Octal-UART Controller with 256-Byte FIFO

IN16C1058

Revision 1.0

IK Semicon Co., Ltd.

CONTENTS

1. Description

IN16C1058 is a octal UART(Universal Asynchronous Receiver/Transmitter) with 256-byte FIFO supporting maximum communication speed of 5.3Mbps. It offers flow control function by hardware or software and signal lines which can open or close the Tx/Rx input/output when communicating by RS-422 or RS-485. It can handle eight internal interrupt signals (INT0, INT1, INT2, INT3, INT4, INT5, INT6 and INT7) with one global interrupt signal line (INT) and offers a new 'Xoff re-transmit' function in addition to Xon any character.

UART can convert 8-bit parallel data to asynchronous serial data and vice versa. It can transmit 5 to 8-bit letters, program I/O interrupt trigger level and has 256-byte I/O data FIFO.
UART can generate any baud rate using clock and programmable divisor, transmit data with even, odd or no parity and 1, 1.5, 2 stop bit, and detect break, idle, framing error, FIFO overflow and parity error in input data.
UART has a software interface for modem controlling.

IN16C1058 offers TQFP128 (20x20 body) packages.

2. Features

- 8 Channel UART
- 3.3V Operation with 5V tolerant Inputs
- Up to 5.3 Mbps Baud Rate (Up to 85 MHz Oscillator Input Clock)
- 256-byte Transmit FIFO
- 256-byte Receive FIFO with Error Flags
- Industrial Temperature Range (-40 ℃ to +85 ℃)
- Programmable and Selectable Transmit and Receive FIFO Trigger Levels for DMA and Interrupt Generation
- Software (Xon/Xoff) / Hardware (nRTS/nCTS) Flow Control
  - Programmable Xon/Xoff Characters
  - Programmable Auto-RTS and Auto-CTS
- Global Interrupt Mask/Poll Control
- Optional Data Flow Resume by Xon Any Character Control
- Optional Data Flow Additional Halt by Xoff Re-transmit Control
- Dedicated pins for automatic bus control of RS-422 and RS-485 communications.
  - RS-422 Point to Point/Multi-Drop Control
  - RS-485 Echo/Non Echo Control
- DMA Signaling Capability for Both Received and Transmitted Data
- Software Selectable Baud Rate Generator
- Prescaler Provides Additional Divide-by-4 Function
- Fast Data Bus Access Time
- Programmable Sleep Mode

- Programmable Serial Interface Characteristics

- 5, 6, 7, or 8-bit Characters
- Even, Odd, or No Parity Bit Generation and Detection
- 1, 1.5, or 2 Stop Bit Generation

- False Start Bit Detection
- Line Break Generation and Detection
- Fully Prioritized Interrupt System Controls
- Modem Control Functions (nRTS, nCTS, nDTR, nDSR, nDCD, and nRI)
- Built-In the Control Logics for multi serial channels
  - Address decoding logic for 8 channels
  - Each serial interface working mode information
  - Option Registers for internal interrupt control
  - Expandable up to 32-port without any glue-logics using MIO Bus

3. Ordering Information

Table 1: Ordering Information

Part Number	Package	Operating Temperature Range	Device Status
IN16C1058-TQ	128-Pin TQFP (20x20)	-40 ℃ to +85 ℃	Active

