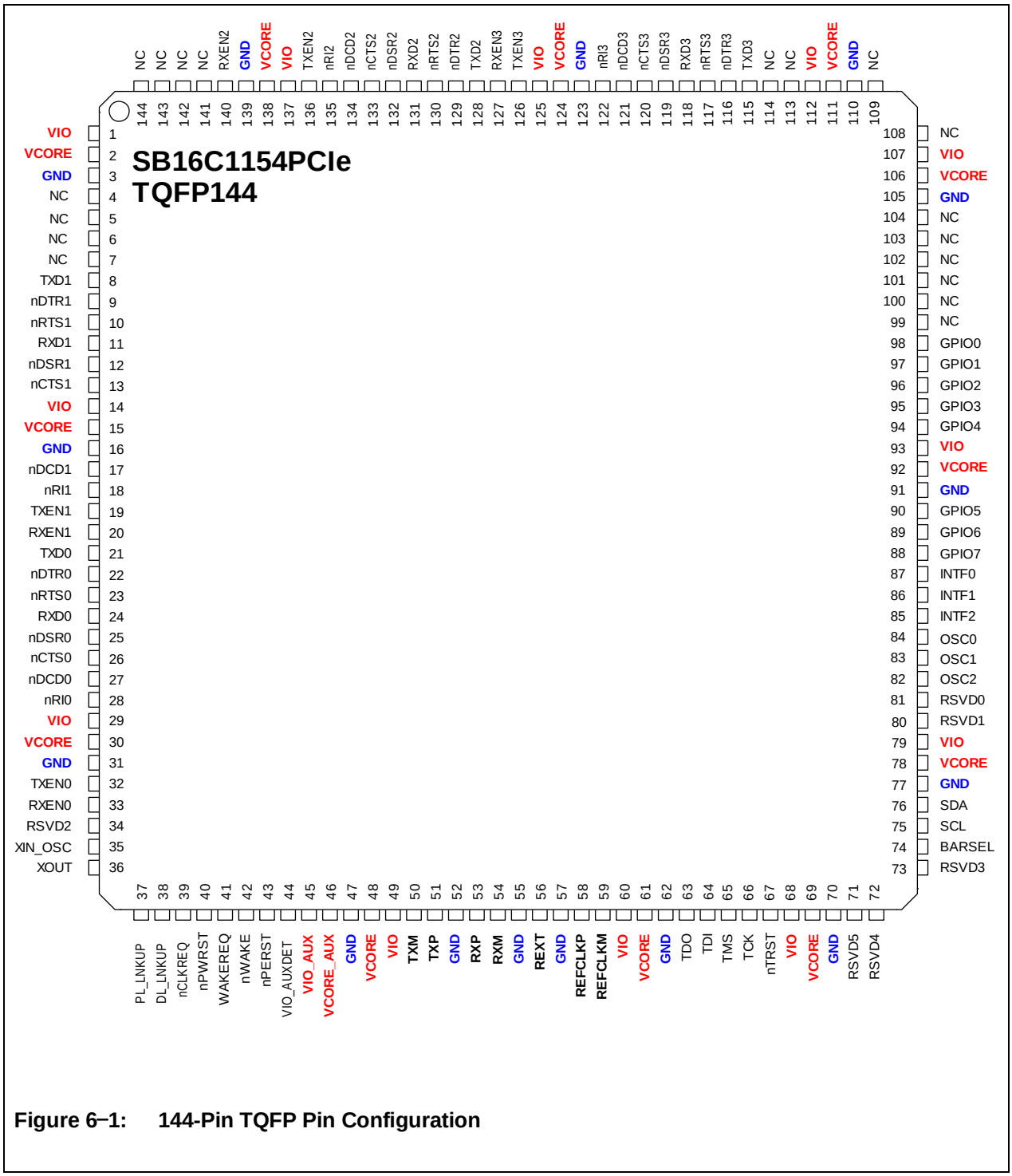


Figure 6–1: 144-Pin TQFP Pin Configuration

6.2 Pin Description

Table 6–1: Pin Description

PCI Express Bus Interface			
Name	**Pin**	**Type**	**Description**
TXP TXM	51 50	O O	**PCI Express Transmitter Differential Pair:** PCI Express differential pair output from PHY. All the transmitter should be AC-coupled to media.
RXP RXM	53 54	I I	**PCI Express Receiver Differential Pair:** PCI Express differential pair input to PHY with 50ohm on-chip termination.
REFCLKP REFCLKM	58 59	I I	**100MHz Reference Clock:** This is the spread spectrum source clock for PCI Express. Differential pair input with 50ohm on-chip termination and nominal frequency of 100MHz±300PPM.
nPERST	43	I	**PCI Express Fundamental Reset:** Reset the SB16C1154PCIe. The inputted signal indicates when the applied main power is within the specified tolerance and stable. This signal is asynchronous.
nWAKE	42	O	**Signal for Link Reactivation:** The open-drain output, active low signal that is driven to reactivate the PCI Express Link hierarchy's main power rails and reference clocks.
REXT	56	O	**Register for the reference current :** Connects 190Ω 1% resistor with GND.
WAKEREQ	41	I	**WAKE Request:** LTSSM of PCI Express Device goes from L2 state to L0 state with the Wake Up Event. This pin receives the event signal needed for the transition from L2 state to L0 state.
nCLKREQ	39	O	**PCI Express Connector Clock Request:** Clock Request Signal
PL_LNKUP	37	O	**Physical Layer Linkup Indicator:** This pin is used to indicate whether PCI Express physical layer linked up or not.
DL_CLKUP	38	O	**Data Link Layer Linkup Indicator:** This pin is used to indicate whether PCI Express physical layer linked up or not.
VIO_AUXDET	44	I	**Detection for Auxiliary VIO power:** Detect Auxiliary VIO power

Modem and Serial I/O Interface			
Name	**Pin**	**Type**	**Description**
TXD0 TXD1 TXD2 TXD3	21 8 128 115	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	24 11 131 118	I (PU) I (PU) I (PU) I (PU)	**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.
nRTS0 nRTS1 nRTS2 nRTS3	23 10 130 117	O O O O	**Request to Send (active low):** These pins indicate that the UART is ready to receive data from the modem, and affect transmit and receive operations only when Auto-RTS function is enabled.

Table 6–1: Pin Description…*continued*

Modem and Serial I/O Interface			
Name	**Pin**	**Type**	**Description**
nCTS0 nCTS1 nCTS2 nCTS3	26 13 133 120	I (PU) I (PU) I (PU) I (PU)	**Clear to Send (active low):** These pins indicate the modem is ready to receive transmitted data from the UART, and affect transmit and receive operations only when Auto-CTS function is enabled.
nDTR0 nDTR1 nDTR2 nDTR3	22 9 129 116	O O O O	**Data Terminal Ready (active low):** These pins indicate UART is ready for data exchange with modem after self initialization.
nDSR0 nDSR1 nDSR2 nDSR3	25 12 132 119	I (PU) I (PU) I (PU) I (PU)	**Data Set Ready (active low):** These pins indicate modem is powered-on and is ready for data exchange with UART.
nDCD0 nDCD1 nDCD2 nDCD3	27 17 134 121	I (PU) I (PU) I (PU) I (PU)	**Carrier Detect (active low):** These pins indicate that a carrier has been detected by modem.
nRI0 nRI1 nRI2 nRI3	28 18 135 122	I (PU) I (PU) I (PU) I (PU)	**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	32 19 136 126	O O O O	**TX Enable:** This pin is for Auto tri-state control of the RS422 or RS485 communication (Auto Toggling). 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	33 20 140 127	O O O O	**RX Enable:** This pin is for Auto tri-state control of the RS422 or RS485 communication (Auto Toggling). When serial date is transmitted to TXD, the value set on ATR[7] is transmitted. These pins eliminate additional glue logic outside.

Function Configuration Interfaces			
Name	**Pin**	**Type**	**Description**
INTF0 INTF1 INTF2	87 86 85	I I I	**Line Interface Type Select:** These pins are used to select the type of Line Transceiver interfaced in Serial 4-port Mode. The inputted value from these pins is shown in IIR0[6:4] of the Option register. INTF[2:0] = 0xxb : RS232 transceiver is selected. INTF[2:0] = 100b : RS422 transceiver with 1:1 mode is selected. INTF[2:0] = 101b : RS422 transceiver with Multi-Drop mode is selected. INTF[2:0] = 110b : RS485 transceiver with Non-Echo mode is selected. INTF[2:0] = 111b : RS485 transceiver with Echo mode is selected.

Function Configuration Interfaces			
Name	**Pin**	**Type**	**Description**
OSC0 OSC1 OSC2	84 83 82	I I I	**Oscillator Select:** These pins are used to select the type of Oscillator. The inputted value from these pins are shown in DIR0[3:0] of the Option register. OSC[2:0] = 000b : 1.8432MHz (up to 115.2Kbps) OSC[2:0] = 001b : 3.6864MHz (up to 230.4Kbps) OSC[2:0] = 010b : 7.3728MHz (up to 460.8Kbps) OSC[2:0] = 011b : 14.7456MHz (up to 921.6Kbps) OSC[2:0] = 100b : 29.4912MHz (up to 1.8432Mbps)
BARSEL	74	I	**Base Address Range Select:** When this pin is 0b, it allocates Base Address Ranges as a I/O space. So you can set the serial card to I/O mapped I/O. When this pin is 1b, it allocates Base Address Ranges as a Memory space. You can set the serial card to Memory mapped I/O (default)

Serial EEPROM Interfaces			
Name	**Pin**	**Type**	**Description**
SDA	76	I/O	**Serial EEPROM Data Input/Output:** Connected to SDA of serial EEPROM.
SCL	75	O	**Serial EEPROM Clock Output:** Connected to SCL of serial EEPROM.

GPIO Interfaces			
Name	**Pin**	**Type**	**Description**
GPIO0 GPIO1 GPIO2 GPIO3 GPIO4 GPIO5 GPIO6 GPIO7	98 97 96 95 94 90 89 88	I/O	**General Purpose Input and Output:** The directions of these pins are controlled by GPOCR of the Option Register. Inputted data is stored in GPIDR of the Option Register. Output data is determined by GPODR of the Option Register.

Other Interfaces			
Name	**Pin**	**Type**	**Description**
XIN_OSC	35	I	**Crystal or External Clock Input:** This input of up to 62.5MHz for data rate of 3.9Mbps at 1.2V. When OSC_SEL=1, SB16C1154PCIe use crystal as clock source. In this case, OSC_IN is not used and have to be wired to 0b.
XOUT	36	O	**Crystal or Buffed Clock Output:** This is output from resonant circuit of crystal.
nPWRST	40	I	**Power On Resets**

Reserved Pins for TEST			
Name	**Pin**	**Type**	**Description**
RSVD0	81	I	**Reserved inputs for test.**
RSVD1	80	I	Please make these reserved pins pull-down.
RSVD2	34	I	
RSVD3	73	I	**Reserved inputs for test mode selection.**
RSVD4	72	I	Please make these reserved pins pull-up with 10K resistor.
RSVD5	71	I	
TDO	63	O	**JTAG Interfaces for TEST.**
TDI	64	I	Please make output pins open.
TMS	65	I	Please make input pins pull-up.
TCK	66	I	
nTRST	67	O	

Power and Ground			
Name	**Pin**	**Type**	**Description**
VIO	1, 14, 29, 49, 60, 68, 79, 93, 107, 112, 125, 137	PWR	**Power Supply for Digital I/O:** Connect to +3.3V and to I/O Ground through 0.1uF capacitors.
VCORE	2, 15, 30, 48, 61, 69, 78, 92, 106, 111, 124, 138	PWR	**Power Supply for Digital Core:** Connect to +1.2V and to Core Ground through 0.1uF capacitors.
VIO_AUX	45	PWR	**Auxiliary Power Supply for Digital I/O:** Connect to +3.3V_aux and to I/O Ground through 0.1uF capacitors.
VCORE_AUX	46	PWR	**Auxiliary Power Supply for Digital Core:** Connect to +1.2V_aux and to Core Ground through 0.1uF capacitors.
GND	3, 16, 31, 47, 52, 55, 57, 62, 70, 77, 91, 105, 110, 123, 139	GND	**Ground for Digital Core:** Connect to ground.
NC	4, 5, 6, 7, 99, 100, 101, 102, 103, 104, 108, 109, 113, 114, 141, 142, 143, 144	-	**No Connect:**

7. Configuration Loader

SB16C1154PCIe can perform system initialization by reading PCI Express Configuration header data from Internal MIO registers or external serial EEPROM. It is decided to download configuration header data through exist of external serial ROM.
SB16C1154PCIe use a 256K I^2C serial EEPROM (24C256). We recommend customers use ATMEL AT24C256 or CATALYST CA24C256 for external serial ROM.
When SB16C1154PCIe is reset after power is granted, Configuration Loader inside SB16C1154PCIe loads Configuration header data and etc from internal MIO registers or external serial EEPROM depending on the existence of external EXT_LOAD pin. If external serial ROM exist, it reads saved data from external serial EEPROM and performs configuration. If external serial ROM do not exist, it reads data from internal MIO registers to perform the configuration.

7.1 Serial EEPROM Information Table

When the Auto Load Tag of the external serial EEPROM have hexadecimal 55 (55h), SB16C1154PCIe start to load configuration data from serial ROM. SB16C1154PCIe can recognize whether external serial EEPROM exist or not using the AUTO Load Tag. Even though the external serial EEPROM is attached, if the value of AUTO Load Tag is not 55h, SB16C1154PCIe think there is no external serial EEPROM on I^2C bus.
When the Auto Load Tag value is not 55h or there is no serial ROM, SB16C1154PCIe didn't load any data from serial ROM. But SB16C1154PCIe load the configuration data from MIO Register instead of external serial EEPROM.

Table 7–1: Serial EEPROM Information Table

Address	Description
00h	Auto Load TAG (55h)
01h	Vendor ID Low Byte
02h	Vendor ID High Byte
03h	Device ID Low Byte
04h	Device ID High Byte
05h	Revision ID
06h	Sub Vendor ID Low Byte
07h	Sub Vendor ID High Byte
08h	Sub System ID Low Byte
09h	Sub System ID High Byte
10h ~	Reserved

Vendor ID: It represents manufacturer of device. It is a unique ID given by PCI SIG. If you do not own Vendor ID, you can use 14A1h given to SystemBase by PCI SIG with permission.

Device ID: It is a unique ID of each device and is assigned at manufacturer's discretion.

Revision ID: It is a value representing device revision.

Sub Vendor ID: It 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.

Sub System ID: You can think of it as a Subsystem manufacturer's own Device ID.

8. PCIe Configuration Space

PCI Express Configuration offers two types of Configuration Space access method.

- PCI Compatible Configuration method
- PCI Express Enhanced 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.

Enhanced Configuration method offers increased Configuration Space to optimize access method. PCI Express Enhanced Configuration method uses Flat Memory-Mapped Address Space to access Configuration Register. In this case, memory address determines configuration register for access and memory data returns contents of the register that was addressed.

PCI Express provides 4096 bytes for Configuration Space per device function. This is a great increase compared to PCI Local Bus Specification, Revision 2.3 which provided 256 bytes only.

From 0 byte up to 256 bytes is called PCI Compatible Configuration Space and from 256 bytes to 4096 bytes is called Enhanced Configuration Space. This part has areas for configuring unique characteristics of PCI Express.

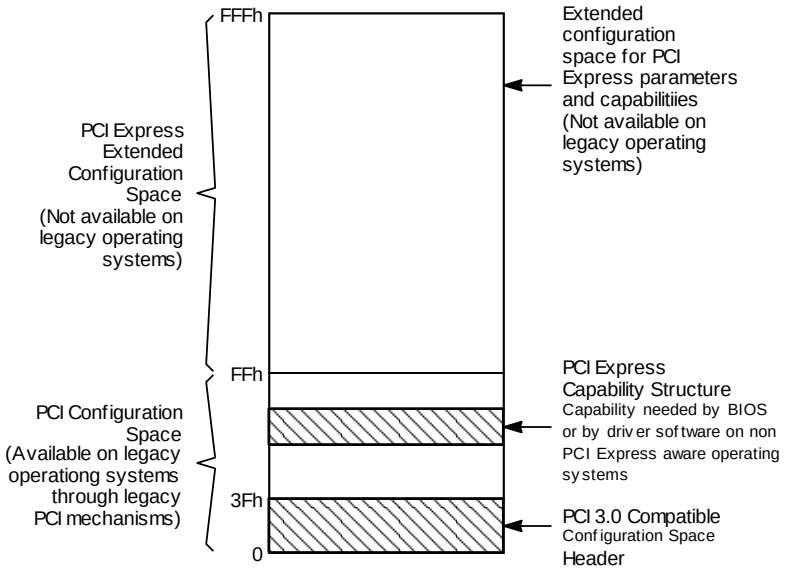


Figure 8–1: PCI Express Configuration Space block Diagram

8.1 Configuration Registers

8.1.1 Configuration Register Types

Field of Configuration Register is allocated to one of following described attributes

Table 8–1: Configuration Register Types

Register Attribute	Description
HwInit	**Hardware Initialized:** Register bits are initialized through firmware or hardware methods like pin strapping or serial EEPROM. (System firmware hardware initialization is only permitted to system integrated devices). Bits became read-only after they are initialized and Reset with Fundamental Reset is only possible. (Write is possible only by firmware)
RO	**Read Only:** Register bits are read-only and cannot be changed by software. Register bits are initialized by hardware methods like pin strapping or serial EEPROM.
RW	**Read-Write register:** Register bits are readable/writeable and can either be set to a desired state or be cleared by software.
RW1C	**Read-Only Status, Write-1-to-Clear Status register:** Register bits display status when read. The set bit indicating a status event can be cleared by writing a 1. Writing 0b to RW1C bit will not have any effect.
ROS	**Sticky-Read-Only register:** Register bits are read-only and cannot be changed by software. Registers cannot be initialized or modified by hot reset. Devices using Aux Power should preserve values of these sticky bits when using it. In these cases, the values cannot be initialized or modified by hot reset, warm reset or cold reset.
RWS	**Sticky-Read-Write register:** Register bits are readable/writeable and can either be set to a desired state or be cleared by software. Bits cannot be initialized or modified by hot reset. Devices using Aux Power should preserve values of these sticky bits when using it. These values cannot be initialized or modified by hot reset, warm reset or cold reset.
RW1CS	**Sticky-Read-Only Status, Write-1-to-Clear Status register:** Register bits display status when read. The set bit indicating a status event can be cleared by writing a 1. Writing 0b to RW1C bit will not have any effect. Devices using Aux Power should preserve values of these sticky bits when using it. These values cannot be initialized or modified by hot reset, warm reset or cold reset.
RsvdP	**Reserved and Preserved:** Secured for later RW implementation. Software needs to store values read in order to write on bits.
RsvdZ	**Reserved and Zero:** Secured space for later RW1C implementation afterwards. Software uses 0b to write on bits.