4. Block Diagram

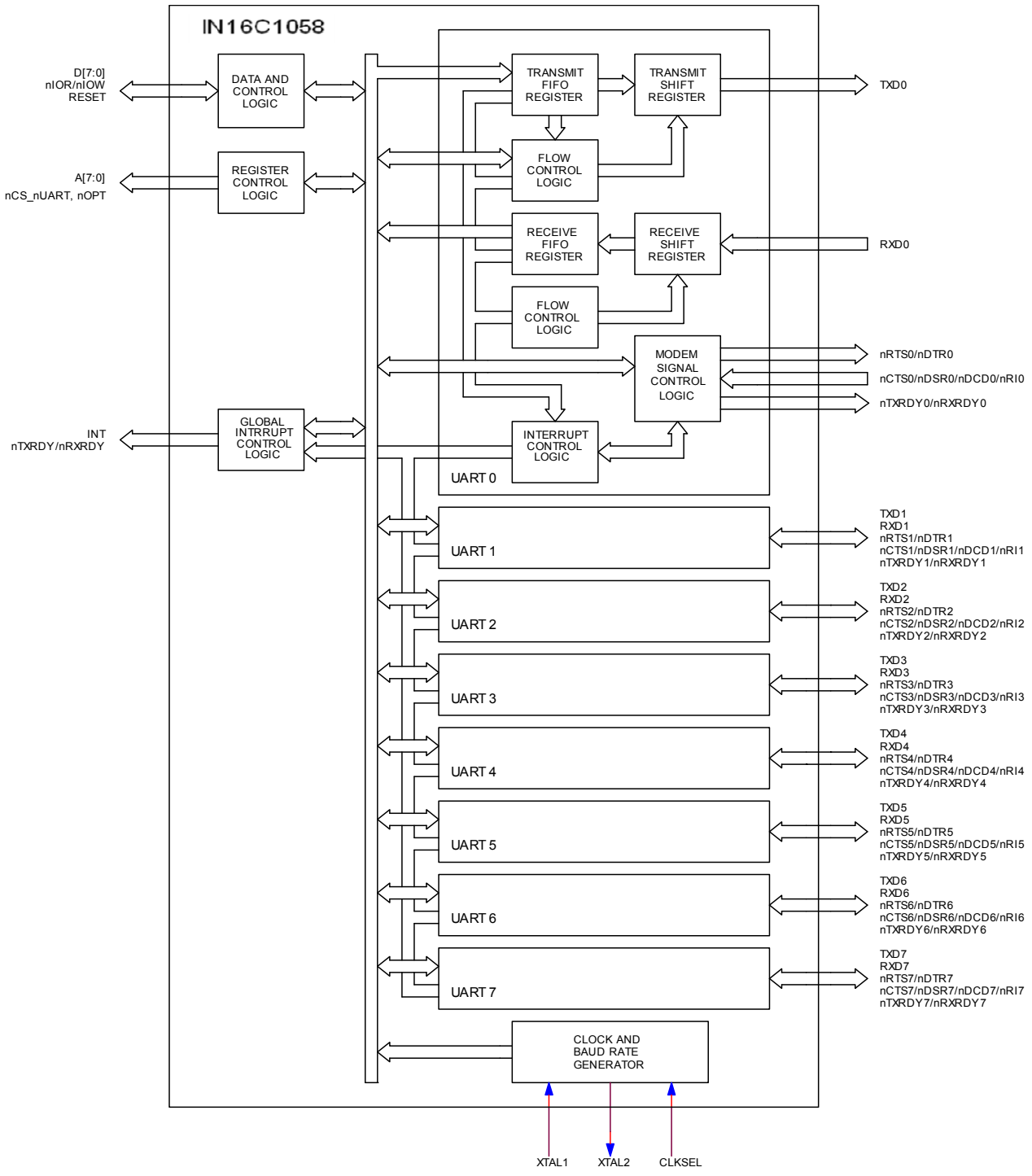


Figure 1: Block Diagram

5. Pin Configuration

5.1 Pin Configuration for 128-Pin TQFP (20x20) Package

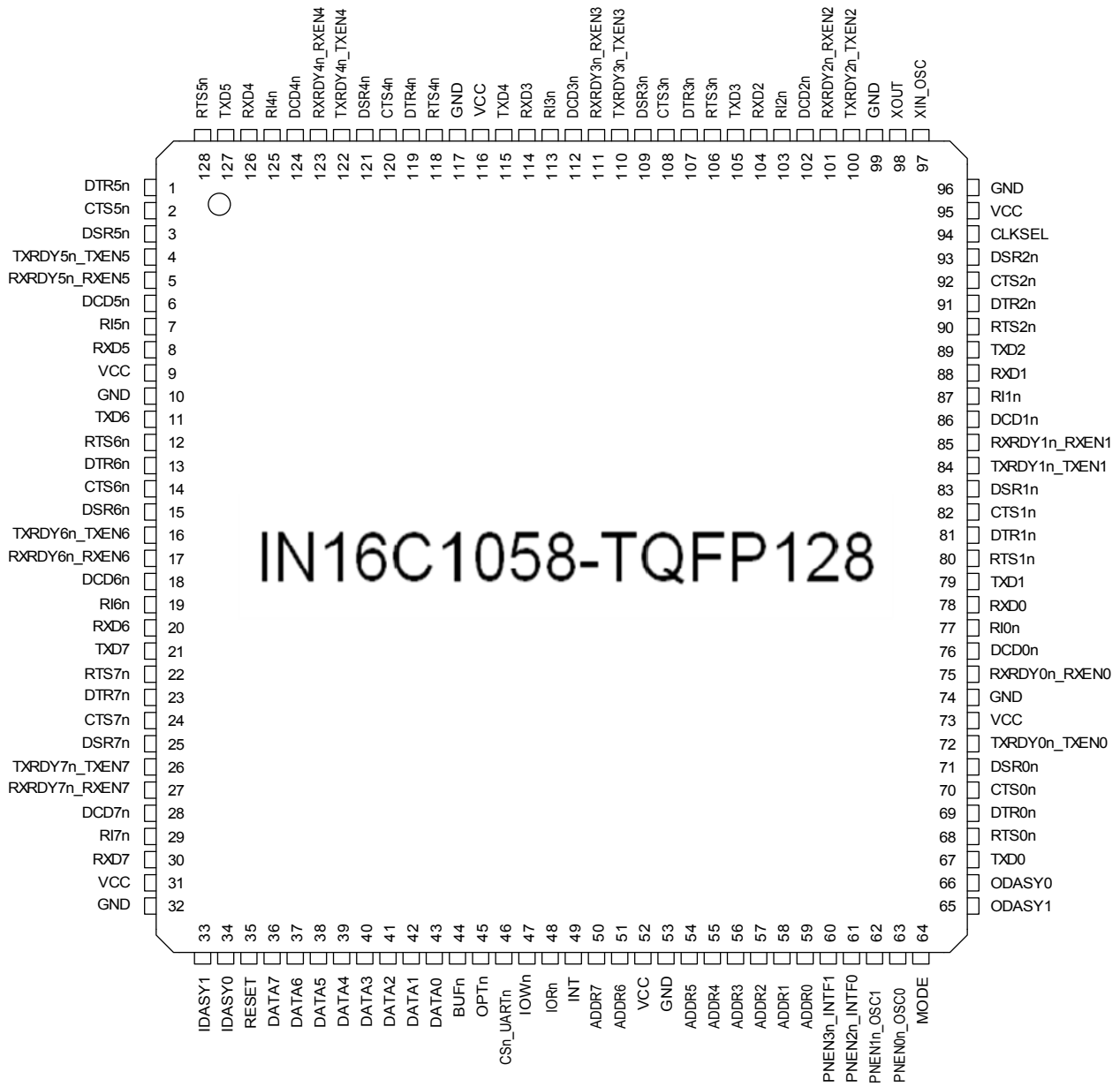


Figure 2: 128-Pin TQFP (20x20) Pin Configuration

5.2 Pin Description

Table 2: Pin Description

Data Bus Interface			
Name	**Pin**	**Type**	**Description**
ADDR0	59	I	Address Bus Lines [7:0].
ADDR1	58	I	Address Bus Lines operates in two modes – Normal mode or MIO mode.
ADDR2	57	I	In the normal mode(MODE=0b), A[5:0] are used and A[7:6] are not used.
ADDR3	56	I	A[5:3] are for the selection of 8 UART channels and A[2:0] are for the
ADDR4	55	I	internal registers of the selected UART channel.
ADDR5	54	I	In the MIO mode(MODE=1b), A[7:0] are all used. A[7:6] are for the
ADDR6	51	I	selection of 4 panels. A[5:0] are same as normal mode.
ADDR7	50	I	
DATA0	43	I/O	Data Bus Lines [7:0].
DATA1	42	I/O	These pins are tri-state data bus for data transfer to or from the controlling
DATA2	41	I/O	CPU.
DATA3	40	I/O	
DATA4	39	I/O	
DATA5	38	I/O	
DATA6	37	I/O	
DATA7	36	I/O	
nIOR	48	I	Read Data (active low strobe). A valid low level on nIOR will load the data of an internal register defined by address lines onto the UART data bus for access by an external CPU.
nIOW	47	I	Write Data (active low strobe). A valid low level on nIOW will transfer the data from external CPU to an internal register that is defined by address lines.
nCS_nUART	46	I	Chip Select (active low). This pin enables data transfers between the external CPU and the UART for the respective channel. In MIO mode, this pin's name is nUART and it does as nCS in normal mode.
nOPT	45	I	Option Select (active low). This pin used in MIO mode only and enables data transfer between the external CPU and internal option registers. If you don't use this pin in normal mode, please pull-up this pin.
nBUF	44	O	Buffer Enable (active low). This pin used in MIO mode only. When IN16C1058 output data to MIO bus, it can control the direction of bus transceivers.
INT	49	O	Interrupt, This pin is a global interrupt for all 8 UART channels. Each internal interrupt, INT0-7 are enabled when MCR[3] is set to '1' and AFR[4] is cleared to '0' (default state). INT's asserted state is determined by AFR[5]. It's asserted state is active high when AFR[5] is set to '1', and active low when AFR[5] is cleared to '0'. The status of the 8 interrupts are shown on IPR(Interrupt Poll Register). The interrupts are masked through IMR(Interrupt Mask Register) and handled.

Table 2: Pin Description…*continued*

Name	Pin	Type	Description
nTXRDY0/TXEN0	72	O	Transmitter Ready/Tx Enable. These pins provide individual channel transmitter ready or transmit enable. nTXRDY0-7 are enabled when ATR[1:0] is cleared to '00' (default state). If ATR[1:0] are set to '11', nTXRDY0-7 operate as TXEN0-7. nTXRDY0-7 (active low) are asserted by TX FIFO/THR status for transmit channels 0-7. TXEN0-7's asserted state is determined by ATR[5:4]. If ATR[4] is cleared to '0', the state holds the same value as ATR[5]. If ATR[4] is set to '1', it is the auto-toggling state based on ATR[5]. If these pins are unused, leave them open.
nTXRDY1/TXEN1	84	O	
nTXRDY2/TXEN2	100	O	
nTXRDY3/TXEN3	110	O	
nTXRDY4/TXEN4	122	O	
nTXRDY5/TXEN5	4	O	
nTXRDY6/TXEN6	16	O	
nTXRDY7/TXEN7	26	O	
nRXRDY0/RXEN0	75	O	Receiver Ready/Rx Enable. These pins provide individual channel receiver ready or receive enable. nRXRDY0-7 are enabled when ATR[1:0] is cleared to '00' (default state). If ATR[1:0] is set to '11', nRXRDY0-7 are changed to RXEN0-7. nRXRDY0-7 (active low) are asserted by RX FIFO/RBR status for receive channels 0-7. RXEN0-7's asserted state is determined by ATR[7:6]. If ATR[6] is cleared to '0', the state holds the same value as ATR[7]. If ATR[6] is set to '1', it is the auto-toggling state based on ATR[7]. If these pins are unused, leave them open.
nRXRDY1/RXEN1	85	O	
nRXRDY2/RXEN2	101	O	
nRXRDY3/RXEN3	111	O	
nRXRDY4/RXEN4	123	O	
nRXRDY5/RXEN5	5	O	
nRXRDY6/RXEN6	17	O	
nRXRDY7/RXEN7	27	O	
Modem and Serial I/O Interface			
Name	**Pin**	**Type**	**Description**
TXD0	67	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.
TXD1	79	O	
TXD2	89	O	
TXD3	105	O	
TXD4	115	O	
TXD5	127	O	
TXD6	11	O	
TXD7	21	O	
RXD0	78	I	Receive Data. These pins are individual receive data input. During the local loop-back mode, the RXD input pin is disabled and RXD data is internally connected to the TXD output.
RXD1	88	I	
RXD2	104	I	
RXD3	114	I	
RXD4	126	I	
RXD5	8	I	
RXD6	20	I	
RXD7	30	I	
nRTS0	68	O	Request to Send (active low). These pins indicate that the UART is ready to send data to the modem, and affect transmit and receive operations only when Auto-RTS function is enabled.
nRTS1	80	O	
nRTS2	90	O	
nRTS3	106	O	
nRTS4	118	O	
nRTS5	128	O	
nRTS6	12	O	
nRTS7	22	O	

Table 2: Pin Description…*continued*

Name	Pin	Type	Description
nCTS0	70	I	Clear to Send (active low). These pins indicate the modem is ready to accept transmitted data from the UART, and affect transmit and receive operations only when Auto-CTS function is enabled.
nCTS1	82	I	
nCTS2	92	I	
nCTS3	108	I	
nCTS4	120	I	
nCTS5	2	I	
nCTS6	14	I	
nCTS7	24	I	
nDTR0	69	O	Data Terminal Ready (active low). These pins indicate UART is ready to transmit or receive data.
nDTR1	81	O	
nDTR2	91	O	
nDTR3	107	O	
nDTR4	119	O	
nDTR5	1	O	
nDTR6	13	O	
nDTR7	23	O	
nDSR0	71	I	Data Set Ready (active low). These pins indicate modem is powered-on and is ready for data exchange with UART.
nDSR1	81	I	
nDSR2	93	I	
nDSR3	109	I	
nDSR4	121	I	
nDSR5	3	I	
nDSR6	15	I	
nDSR7	25	I	
nDCD0	76	I	Carrier Detect (active low). These pins indicate that a carrier has been detected by modem.
nDCD1	86	I	
nDCD2	102	I	
nDCD3	112	I	
nDCD4	124	I	
nDCD5	6	I	
nDCD6	18	I	
nDCD7	28	I	
nRI0	77	I	Ring Indicator (active low). These pins indicate the modem has received a ringing signal from telephone line. A low to high transition on these input pins generates a modem status interrupt, if enabled.
nRI1	87	I	
nRI2	103	I	
nRI3	113	I	
nRI4	125	I	
nRI5	7	I	
nRI6	19	I	
nRI7	29	I	