8.1.2 Configuration Space Map of SB16C1154PCIe

Table 8–2: Configuration Space Map

	Bit[31:24]	Bit[23:16]	Bit[15:8]	Bit[7:0]
Reg000	Device ID		Vendor ID	
Reg004	Status Register		Command Register	
Reg008	Class Code			Revision
Reg00C	BIST	Header Type	Latency Timer	Cache Line Size
Reg010	BAR0 (UART)			
Reg014	BAR1 (OPTION REGISTER)			
Reg018	BAR2 (Reserved)			
Reg01C	BAR3 (Reserved)			
Reg020	BAR4 (Reserved)			
Reg024	BAR5 (Reserved)			
Reg028	CardBus CIS Pointer			
Reg02C	Subsystem ID		Subsystem Vendor ID	
Reg030	Expansion ROM BAR			
Reg034	Reserved			Cap. Pointer
Reg038	Reserved			
Reg03C	Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line
(You can access the PCI Standard Capability Structures using capability pointer)				
	Power Management Capability			
+0x4	Power Management Control & Status			
	MSI Capability			
+0x4	MSI Low 32-bit Address			
+0x8	MSI High 32-bit Address			
+0xC	Reserved		MSI Data	
+0x10	Mask Bits Register			
+0x14	Pending Bits Register			
	PCIe Capabilities Register		Next Cap Pointer	PCIe Cap. ID
+0x4	PCIe Device Capabilities Register			
+0x8	Device Status		Device Control	
+0xC	PCIe Link Capabilities Register			
+0x10	Link Status		Link Control	
+0x24	Device Capabilities 2			
+0x28	Device Status 2		Device Control 2	
+0x2C	Link Capabilities 2			
+0x30	Link Status 2		Link Control 2	

	VPD Capability ID
+0x0	VPD Control and Capabilities Register
+0x4	VPD Data Register

Configuration Space of SB16C1154PCIe can be divided into 4 following functions.

- PCI Compatible Configuration Registers
- Power Management Registers
- MSI(Message Signaled Interrupt) Registers
- PCI Express Specific Configuration Registers

PCI Compatible Configuration Registers are from Reg000 to Reg03C and these parts are compatible with existing PCI Configuration Registers. Power Management Registers, MSI Registers and PCI Express Specific Configuration Registers could be accessible by capability pointer from PCI configuration Registers.

8.2 PCI Compatible Configuration Registers of SB16C1154PCIe

Table 8–3: Compatible Configuration Registers

	Bit[31:24]	**Bit[23:16]**	**Bit[15:8]**	**Bit[7:0]**
Reg000	Device ID		Vendor ID	
Reg004	Status Register		Command Register	
Reg008	Class Code			Revision
Reg00C	BIST	Header Type	Latency Timer	Cache Line Size
Reg010	BAR0 (UART)			
Reg014	BAR1 (OPTION REGISTER)			
Reg018	BAR2 (Reserved)			
Reg01C	BAR3 (Reserved)			
Reg020	BAR4 (Reserved)			
Reg024	BAR5 (Reserved)			
Reg028	CardBus CIS Pointer			
Reg02C	Subsystem ID		Subsystem Vendor ID	
Reg030	Expansion ROM BAR			
Reg034	Reserved			Cap. Pointer
Reg038	Reserved			
Reg03C	Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line

SB16C1154PCIe uses Configuration Register of Header Type0 which is used as Endpoint.
Following is a detailed description of PCI Compatible Configuration Register.

8.2.1 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. If you do not own a

Vendor ID, it is fine to use 14A1h given to SystemBase by PCI SIG. [HwInit]

8.2.2 Device ID

A 16-bit unique ID of each device given by the Function Manufacturer which can be assigned by the manufacturer freely.
It is related to software driver installation/recognition. [HwInit]

8.2.3 Command Register

Table 8–4: Command Register

Bit	Type	Description
15:11	RO	**Reserved:**
10	RW	**INTx Assertion Disable:** This bit controls PCI Express function's INTx interrupt message creation ability. When it is 0b, function can create INTx interrupt message. When it is 1b, function cannot create INTx interrupt message. Although this bit is set after the Assert_INTx emulation interrupt message is transmitted, Deassert_INTx interrupt message must be sent to finish the Assert_INTx that was sent earlier. Default value of this bit is 0b.
9	RO	**Fast Back-to-Back Enable:** It is not applied to PCI Express and 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	**IDSEL Stepping/Wait Cycle Control:** It is not applied to PCI Express and 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	**VGA Palette Snoop:** It is not applied to PCI Express and hardwired to 0b.
4	RO	**Memory Write and Invalidate:** It is not applied to PCI Express and hardwired to 0b.
3	RO	**Special Cycle Enable:** It is not applied to PCI Express and hardwired to 0b.
2	RW	**Bus Master Enable:** When it is 0b, it cannot operate as master. When it is 1b, it can start transaction as master. Enables the device to operate as master but is hardwired to 0b since this device does not create transaction.
1	RW	**Memory Space Enable:** When it 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 Space Enable:** When it 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.

8.2.4 Status Register

Table 8–5: Status Register

Bit	Type	Description
15	RW1C	**Detected Parity Error:** This bit is set when function receives poisoned TLP. It is not related with Command Register's Parity Error Enable bit status. Default value of this bit is 0b.
14	RW1C	**Signaled System Error:** This bit is set when Command Register's SERR Enable bit is set and function transmits ERR_FATAL or ERR_NONFATAL message. Default value of this bit is 0b.
13	RW1C	**Received Master Abort:** This bit is set when requester receives Completion of Unsupported Request Completion Status. Default value of this bit is 0b.
12	RW1C	**Received Target Abort:** This bit is set when requester receives Completion of Completer Abort Completion Status. Default value of this bit is 0b.
11	RW1C	**Signaled Target Abort:** This bit is set when a request is completed by Completer Abort Completion Status. Default value of this bit is 0b.
10:9	RO	**DEVSEL Timing.** It is not applied to PCI Express and hardwired to 0b.
8	RW1C	**Master Data Parity Error:** It is set when Parity Error Enable bit of Command Register is set and following two cases occur. - When requester receives Poisoned Completion. - When requester's write request is Poisoned. If Parity Error Enable bit is cleared, this bit is not set.
7	RO	**Fast Back-to-Back Capable:** It is not applied to PCI Express and hardwired to 0b.
6	RO	**Reserved.**
5	RO	**66MHz-Capable:** It is not applied to PCI Express and hardwired to 0b.
4	RO	**Capabilities List:** Shows that there is 1 or more extended capability register set. This bit should be set to 1b since SB16C1154PCIe possesses Extended Capability register range.
3	RO	**INTx Status:** Indicates that the function has an interrupt request that has not been processed yet. (Function is waiting to be serviced after sending an interrupt message. In other words, it is 1b when Assert_INTx is sent and Deassert_INTx is not.) This bit is only related to INTx message and has no meaning when using Message Signaled Interrupt.
2:0	RO	**Reserved.**

8.2.5 Revision

This register shows device revision. Manufacturer can assign it freely. It is also related to software device driver installation.

8.2.6 Class Code

This register contains descriptions on functions the device implements. It is divided as Base Class, Sub Class and Programming Interface in bytes. It must be set to the values provided by PCI Express Bus Specification.
SB16C1154PCIe gets 07_00_02h since it is 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).

8.2.7 Cache Line Size

This register assigns size of system's Cache Line. It is implemented as [RW] for compatibility with existing PCI and does not affect functioning of PCI Express.

8.2.8 Latency Timer

This register assigns latency clock related to bus master which does burst access. It is not applied to PCI Express and is hardwired to 0b.

8.2.9 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.

8.2.10 BIST(Built-In Self Test)

Table 8–5: BIST

Bit	Type	Description
7	RO	**BIST Capable:** Displays 1b if BIST is implemented and displays 0b if it is not implemented.
6	RO	**Start BIST:** Function's BIST starts when this bit is set to 1b and after BIST finishes, the bit is cleared to 0b.
5:4	RO	**Reserved:**
3:0	-	**Completion Code:** Display 0b for successful completion and other values for Function-specific errors.

8.2.11 Base Address Registers

These are spaces for assigning Base address for accessing I/O device or memory on PCI Express Local Bus. There are 2 spaces from Base Address Register 0 to 5 but spaces from Base Address Register 2 to 5 are set as unused reserved area.
In SB16C1154PCIe, Base Address Register 0 is used for UART Space and Base Address Register 1 is used for Option Registers Space. 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. You can select the memory space or I/O space of BAR0 and BAR1 using 'BARSEL' input pin.

8.2.11.1 Base Address Register0

SB16C1154PCIe operates Serial 4-port Mode. The size of memory or I/O address space is from 00h to 7Fh. (1 channel has 00h ~ 1Fh space.)
See *'9. SB16C1154PCIe Register Description'* for more details.

8.2.11.2 Base Address register1

Aside from UART area, SB16C1154PCIe contains Option Registers area which controls overall operations of the SB16C1154PCIe. SB16C1154PCIe sets this area with Base Address Register1(BAR1. Memory or I/O Address space size of the Option I/O Register is from 00 to 1Fh.
See *'9. SB16C1154PCIe Register Description'* for more details.

8.3 Power Management Registers of SB16C1154PCIe

Sometimes controlling over power is needed on PCI Express Bus applied systems. Especially in cases when system uses independent power source like mobile system or when PCI device occupies a lot of power but it is not used, the system must limit power supply to PCI device for making a power efficient system. For this reason, PCI specification provides Power Management Interface Specification making Power Management more convenient. This Power Management Capability Structures Registers could be accessed by capability pointer through linked list of capability register sets. SB16C1154PCIe supports *'PCI Power Management Interface Specification Revision 1.2.'* Content of registers that are implemented at Configuration Space Header is shown below. See *'PCI Power Management Interface Spec. Rev 1.2'* for more details.

Table 8–7: Power Management Register Block

	Power management Capabilities (PMC)		Next Item Ptr	Capability ID (0x01)
+0x4	Data	PMCSR_BSE Bridge Support Extensions	Power Management Control/Status Register (PMCSR)	

8.3.1 Capability ID (offset: 0x0)

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

8.3.2 Pointer to Next Capability (offset: 0x01)

A pointer storing address of register which has information about next Capability. You can find next Capability from other different Capability List by following this address. SB16C1154PCIe supports MSI Capability List and has the pointer to it which has the value 48h. See *'PCI Spec. Rev 3.0'* and *'PCI Express Base Spec. Rev 1.1'* for more details.

8.3.3 Power Management Capabilities (offset: 0x02)

Table 8–8: Power Management Capabilities

Bit	Type	Description
15:11	RO	**PME_Support:** Shows condition of PME Message transmission. SB16C1154PCIe can receive PME Message in $D3_{hot}$ and $D3_{cold}$ states and has value of 1_1001b. Bit11: When set, PME Messages can be generated from D0 Bit12: When set, PME Messages can be generated from D1 Bit13: When set, PME Messages can be generated from D2 Bit14: When set, PME Messages can be generated from D3hot Bit15: When set, PME Messages can be generated from D3cold
10	RO	**D2_Support:** Tells whether function supports D2 Power Management State. This device does not support D2 state and has value of 0b.
9	RO	**D1_Support:** Tells whether function supports D1 Power Management State. This device does not support D1 state and has value of 0b.
8:6	RO	**Aux Current:** Report VIO_AUX auxiliary current requirements for this function. Function of this device is configured to require 375mA which is maximum support capacity of an electric current supply and has value of 111b.
5	RO	**Device-Specific Initialization (DSI):** Shows need of DSI after transition from D3 to D0 uninitialized state. It should be set to 1b since initial value configuration for UART's communication is needed here.
4	RO	**Reserved:**
3	RO	**PME Clock:** It is not applied to PCI Express and hardwired to 0b.
2:0	RO	**Version:** Compatible with PCI PM Specification V1.2 and has value of "011".

8.3.4 Power Management Control/Status Register (offset: 0x04)

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

Table 8–9: Power Management Control/Status Register

Bit	Type	Description
15	RW1C	**PME_Status:** PME_Status is set to 1b after receiving PME Message and when PM S/W writes 1b, it is cleared.
14:13	RO	**Data Scale:** This device did not implement Data Scale and it is 00b.
12:9	RO	**Data Select:** This device did not implement Data Select and it is 0000b
8	RWS	**PME_EN:** If PME_En equals 1b, receiving PME Message becomes possible. If PME_En equals 0b, PME Message cannot be received. If receiving PME Message in $D3_{cold}$ state becomes possible, this value is unclear at OS booting time and it is Read/Written by PM S/W. (RWS) If receiving PME Message in $D3_{cold}$ state is not possible, this bit is 0b.
7:4	RO	**Reserved:** Always zero.
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. Its value is 0b.
2:0	RO	**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 state

8.3.5 PMCSR PCI-to-PCI Bridge Support Extensions (offset: 0x06)

PMCSR_BSE supports PCI bridge specific functions and is essential to all PCI-to-PCI Bridge. SB16C1154PCIe does not support it since it is an Endpoint. (00h)

8.3.6 Power Management Data Register (offset: 0x07)

Data register is an option and is a 8bit read only register. It shows power consumed and heat dissipation. SB16C1154PCIe does not support it and when read, 00h is returned.

8.4 Message Signaled Interrupt Registers of SB16C1154PCIe

Existing PCI Bus used control signals of Side Band for various kinds of required controls. Interrupt was also controlled with INTA, INTB, INTC, INTD signals of Side Band signal. However, all Side Band signals disappeared from PCI Express since it uses Serial Bus and the use of In Band Bus signals became necessary. Consequently, all interrupts are handled in Message Type.

Two ways of interrupt transmission of PCI Express Bus are shown below.

- Native PCI Express Interrupt Delivery – MSI
- Legacy PCI Interrupt Delivery – INTx Message

We provide MSI Capability to support Message Signaled Interrupt which is used by Native PCI Express Device. This MSI(Message Signaled Interrupt) Capability Structures Registers could be accessed by capability pointer through linked list of capability register sets. And this Capability Block is compatible with existing MSI of PCI 3.0 and provides the same functions. Following is a Map of MSI Capability List.

Table 8–10: Message Signaled Interrupt Register Block

	Message Control Register	Next Item Ptr	Capability ID
+0x4	MSI Low 32-bit Address		
+0x8	MSI High 32-bit Address		
+0xC	Reserved	MSI Data	
+0x10	Mask Bits Register		
+0x14	Pending Bits Register		

8.4.1 Capability ID (offset: 0x0)

Capability ID for MSI register and default value is 05h.

8.4.2 Pointer to Next Capability (offset: 0x01)

Pointer storing address of register which has information about next Capability. You can find next Capability from other different Capability List by following this address. SB16C1154PCIe supports PCI Express Capability List and has the pointer to it that has the value 60h. See PCI Express Base Spec. for more details.

8.4.3 Message Control Register (offset: 0x02)

MSI Low 32-bit Address & MSI High 32-bit Address
Since each one is 32-bit Address Register, maximum 64-bit Address can be supported. 64-bit Address is used when accessing spaces larger than 4GB.

Table 8–11: Message Control Register

Bit	Type	Description
15:8	RO	**Reserved:** Always zero.
8	RO	**PVM_EN:** MSI Per Vector Masking (PVM) supported
7	RO	**MSI 64 Enable:** 64-bit Address Capable
6:4	RW	**Multiple Message Enable:** Number of Multiple Message that will be used by this device.
3:1	RO	**Multiple Message Capable:** Number of Multiple Message that this device can use. only 1 is supported.
0	RO	**64-bit Address Capable:** It has value 1b since this device supports 64-bit Address.

8.4.4 MSI Data Register (offset: 0x08; if MSI 64 EN is 1b, offset: 0x0C)

System Software can set 16-bit Base Message Data Pattern which will be allocated to device.

8.4.5 MSI Mask Bits Register (offset: 0x0C; if MSI 64 EN is 1b, offset: 0x10)

For each Mask bit that is set, the function is prohibited from sending the associated message.

8.4.6 MSI Pending Bits Register (offset: 0x10; if MSI 64 EN is 1b, offset: 0x14)

For each Pending bit that is set, the function has a pending associated message.

8.5 PCI Express Specific Configuration Registers of SB16C1154PCIe

PCI Express Device must have PCI Express Specific Configuration Register space and these Capabilities also have Linked List structure. These registers are provided to control and manage unique characteristics of PCI Express.

Following table shows Map of PCI Express Capability Register Set. In this device, This PCI Express Specific Configuration Capability Structures Registers could be accessed by capability pointer through linked list of capability register sets. This device has essential Capabilities that Endpoint must have.

Table 8-12: PCI Express Specific Configuration Registers

	PCIe Capabilities Register	Next Item Ptr	Capability ID
+0x4	PCIe Device Capabilities Register		
+0x8	Device Status	Device Control	
+0xC	PCIe Link Capabilities Register		
+0x10	Link Status	Link Control	

8.5.1 Capability ID (offset: 0x0)

Capability ID for PCI Express Capability and the value is 10h. (default)

8.5.2 Pointer to Next Capability (offset: 0x01)

Pointer storing address of register which has information about next Capability. You can find next Capability from other different Capability List by following this address.

SB16C1154PCIe is device has 00h value since this Capability is the last for this device.

8.5.3 PCI Express Capabilities Register (offset: 0x02)

Table 8–13: PCI Express Capabilities Register

Bit	Type	Description
15:14	RO	**Reserved:** Always zero
13:9	RO	**Interrupt Message Number:** If this Function has one or more MSI Interrupt Message value, this register gets MSI Data value. Since this device gets only one MSI Interrupt Message value, it is 00000b.
8	HwInit	**Slot Implemented:** Tells whether Add-in Card slot is connected to Root Port or Switch downstream port. But since this device is an Endpoint, it is 0b.
7:4	RO	**Device/Port Type:** This device is Legacy PCI Express Endpoint that supports IO Transaction and so it is 0001b.
3:0	RO	**Capability Version:** PCI Express Capability Structure Version defined by PCI-SIG and it is 1h.

8.5.4 PCI Express Device Capabilities Register (offset: 0x04)

Table 8–14: PCI Express Device Capabilities Register

Bit	Type	Description
31:29	RO	**Reserved:** Always zero
28	RO	**Function Level Reset Capability**
27:26	RW	**Captured Slot Power Limit Scale:** Shows scale of power limit and has one of following values. 00b = 1.0x, 01b = 0.1x, 10b = 0.01x, 11b = 0.001x
25:18	RW	**Captured Slot Power Limit Value:** In combination with Slot Power Limit Scale value, it shows maximum power limit. Power Limit (W) = Slot Power Limit Value X Slot Power Limit Scale Value
17:16	RO	**Reserved:** All zero
15	RO	**Role-Based Error Reporting:** Required to be set for device compliant to 1.1 spec and later.
14	RO	**Power Indicator Present:** Not used in this device. 0b
13	RO	**Attention Indicator Present:** Not used in this device. 0b
12	RO	**Attention Button Present:** Not used in this device. 0b
11:9	RO	**Endpoint L1 Acceptable Latency:** Shows changing time that can be accepted as normal when this device changes from L1 state to L0 states regarding ASPM. This device has 100b which means it should change from L1 to L0 in 8us ~ 16us.
8:6	RO	**Endpoint L0s Acceptable Latency:** Shows changing time that can be accepted as normal when this device changes from L0s state to L0 states regarding ASPM. This device has 100b which means it should change from L0s to L0 in 512ns ~ 1us.
5	RO	**Extended Tag Field Supported:** Shows maximum Tag field size when function operates as Requester. 0b means 5-bit Tag Field Supported(Max. 32). 1b means 8-bit Tag Field Supported (Max. 256). This device does not support Extended Tag so its value is 0b.
4:3	RO	**Phantom Function Supported:** Each Function can send maximum 32 requests without receiving completion. However, some functions need larger value than this in some cases. If Extended Tag Field Enable Bit of Device Control Register is enabled, function can send maximum 256 requests. If function needs to send more than 256 requests, this Phantom Function can be used. This device does not support Phantom Function so its value is 00b.
2:0	RO	**Max Payload Size Supported:** Maximum payload size that function which support TLP can support. Since this device transmits serial data, it is set to support 512byte max payload size which is the minimum and has the value of 010b.