Table 2: Pin Description…*continued*

Multiport I/O Interfaces			
Name	**Pin**	**Type**	**Description**
IDASY0 IDASY1	34 33	I I	Daisy Chain Input [1:0]. These pins are used only in MIO mode. SystemBase's MIO Bus can be expanded up to 32 serial ports by 8 ports. The ports are managed with Daisy Chain in order for 8 ports' install information to be automatically recognized. These pins are the input of Daisy Chain.
ODASY0 ODASY1	66 65	O O	Daisy Chain Output [1:0]. These pins are used only in MIO mode. SystemBase's MIO Bus can be expanded up to 32 serial ports by 8 ports. The ports are managed with Daisy Chain in order for 8 ports' install information to be automatically recognized. These pins are the output of Daisy Chain.
nPNEN0_OSC0 nPNEN1_OSC1	63 62	I/O	These pin are dual mode pins. After power is supplied to the chip, the pin is set to input mode for a while and receive OSC[1:0] input. After that, the pins are set to output mode and outputs nPNEN[1:0] values. The pins are only used in MIO mode. In input mode, OSC[1:0] values mean OSC[1:0] = 00b : Use 1.8432MHz UART Clock. OSC[1:0] = 01b : Use 3.6864MHz UART Clock. OSC[1:0] = 10b : Use 7.3728MHz UART Clock. OSC[1:0] = 11b : Use 14.7456MHz UART Clock. In output mode, nPNEN[1:0] values mean nPNEN[3:0] = 0000b : No serial port on MIO Bus. nPNEN[3:0] = 0001b : 8 serial ports used on MIO Bus. nPNEN[3:0] = 0011b : 16 serial ports used on MIO Bus. nPNEN[3:0] = 0111b : 24 serial ports used on MIO Bus. nPNEN[3:0] = 1111b : 32 serial ports used on MIO Bus.
nPNEN2_INTF0 nPNEN3_INTF1	61 60	I/O	These pin are dual mode pins. After power is supplied to the chip, the pin is set to input mode for a while and receive INTF[1:0] input. After that, the pins are set to output mode and outputs nPNEN[3:2] values. The pins are only used in MIO mode. In input mode, OSC[1:0] values mean INTF[1:0] = 00b : Use RS232 Interface INTF[1:0] = 01b : Use RS422 Interface INTF[1:0] = 10b : Use RS485 Interface INTF[1:0] = 11b : Use Unknown Interface In output mode, nPNEN[1:0] values mean nPNEN[3:0] = 0000b : No serial port on MIO Bus. nPNEN[3:0] = 0001b : 8 serial ports used on MIO Bus. nPNEN[3:0] = 0011b : 16 serial ports used on MIO Bus. nPNEN[3:0] = 0111b : 24 serial ports used on MIO Bus. nPNEN[3:0] = 1111b : 32 serial ports used on MIO Bus.

Table 2: Pin Description…*continued*

Multiport I/O Interfaces			
Name	**Pin**	**Type**	**Description**
MODE	64	I	UART Mode Input MODE = 0b : Normal UART mode MODE = 1b : MIO UART mode
Other Interfaces			
Name	**Pin**	**Type**	**Description**
XIN_OSC	97	I	Crystal or External Clock Input.
XOUT	98	O	Crystal or Buffered Clock Output.
CLKSEL	94	I	Clock Select. This pin selects the divide-by-1 or divide-by-4 prescalable clock. During the reset, The high on CLKSEL selects the divide-by-1 prescaler. The low on CLK selects the divide-by-4 prescaler. The inverting value of CLKSEL is latched into MCR[7] at the trailing edge of RESET.
RESET	35	I	Reset (active high). This pin will reset the internal registers and all the outputs.
VCC	9 31 52 73 95 116	I	Power Supply Input. 3.3V (2.7V ~ 3.6V)
GND	10 32 53 74 96 99 117	I	Signal and Power Ground.

6. Functional Description

IN16C1058 UART supports Normal mode in which the chip operates as other common Octal-UARTs and MIO mode which supports SystemBase's MIO mode. The mode can be selected by MODE input. Furthermore, the UART supports 256-byte FIFO which enhances system performance and prevents Overrun Errors in multiple serial communication system. When FIFO is enabled, it has a register configuration compatible with 64-byte FIFO and 16C654, so it becomes compatible with 16C654. If you enable 256-byte FIFO, you use the unique supreme function that IN16C1058 offers. It offers communication speed up to 5.3Mbps and more enhanced functions that other UARTs with 128-byte FIFO do not.

IN16C1058 can select hardware/software flow control. Hardware flow control significantly reduces software overhead and increases system efficiency by automatically controlling serial data flow using the nRTS output and nCTS input signals. Software flow control automatically controls data flow by using programmable Xon/Xoff characters. In addition, IN16C1058 has to control communication bus in RS422/485 communication in order to have a stable data communication. By automatically controlling this on hardware level , IN16C1058 allows users to use the chip more easily.

6.1 Normal mode and MIO mode

IN16C1058 can be configured as Normal mode or MIO mode depending on the MODE input. In a Normal application, the device operates in Normal mode with MODE = 0b and operates in MIO mode with MODE = 1b.

When IN16C1058 is operating in Normal mode, only ADDR[5:0], DATA[7:0], nIOR, nIOW and nCS signals are used. The internal Options Registers structure, controls pins for MIO Bus cannot be used and only internal UART Registers are accessible.

In Normal mode, the device operates just as other Octal-UARTs that can be found elsewhere. Some pins are not used in this mode and the pins are listed below in 'Table 3: Pin Usage between Normal mode and MIO mode'.

MIO mode is a structure designed by SystemBase in which UART region and Option region can be accessed through MIO(MultiPort I/O) Bus and I/O Bus. Refer to table 'Table 3' to see its usage. Each channel's Internal Registers for Octal-UART can be accessed through UART region. SystemBase has designed Option Registers to control serial communication and this can be accessed through Option region.

With the use of MIO Bus structure and Option Registers, up to 32 ports can be expanded by 8 ports.

Table 3: Pin Usage between Normal mode and MIO mode

PIN Name	Normal mode	MIO mode	Description
ADDR0	ADDR0	ADDR0	Normal mode (nCS) ADDR[2:0] : UART registers selection ADDR[5:3] : 8 UARTs selection MIO mode (nUART) ADDR[5:0] : same as the normal mode ADDR[7:6] : 4 Panels selection MIO mode (nOPT) ADDR[5:0] : option registers selection
ADDR1	ADDR1	ADDR1	
ADDR2	ADDR2	ADDR2	
ADDR3	ADDR3	ADDR3	
ADDR4	ADDR4	ADDR4	
ADDR5	ADDR5	ADDR5	
ADDR6	Not Used	ADDR6	
ADDR7	Not Used	ADDR7	
DATA[7:0]	DATA[7:0]	DATA[7:0]	
nIOR	nIOR	nIOR	
nIOW	nIOW	nIOW	
nCS_nUART	nCS	nUART	
nOPT	Not Used	nOPT	Option Register Access Enable
nBUF	Not Used	nBUF	MIO Bus Read Enable for 245 Buffer
IDASY[1:0]	Not Used	IDASY[1:0]	Input Daisy Chain for 8-port Unit Block
ODASY[1:0]	Not Used	ODASY[1:0]	Output Daisy Chain for 8-port Unit Block
nPNEN0_OSC0	Not Used	nPNEN0_OSC0	Input @ Initial Time : OSC[1:0], INTF[1:0] Output @ Normal : nPNEN[3:0]
nPNEN1_OSC1	Not Used	nPNEN1_OSC1	
nPNEN2_INTF0	Not Used	nPNEN2_INTF0	
nPNEN3_INTF1	Not Used	nPNEN3_INTF1	

6.2 MIO mode

IN16C1058 operates in MIO mode with MODE = 1b.

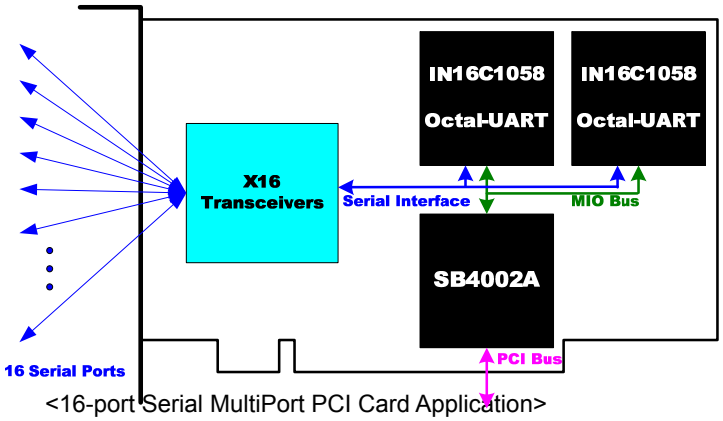
IN16C1058 contains built-in Control Logics that allow expansion of up to 32 ports (by 8 ports) by adding MIO Bus Interface and Option Registers structure on a common Octal-UART. Through this method, glue-logics are unnecessary when expanding ports. With SystemBase's PCI Bridge Controller SB4002A, 8, 16, 24 and 32 port serial communication PCI Card Adapter application can be made easily with low cost.

In MIO mode, ADDR[7:0] 8-bits are used. ADDR[7:6] are used to select one of 4 Octal UARTS, ADDR[5:3] are used to select one of the 8 UART channels and ADDR[2:0] are used to select UART channel's internal registers. To manage the install information of Octal UARTs that are managed by 8 ports on MIO Bus, Daisy Chain is used and the connections are IDASY[1:0] and ODASY[1:0]. nPNEN0_OSC0, nPNEN_OSC1, nPNEN_INTF0 and nPNEN_INTF1 are used for management of Oscillator and Interface

configuration and panels.

16-port serial card and expandable 32-port serial card can be made. 16-port serial card is designed to expand 16 serial ports with two IN16C1058 Octal-UARTs connected to SB4002A by MIO Bus as shown in below diagram. 32-port serial card is designed as shown in below diagram. It is composed of one SB4002A and is capable of expanding by 8 ports using 8-port panels.

When serial communication port is expanded over 8 ports, there is higher chance of the FIFO Buffer getting full in the UART as the result of PCI Bus' performance problem and as data is overwritten, overrun errors are more likely to occur. Since IN16C1058 uses 256-Byte FIFO, overrun errors and the loss of data can be prevented. Therefore, IN16C1058 can be considered the optimum solution for serial cards with multiple ports.



<16-port Serial MultiPort PCI Card Application>

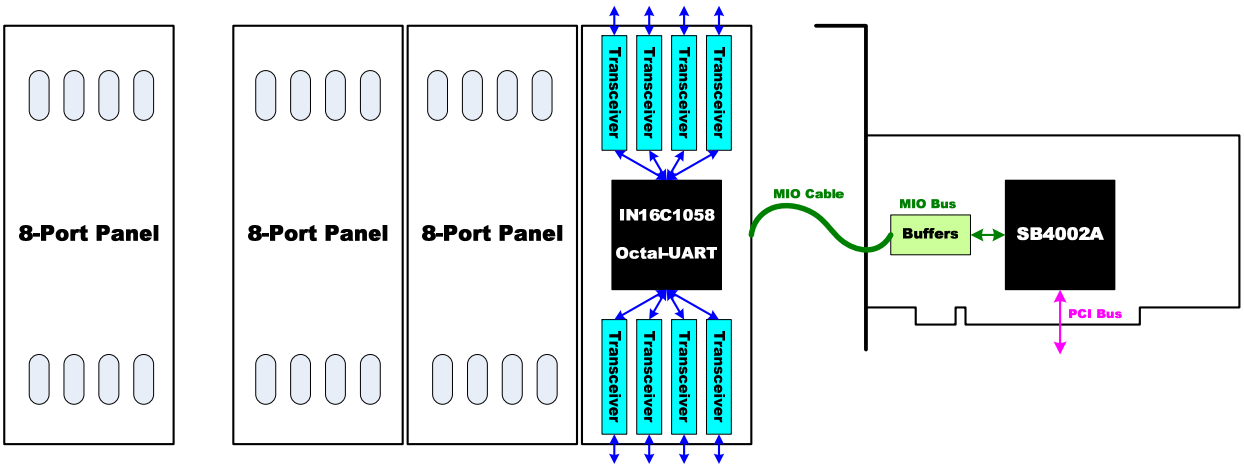


Figure 3: Expandable 32-port Serial MultiPort PCI Card Application

6.3 FIFO Operation

IN16C1058's FIFO has two modes, 64-byte FIFO mode and 256-byte FIFO mode. Setting FCR[0] to '1' enables FIFO, and if AFR[0] is set to '0', it operates in 64-byte FIFO mode(default). In this mode, Transmit Data FIFO, Receive Data and Receive Status FIFO are 64 bytes. 64-byte FIFO mode allows you to select the Transmit Interrupt Trigger Level from 8, 16, 32, or 56. You can verify this Interrupt Trigger Level by TTR and RTR. In this mode TTR and RTR are Read Only.
And by FCR[5:4], XOFF Trigger Level can be selected to either 8, 16, 56, or 60, and XON Trigger Level to either 0, 8, 16, or 56 by FCR[7:6]. You can verify XON and XOFF Trigger Level by FUR and FLR. In 64-byte FIFO mode TTR and RTR are Read Only.