8.5.5 Device Control Register (offset: 0x08)

Table 8–15: Device Control Register

Bit	Type	Description
15	RO	**Initiate FLR:** Always zero
14:12	RW	**Max_Read_Request_Size:** Assign maximum read data size when device operates as requester. Default value is 010b. 000b = 128 byte max read request size 001b = 256 byte max read request size 010b = 512 byte max read request size 011b = 1K byte max read request size 100b = 2K byte max read request size 101b = 4K byte max read request size 110b = Reserved 111b = Reserved
11	RW	**Enable No Snoop:** If Software set this bit, memory slot for Requester Access is not cached by Processor.
10	RWS	**Auxiliary Power PM Enable:** When this bit is set, you can use Aux Power from PCI Express slot. Since this device support PCI Express Power Management, this bit can be configured.
9	RW	**Phantom Functions Enable:** This bit is used when you want to send more than 256 requests without completion. However, this device does not support it so it is fixed to 0b.
8	RW	**Extended Tag Field Enable:** When this bit is set, it is possible to use 8-bit Tag field. If it is 0b, 5-bit Tag field should be used. Since this device does not support Extended Tag Field, it is fixed to 0b.
7:5	RW	**Max Payload Size:** This field configures maximum TLP payload size for device. Device as a receiver should handle TLPx that has been configured and as a transmitter, sending TLP larger than configured value should be prohibited. Writing on this field is determined by Max_Payload_Size Supported of Device Capabilities register. Since this device is a controller sending one byte serial data, this device is configured to have 128 byte max payload size which is the minimum and the value is 000b.
4	RW	**Enable Relaxed Ordering:** When this bit is set to 1b, Relaxed Ordering Bit of Attributes field of TLP can be set. Default value is 1b. When Ordering Bit of Attributes field of TLP is 0b, it means using PCI Strongly Ordered Model and when it is 1b, it means using PCI-X Relaxed Ordered Model.
3	RW	**Unsupported Request Reporting Enable:** This bit decides whether to report Unsupported Request. Default value is 0b.
2	RW	**Fatal Error Reporting Enable:** This bit decides whether to report Fatal Error. Default value is 0b.
1	RW	**Non-Fatal Error Reporting Enable:** This bit decides whether to report Non-Fatal Error. Default value is 0b.
0	RW	**Correctable Error Reporting Enable:** This bit decides whether to report correctable error. Default value is 0b.

8.5.6 Device Status Register (offset: 0x0A)

Table 8–16: Device Status Register

Bit	Type	Description
15:6	RO	**Reserved:** Always zero
5	RO	**Transactions Pending:** When this bit is set, it means function could not complete non-post request packet. Function clears this bit when request is terminated by Completion Timeout mechanism or when non-post request is completed.
4	RO	**Aux Power Detected:** If Aux Power PM Enable Bit of Device Control Register is set and aux power is detected, this shows the state.
3	RW1C	**Unsupported Request Detected:** If Unsupported Request Reporting Enable Bit of Device Control Register is set and one or more unsupported request is detected, this shows the state. When it is set to 1 and software writes 1 afterwards, it is cleared.
2	RW1C	**Fatal Error Detected:** If Fatal Error Reporting Enable Bit of Device Control Register is set and one or more fatal error is detected, this shows the state. When it is set to 1 and software writes 1 afterwards, it is cleared.
1	RW1C	**Non-Fatal Error Detected:** If Non-Fatal Error Reporting Enable Bit of Device Control Register is set and one or more non-fatal error is detected, this shows the state. When it is set to 1 and software writes 1 afterwards, it is cleared.
0	RW1C	**Correctable Error Detected:** If Correctable Error Reporting Enable Bit of Device Control Register is set and one or more correctable error is detected, this shows the state. When it is set to 1 and software writes 1 afterwards, it is cleared.

8.5.7 Link Capabilities Register (offset: 0x0C)

Table 8–17: Link Capabilities Register

Bit	Type	Description
31:24	HwInit	**Port Number:** Shows number of ports related to the link and this device has 00h.
23	RO	**Reserved:** Always zero
22	RO	**ASPM Optionality Compliance**
21	RO	**Link Bandwidth Notification Capability**: Always zero
20	RO	**Data Link Layer Active Reporting Capability**: Always zero
19	RO	**Surprise Down Error Reporting Capable:** Not supported, hardwired to 0x0.
18	RO	**Clock Power Management**
17:15	RO	**L1 Exit Latency:** This field has value of 101b and shows that this device has latency between 16us to 32us when it is changing from L1 to L0.
14:12	RO	**L0s Exit Latency:** This field has value of 101b and shows that this device has latency between 1us to 2us when it is changing from L0s to L0.
11:10	RO	**Active State Power Management (ASPM) Support:** This field has value of 01b. This device supports only L0s state.
9:4	RO	**Maximum Link Width:** This field has value of 01h and supports x1 single lane link width.
3:0	RO	**Maximum Link Speed:** This field has value of 0001b and supports 2.5Gbps link speed. Indicates the supported maximum Link speeds of the associated Port. The encoding is the binary value of the bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 Register) that corresponds to the maximum Link speed.

8.5.8 Link Control Register (offset: 0x10)

Table 8–18: Link Control Register

Bit	Type	Description
15:12	RO	**Reserved:** Always zero
11	RW	**Link Autonomous Bandwidth Interrupt Enable:** When set, this bit enables the generation of an interrupt to indicate that the Link Autonomous Bandwidth Status bit has been set.
10	RW	**Link Bandwidth Management Interrupt Enable:** When set, this bit enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been set.
9	RO	**Hardware Autonomous Width Disable:** Not supported, hardwired to 0.
8	RW	**Enable Clock Power Management:** Hardwired to 0 if Clock Power Management is disabled in the Link Capabilities Register.
7	RW	**Extended Sync:** If this bit is set to 1b, - Send 4096 FTS Ordered Sets during L0s state. - Send one SKP Ordered Set before entering L0 state. - Send 1024 TS1 Ordered Sets before entering Recovery state from L1 state. It is used to improve bit lock and symbol lock. Default value of this bit is 0b.
6	RW	**Common Clock Configuration:** 1b when components on each end of link use same reference clock. 0b when components on each end of link use different reference clock
5	RW	**Retrain Link.** Reserved for Endpoint.
4	RW	**Link Disable.** Reserved for Endpoint.
3	RW	**Read Completion Boundary (RCB):** It is set by Configuration Software but it is hardwired to 0b since it is not used in this device.
2	RO	**Reserved:** Always zero
1:0	RW	**Active State Power Management (ASPM) Control:** Control ASPM level supported to link. 00b = Disabled. 01b = L0s Entry Enabled. 10b = L1 Entry Enabled. 11b = L0s and L1 Entry Enabled.

8.5.9 Link Status Register (offset: 0x12)

Table 8–19: Link Status Register

Bit	Type	Description
15	RO	**Link Autonomous Bandwidth Status:**
14	RO	**Link Bandwidth Management Status:**
13	RO	**Data Link Layer Active:**
12	HwInit	**Slot Clock Configuration:** This bit is set to 1b when components of each end of link use reference clock provided by platform and cleared to 0b if they use different reference clock for themselves.
11	RO	**Link Training:** It is 1b when Link Training is in process. It is cleared to 0b when Link Training is successfully completed.
10	RO	**Training Error:** It is set to 1b if Link Training Error occurs. It is cleared to 0b when hardware changes to L0 state normally after Link Training succeeds.
9:4	RO	**Negotiated Link Width:** Shows each end negotiated Link Width. Since this device supports x1 single lane, it is 000001b**.**
3:0	RO	**Link Speed:** Shows each end negotiated Link Speed. This device has value of 0001b and speed of 2.5Gbps.

8.5.10 Device Capabilities 2 Register (offset: 0x24)

Table 8–20: Device Capabilities 2 Register

Bit	Type	Description
31:20	RO	**Reserved:** Always zero
19:18	RO	**OBFF Supported:**
17:14	RO	**Reserved:**
13:12	RO	**TPH Completer Supported:**
11	RO	**LTR Mechanism Supported**:
10	HwInit	**No RO-enabled PR-PR Passing:**
9	RO	**128-bit CAS Completer Supported:**
8	RO	**64-bit AtomicOp Completer Supported:**
7	RO	**32-bit AtomicOp Completer Supported:**
6	RO	**AtomicOp Routing Supported. Not applicable for EP:**
5	RO	**ARI Forwarding Supported**
4	RO	**Completion Timeout Disable Supported:**
3:0	RO	**Completion Timeout Ranges Supported:** This field is applicable only to Root Ports, Endpoints that issue Requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of Requests issued on PCI Express.

8.5.11 Device Control 2 Register (offset: 0x28)

Table 8–21: Device Control 2 Register

Bit	Type	Description
31:15	RW	**Reserved:**
14:13	RW	**OBFF Enable:**
12:11	RW	**Reserved:**
10	RW	**LTR Mechanism Enable:**
9	RW	**IDO Completion Enable:**
8	RW	**IDO Request Enable:**
7	RW	**AtomicOp Egress Blocking:**
6	RW	**AtomicOp Requester Enable:**
5	RW	**ARI Forwarding Supported:**
4	RW	**Completion Timeout Disable:**
3:0	RW	**Completion Timeout Value:** 0000b: 50us to 50ms (default) 0001b: 50us to 100us 0010b: 1ms to 10ms 0101b: 16ms to 55ms 0110b: 65ms to 210ms 1001b: 260ms to 900ms 1010b: 1s to 3.5s 1101b: 4s to 13s 1110b: 17s to 64s

8.5.12 Link Capabilities 2 Register (offset: 0x2C)

Table 8–22: Link Capabilities 2 Register

Bit	Type	Description
31:9	RO	**Reserved:**
8	RO	**Crosslink Supported:**
7:1	RO	**Supported Link Speeds Vector:** Indicates the supported Link speeds of the associated Port. For each bit, a value of 1b indicates that the corresponding Link speed is supported. Otherwise, the Link speed is not supported. Bit definitions are: Bit 1 2.5Gbps Bit 2 5.0Gbps Bit 3 8.0Gbps Bits 7:4 reserved
0	RO	**Reserved:**

8.5.13 Link Control 2 Register (offset: 0x30)

Table 8–23: Link Control 2 Register

Bit	Type	Description
15:13	RO	**Reserved:** Always zero
12	HwInit	**Slot Clock Configuration:** This bit is set to 1b when components of each end of link use reference clock provided by platform and cleared to 0b if they use different reference clock for themselves.
11	RO	**Link Training:** It is 1b when Link Training is in process. It is cleared to 0b when Link Training is successfully completed.
10	RO	**Training Error:** It is set to 1b if Link Training Error occurs. It is cleared to 0b when hardware changes to L0 state normally after Link Training succeeds.
9:4	RO	**Negotiated Link Width:** Shows each end negotiated Link Width. Since this device supports x1 single lane, it is 000001b**.**
3:0	RW	**Target Link Speed:** For Downstream ports

8.5.14 Link Status 2 Register (offset: 0x32)

Table 8–24: Link Status 2 Register

Bit	Type	Description
15:13	RO	**Reserved:** Always zero
12	HwInit	**Slot Clock Configuration:** This bit is set to 1b when components of each end of link use reference clock provided by platform and cleared to 0b if they use different reference clock for themselves.
11	RO	**Link Training:** It is 1b when Link Training is in process. It is cleared to 0b when Link Training is successfully completed.
10	RO	**Training Error:** It is set to 1b if Link Training Error occurs. It is cleared to 0b when hardware changes to L0 state normally after Link Training succeeds.
9:4	RO	**Negotiated Link Width:** Shows each end negotiated Link Width. Since this device supports x1 single lane, it is 000001b**.**
3:0	RO	**Link Speed:** Shows each end negotiated Link Speed. This device has value of 0001b and speed of 2.5Gbps.

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 Laptop to Server. 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 Express Power Management

9.1.1 PCIe Function Power State

Total 4 power states are defined for PCIe 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 PCIe device. D3 is classified into two states depending on existence or absence of Vcc(VIO & VCORE). Those states are $D3_{hot}$ and $D3_{cold}$.

$D3_{hot}$ is the state where Vcc(VIO & VCORE) 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 (VIO_AUX & VCORE_AUX) power.

At Power Off state, device's main power and Vaux are cut off and execution of Wakeup protocol is not possible. It is L3 state of LTSSM. 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 L2 state of LTSSM and Wakeup protocol can be executed. To get out of this state, assign nWAKE signal to Root Complex. The system waken up by this can change to D0 state by re-assigning Vcc to this device and assigning nPERST.

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 PCIe function must support D0, $D3_{hot}$ and $D3_{cold}$. SB16C1154PCIe also support D0, $D3_{hot}$ and $D3_{cold}$ and do not support D1 and D2.

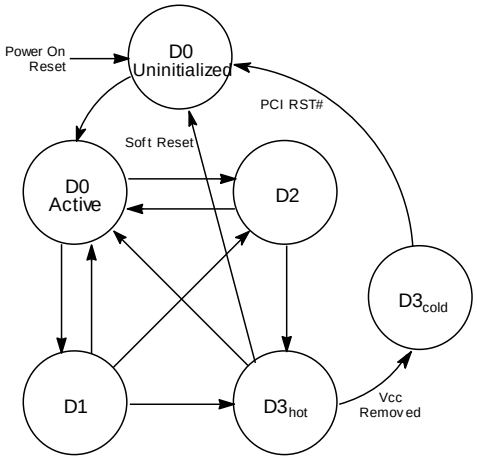


Figure 9–1: PCI Express Function Power Management State Transition

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.

9.1.2 Sticky Bits (PME Context)

When main power is cut off and Vaux is supplied (sleep state), contents of some bits of Configuration Register remain. These bits are called “Sticky Bits” and they are designed as ROS, RWS, and RW1CS in PCIe setting. These sticky bits must remain their state even though main power cuts off or warm reset or cold reset is executed.

Values of Sticky Bits must be preserved at nPERST signal at time of restoration from $D3_{cold}$ to D0.

Following values must be maintained as long as power is supplied.

- pm_pme_status_o (RW1CS)
- pm_pme_enable_o (RWS)
- aux_power_pm_enable (RWS)
- Requester ID

9.1.3 Wakeup Protocol

Some devices have hardware wake-up function which wakes up computer that was in sleep state when external event occurs. These devices include power button, modem or network card. In sleep state, external event happen when modem receives a call or network card receive a packet. It is usually called as WOL(Wake On LAN) or WOM(Wake On MoDem).

Mechanism for Wake Event assigns nWAKE signal line by following Wakeup protocol to wake up system in sleep state. Woken up system assigns main power to all devices and executes Link Training and Flow Control initialization. After successfully executing Wakeup protocol, PM_PME Message is sent from Endpoint to Root Complex.

For a function to execute Wakeup protocol, PME software must set PME_En bit of PMCSR to 1b.

9.1.4 PME Service

PCIe Core that is in power-saving state($D3_{hot}$ & $D3_{cold}$) when Wakeup Event occurred, creates and sends PME Message. PM_PME Message Format is as follows.

Table 9–1: PM_PME Message Format

Byte0	Byte1	Byte2	Byte3	
30h (Fmt, Type)	00h	00h	00h	Double Word 0
00h	00h	00h	18h (Msg Code)	Double Word 1
00h	00h	00h	00h	Double Word 2
00h	00h	00h	00h	Double Word 3

PM_PME Message can be generated in following states.

- Device is in $D3_{hot}$ state and Wakeup Event is generated,
  When SB16C1154PCIe is in $D3_{hot}$ state, main power and reference clock are supplied but it does no operation. UARTs inside SB16C1154PCIe enter sleep mode since it must be in maximum power-saving mode. In this case, SB16C1154PCIe does not separately organize logic for power-saving like memory inside Core for quick restoration. In this case, it enters $D3_{hot}$ state after S/W device driver sets all UARTs to enter sleep mode. Link State of device is in L1 state when it is in $D3_{hot}$ state. Interrupt generation on UART which is in Sleep Mode by receiving Serial Data for Wakeup Event makes Link State to change from L1 to L0 and sends PM_PME Message afterwards. two wake-up events are offered when device is in $D3_{hot}$ state. One is input signal to WAKEREQ input pin and the other is interrupt signal generated at UART. This is selected by PMRR (Power Management Resource Register) of the Option Registers.

- Device is in $D3_{cold}$ state and Wakeup Event is generated,
  When SB16C1154PCIe is in $D3_{cold}$ state, it uses Gated Clock to stop assigning of clock of UART's clock input. (Fixing Gated Clock to 0b) And it assigns Reset signal of UART to stop all situations. Then it stops functioning of Completion Memory which is a memory in Transaction Layer, VC Memory and Retry Memory. For power-saving purpose, use Gated Clock for clock assigned to memories.
  Since Reset is assigned to UART, all internal configurations of UART are uninitialized and thus re-initialization is required at time of restoration to D0.
  When it is in $D3_{cold}$ state and Wakeup Event occurs, it asserts nWAKE signal and wakes up Root Complex. Main power and Reference Clock are re-supplied to the woken up Root Complex. It asserts with nPERST and initializes all devices. When it becomes L0 state and it is able to communicate, it sends PM_PME Message.
  Wakeup Event could be Ring Indicator signal through modem.

9.1.5 PME Service Timeout Protocol

Reason for PME Service Timeout protocol is to prevent deadlock. Root Complex can buffer limited amount of message PM_PME requester transaction packet. If excessive packets arrive at upstream, switch must buffer the packets after Root Complex are fully filled. When the queue is full with PME Messages sent by the first PME Requester, PME Messages by other PME Requesters are waiting in Switch's Queue. To service the first PME Message, Root Complex sends Configuration Read Request to read PME_Status of the first PME Requester's PMCSR. Completion for this request which is waiting in switch queue cannot be delivered to Root Complex due to previous PME Messages.
By the transaction ordering rule, completer transaction is places back of buffered message PM_PME requester transactions. Root Complex therefore, goes to deadlock since it cannot complete the current PME Service.

When Message PM_PME requester transaction packet is sent, PME_Status bit is set to 1b and PME service timer is cleared to 0b and the timer is started. As a reply to handling of message PM_PME requester transaction packet which is received by Root Complex, PM software clears PME_Status bit to 0b. If PME_Status bit is not cleared to 0b in 95㎳ ~ 150㎳ time by PME Service Timer, function must retransmit message PM_PME requester transaction packet and clear PME Service Timer to 0b and restart the timer. This procedure must be repeated until PME_Status bit is cleared to 0b by PM software before PME Service Timer expires.

9.2 SB16C1154PCIe Power Management Pins and Functions

9.2.1 SB16C1154PCIe Pins for Power Management

Table 9–2 SB16C1154PCIe Pin Table for PM

Pin Name	Type	Description
WAKEREQ	I	It is a Input of WAKEREQ(Wake Request). For example, it receives Wakeup Event generated by Ring Indicator.
nWAKE	O	It is a Side band signal that Wakes Root Complex up to restore main power and reference clock that have been removed to implement Wakeup Protocol.
VIO VCORE	PWR	Vcc Power pin to catch the status of main power that is supplied to slot. $D3_{hot}$ state and $D3_{cold}$ state can be classified with this.
VIO_AUX VCORE_AUX	PWR	Vaux Power pin to catch the status of Vaux power that is supplied to slot. Status of $D3_{cold}$ can be understood with this and it reflects Aux Power Detected part of PCIe Device Capability Register

9.2.2 SB16C1154PCIe Power Management Wakeup implementation

Above 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 SB16C1154PCIe and handled as Wakeup Event. And if it is in $D3_{cold}$ state, this signal is asserted as nWAKE side band signal.