If you select 256-byte FIFO mode, you can experience more powerful features of IN16C1058. Setting both FCR[0] and AFR[0] to '1' will enable this mode. In this mode, Transmit Data FIFO, Receive Data and Receive Status FIFO are 256 bytes. Interrupt Trigger Level and XON, XOFF Trigger Level are controlled by TTR, RTR, FUR and FLR, not by FCR[7:4]. That is, TTR, RTR, FUR and FLR can both read and write. You can verify free space of Transmit FIFO and the number of characters received in Receive FIFO by TCR, RCR and ISR[7:6].
While TX FIFO is full, the value sent to THR by CPU disappears. And while RX FIFO is full, the data coming from external devices disappear as well, provided that flow control function is not used.
For more information, refer to Register Description.

6.4 Hardware Flow Control

Hardware flow control is executed by Auto-RTS and Auto-CTS. Auto-RTS and Auto-CTS can be enabled/disabled independently by programming EFR[7:6]. If Auto-RTS is enabled, it reports that it cannot receive more data by asserting 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 '0', transmitting data to TX FIFO has to be suspended because external device has reported that it cannot accept more data. When data transmission has been suspended and nCTS is set to '1', data in TX FIFO is retransmitted because external device has reported that it can accept more data. These operations prevent overrun during communication and if hardware flow control is disabled and transmit data rate exceeds RX FIFO service latency, overrun error occurs.

6.4.1 Auto-RTS

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

changed to '1' until the amount in RX FIFO exceeds the value set in FUR. But if NRTS is '1' now, NRTS is not changed to '0' until the amount in RX FIFO equals the value written in FLR.

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

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

In Figure 4, 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.

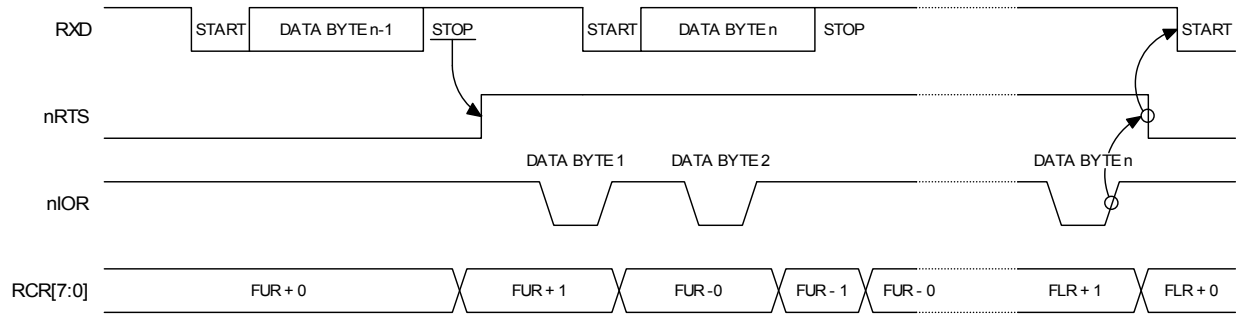


Figure 4: NRTS Functional Timing

6.4.2 Auto-CTS

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

generated by Auto-CTS are removed if MSR is read.

6.5 Software Flow Control

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

Table 4: Software flow control options (EFR[3:0])

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

6.5.1 Transmit Software Flow Control

To make Transmit Software Flow Control enabled, EFR[3:2] must be set to '01', '10' or '11'. Unlike Auto-RTS in which '0' is outputted on 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 '00', '01', '10', and '11', FUR is 8, 16, 56, and 60, respectively. And if FCR[5:4] is '00', '01', '10', and '11', FLR is 0, 8, 16, and 56, respectively in 64-byte FIFO. In 256-byte FIFO mode, users can input values in FUR and FLR as they want and use them. But the value in FUR must be larger than that of FLR. While TX software flow control is active, its status (if Xon or Xoff) can be verified by FSR[4]. If FSR[4] is '0', the status is Xon and if '1', the status is Xoff. It can be verified by FSR[4] only. And for there is no condition to generate interrupt, interrupt doesn't occur. It is different from that interrupt is generated by IER[5] when RX software flow control is enabled.

6.5.2 Receive Software Flow Control

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

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

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

character.

- If Xon and Xoff character are set to same, both characters are treated as Xon character.

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

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

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

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

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

  In all these cases no XON/XOFF characters are treated as data.
  Refer to Table 5 below.

Table 5: Xon/Xoff Character Recognition Logic Table

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

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

If Xon, Xoff characters are used as one character,

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

If Xon, Xoff characters are used as two characters,

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

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

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

6.5.3 Xon Any Function

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

Details about it are described in 6.3.2 Receive Software Flow Control.

6.5.4 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 IN16C1058 does with Xoff Re-transmit function.

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

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

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

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

6.6 Interrupts

As there are eight independent channel UARTs in IN16C1058, so there are eight internal interrupts. Interrupts are assigned internal interrupts: INT0, INT1, INT2, INT3, INT4, INT5, INT6 and INT7 for each channel. Each interrupt has six prioritized level's interrupt generation capability. The IER enables each of the six types of interrupts and INT signal in response to an interrupt generation. When an interrupt is generated, the ISR indicates that an interrupt is pending and provides the type of interrupt. And IN16C1058 can handle for eight interrupts with one global interrupt. Global interrupt treats eight of each interrupt as one interrupt, so it is useful when external system has few interrupt resource. GICR determines whether global interrupt occurs or not. While GICR[x] is set to '1', an interrupt that is generated in one of eight channel UARTs and treated as UNMASK is transmitted to GINT. But if GICR[x] is cleared to '0', an interrupt is not transmitted to GINT though interrupts are generated in one of eight channel UARTs and treated as MASK. So this interrupt is not transmitted to external CPU. GISR is the status of each channel UART. It just show the status of eight channels whether interrupt is generated or not. If GISR[0] is cleared to '0', it means that interrupt is not generated in the UART and if set to '1', it means that interrupt is generated. The value of GISR[0] shows the status of interrupt generated in the UART.