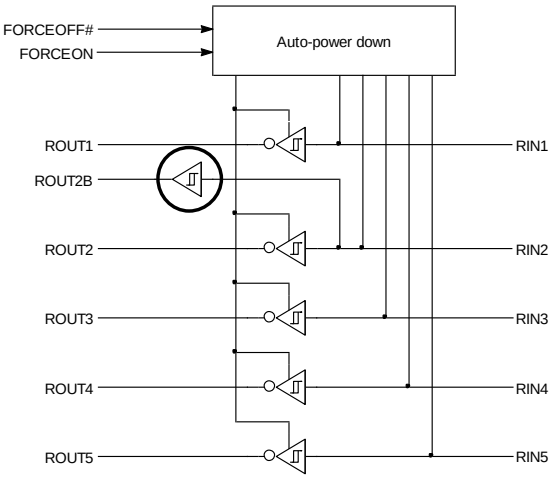
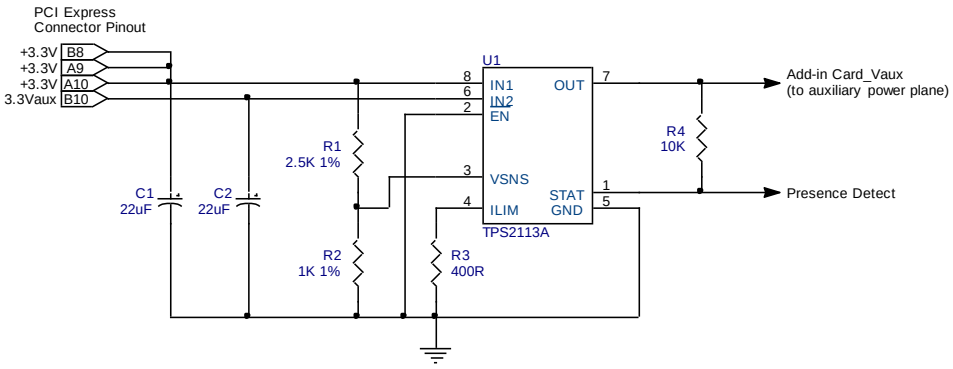


Figure 9–2: Logic Diagram of MAX3243

9.2.3 VIO_AUX Presence Detection & Power Routing

PCI Express Add-In Card that implements a function which can generate Power Management Event from D3cold must decide existence of 3.3V on Pin B10(VIO_AUX) of PCI Express Bus. When weak pull-down attached to Pin B10 is implemented on system board that does not support supply of VIO_AUX, it is there to make logic low reference and it must be implemented in all Add-In Card. On systems that do not supply VIO_AUX through Pin B10, PCIe 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 VIO_AUX 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 Space

It is determined by Base Address Register 0 (10h ~ 13h) from PCIe Configuration Space. This is BAR0 area of Configuration Space and this area is for accessing actual physical UARTs.

This space could be set to 'I/O mapped I/O space' or 'memory mapped I/O space' by BARSEL input pin. When BARSEL is '0', this UART space will be set to 'I/O mapped I/O space'. When BARSEL is '1', this UART space will be set to 'memory mapped I/O space'.

10.1 UART I/O Address Map

Conventional UART core has 8byte range for each channel. As UART has many functions, the internal registers became complicated structure with pages. So s/w engineer have to move the page of internal register for the setting of various functions. This may become overhead of program and has bad performance. But SB16C1150 core make flatten these pages of internal registers to just one page. So it has 32 bytes range for each channel. It enable to access internal registers easily and set various functions quickly. So the performance is better.

Because SB16C1150 have the flattened register structure, it has 32 bytes (00h ~ 1Fh) per port. It also can be accessed by 8 byte addressing as 16C550 compatible device.

I/O or Memory area of BAR0 increased with number of port. In case of SB16C1154PCIe, the port numbers are 4, I/O or Memory area is 128 bytes (32 bytes * 4 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.

Table 10–1: UART I/O Address Map

I/O Address	8-serial Mode
00h ~ 1Fh	UART0
20h ~ 3Fh	UART1
40h ~ 5Fh	UART2
60h ~ 7Fh	UART3

11. Option I/O Space

It is determined by Base Address Register 1 (14h ~ 17h) from PCIe Configuration Space.
Contents of 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

I/O Address	Register Name	I/O
00h	GIR0 (General Information Register – Port Number)	Read Only
01h	GIR1 (General Information Register – Product Version)	Read Only
02h	GIR2 (General Information Register – Operation Mode)	Read Only
03h	GIR3 (General Information Register – PCI Express Core Version)	Read Only
03h	SRR (Software Reset Register)	Write Only
04h	DIR0(Port1 ~ Port8, Device Information Register)	Read Only
05h	DIR1(Port9 ~ Port16, Device Information Register), Not Used	Read Only
06h	DIR2(Port17 ~ Port24, Device Information Register) , Not Used	Read Only
07h	DIR3(Port25 ~ Port32, Device Information Register) , Not Used	Read Only
08h	IIR0(Port1 ~ Port8, Interface Information Register)	Read/Write
09h	IIR1(Port9 ~ Port16, Interface Information Register) , Not Used	Read/Write
0Ah	IIR2(Port17 ~ Port24, Interface Information Register) , Not Used	Read/Write
0Bh	IIR3(Port25 ~ Port32, Interface Information Register) , Not Used	Read/Write
0Ch	IMR0(Port1 ~ Port8, Interrupt Mask Register)	Read/Write
0Dh	IMR1(Port9 ~ Port16, Interrupt Mask Register) , Not Used	Read/Write
0Eh	IMR2(Port17 ~ Port24, Interrupt Mask Register) , Not Used	Read/Write
0Fh	IMR3(Port25 ~ Port32, Interrupt Mask Register) , Not Used	Read/Write
10h	IPR0(Port1 ~ Port8, Interrupt Poll Register)	Read Only
11h	IPR1(Port9 ~ Port16, Interrupt Poll Register) , Not Used	Read Only
12h	IPR2(Port17 ~ Port24, Interrupt Poll Register) , Not Used	Read Only
13h	IPR3(Port25 ~ Port32, Interrupt Poll Register) , Not Used	Read Only
14 ~ 15h	Reserved	-
16h	APPSR (Auto Toggle Pin Select Register)	Read/Write
17h	Reserved	-
18h	PPMRR(PM_PME Message Resource Register in D3hot)	Read/Write
19 ~ 1Fh	Reserved	-
20h	GPOCR (General Purpose Output Control Register, GPIO0~GPIO7)	Read/Write
21h	GPODR (General Purpose Output Data Register, GPIO0~GPIO7)	Read/Write
22h	GPIDR (General Purpose Input Data Register, GPIO0~GPIO7)	Read Only
23h	Reserved	-
24h	GPOCR2 (General Purpose Output Control Register, GPIO8~GPIO15)	Read/Write
25h	GPODR2 (General Purpose Output Data Register, GPIO8~GPIO15)	Read/Write
26h	GPIDR2 (General Purpose Input Data Register, GPIO8~GPIO15)	Read Only
27h ~	Reserved	-

11.1 General Information Register0 – Port Number (GIR0)

Port Number: Shows how many ports are installed on current Serial Multi-Port.
In this case, Serial 4-port Mode has 4 ports. It has '04h'.

11.2 General Information Register1 – Product Version (GIR1)

Serial shows the version of Multi-Port Controller. (Currently 10h meaning version 1.0)
Higher 4 bits represent Major version and lower 4 bits represents Minor version.

11.3 General Information Register2 – Operation Mode (GIR2)

It means the number of operational parallel port in SB16C1154PCIe. So it has '00h'

11.4 General Information Register3 – Core Version (GIR3)

Shows PCI Express Target Interface Core version (Currently 2.4). It has '24h'.

11.5 Software Reset Register

If 53h("R") is written on SRR, Reset is outputted to Serial Multi-Port I/O Bus and this means PCIe UART goes to Reset state. If values other than 53h are written on SRR, Reset state is cleared.

11.6 Device Information Register0 ~ 3 (DIR0 ~ 3)

DIR0 : Device information of Port1 ~ Port8 (in this case, Port1 ~ Port4)
DIR0 have settings of UART controller and Oscillator Information in SB16C1154PCIe.
DIR1~DIR3 is not used in SB16C1154PCIe Serial 4-port Mode.

Table 11–3: Device Information Register 0 Description

Bit	Symbol	Description
7:4	DIR0[7:4]	**UART Select**: Content of U[3:0] shows type of UART. 000b: 16C550 (Not Supported) 001b: 16C1050 (Not Supported) 010b: 16C1050A (Not Supported) 011b: 16C1150 (Supported. 9-bit & enhanced auto toggle) 100b ~ 111b: Reserved
3:0	DIR0[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 (1.8432Mbps) 0101b ~ 1111b: Reserved

11.7 Interface Information Register0 ~ 3 (IIR0 ~ 3)

IIR0: indicates interface information of Port1 ~ Port8 (Actually Port1 ~ Port4)
IIR1: indicates interface information of Port9 ~ Port16 (No Used)
IIR2: indicates interface information of Port17 ~ Port24 (No Used)
IIR3: indicates interface information of Port25 ~ Port32 (No Used)

Table 11–4: Interface Information Register 0 Description

Bit	Symbol	Description
7:6	IIR0[7:6]	**Reserved:** Always zero
5:4	IIR0[5:4]	**Interface Type Indicator:** 0xxb: RS232 interface is selected on Serial 4-port mode by INTF0[20]=0b. 100b: RS422 Point-To-Point mode interface is selected on Serial 4-port mode. 101b: RS422 Multi-Drop mode interface is selected on Serial 4-port mode. 110b: RS485 Non-Echo mode interface is selected on Serial 4-port mode. 111b: RS485 Echo mode interface is selected on Serial 4-port mode.

11.8 Interrupt Mask Register0 ~ 3 (IMR0 ~ 3)

IMR0: Enables or disables interrupt of Port1 ~ Port8 in Serial 4-port Mode.
IMR1: Enables or disables interrupt of Port9 ~ Port16. Not Used.
IMR2: Enables or disables interrupt of Port17 ~ Port24. Not Used.
IMR3: Enables or disables interrupt of Port25 ~ Port32. Not Used.

Table 11–6: Interrupt Mask Register0 Description

Bit	Symbol	Description
7	IMR0[7]	Not used.
6	IMR0[6]	Not used.
5	IMR0[5]	Not used.
4	IMR0[4]	Not used.
3	IMR0[3]	0b: Disables Port4 interrupt on Serial 4-port Mode. 1b: Enables Port4 interrupt on Serial 4-port Mode.
2	IMR0[2]	0b: Disables Port3 interrupt on Serial 4-port Mode. 1b: Enables Port3 interrupt on Serial 4-port Mode.
1	IMR0[1]	0b: Disables Port2 interrupt on Serial 4-port Mode. 1b: Enables Port2 interrupt on Serial 4-port Mode.
0	IMR0[0]	0b: Disables Port1 interrupt on Serial 4-port Mode. 1b: Enables Port1 interrupt on Serial 4-port Mode.

11.9 Interrupt Poll Register0 ~ 3 (IPR0 ~ 3)

IPR0: Indicates interrupt generation state of Port 1 ~ Port 8. (Actually Port1 ~ Port4)
IPR1: Indicates interrupt generation state of Port 9 ~ Port 16. Not Used.
IPR2: Indicates interrupt generation state of Port 17 ~ Port 24. Not Used.
IPR3: Indicates interrupt generation state of Port 25 ~ Port 32. Not Used.

Table 11-10: Interrupt Poll Register0 Description

Bit	Symbol	Description
7	IPR0[7]	Not used.
6	IPR0[6]	Not used.
5	IPR0[5]	Not used.
4	IPR0[4]	Not used.
3	IPR0[3]	0b: Port4 interrupt has occurred in Serial 4-port Mode. 1b: Port4 interrupt has not occurred in Serial 4-port Mode.
2	IPR0[2]	0b: Port3 interrupt has occurred in Serial 4-port Mode. 1b: Port3 interrupt has not occurred in Serial 4-port Mode.
1	IPR0[1]	0b: Port2 interrupt has occurred in Serial 4-port Mode. 1b: Port2 interrupt has not occurred in Serial 4-port Mode.
0	IPR0[0]	0b: Port1 interrupt has occurred in Serial 4-port Mode. 1b: Port1 interrupt has not occurred in Serial 4-port Mode.

11.10 Auto Toggle Pin Select Register (ATPSR)

Select a pin among dedicated auto toggle pin, nRTS and nDTR as the auto toggle pin in RS422 and RS485 mode.

Table 11-11: Auto Toggle Pin Select Register Description

Bit	Symbol	Description
7:2		Reserved.
1:0	ATPSR[1:0]	00b: Auto Toggle mode is disabled. (default). In this case, nRTS_TXEN operates as nRTS, nDTR_TXEN operates as nDTR, nTXRDY_TXEN operates as nTXRDY and nRXRDY_RXEN operates as nRXRDY. 01b: Select 'nDTR_TXEN' as Auto Toggle control signal instead of dedicated Auto Toggle pins, nTXRDY_TXEN and nRXRDY_RXEN. In this case, nTXRDY_TXEN pin operates as nTXRDY and nRXRDY_RXEN pin operates as nRXRDY. 10b: Select 'nRTS_TXEN' as Auto Toggle control signal instead of dedicated Auto Toggle pins, nTXRDY_TXEN and nRXRDY_RXEN. In this case, nTXRDY_TXEN pin operates as nTXRDY and nRXRDY_RXEN pin operates as nRXRDY. 11b: Select dedicated Auto Toggle pins, nTXRDY_TXEN and nRXRDY_RXEN as Auto Toggle control signal. In this case, nRTS_TXEN operates as nRTS and nDTR_TXEN operates as nDTR.

11.11 PM_PME Message Resource Register (PPMRR)

Select Event to wake-up Root Complex in D3hot state.

Table 11-12: PM_PME Message Resource Register Description

Bit	Symbol	Description
1:7		Reserved.
0	PPMRR[0]	0b: WAKEREQ pin is not selected as Wake-up Event for waking up Root Complex (default). 1b: WAKEREQ pin is selected as Wake-up Event for waking up Root Complex.

If PPMRR[0] is set as 1b which means D3hot-W are set, PM_PME message is sent when only the events occurs.

11.12 General Purpose Outputs Control Register (GPOCR)

Select input/output operation mode from GPIO0 to GPIO7.

Table 11-13: General Purpose Outputs Control Register Description

Bit	Symbol	Description
7	GPOC[7]	0b: Pin GPIO7 operates as input mode. (default) 1b: Pin GPIO7 operates as output mode.
6	GPOC[6]	0b: Pin GPIO6 operates as input mode. (default) 1b: Pin GPIO6 operates as output mode.
5	GPOC[5]	0b: Pin GPIO5 operates as input mode. (default) 1b: Pin GPIO5 operates as output mode.
4	GPOC[4]	0b: Pin GPIO4 operates as input mode. (default) 1b: Pin GPIO4 operates as output mode.
3	GPOC[3]	0b: Pin GPIO3 operates as input mode. (default) 1b: Pin GPIO3 operates as output mode.
2	GPOC[2]	0b: Pin GPIO2 operates as input mode. (default) 1b: Pin GPIO2 operates as output mode.
1	GPOC[1]	0b: Pin GPIO1 operates as input mode. (default) 1b: Pin GPIO1 operates as output mode.
0	GPOC[0]	0b: Pin GPIO1 operates as input mode. (default) 1b: Pin GPIO1 operates as output mode.

11.13 General Purpose Outputs Data Register (GPODR)

Decide output data of GPIO pins from GPIO0 to GPIO7.

Table 11-13: General Purpose Outputs Data Register Description

Bit	Symbol	Description
7	GPOD[7]	Output data bit to GPIO7 pin
6	GPOD[6]	Output data bit to GPIO6 pin
5	GPOD[5]	Output data bit to GPIO5 pin
4	GPOD[4]	Output data bit to GPIO4 pin
3	GPOD[3]	Output data bit to GPIO3 pin
2	GPOD[2]	Output data bit to GPIO2 pin
1	GPOD[1]	Output data bit to GPIO1 pin
0	GPOD[0]	Output data bit to GPIO0 pin

11.14 General Purpose Inputs Data Register (GPIDR)

Indicates inputted data of GPIO pins from GPIO0 to GPIO7.

Table 11-14: General Purpose Inputs Data Register Description

Bit	Symbol	Description
7	GPID[7]	Inputted data bit from GPIO7 pin
6	GPID[6]	Inputted data bit from GPIO6 pin
5	GPID[5]	Inputted data bit from GPIO5 pin
4	GPID[4]	Inputted data bit from GPIO4 pin
3	GPID[3]	Inputted data bit from GPIO3 pin
2	GPID[2]	Inputted data bit from GPIO2 pin
1	GPID[1]	Inputted data bit from GPIO1 pin
0	GPID[0]	Inputted data bit from GPIO0 pin

11.12 General Purpose Outputs Control Register 2 (GPOCR2)

Select input/output operation mode from GPIO8 to GPIOF.

Table 11-13: General Purpose Outputs Control Register 2 Description

Bit	Symbol	Description
7	GPOC[F]	0b: Pin GPIOF operates as input mode. (default) 1b: Pin GPIOF operates as output mode.
6	GPOC[E]	0b: Pin GPIOE operates as input mode. (default) 1b: Pin GPIOE operates as output mode.
5	GPOC[D]	0b: Pin GPIOD operates as input mode. (default) 1b: Pin GPIOD operates as output mode.
4	GPOC[C]	0b: Pin GPIOC operates as input mode. (default) 1b: Pin GPIOC operates as output mode.
3	GPOC[B]	0b: Pin GPIOB operates as input mode. (default) 1b: Pin GPIOB operates as output mode.
2	GPOC[A]	0b: Pin GPIOA operates as input mode. (default) 1b: Pin GPIOA operates as output mode.

Bit	Symbol	Description
1	GPOC[9]	0b: Pin GPIO9 operates as input mode. (default) 1b: Pin GPIO9 operates as output mode.
0	GPOC[8]	0b: Pin GPIO8 operates as input mode. (default) 1b: Pin GPIO8 operates as output mode.

11.13 General Purpose Outputs Data Register 2 (GPODR2)

Decide output data of GPIO pins from GPIO8 to GPIOF.

Table 11-13: General Purpose Outputs Data Register 2 Description

Bit	Symbol	Description
7	GPOD[F]	Output data bit to GPIOF pin
6	GPOD[E]	Output data bit to GPIOE pin
5	GPOD[D]	Output data bit to GPIOD pin
4	GPOD[C]	Output data bit to GPIOC pin
3	GPOD[B]	Output data bit to GPIOB pin
2	GPOD[A]	Output data bit to GPIOA pin
1	GPOD[9]	Output data bit to GPIO9 pin
0	GPOD[8]	Output data bit to GPIO8 pin

11.14 General Purpose Inputs Data Register 2 (GPIDR2)

Indicates inputted data of GPIO pins from GPIO8 to GPIOF.

Table 11-14: General Purpose Inputs Data Register 2 Description

Bit	Symbol	Description
7	GPID[F]	Inputted data bit from GPIOF pin
6	GPID[E]	Inputted data bit from GPIOE pin
5	GPID[D]	Inputted data bit from GPIOD pin
4	GPID[C]	Inputted data bit from GPIOC pin
3	GPID[B]	Inputted data bit from GPIOB pin
2	GPID[A]	Inputted data bit from GPIOA pin
1	GPID[9]	Inputted data bit from GPIO9 pin
0	GPID[8]	Inputted data bit from GPIO8 pin

12. UART(SB16C1150) Functional Description

SB16C1150 is a 32-bit accessible UART Core with 256-byte Tx/Rx FIFO. It also supports 8-bit and 16-bit access. When you use 32-bit access of UART core, you can get the best performance, exactly 4-times faster than existence UART cores. It supports burst transfer up to 512 bytes. So it transfer 256 bytes of TX data. And it also transfer 256-bytes of RX data and 256-byte of RX status.

If you enable 256-byte FIFO, you use the unique supreme function that SB16C1150 offers. It offers communication speed up to 8.7Mbps and more enhanced functions that other UARTs with 128-byte FIFO do not.

SB16C1150 can support 9-bit serial transmission, auto toggling and H/W & S/W flow control. Hardware flow control significantly reduces software overhead and increases system efficiency by automatically controlling serial data flow using the nRTS output and nCTS input signals. Software flow control automatically controls data flow by using programmable Xon/Xoff characters.

SB16C1150 supports 9-bit data communication and automatic address detection function. It also supports IrDA communication data with IrDA encoder/decoder.

Especially SB16C1150 can be accessed with 8-bit/ 16-bit/ 32-bit width data. Because PCI Express Controller handle with data with 32-bit width, if UART controller could be serviced with 32-bit width data, the performance will be better than 8-bit width handling.

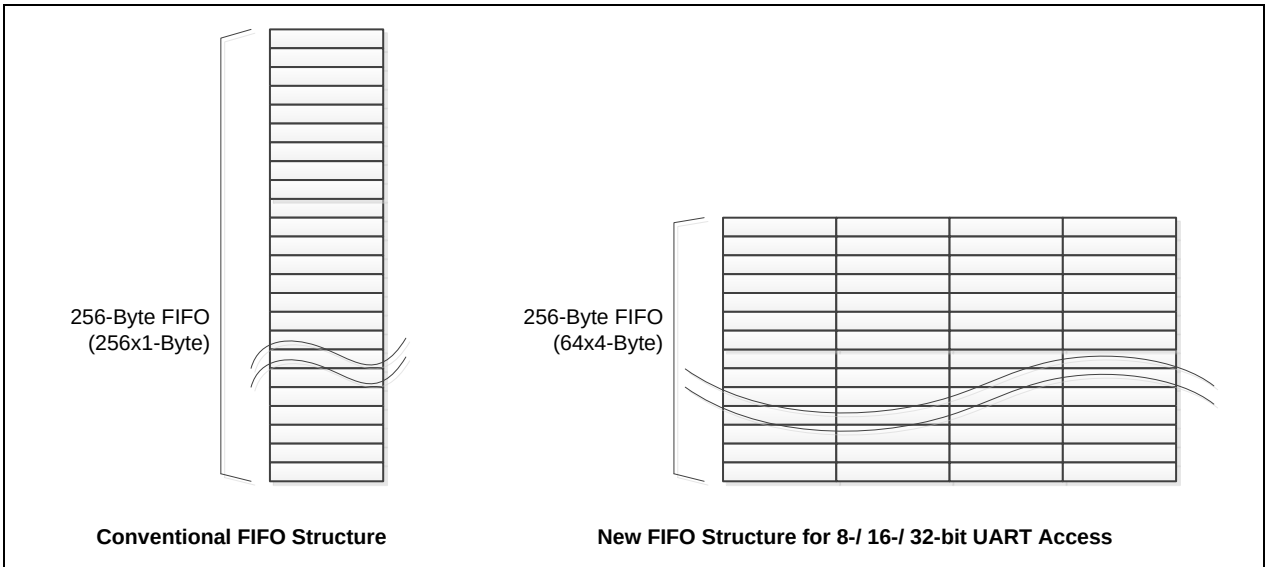


Figure 12–1: New FIFO Structure for 8-/ 16-/ 32-bit UART Access

12.1 FIFO Operation

SB16C1150 UART Core has 256-byte FIFO, but it is uniquely structured because it can be accessed by 8-bit, 16-bit or 32-bit Data Read/Write commands. It has special FIFO structure as Figure 12-1 New FIFO Structure for 8-/ 16-/ 32-bit UART Access.

It shows the special FIFO structure, 64x4-Byte. Not only SB16C1150 UART core can be

accessed with conventional 8-bit access but also with 16-bit or 32-bit access.
When CPU reads 32-bit data from Rx FIFO, there are two methods. One is reading the 32-bit received data from Rx FIFO at once. This is possible only when there is no error in Rx FIFO. The other is reading 2-Byte received data and 2-Byte receive status values from RxFIFO. This is happened when there are errors in Rx FIFO. In this case, the programmer have to know which byte has error, so the CPU can read 1- or 2-byte and its Status Register values from Rx FIFO at once.
With 32-bit UART access function, SB16C1154PCIe is 4 times faster than existing UARTs. Additionally because SB16C1154PCIe can support 256-byte PCI Express Burst Transfer, the speed is much faster than 4 times.
In the 256-byte FIFO 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 13. UART Register Descriptions.

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 nRTS 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 nRTS when the amount of existing data in RX FIFO is less than the written value in FLR. When Auto-CTS is enabled and nCTS 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 nCTS 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, nRTS outputs 0b. If the number of received data in RX FIFO is larger than the value stored in FUR, nRTS will be changed to 1b and if not, holds 0b. This state indicates that RX FIFO can accept more data. After nRTS 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, nRTS changes to 0b and reports that it can accept more data. That is, if nRTS is 0b now, nRTS is not changed to 1b until the amount in RX FIFO exceeds the value set in FUR. But if nRTS is 1b now, nRTS 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 nRTS is 0b or 1b by FSR[5]. If FSR[5] is 0b, nRTS is 0b and if 1b, nRTS is 1b, too.

When IER[6] is set to 1b and nRTS 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. nRTS is changed from 0b to 1b after the first STOP bit is received. Figure 12–2 shows the nRTS timing chart while Auto-RTS is enabled.

In Figure 12–2, Data Byte n-1 is received and nRTS 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 nRTS 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, nRTS is asserted for external UART to transmit new data.

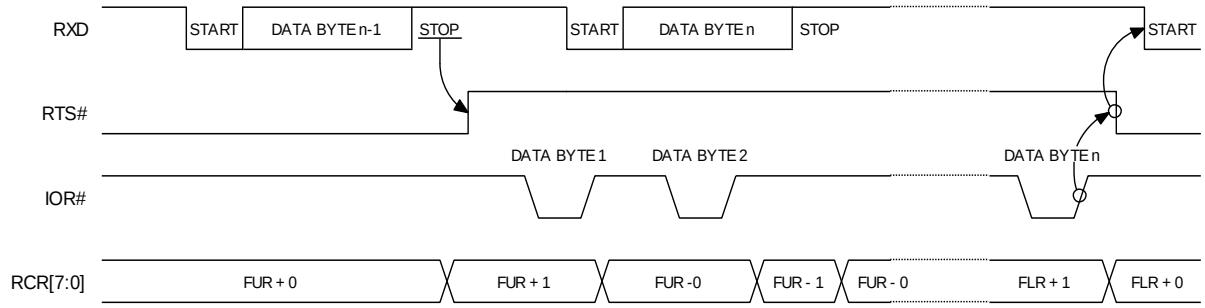


Figure 12–2: nRTS Functional Timing

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 nCTS. 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 nCTS by FSR[1]. If 0b, nCTS is 0b and it means external UART can accept new data, If 1b, nCTS 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 nCTS 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.

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

12.3.1 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 nRTS 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.

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 nCTS 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 SB16C1150 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 SB16C1150 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.
- nCTS, nDSR, nDCD, and nRI 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 SB16C1150 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.
- nCTS, nDSR, nDCD, and nRI inputs are changed.

12.5 Programmable Baud Rate Generator

The SB16C1150 has a programmable baud rate generator with a prescaler. The prescaler is controlled by MCR[7], as shown in Figure 12−3. 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:

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

When MCR[7] is cleared to 0b, prescaler is 1. This is default.

When MCR[7] is set to 1b, prescaler is 4.

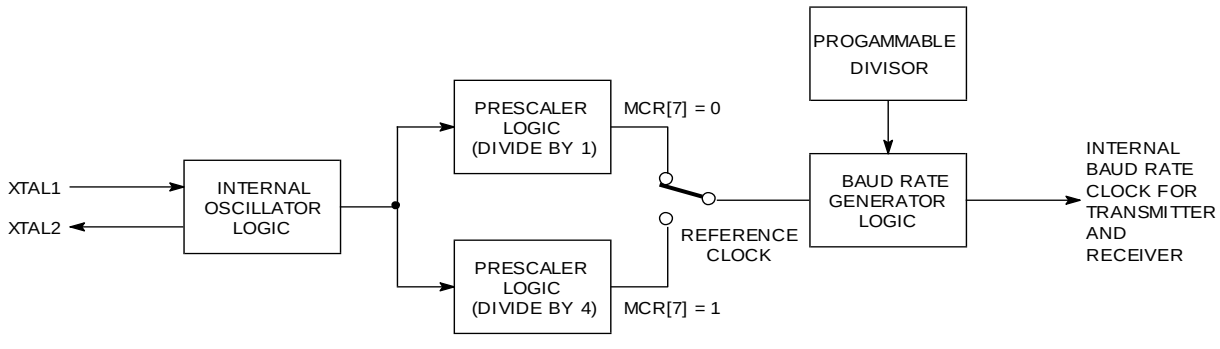


Figure 12−3: Prescaler and Baud Rate Generator Block Diagram

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 SB16C1150 is effectively disabled, as no baud clock will be generated.
For example, when you want to communicate at 2000bps with 29.4912MHz clock, the baud rate divisor have 0399h. In this case, DLM is 03h and DLL is 99h.

Table 12–3 shows the baud rate and divisor value for prescaler with divide by 1 as well as oscillator with frequency 1.8432MHz, 3.6864MHz, 7.3728MHz, 14.7456MHz and 29.4912MHz respectively. When crystal used to this chip, only crystal with 29.4912MHz can be used.

Figure 12–4 shows the crystal clock circuit reference. (Use 29.4912MHz crystal only.)

Table 12–3: Baud Rates

Desired Baud Rate	16X Digit Divisor for Prescaler with Divide by 1				
(bps)	1.8432MHz	3.6864MHz	7.3728MHz	14.7456MHz	29.4912MHz
50	0900h	1200h	2400h	4800h	9000h
75	0600h	0C00h	1800h	3000h	6000h
150	0300h	0600h	0C00h	1800h	3000h
300	0180h	0300h	0600h	0C00h	1800h
600	00C0h	0180h	0300h	0600h	0C00h
1200	0060h	00C0h	0180h	0300h	0600h
1800	0040h	0080h	0100h	0200h	0400h
2000	003Ah	0074h	00E8h	01D0h	0399h
2400	0030h	0060h	00C0h	0180h	0300h
3600	0020h	0040h	0080h	0100h	0200h
4800	0018h	0030h	0060h	00C0h	0180h
7200	0010h	0020h	0040h	0080h	0100h
9600	000Ch	0018h	0030h	0060h	00C0h
19.2K	0006h	000Ch	0018h	0030h	0060h
38.4K	0003h	0006h	000Ch	0018h	0030h
57.6K	0002h	0004h	0008h	0010h	0020h
115.2K	0001h	0002h	0004h	0008h	0010h
230.4K	–	0001h	0002h	0004h	0008h
460.8K	–	–	0001h	0002h	0004h
921.6K	–	–	–	0001h	0002h
1.8432M	–	–	–	–	0001h

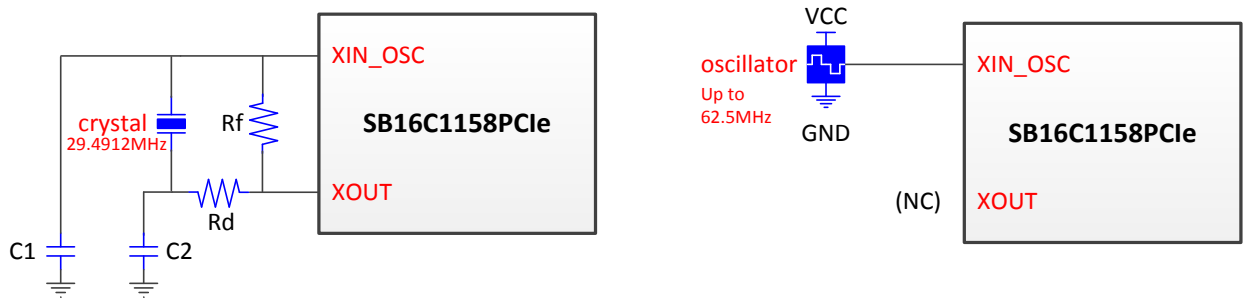


Figure 12–4: Crystal and Oscillator Clock Circuit Reference Diagram

Table 12–4: Component Values for crystal oscillating

Frequency Range (MHz)	C1 (pF)	C2 (pF)	R1 (Ω)	R2(Ω)
29.4912 (recommend)	16	16	1M	33
29.4912 (range)	10~50	10~50	220K ~ 2.2M	10 ~ 100

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 12–5 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. SB16C1150 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.

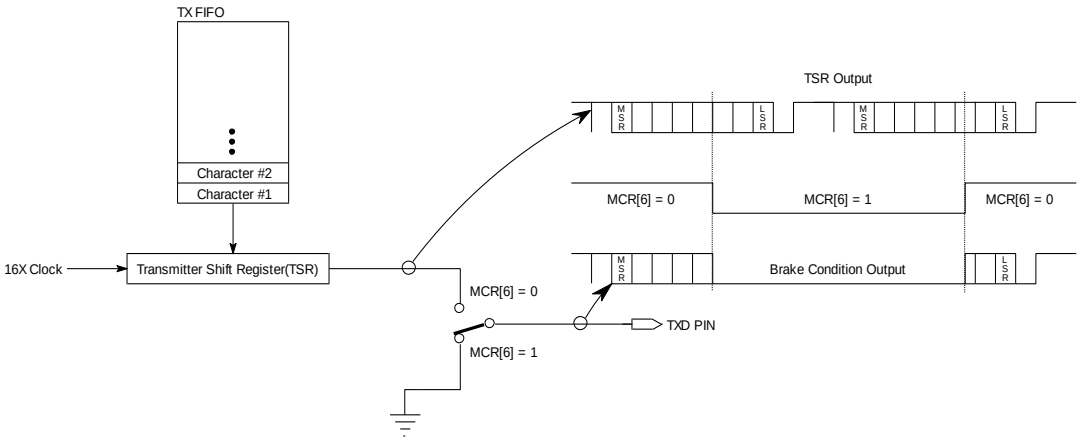


Figure 12–5: Break Condition Block Diagram

12.7 Multi Drop Mode (9-bit Data Transmission)

Some micro-controllers use 9-bit serial communication for several years. It is started when Intel's 8051 micro-controller used 9-bit serial communication, and then, it has been spread out in the market. Serial communication transmits 8-bit data generally. But, 9-bit Serial communication is that transmitting 9-bit data literally.

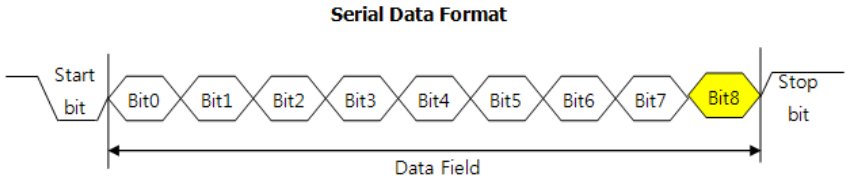


Figure 12–6: Serial Data Format

A multi-drop system consists of master device and slave device in RS422 or RS485 communication. That is because several slave devices are used common data bus line for common. In multi-drop mode, each slave uses distinct address (ID) and a master can communicate with each slave separately using this address.

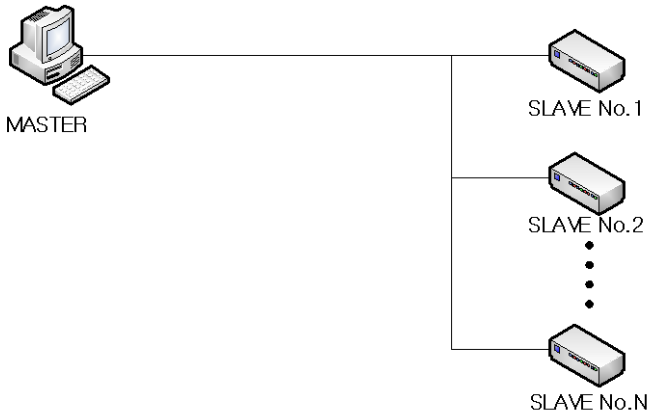


Figure 12–7: Normal RS422 Network

The slave reads all data from bus and compares all data with address itself. If it is equal to address of slave, the slave accept the data. Then, because the slave doesn't know what data from slave is address or data, comparison between data and its own address is operated to find a packet indicated its own among the data from bus.

Because of this, the processing time is consumed in a lot slave and the performance of device is reduced. In order to resolve this problem, 9-bit communication is used to discriminate between address and data. If 9-bit data from master is '1', it means address. Also, 9-bit data from master is '0, it' means data. The processing time can be reduced because the address is compared to its own address if 9-bit is '1'. If the address is same to its own address, a slave accepts data. But, it is not same to its own address, a slave ignores data.

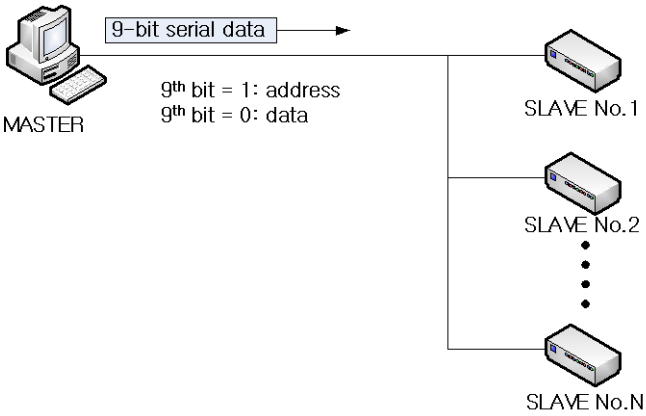


Figure 12–8: Usage of the 9-bit communication on RS422 Network

When RS422/RS485 communication is worked, slaves check the data from the master. If the address is same to its own address, the slave accepts following data and it is not same to its own address, the slave doesn't accept the data. Then, because it is not precise when address is transmitted, a slave device should check RS422 or RS485 Bus's data continually.

But, if 9-bit communication is used, 9-bit data in serial data has only to '1', it is recognized for address. Data doesn't have to be checked because the data is judged that transmitted for its own after the address is compared to its own address. Because of this, the slave device doesn't have to do unnecessary comparison process. As a result, the performance is improved because device's overhead is reduced and the device can be used in many ways.

SB16C1150 UART Core developed in SystemBase is supported 9-bit communication and 3 convenience functions are offered additionally.