Each internal interrupt is decided by the value of 'GICR x GISR'. In other words, when the both of them have logic '1', internal interrupt of the channel is generated. And the global interrupt is decided by logic AND for each internal interrupt. If one of eight internal interrupt is generated, the global interrupt is generated.

6.7 DMA Operation

Transmitter and Receiver DMA operation is available through nTXRDY, nRXRDY, nTXRDY[7:0], and nRXRDY[7:0]. There are two modes of DMA operation, DMA mode 0 or DMA mode 1, selected by FCR[3].

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

6.7.1 Single DMA transfer (DMA Mode 0/FIFO Disable)

Transmitter: There are no character in TX FIFO or THR. And the nTXRDY[7:0] signals will be in assert state. nTXRDY[7:0] will switch to deassert state after one character is loaded into TX FIFO or THR.

Receiver: There is at least one character in RX FIFO or RHR. And the nRXRDY[7:0] signals will be in assert state. Once nRXRDY is asserted, nRXRDY[7:0] signal will switch to deassert state when there are no more characters in RX FIFO or RBR.

Figure 5 shows nTXRDY, nTXRDY[7:0], nRXRDY, and nRXRDY[7:0] in DMA mode 0/FIFO disable.

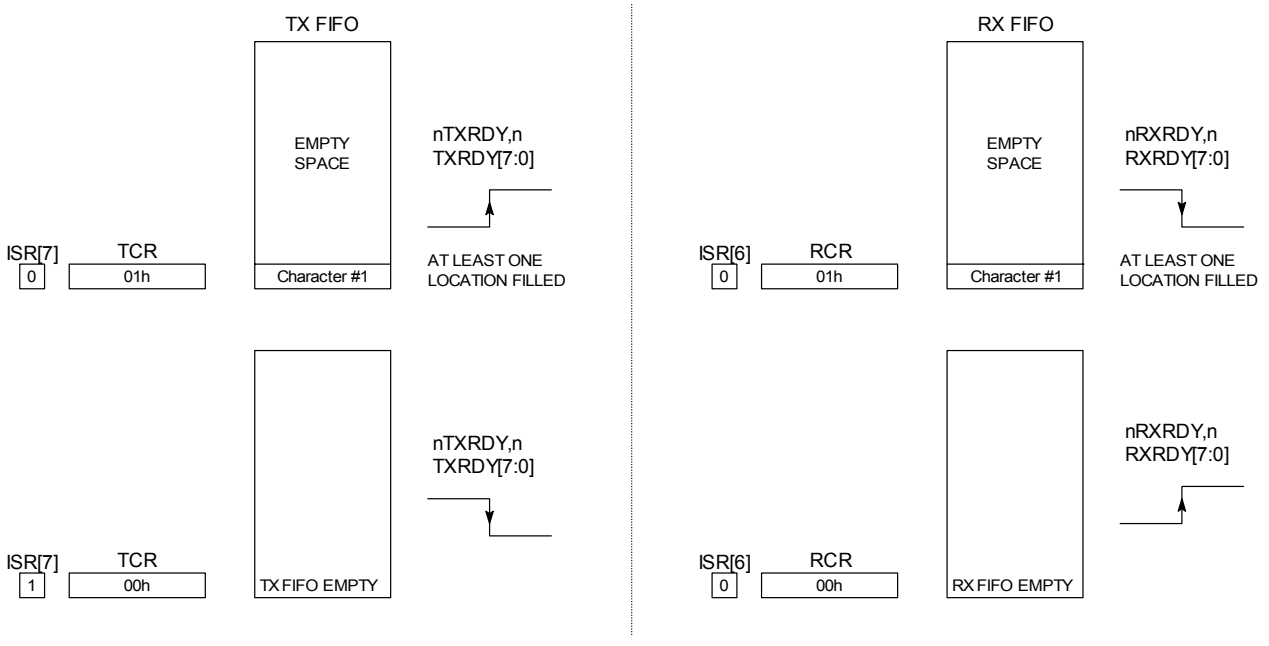


Figure 5: nTXRDY/nTXRDY[7:0] and nRXRDY/nRXRDY[7:0] in DMA mode 0/FIFO disable.

6.7.2 Block DMA transfer (DMA Mode 1)

Transmitter: When the characters in TX FIFO are less than the trigger level that is set in TTR, nTXRDY or nTXRDY[7:0] signal is asserted. When TX FIFO is full, nTXRDY or nTXRDY[7:0] signal is deasserted.

Receiver: When the characters in RX FIFO are more than the trigger level that is set in RTR, nRXRDY or nRXRDY[7:0] signal is asserted. When RX FIFO is empty, RXRDY or RXRDY[7:0] signal is deasserted.

The figure 6 below shows nTXRDY, nTXRDY[7:0] and nRXRDY, nRXRDY[7:0] in DMA mode 1.