12.7.1 Transmit 9-bit Address Register (TAR) / Transmit 9-bit Data Register(TDR)

In SB16C1150, By utilizing for SPR(Scratch Pad Register, 7h) that is used for program buffer and is not effective to UART operation, 9-bit serial data is transmitted more convenient. If SB16C1150 is operated as Multi-Drop mode, 7h is operated as not SPR but TAR(Transmit 9-bit Address Register, 7h). As a result of this, 9-bit serial data is transmitted conveniently. Also, THR(Transmit Holding Register, 0h) is operated as TDR(Transmit 9-bit Data Register, 0h), it helps that 9-bit serial data is transmitted. In other words, if Multi-Drop mode is set, TDR(0h) has only to be written byte data, it transmitted '0' in 9th bit automatically and TAR(7h) has only to be written byte data, it transmitted '1' in 9th bit automatically.

The below picture is UART transmission using THR like the existing 16C550 UART.

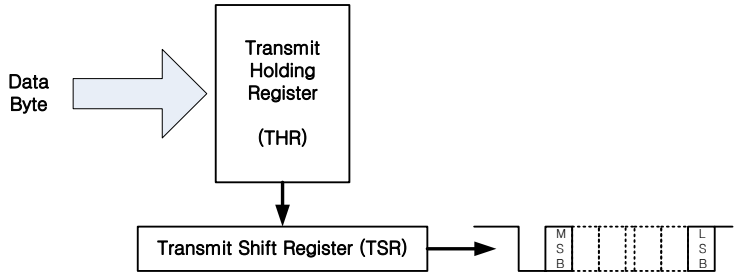


Figure 12–9: Normal Transmit Data processing

But MDR(Multi-Drop mode Register)'s MDE(Multi-Drop mode Enable) bit is set, SB16C1150 is worked as Multi-Drop mode, this can 9-bit serial data transmission. In order to transmit 9-bit address, TAR(Transmit 9-bit Address Register) is active and 9-bit serial information can be transmitted external through TSR(Transmit Shift Register) like below picture.

When 9th bit can be selected following NPS(Ninth-bit Polarity Select) bit information of MDR(Multi-Drop mode Register).

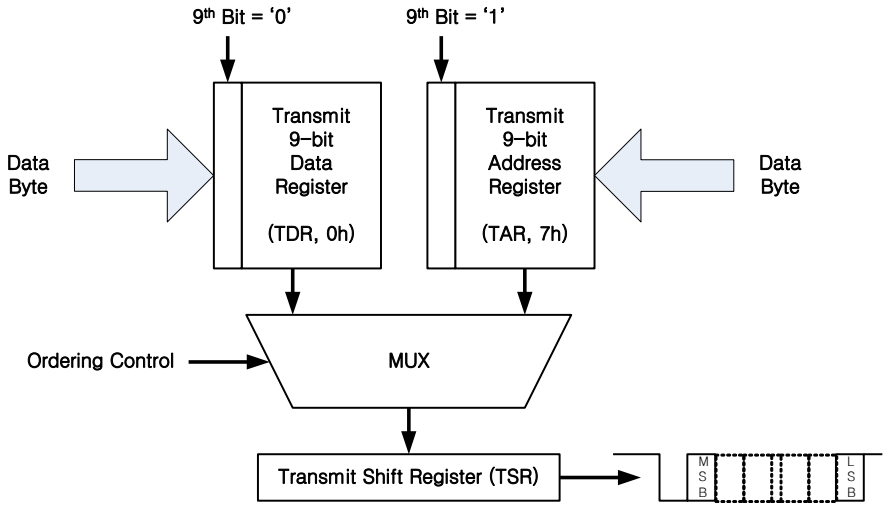


Figure 12–10: 9-bit Transmit Data processing

In general system to support 9-bit communication, when serial data is sent, software command procedure is like below.

① Indicate transmitting address by setting 9th bit to '1' through THR
② Write byte data in THR
③ Indicate transmitting data by clearing 9th bit to '0' through THR
④ Write byte data in THR
⑤ End communication

But, if the way that we suggest is used, software command procedure is like below.

① Write byte data in TAR(7h) (9th bit is set to '1')
② Write byte data in TDR(0h, same to THR) (9th bit is cleared to '0')
③ End communication

If the way that we suggest is used, when 9-bit data is sent, software command procedure is more simplified and twice writing is reduced when RS422 and RS485 communication packet is transmitted. Because of this thing, the performance can be more improved and the serial communication system efficiency can be more upgraded.

12.7.2 Automatic Address Compare

In SB16C1150 UART Core, 2 interrupt sources are added by providing 9-bit communication.
First, if 9-bit serial information is sent in RBR, an interrupt is occurred to detecting 9-bit address. Second, if data of SCR (Special Character Register) is same to 9-bit data, the interrupt is occurred.

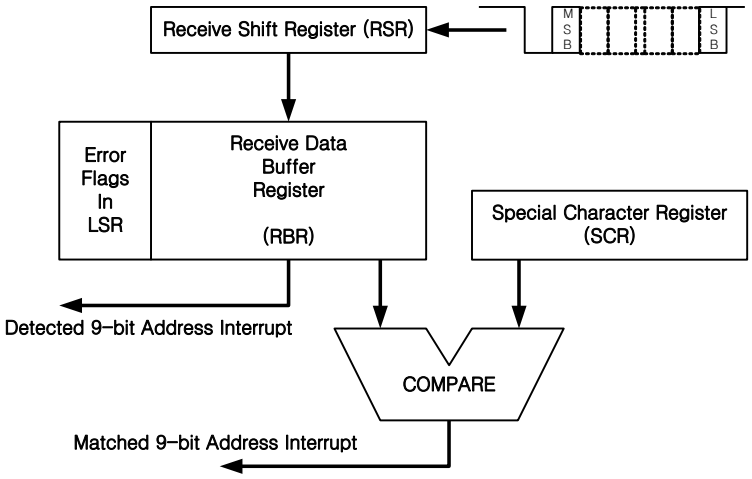


Figure 12–11: Automatic Address Detection in 9-bit communication

If 9th bit is checked for '1' through 9-bit communication, it is regarded as address, it can be differentiated whether packet for right slave if it is compared to slave device. Generally, this comparison is worked in software level, but SB16C1150 UART Core provides this comparison function to operate in hardware level. If 9th bit is '1', its address has only to be stored in advance in SCR(Special Character Register), it can be compared to address in hardware. Of course, if the address designated itself is transmitted in SCR, the software can be signaled by emerging interrupt. SCR is used for Xon/Xoff that is software flow control originally and is used for space to store Xoff/Xon character, in case of Multi-Drop mode, it is used for space to store its address. With this, software and driver's overhead is reduced and the performance is improved through automatic comparison in hardware.

12.7.3 Changed Register Map

Address ADDR[4:0]	Register Name	Bus Size	Description
00h	THR TDR RBR	8-bit	Transmit Holding Register Transmit 9-bit Data Register Receiver Buffer Register
01h	IER	8-bit	Interrupt Enable Register
02h	FCR ISR	8-bit	FIFO Control Register Interrupt Status Register
03h	LCR	8-bit	Line Control Register
04h	MCR	8-bit	Modem Control Register
05h	STR LSR	8-bit	Software Toggle Register Line Status Register
06h	MDR MSR	8-bit	Multi-Drop mode Register Modem Status Register
07h	SPR TAR	8-bit	Scratch Pad Register Transmit 9-bit Address Register
08h	DLL	8-bit	Divisor Latch LSB
09h	DLM	8-bit	Divisor Latch MSB
0Ah	EFR	8-bit	Enhanced Feature Register
0Bh	CPR	8-bit	Clock Pre-scaler Register
0Ch	RMR	8-bit	Receiver Mode-selection Register
0Dh	ATR	8-bit	Auto Toggle Register
0Eh	FSR	8-bit	Flow Status Register
0Fh	XRCR (TOR)	8-bit	Xoff Re-transmit Count Register Receiver Timeout Register
10h	TTR	8-bit	Transmitter FIFO Trigger Level Register
11h	RTR	8-bit	Receiver FIFO Trigger Level Register
12h	FUR	8-bit	Flow Control Upper Threshold Register
13h	FLR	8-bit	Flow Control Lower Threshold Register
14h	TCRL XON1	8-bit/16-bit_0	Transmit FIFO Count Register LSB First XON character Registers
15h	TCRH XON2	8-bit/16-bit_1	Transmit FIFO Count Register MSB Second XON character Register
16h	RCRL XOFF1	8-bit/16-bit_0	Receive FIFO Count Register LSB First XOFF character Register
17h	RCRH XOFF2	8-bit/16-bit_1	Receives FIFO Count Register MSB Second XOFF character Register
18h	RX0 RX0	8-bit/32-bit_0	Receive FIFO Register 0

19h	RX1 RXS0	8-bit/32-bit_1	Receive FIFO Register 1 Receive FIFO Status Register 0
1Ah	RX2 RX1	8-bit/32-bit_2	Receive FIFO Register 2 Receive FIFO Register 1
1Bh	RX3 RXS1	8-bit/32-bit_3	Receive FIFO Register 3 Receive FIFO Status Register 1
1Ch	TX0	8-bit/32-bit_0	Transmit FIFO Register 0
1Dh	TX1	8-bit/32-bit_1	Transmit FIFO Register 0
1Eh	TX2	8-bit/32-bit_2	Transmit FIFO Register 0
1Fh	TX3	8-bit/32-bit_3	Transmit FIFO Register 0

The registers with Bold Character in the upper table are new register or changed register for 9-bit data transmission. For getting detail description. You can see the detail description about the register with bold in the *Chapter 13. UART Register Descriptions.*
Follows are new or changed registers:

- TDR (Transmit 9-bit Data Register) / TAR (Transmit 9-bit Address Register)
- ISR (Interrupt Status Register)
- ACR (Auto Toggle Control Register)
- MDR (Multi Drop mode Register)
- ATR (Auto Toggling Register)
- EFR (Enhanced Feature Register)

13. UART(SB16C1150) Register Descriptions

Each UART channel in the SB16C1154PCIe has its own set of registers selected by address lines A4, A3, A2, A1 and A0 with a specific channel selected. User can select the flattened 32 Bytes Registers of a UART. The complete register set of SB16C1150 is shown on Table 13–1 and Table 13–2.

Table 13–1: Internal Registers Map

Address [4:0]	Register Name	Bus Width	Read/Write (Op. mode)	Description
00h	THR	8-bit	Write Only	Transmitter Holding Register
	RBR	8-bit	Read Only	Receive Buffer Register
	TDR	8-bit	Write Only (9-bit mode)	Transmit 9-bit Data-Byte Register
01h	IER	8-bit	Read/Write	Interrupt Enable Register
02h	FCR	8-bit	Write Only	FIFO Control Register
	ISR	8-bit	Read Only	Interrupt Status Register
03h	LCR	8-bit	Read/Write	Line Control Register
04h	MCR	8-bit	Read/Write	Modem Control Register
05h	STR	8-bit	Write Only	Software Toggle Register
06h	LSR	8-bit	Read Only	Line Status Register
	MDR	8-bit	Write Only	Multi-Drop Mode Register
	MSR	8-bit	Read Only	Modem Status Register
07h	SPR	8-bit	Read/Write (normal mode)	Scratch Pad Register
	TAR	8-bit	Read/Write (9-bit mode)	Transmit 9-bit Address-Byte Register
08h	DLL	8-bit	Read/Write	Divisor Latch LSB
09h	DLM	8-bit	Read/Write	Divisor Latch MSB
0Ah	EFR	8-bit	Read/Write	Enhanced Feature Register
0Bh	CPR	8-bit	Read/Write	Clock Prescaler Register
0Ch	RMR	8-bit	Read/Write	Receiver Mode Selection Register
0Dh	ATR	8-bit	Read/Write	Auto Toggle Register
0Eh	FSR	8-bit	Read Only	Flow Status Register
0Fh	XRCR(TOR)	8-bit	Read/Write	Xoff Re-transmit Count Register (Receiver Timeout Register)
10h	TTR	8-bit	Read/Write	Transmitter FIFO Trigger Level Register
11h	RTR	8-bit	Read/Write	Receive FIFO Trigger Level Register
12h	FUR	8-bit	Read/Write	Flow Control Upper Threshold Register

13h	FLR	8-bit	Read/Write	Flow Control Lower Threshold Register
14h	TCRL	8-bit/16-bit_0	Read Only	Transmitter FIFO Count Register LSB
14h	XON1	8-bit/16-bit_0	Write Only	1st XON Character for software flow control
15h	TCRH	8-bit/16-bit_1	Read Only	Transmitter FIFO Count Register MSB
	XON2	8-bit/16-bit_1	Write Only	2nd XON Character for software flow control
16h	RCRL	8-bit/16-bit_0	Read Only	Receive FIFO Count Register LSB
	XOFF1	8-bit/16-bit_0	Write Only	1st XOFF Character for software flow control
17h	RCRH	8-bit/16-bit_1	Read Only	Receive FIFO Count Register MSB
	XOFF2	8-bit/16-bit_1	Write Only	2nd XOFF Character for software flow control
18h	RX0/RX0	8-bit/ 16-bit_0/ 32-bit_0	Read Only	Receive FIFO Register 0
19h	RX1	16-bit_1/ 32-bit_1	Read Only	Receive FIFO Register 1
19h	RXS0	16-bit_1/ 32-bit_1	Read Only	Receive Status FIFO Register 0
1Ah	RX2	32-bit_2	Read Only	Receive FIFO Register 2
	RX1	32-bit_2	Read Only	Receive FIFO Register 1
1Bh	RX3	32-bit_3	Read Only	Receive FIFO Register 3
	RXS1	32-bit_3	Read Only	Receive Status FIFO Register1
1Ch	TX0	8-bit/ 16-bit_0/ 32-bit_0	Write Only	Transmitter FIFO Register 0
1Dh	TX1	16-bit_1/32-bit_1	Write Only	Transmitter FIFO Register 1
1Eh	TX2	32-bit_2	Write Only	Transmitter FIFO Register 2
1Fh	TX3	32-bit_3	Write Only	Transmitter FIFO Register 3

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
00h	THR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	TDR	Transmit 9-bit Data Register with logic “0” of 9th bit (Accessible when MDE is set to ‘1’)							
00h	RBR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
01h	IER	nCTS Interrupt Enable	nRTS Interrupt Enable	Xoff Special Char 9-bit address Interrupt Enable	Sleep Mode Enable	Modem Status Interrupt Enable	Receive Line Status Interrupt Enable	THR TX FIFO Empty Interrupt Enable	Receive Data Available Interrupt Enable
02h	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
	FCR	0	0	0	0	0	TX FIFO Reset	RX FIFO Reset	FIFO Enable
03h	LCR	0	Set TX Brake	Set Stick Parity	Parity Type Select	Parity Enable	Stop Bits	Word Length Bit 1	Word Length Bit 0
04h	MCR	Xoff Re-transmit Enable	IR Mode Enable	Fast IR Mode Enable	Internal Loopback Enable	Xon-Any Enable	OUT1	nRTS Output	nDTR Output
05h	LSR	RX FIFO Data Error	THR TX FIFO TSR Empty	THR TX FIFO Empty	Receive Break Indicator	Framing Error Indicator	Parity Error Indicator	Overrun Error Indicator	Receive Data Ready Indicator
	STR	RXEN Polarity Select	0	TXEN Polarity Select	0	0	0	0	0
06h	MSR	nDCD	nRI	nDSR	nCTS	ΔnDCD	ΔnRI	ΔnDSR	ΔnCTS
	MDR	0	0	0	0	9-bit Polarity Select	0	9-bit Auto Multi Drop Enable	9-bit Multi Drop Enable
07h	SPR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	TAR	Transmit 9-bit Address Register with logic “1” of 9th bit (Accessible when MDE is set to ‘1’)							
08h	DLL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
09h	DLM	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
0Ah	EFR	Auto-nCTS Flow Control Enable	Auto-nRTS Flow Control Enable	Special Character Detect Enable	Xon/Xoff Parity Check	Software Flow Control Bit 3	Software Flow Control Bit 2	Software Flow Control Bit 1	Software Flow Control Bit 0
0Bh	CPR	0	0	0	Clock Prescaler Select for A/T	0	0	Clock Prescaler Select 1 for Baud	Clock Prescaler Select 0 for Baud
0Ch	RMR	0	0	0	TCR Operation mode	0	0	0	Receiver FIFO Operation Mode

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
0Dh	ATR	RXEN Polarity Select	RXEN Control Mode Select	TXEN Polarity Select	TXEN Control Mode Select	Auto Toggle Deassertion Mode1	Auto Toggle Deassertion Mode0	Auto Toggle Assertion Mode 1	Auto Toggle Assertion Mode 0
0Eh	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
0Fh	XRCR	0	0	Rx Timeout Interrupt Interval Select 1	Rx Timeout Interrupt Interval Select 0	0	0	Xoff Retransmit Count Select 1	Xoff Retransmit Count Select 0
10h	TTR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
11h	RTR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
12h	FUR	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
13h	FLR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
14h	TCRH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	XON1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
15h	TCRL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	XON2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
16h	RCRH	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	XOFF1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
17h	RCRL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	XOFF2	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
18h	RX0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	RX0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
19h	RX1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
	RXS0	RX FIFO Data Error	THR TX FIFO TSR Empty	THR TX FIFO Empty	Receive Break Indicator	Framing Error Indicator	Parity Error Indicator	Overrun Error Indicator	Receive Data Ready Indicator
1Ah	RX2	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
	RX1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
1Bh	RX3	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
	RXS1	RX FIFO Data Error	THR TX FIFO TSR Empty	THR TX FIFO Empty	Receive Break Indicator	Framing Error Indicator	Parity Error Indicator	Overrun Error Indicator	Receive Data Ready Indicator
1Ch	TX0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1Dh	TX1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
1Eh	TX2	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
1Fh	TX3	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24

13.1 Transmit Holding Register (THR, 00h, Output Port)

The transmitter section consists of the Transmitter Holding Register (THR) and Transmitter Shift Register (TSR). The THR is actually a 1-byte buffer or a 256-byte FIFO. The THR receives data and shifts it into the TSR, where it converted to serial data and moved out on the TXD pin. If the FIFO is enabled (256-byte FIFO Mode), THR means location zero of the FIFO is used to store characters but if the FIFO is disabled (1-byte Buffer Mode), THR is the only 1-byte buffer. Characters are lost if overflow occurs.

In the 9-bit transmission mode, this register works as TDR (Transmit 9-bit Data Register). Please refer description of TAR, *13.14 Transmit 9-bit Address Register*.

13.2 Transmit 9-bit Data Byte Register (TDR, 00h, Output Port)

If MDR[0] is set to 1b, the SB16C1150 operates to 9-bit multi drop mode. In the 9-bit multi drop mode, THR is used for Transmit 9-bit Data-Byte Register. TDR is actually a 1-byte buffer or 256-byte FIFO. If TDR receives data from CPU, it shifts received data to the TSR. Complement of MDR[3] is located on the position of parity-bit to represent 9-bit data-byte. In other words, MDR[3] represented data-byte is included on the position of parity bit.

13.3 Receive Buffer Register (RBR, 00h, Input Port)

The receiver section consists of the Receive Buffer Register (RBR) and Receive Shift Register (RSR). The RBR is actually a 1-byte buffer or a 256-byte FIFO. The RSR receives serial data from RXD pin. The serial data is converted to parallel data and id transferred to the RBR. This receiver section is controlled by the line control register. RBR means the location zero of FIFO is used to store characters but if the FIFO is disabled (1-byte Buffer Mode), RBR is the only 1-byte buffer. If overflow occurs, characters are lost. The RBR also stores the error status bits associated with each character.

13.4 Interrupt Enable Register (IER, 01h, Input/Output Port)

IER enables each of the seven types of interrupt, namely receive data ready, transmit empty, line status, modem status, Xoff received, nRTS state transition from low to high, and nCTS state transition from low to high. All interrupts are disabled if bit[7:0] are cleared. The interrupt is enabled by setting appropriate bits. Below table shows IER bit settings.

Table 13–3: Interrupt Enable Register Description

Bit	Symbol	Description
7	IER[7]	**nCTS Interrupt Enable** 0b: Disable the nCTS interrupt (default). 1b: Enable the nCTS interrupt.
6	IER[6]	**nRTS Interrupt Enable** 0b: Disable the nRTS interrupt (default). 1b: Enable the nRTS interrupt.
5	IER[5]	**Xoff/Special character/9-bit address-byte Interrupt Enable** 0b: Disable the Xoff interrupt and 9-bit address-byte matching interrupt (default). 1b: Enable the Xoff interrupt or 9-bit address-byte matching 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 and 9-bit address-byte incoming interrupt (default). 1b: Enable the receive line status interrupt or 9-bit address-byte incoming interrupt.
1	IER[1]	**Transmit Holding Register or TX FIFO Interrupt Enable:** 0b: Disable the THR or TX FIFO interrupt (default). 1b: Enable the THR or TX FIFO interrupt.
0	IER[0]	**Receive Buffer Register Interrupt Enable:** 0b: Disable the RBR interrupt (default). 1b: Enable the RBR interrupt.

13.5 Interrupt Status Register (ISR, 02h, Input Port)

The Sb16C1150 provides multiple levels of prioritized interrupts to minimize software work load. ISR provides the source of interrupt in a prioritized manner.

Below table shows ISR[7:0] bit settings.

ISR[7:6]: These bits are always cleared in non-FIFO modem. They are set when bit 0 of the FCR is set.

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. Mirror the content of FCR[0]. (default). 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 Address Incoming Event (9th bit is '1') 9-bit address-byte incoming event regardless of address-byte matching with character of Xoff2 register.	Reading LSR.
00_0100	2	Receive Data Available	Receiver data available, reaches trigger level.	Reading RBR or RX[3:0] or RCR falls below trigger level.
00_1100	2	Character Timeout Indication	At least one data is in RX FIFO and there is no more data in FIFO during four character time.	Reading the RBR. Or RX[3:0]
00_0010	3	THR (1byte buffer mode) or TX FIFO (256byte FIFO mode) Empty	When THR or TX FIFO is empty or TCR passes below trigger level after TCR passed above trigger level only in 256byte FIFO mode.	Reading the ISR or writing data on THR or TX[3:0]
00_0000	4	Modem Status	nCTS, nDSR, nDCD, nRI	Reading MSR.
01_0000	5	Receive Xoff or Special Character or Address byte of 9-bit	Detection of Xoff, special character or 9-bit address byte matching with character of Xoff2 register.	Reading ISR.
10_0000	6	nRTS, nCTS Status during Auto RTS/CTS flow control	nRTS pin or nCTS pin change state from 0b to 1b.	Reading ISR.

13.6 FIFO Control Register (FCR, 02h, Output Port)

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[7:3]	**Not Used. Always 0000_0b:**
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 TX 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 RX FIFO.
0	FCR[0]	**FIFO enable:** 0b: Disable the 256-byte TX and RX FIFO (default). 1b: Enable the 256-byte TX and RX FIFO

13.7 Line Control Register (LCR, 03h, Input/Output Port)

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]	**Not used. Always 0b:**
6	LCR[6]	**Set Break:** 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.

13.8 Modem Control Register (MCR, 04h, Input/Output Port)

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]	**Xoff Re-transmit Enable:** 0b: Xoff re-transmit disable. (default) 1b: Xoff re-transmit enable.
6	MCR[6]	**IrDA mode Enable:** 0b: Enable the standard receiver and transmitter data format (default). 1b: Enable IrDA receiver and transmitter data format. During this mode, the Tx/Rx outputs/inputs are routed to the infrared encoder/decoder.
5	MCR[5]	**Fast IrDA mode Enable:** 0b: if IrDA mode is enabled, IrDA pulse width will be 3/16 of bit time (default). 1b: if IrDA mode is enabled, IrDA pulse width will be 1/4 of bit time
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]	**Xon Any Enable:** 0b: Disable the Xon-any (default) During the loop back mode, it controls MSR[7] to 1b. 1b: Enable the Xon-any. During the loop back mode, it controls MSR[7] to 0b.
2	MCR[2]	**OUT1:** 0b: During loop back mode, OUT1 outputs to 0b and it controls MSR[6] to 1b. (default) 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 SB16C1158.
1	MCR[1]	**nRTS Output:**

		0b: Force nRTS outputs to 1b. During loop back mode, MCR[1] outputs to 0b and it controls MSR[5] to 0b. 1b: Force nRTS outputs to 0b. During loop back mode, MCR[1] outputs to 1b and it controls MSR[5] to 1b.
0	MCR[0]	**nDTR Output:** 0b: Force nDTR outputs to 1b. During loop back mode, MCR[0] outputs to 0b and it controls MSR[4] to 0b. 1b: Force nDTR outputs to 0b. During loop back mode, MCR[0] outputs to 1b and it controls MSR[4] to 1sb.

13.9 Line Status Register (LSR, 05h, Input Port)

LSR provides the status of data transfers between the SB16C1150 and 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 ten reading RBR. Reading LSR does not lead to 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 character in the RX FIFO.
6	LSR[6]	**THR(1-byte buffer mode), TX FIFO(256-byte FIFO mode) and TSR Empty Indicator:** 0b: THR, TX FIFO or TSR is not empty. 1b: THR, TX FIFO and TSR are empty.
5	LSR[5]	**THR(1-byte buffer mode), TX FIFO(256-byte FIFO mode) Empty Indicator:** 0b: THR or TX FIFO is not empty. 1b: THR and TX FIFO are empty. It indicates that SB16C1150 is ready to accept a new character for transmission. In addition, it uses the SB16C1150 to generate an interrupt to the CPU when the THR or TX FIFO 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). In 9-bit mode, it indicates the detection of 9-bit data-byte. 1b: Parity error. It indicates that the receive character did not have the correct even or odd parity, as selected by the LCR[5:3] In 9-bit mode, it indicates the detection of 9-bit address-byte.
1	LSR[1]	**Overrun Error Indicator:** 0b: No overrun error (default). 1b: Overrun error. It indicates that the character in the RSR is overwritten. A character overrun error generated in the RSR. This happens when additional character arrives while the RBR or RX FIFO is full. In this case, the previous character in the RSR is overwritten. Note that under this condition, the character in the RSR is not transferred into the RBR or RX FIFO, therefore the character in the RBR or RX FIFO is not corrupted by the error.
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.

13.10 Software Toggle Register (STR, 05h, Output Port)

STR controls the signals for controlling input/output signals when using Line Interface as RS422 or RS485 by software.

Below table shows STR bit settings.

Table 13–9: Auto Toggle Control Register Description

Bit	Symbol	Description
7	STR[7]	**RXEN Polarity Select:** 0b: Asserted output of RXEN is 0b. 1b: Asserted output of RXEN is 1b. (default) This bit is not asserted on value of ATR[6] set 1b and is the same as ATR[7].
6	STR[6]	Not used. Always 0b.
5	STR[5]	**TXEN Polarity Select:** 0b: Asserted output of TXEN is 0b. (default) 1b: Asserted output of TXEN is 1b. This bit is not asserted on value of ATR[4] set 1b and is the same as ATR[5].
4:0	STR[4:0]	Not used. Always 0_0000b.

13.11 Modem Status Register (MSR, 06h, Input Port)

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 13–10 shows MSR bit settings.

Table 13–10: Modem Status Register Description

Bit	Symbol	Description
7:4	MSR[7]	**DCD Input Status:** Complement of Data Carrier Detect (nDCD) input. In loop back mode this bit is equivalent to OUT2 in the MCR.
6	MSR[6]	**RI Input Status:** Complement of Ring Indicator (nRI) 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 (nDSR) 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 (nCTS) input. In loop back mode this bit is equivalent to RTS in the MCR.
3	MSR[3]	**ΔDCD Input Status:** 0b: No change on nDCD input (default). 1b: Indicates that the nDCD input has changed state.
2	MSR[2]	**ΔRI Input Status:** 0b: No change on nRI input (default). 1b: Indicates that the nRI input has changed state from 0b to 1b.
1	MSR[1]	**ΔDSR Input Status:** 0b: No change on nDSR input (default). 1b: Indicates that the nDSR input has changed state.
0	MSR[0]	**ΔCTS Input Status:** 0b: No change on nCTS input (default). 1b: Indicates that the nCTS input has changed state.

13.12 Multi Drop mode Register (MDR, 06h, Output Port)

Address 6h is used for MSR (Modem Status Register) in the existent UART Core with R/W permission. But we change to use address 6h with write permission to MDR (Multi Drop mode Register). This is used for setting 9-bit transmission mode of multi drop in RS422 and RS485 network.

Table 13–11: Multi Drop mode Register Description

Bit	Symbol	Description
7:4	MDR[7:4]	Not used. Always 0000b.
3	MDR[3]	**9th Bit Polarity Select (NPS):** 0b: address-byte bit is set to 0b and data-byte bit is set to 1b. 1b: address-byte bit is set to 1b and data-byte bit is set to 0b.

		(default)
2	MDR[2]	Not used. Always 0b.
1	MDR[1]	**9-bit Auto Multi-drop Enable (AME):** 0b: 9-bit auto multi drop mode is disabled (default). 1b: 9-bit auto multi drop mode is enabled. When the auto multi drop mode is enabled, the SB16C1150 checks whether received address-byte on the RSR(Receive Shift Register) matches character of the Xoff2 register or not. If the address-byte matches character of the Xoff2 register, the SB16C1150 automatically stores the address-byte on RX FIFO. Then the next received character (data-byte) on RSR is stored on RX FIFO until the next address-byte is received. If the address-byte doesn't match character of the XOff2 register, the SB16C1150 automatically doesn't store address-byte and data-bytes on RX FIFO until next address-byte is received. When the auto multi drop mode is disabled, the SB16C1150 doesn't check if received address-byte on the RSR matches character of the Xoff2 register, received address-byte and data-bytes stored on RX FIFO regardless of address-byte matching.
0	MDR[0]	**9-bit Multi Drop Enable (MDE):** 0b: 9-bit Multi-Drop mode is disabled. (default). 1b: 9-bit Multi-Drop mode is enabled. If MDR[0] is set to 1b, the SB16C1150 will generate receive-line-status interrupt when received address-byte matched character of Xoff2 register.

13.13 Scratch Pad Register (SPR, 07h, Input/Output Port)

This 9-bit Read/Write Register does not control the SB16C1150 in anyway. It is intended as a scratch pad register to be used by the programmer to hold data temporarily.

13.14 Transmit 9-bit Address Register (TAR, 07h, Output Port)

Address 7h is used for SPR (Scratch Pad Register) in the existent UART Core with R/W permission. But when UART core work in 9-bit transmission mode(MDE, bit0 of MDR bit is set to '1'), this register is used for sending address bye with 9th bit, 1.

If MDR[0] is set to 1b, the SB16C1150 operates to 9-bit multi drop mode. In the 9-bit multi drop mode, SPR is used for Transmit 9-bit Address-Byte Register. The TAR is actually a 1-byte buffer or 256-byte FIFO. If the TAR receives data from CPU, it shifts received data to the TSR. The value of MDR[3] is located on the position of parity-bit to represent 9-bit address-byte. In other words, MDR[3], a bit which represents data-byte is included on the position of parity bit.

13.15 Divisor Latches (DLL/DLM, 08h/09h, Input/Output Port)

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 12.5 describes the details of divisor latches.

13.16 Enhanced Features Register (EFR, 0Ah, Input/Output Port)

EFR enables or disables the enhanced features of SB16C1150.

Below table shows EFR bit settings.

Table 13–12: 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 nCTS pin is inputted 1b. Transmission resumes when nCTS 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 nRTS pin outputs 1b when data in RX FIFO fills above the FUR. nRTS pin outputs 0b when data in RX FIFO fall below the FLR.
5	EFR[5]	**Special Character Detect:** When bit5 of EFR is set to '1', UART core compare received address byte with Special Character (Xoff2) automatically in 9-bit transmission mode. This register used for selecting H/W address matching method or S/W. If bit0 of MDR is not set, this bit can't affect to special character detection. 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]	**XON/XOFF Parity Check:** 0b: Xon/Xoff characters are valid flow control characters even if they have parity errors. 1b: Xon/Xoff characters are not valid flow control characters even if they have parity errors. (default)
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])".

13.17 Clock Prescaler Register (CPR, 0Bh, Input/Output)

CPR enables or disables the enhanced features of SB16C1150.
Below table shows CPR bit settings.

Table 13–13: Clock Prescaler Register Description

Bit	Symbol	Description
7:5	CPR[7:5]	Not used. Always 000b.
4	CPR[4]	**Clock Prescaler Select for Auto Toggling Controller:** 0b: Divided by 1. The external UART clock is supplied directly to the Auto Toggling Controller without further modification. (default). 1b: Divided by 2. The Prescaler divides the external UART clock by 2 and supplies it to the Auto Toggling Controller.
3:2	CPR[3:2]	Not used. Always 00b.
1:0	CPR[1:0]	**Clock Prescaler Select for Programmable Baud Generator:** 00b: Divided by 1. The external UART clock is supplied directly to the Programmable Baud Rate Generator without further modification. 01b: Divided by 4. The prescaler divides the external UART clock by 4 and supplies it to the Programmable Baud Rate Generator. 10b: Divided by 8. The prescaler divides the external UART clock by 8 and supplies it to the Programmable Baud Rate Generator. 11b: Divided by 16. The prescaler divides the external UART clock by 16 and supplies it to the Programmable Baud Rate Generator.

13.18 Receiver Mode Selection Register (RMR, 0Ch, Input/Output Port)

RMR select the FIFO operation mode of SB16C1150.
Below table shows RMR bit settings.

Table 13–14: Receiver Mode Selection Register Description

Bit	Symbol	Description
7:5	RMR[7:5]	Not used. Always 000b.
4	RMR[4]	**TCR Operation Mode:** 0b: TCR indicates the number of characters that can be emptied in TX FIFO. (default) 1b: TCR indicates the number of characters that can be stored in TX FIFO.
3:1	RMR[3:1]	Not used. Always 000b.
0	RMR[0]	**Receiver FIFO Operation Mode:** 0b: RX0/RX0, RX1/RXS0, RX2/RX1, RX3/RXS1 is operated

	to RX0, RX1, RX2, RX3. (default) 1b: RX0/RX0, RX1/RXS0, RX2/RX1, RX3/RXS1 is operated to RX0, RXS0, RX1, RXS1. If there is any error in Rx FIFO, user can read all received data in Rx FIFO with burst access up to 256-byte data and 256-byte status values (LSR).

13.19 Auto Toggle Register (ATR, 0Dh, Input/Output Port)

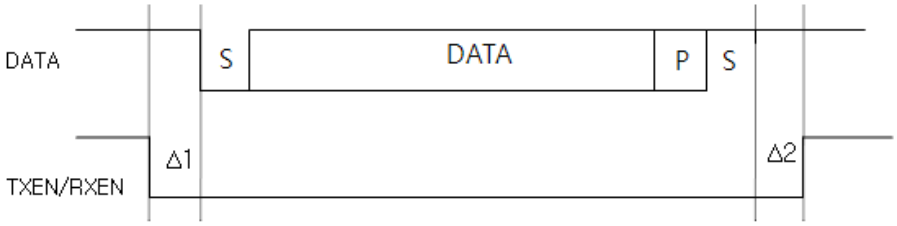
ATR controls the signals for controlling input/output signals when using Line Interface as RS422 or RS485, so eliminates additional glue logic outside.
Below table shows ATR bit settings.

Table 13–15: Auto Toggle 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:** 0b: RXEN is outputted same as ATR[7], irrespective of TXD signal (default). 1b: RXEN is outputted same as ATR[7] when TXD signal is transmitted, and outputted as complement of ATR[7] when TXD signal is not transmitted.
5	ATR[5]	**TXEN Polarity Select:** 0b: Asserted output of TXEN is 0b. (default) 1b: Asserted output of TXEN is 1b.
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 ATR[5] when TXD signal is transmitting, and outputted as complement of ATR[7] when TXD signal is not transmitting.
3:2	ATR[3:2]	**Delayed De-assert Time Select for Auto Toggle:** 00b: maximum delayed de-assert time is 0-clock time of external UART clock. (default) 01b: maximum delayed de-assert time is 1-clock time of external UART clock. 10b: maximum delayed de-assert time is 2-clock time of external UART clock. 11b: maximum delayed de-assert time is 4-clock time of external UART clock.
1:0	ATR[1:0]	**Delayed De-assert Time Select for Auto Toggle:** When ATR[3:2] is not set to 11b (For normal RS422/RS485 transceiver)

00b: maximum delayed assert time is 0-clock time of external UART clock. (default)
01b: maximum delayed assert time is 1-clock time of external UART clock.
10b: maximum delayed assert time is 2-clock time of external UART clock.
11b: maximum delayed assert time is 4-clock time of external UART clock.

When ATR[3:2] is set to 11b
(For slew-rate limit RS422/RS485 transceiver)
00b: maximum delayed assert time is 8-clock time of external UART clock. (default)
01b: maximum delayed assert time is 16-clock time of external UART clock.
10b: maximum delayed assert time is 22-clock time of external UART clock.
11b: maximum delayed assert time is 37-clock time of external UART clock.



13.20 Flow Control Status Register (FSR, 0Eh, Input Port)

FSR show the status of operation of TX Hardware Flow Control, RX Hardware Flow Control, TX Software Flow Control, and RX Software Flow Control.
Below table shows FSR bit settings.

Table 13–16: 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 are 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 reads 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 are 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 reads 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, '0' is inputted in NCTS 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 are enabled, '1' is inputted in NCTS 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 are 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 are 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.

13.21 Xoff Re-transmit Count Register (XRCR, 0Fh, Input/Output Port)

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.

Below Table shows XRCR bit settings. XRCR is consist of TOR(Receiver Time-Out Register) field and XRCR(Xoff Re-transmit Count Register) field.

Table 13–17: Xoff Re-Transmit Count Register Description

Bit	Symbol	Description
7:6	TOR[3:2]	Not used. Always 00b. (Receiver Time-Out Register field)
5:4	TOR[1:0]	**Receiver Time-out Interrupt Interval Select:** 00b: Receiver time-out is generated by no character has been removed from RX FIFO during last one character time, and there is at least one character in it during this time (default). 01b: Receiver time-out is generated by no character has been removed from RX FIFO during last two character time, and there is at least one character in it during this time. 10b: Receiver time-out is generated by no character has been removed from RX FIFO during last three character time, and there is at least one character in it during this time 01b: Receiver time-out is generated by no character has been removed from RX FIFO during last four character time, and there is at least one character in it during this time.
3:2	XRCR[3:2]	Not used. Always 00b.
1:0	XRCR[1:0]	**Xoff Re-transmit Count Select:** 00b: If EFR[3:0] is not set to 11x1b, It transmits Xoff character whenever the number of received data is 1 during XOFF status (default). If EFR[3:0] is set to 11x1b, It transmits Xoff character whenever the number of received data is 2 during XOFF status. 01b: It transmits Xoff character whenever the number of received data is 4 during XOFF status. 10b: It transmits Xoff character whenever the number of received data is 8 during XOFF status. 11b: It transmits Xoff character whenever the number of received data is 16 during XOFF status.

13.22 Transmit FIFO Trigger Level Register (TTR, 10h, Input/Output)

It operates only when 256-byte FIFO mode is enabled. It sets the trigger level of 256-byte TX FIFO for generating to transmit interrupt. The interrupt is generated when the number of data remained in TX FIFO after transmitting through TXD pin is less than the value of TTR. Default value is 32d, 0010_0000b. And 0000_0000b must not be written. If it is written, unexpected operation may occur.

13.23 Receive FIFO Trigger Level Register (RTR, 11h, Input/Output)

It operates only when 256-byte FIFO mode is enabled. It sets the trigger level of 256-byte RX FIFO for generating to receive interrupt. The interrupt is generated when the number of data remained in RX FIFO exceeds the value of RTR(this time, timeout or interrupt is valid). Default value is 16d, 0001_0000b.

13.24 Flow Control Upper Threshold Register (FUR, 12h, Input/Output)

It can be written only when 256-byte FIFO mode is enabled and one of TX software flow controls or Auto-RTS is enabled. While TX software flow control is enabled, Xoff character is transmitted when the number of data in RX FIFO exceeds the value of FUR. If Auto-RTS is enabled, '1' is outputted on nRTS pin to report that it cannot receive data any more. If both TX software flow control and Auto-RTS are enabled, Xoff character is transmitted and '1' is outputted on nRTS pin. The value of FUR must be larger than that of FLR. Default value is 224d, 1110_0000b.

13.25 Flow Control Lower Threshold Register (FLR, 13h, Input/Output)

It can be written only when 256-byte FIFO mode is enabled and one of TX software flow controls or Auto-RTS is enabled. 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 FLR only if Xoff character is transmitted before. If Auto-RTS is enabled, '0' is outputted on nRTS pin to report that it can receive more data. If both TX software flow control and Auto-RTS are enabled, Xon character is transmitted only if Xoff character is transmitted before and '0' is outputted on nRTS pin. The value of FLR must be less than that of FUR. Default value is 32d, 0100_0000b.

13.26 Transmitter FIFO Count Register LSB (TCRL, 14h, Input)

TCR(TCRH + TCRL) shows the number of characters stored in TX FIFO. If the number of characters stored in TX FIFO is 0, it is shown as 00000000(TCRH)_00000000b(TCRL) and if it is 255, it is shown as 0000000_11111111b. And in case of the maximum number 256, it is shown as 00000001_00000000b.

13.27 Transmitter FIFO Count Register MSB (TCRH, 15h, Input)

TCR(TCRH + TCRL) shows the number of characters stored in TX FIFO. If the number of characters stored in TX FIFO is 0, it is shown as 00000000(TCRH)_00000000b(TCRL) and if it is 255, it is shown as 0000000_11111111b. And in case of the maximum number 256, it is shown as 00000001_00000000b.

13.28 Receiver FIFO Count Register LSB (RCRL, 16h, Input)

RCR(RCRH + RCRL) shows the number of characters stored in RX FIFO. If the number of characters stored in RX FIFO is 0, it is shown as 00000000(RCRH)_00000000b(RCRL) and if it is 255, it is shown as 00000000_11111111b. And in case of the maximum number 256, it is shown as 00000001_00000000b.

13.29 Receiver FIFO Count Register MSB (RCRH, 17h, Input)

RCR(RCRH + RCRL) shows the number of characters stored in RX FIFO. If the number of characters stored in RX FIFO is 0, it is shown as 00000000(RCRH)_00000000b(RCRL) and if it is 255, it is shown as 00000000_11111111b. And in case of the maximum number 256, it is shown as 00000001_00000000b.

13.30 Xon1 Character Register (XON1, 14h, Output)

XON1 is used to program the Xon1 control character. Default value is 00d, 0000_0000b.

13.31 Xon2 Character Register (XON2, 15h, Output)

XON2 is used to program the Xon2 control character. Default value is 00d, 0000_0000b.

13.32 Xoff1 Character Register (XOFF1, 16h, Output)

XOFF1 is used to program the Xoff1 control character. Default value is 00d, 0000_0000b.

13.33 Xoff2 Character Register (XOFF2, 17h, Output)

XOFF2 is used to program the Xoff1 control character. Default value is 00d, 0000_0000b.

13.34 Receive FIFO Register 0 (RX0, 18h, Input)

Please refer 13.37. 18h~1Bh of UART is used to Receive FIFO Register or Receive Line Status Register for 8-/ 16-/ 32-bit access.

13.35 Receive FIFO Register 1 / Receive Line Status Register 0 (RX1/RXS0, 19h, Input)

Please refer 13.37. 18h~1Bh of UART is used to Receive FIFO Register or Receive Line Status Register for 8-/ 16-/ 32-bit access.

13.36 Receive FIFO Register 2 / Receive FIFO Register 1 (RX2/RX1, 1Ah, Input)

Please refer 13.37. 18h~1Bh of UART is used to Receive FIFO Register or Receive Line Status Register for 8-/ 16-/ 32-bit access.

13.37 Receive FIFO Register 3 / Receive Line Status Register 0 (RX3/RXS1, 1Bh, Input)

When Receiver FIFO Operation Mode(RMR[0]) is ‘0’, these registers are operated to RX0/RX1/RX2/RX3 and when Receiver FIFO Operation Mode(RMR[0]) is ‘0’, these registers are operated to RX0/RXS0/RX1/RXS1.

RX0/RX1/RX2/RX3 provide software with the option of accessing the RX_FIFO in 8-bit, 16-bit or 32-bit form and RX0/RXS1/RX1/RXS1 provide software with the option of accessing the RX_FIFO and RXS_FIFO in 16-bit or 32-bit form.

The CPU to make 8-bit accesses to the location of RX0 (do not allow to access to the location of RX1, RX2, RX3), which the SB16C1150 core accesses to 8-bit data size of RX_FIFO with constructing of 8-bit x 4-block x 64-depth.
For a 8-bit read (assuming above 1 byte exist in RX_FIFO), the relations are:

RX_FIFO[GET_PTR] to RX[0]

The CPU to make 16-bit accesses to the location of RX0 (do not allow to access to the location of RX1, RX2, RX3), which the SB16C1150 core accesses to 16-bit data size of RX_FIFO or 8-bit data size of RX_FIFO/RXS_FIFO with constructing of 8-bit x 4-block x 64-depth.
For a 16-bit read on RMR[0] is set to 0b (assuming above 2 bytes exist in RX_FIFO), the relations are:

RX_FIFO[GET_PTR + 1] to RX[1]
RX_FIFO[GET_PTR + 0] to RX[0]

For a 16-bit read on RMR[0] is set to 1b (assuming above 1 byte exist in RX_FIFO), the relations are:

RXS_FIFO[GET_PTR + 0] to RX[1]
RX_FIFO[GET_PTR + 0] to RX[0]

The CPU to make 32-bit accesses to the location of RX0 (do not allow to access to the location of RX1, RX2, RX3), which the SB16C1150 core accesses to 32-bit data size of RX_FIFO or 16-bit data size of RX_FIFO/RXS_FIFO with constructing of 8-bit x 4-block x 64-depth.
For a 32-bit read on RMR[0] is set to 0b (assuming above 4 bytes exist in RX_FIFO), the relations are:

RX_FIFO[GET_PTR + 3] to RX[3]
RX_FIFO[GET_PTR + 2] to RX[2]
RX_FIFO[GET_PTR + 1] to RX[1]
RX_FIFO[GET_PTR + 0] to RX[0]

For a 32-bit read on RMR[0] is set to 1b (assuming above 4 bytes exist in RX_FIFO), the relations are:

RXS_FIFO[GET_PTR + 1] to RX[3]
RX_FIFO[GET_PTR + 1] to RX[2]
RXS_FIFO[GET_PTR + 0] to RX[1]
RX_FIFO[GET_PTR + 0] to RX[0]

13.38 Transmit FIFO Register 0 (TX0, 1Ch, Output)

Please refer 13.41. 1Ch~1Fh of UART is used to Transmit FIFO Register for 8-/ 16-/ 32-bit access.

13.39 Transmit FIFO Register 1 (TX1, 1Dh, Output)

Please refer 13.41. 1Ch~1Fh of UART is used to Transmit FIFO Register for 8-/ 16-/ 32-bit access.

13.40 Transmit FIFO Register 2 (TX2, 1Eh, Output)

Please refer 13.41. 1Ch~1Fh of UART is used to Transmit FIFO Register for 8-/ 16-/ 32-bit access.s

13.41 Transmit FIFO Register 3 (TX3, 1Fh, Output)

TX[3:0] provide software with the option of accessing the TX_FIFO in 8-bit, 16-bit or 32-bit form.

The CPU to make 8-bit accesses to the location of TX0 (do not allow to access to the location of TX1, TX2, TX3), which the SB16C1150 core accesses to 8-bit data size of TX_FIFO with constructing of 8-bit x 4-block x 64-depth.
For a 8-bit read (assuming above 1 byte free in TX_FIFO), the relations are:

TX_FIFO[PUT_PTR] to TX[0]

The CPU to make 16-bit accesses to the location of TX0 (do not allow to access to the location of TX1, TX2, TX3), which the SB16C1150 core accesses to 16-bit data size of TX_FIFO with constructing of 8-bit x 4-block x 64-depth.
For a 16-bit read (assuming above 2 bytes free in TX_FIFO), the relations are:

TX_FIFO[PUT_PTR + 1] to TX[1]
TX_FIFO[PUT_PTR + 0] to TX[0]

The CPU to make 32-bit accesses to the location of TX0 (do not allow to access to the location of TX1, TX2, TX3), which the SB16C1150 core accesses to 32-bit data size of TX_FIFO with constructing of 8-bit x 4-block x 64-depth.
For a 32-bit read (assuming above 4 bytes free in TX_FIFO), the relations are:

TX_FIFO[PUT_PTR + 3] to TX[3]
TX_FIFO[PUT_PTR + 2] to TX[2]
TX_FIFO[PUT_PTR + 1] to TX[1]
TX_FIFO[PUT_PTR + 0] to TX[0]

14. Electrical Information ← 내용 아이칩스에서 받아서 적용 예정

14.1 Absolute Maximum Ratings

Symbol	Description	Rating	Unit
VCCK	Core Power Supply	-0.3 to 2.16	V
VCC3IO	Power Supply of 3.3V I/O	-0.3 to 4.0	V
V_{IN3}	Input Voltage of 3.3V I/O	-0.3 to 4.0	V
	Input Voltage of 3.3V I/O with 5V tolerance	-0.3 to 5.8	V
T_{STG}	Storage Temperature	-40 to 150	℃
I_{IN}	DC Input Current	20	mA
I_{OUT}	Output Short Circuit Current	20	mA
P_{MAX}	Maximum Power Consumption - Most of Power Consumption took a place in internal Core(1.8V)	434	mW

14.2 Recommended Operating Conditions

Symbol	Description	Min	Typ	Max	Unit
VCCK	Core Power Supply	1.62	1.8	1.98	V
VCC3IO	Power Supply of 3.3V I/O	2.97	3.3	3.63	V
V_{IN3}	Input Voltage of 3.3V I/O	0	3.3	3.63	V
	Input Voltage of 3.3V I/O with 5V tolerance	0	3.3	5.25	V
T_J	Junction Operating Temperature	-40	25	125	℃

14.3 DC Characteristics of I/O

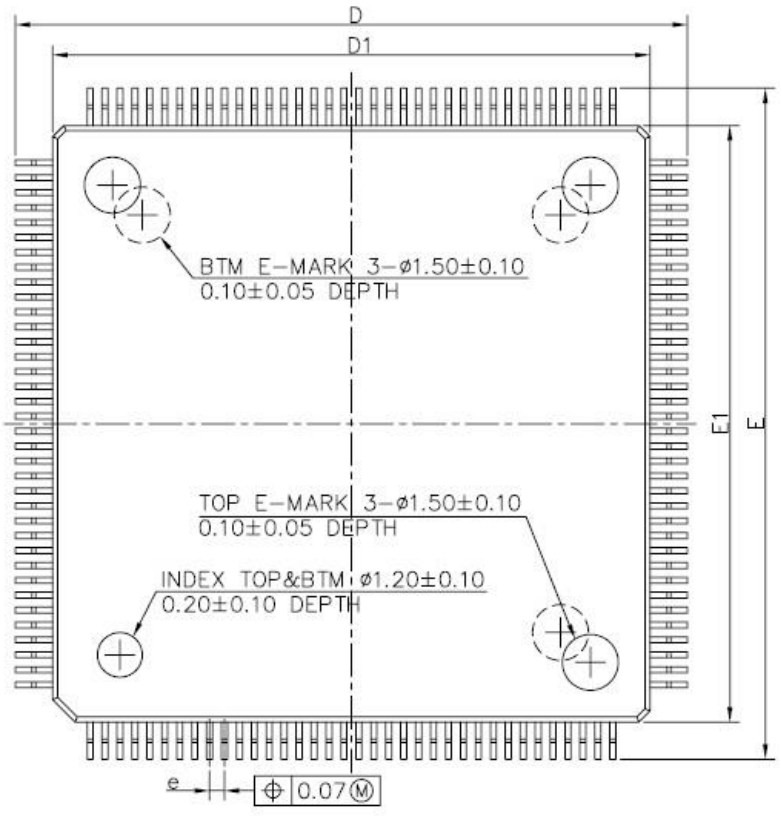
Symbol	Parameter	Condition	Min	Typ	Max	Unit
VCCK	Core Power Supply	1.8V	1.62	1.8	1.98	V
VCC3IO	Power Supply	3.3V	2.97	3.3	3.63	V
T_J	Junction Temperature		-40	25	125	℃
V_{IL}	Input Low Voltage	LVTTL	-	-	0.8	V
V_{IH}	Input High Voltage		2.0	-	-	V
V_T	Switching Threshold	LVTTL	-	1.5	-	V
V_{T-}	Schmitt Trigger - Threshold	Neg. going	0.8	1.1	-	V
V_{T+}	Schmitt Trigger + Threshold	Pos. going	-	1.6	2.0	V
V_{OL}	Output Low Voltage	2~16mA	-	-	0.4	V
V_{OH}	Output High Voltage	-2~-16mA	2.4	-	-	V
R_{PU}	Input pull-up Resistance	V_{IN} = 0V	40	75	190	KΩ
R_{PD}	Input pull-down Resistance	V_{IN} = 3.3V	40	75	190	KΩ
I_{IN}	Input Leakage Current		-10	±1	10	uA
	Input Leakage Current with pull-up resistance	V_{IN} = 0V	-15	-45	-85	uA
	Input Leakage Current with pull-down resistance	V_{IN} = 3.3V	15	45	85	uA
I_{OZ}	Tri-State Output Leakage Current		-10	±1	10	uA

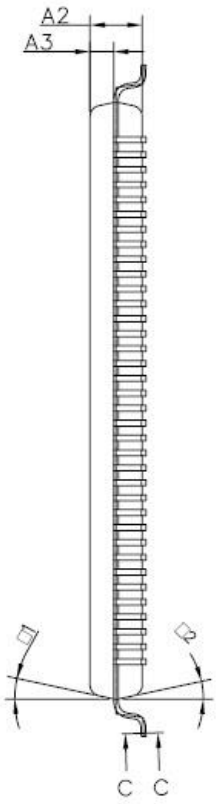
14.4 I/O Classification

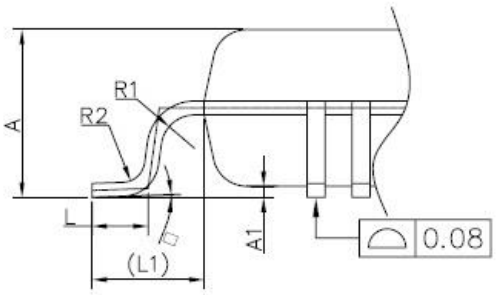
Group 1	V_{IL}	0.00	MA0, MA1, MA2, MA3, MA4, MA5, MA6, MA7, TXD2, RTS2n, DTR2n, TXEN2, RXEN2, TXD3, RTS3n, DTR3n, TXEN3, RXEN3, SM2, SM3, MRESET, MCS4n, MCS5n, MCS6n, MCS7n, OPT_ENn, UART_ENn, MIO_ENn, MWRn, MRDn
	V_{IH}	0.00	
	V_{OL}	0.40	
	V_{OH}	2.40	
	I_{OL}	8.00	
	I_{OH}	-8.00	
Group 2	V_{IL}	0.00	TXD0, RTS0n, DTR0n, TXEN0, RXEN0, TXD1, RTS1n, DTR1n, WAKEn, RCS, SM0, SM1, RSVD_C0, RSVD_C1, RSVD_C2, RSVD_C3, RSVD_C4, RSVD_C5, RSVD_C6, RSVD_C7
	V_{IH}	0.00	
	V_{OL}	0.40	
	V_{OH}	2.40	
	I_{OL}	16.00	
	I_{OH}	-16.00	
Group 3	V_{IL}	0.80	RXD0, CTS0n, DSR0n, DCD0n, RI0n, RXD1, CTS1n, DSR1n, DCD1n, RI1n, RXD2, CTS2n, DSR2n, DCD2n, RI2n, RXD3, CTS3n, DSR3n, DCD3n, RI3n, PERSTn, VAUXGOOD, MWAIT, OSC_IN, OSC_SELn, OSC[1:0], MINT[7:4], GINT[3:0]n, TRXSSEL, IQUART, PN_EN[3:0]n, INTF4[1:0], INTF20, INTF10, CLKSEL, RSVD_O[2:0], EXTLOAD, INTF3[1:0], PORT[3:0], WAKEREQ, TSTCKEN,
	V_{IH}	2.00	
	V_{OL}	0.00	
	V_{OH}	0.00	
	I_{OL}	0.00	
	I_{OH}	0.00	
Group 4	V_{IL}	0.80	TXEN1, RXEN1, RSK, RDI, RDO, PMS_TSTCLK;
	V_{IH}	2.00	
	V_{OL}	0.40	
	V_{OH}	2.40	
	I_{OL}	16.00	
	I_{OH}	-16.00	
Group 5	V_{IL}	0.80	INTF21, INTF11
	V_{IH}	2.00	
	V_{OL}	0.40	
	V_{OH}	2.40	
	I_{OL}	4.00	
	I_{OH}	-4.00	
Group 6	V_{IL}	0.80	MD[7:0]
	V_{IH}	2.00	
	V_{OL}	0.40	
	V_{OH}	2.40	
	I_{OL}	8.00	
	I_{OH}	-8.00	

15. Package Outline

144Pin TQFP: Thin Quad Flat Package; Body 16 x 16 x 1.6 mm







COMMON DIMENSIONS
(UNITS OF MEASURE=MILLIMETER)

SYMBOL	MIN	NOM	MAX
A	–	–	1.60
A1	0.05	–	0.15
A2	1.35	1.40	1.45
A3	0.59	0.64	0.69
b	0.14	–	0.23
b1	0.13	0.16	0.19
c	0.13	–	0.18
c1	0.115	0.125	0.135
D	17.80	18.00	18.20
D1	15.90	16.00	16.10
E	17.80	18.00	18.20
E1	15.90	16.00	16.10
e	0.35	0.40	0.45
2 L	0.35	0.50	0.65
L1		1.00REF	
R1	0.08	–	–
R2	0.08	–	0.30
θ	0°	2°	8°
θ 1	11°	12°	13°
θ 2	11°	12°	13°

Note :

All dimensions are in millimeters.