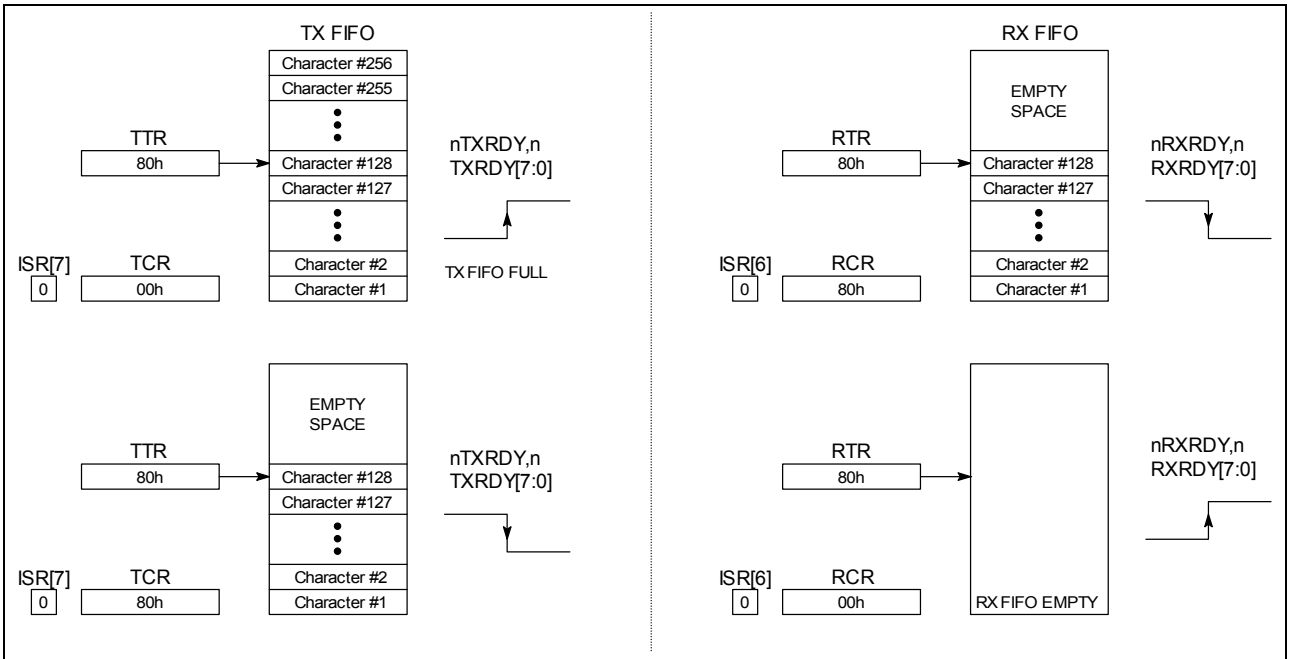


Figure 6: nTXRDY/nTXRDY[7:0] and nRXRDY/nRXRDY[7:0] in DMA mode 1.

6.8 Sleep Mode with Auto Wake-Up

The IN16C1058 provides sleep mode operation to reduce its power consumption when sleep mode is activated. Sleep mode is enabled when EFR[4] and IER[4] are set to '1'.

Sleep mode is activated when:

- RXD input is in idle state.
- 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 IN16C1058 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.

6.9 Programmable Baud Rate Generator

The IN16C1058 has a programmable baud rate generator with a prescaler. The prescaler is controlled by MCR[7], as shown in Figure 7. The MCR[7] sets the prescaler to divide the clock frequency by 1 or 4. And the baud rate generator further divides this clock frequency by a programmable divisor (DLL and DLM) between 1 and (2^{16} – 1) to obtain a 16X sampling rate clock of the serial data rate. The sampling rate clock is used by transmitter for data bit shifting and receiver for data sampling.

The divisor of the baud rate generator is:

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

MCR[7] is cleared to '0' (prescaler = 1), when CLKSEL input is in high state after reset.
MCR[7] is set to '1' (prescaler = 4), when CLKSEL input is in low state after reset.

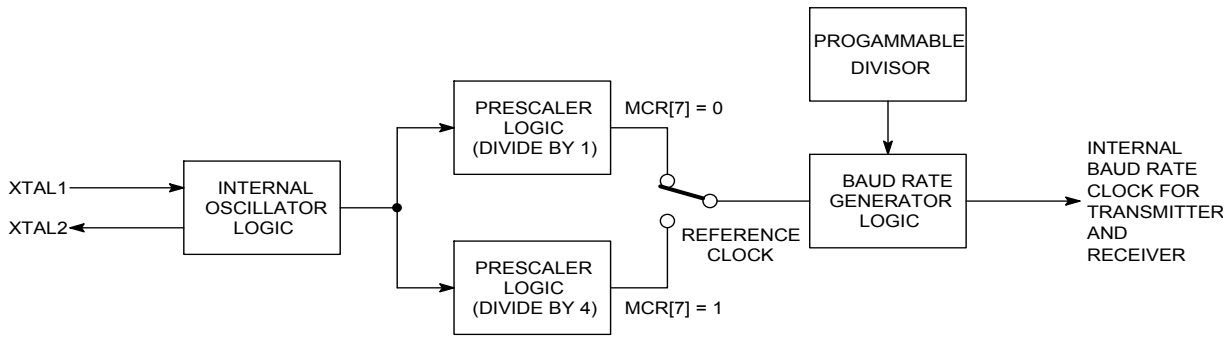


Figure 7: Prescaler and Baud Rate Generator Block Diagram

DLL and DLM must be written to in order to program the baud rate. DLL and DLM are the least and most significant byte of the baud rate divisor, respectively. If DLL and DLM are both zero, the IN16C1058 is effectively disabled, as no baud clock will be generated.

Table 6 shows the baud rate and divisor value for prescaler with divide by 1 as well as crystal with frequency 1.8432MHz, 3.6864MHz, 7.3728MHz, and 14.7456MHz, respectively.

Figure 8 shows the crystal clock circuit reference.

Table 6: Baud Rates

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

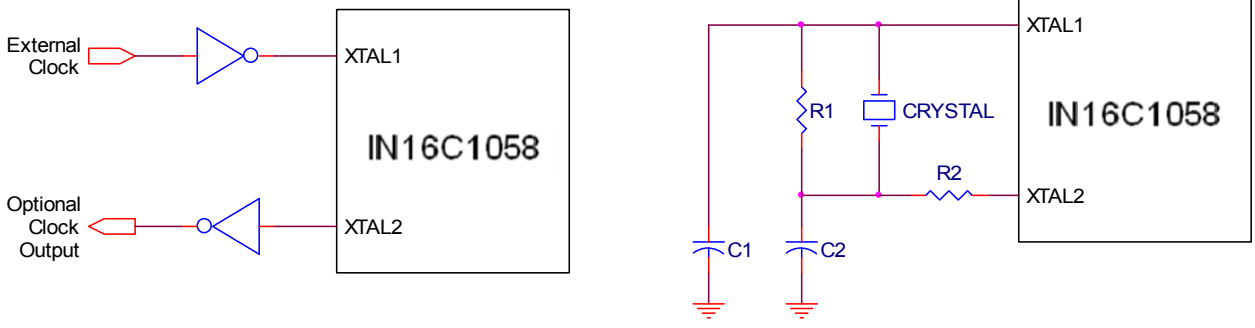


Figure 8: Crystal Clock Circuit Diagram

Table 7: Component Values

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

6.9 Break and Time-out Conditions

Break Condition:

Break Condition is occurred when TXD signal outputs '0' and sustains for more than one character.

It is occurred if LCR[6] is set to '1' and deleted if '0'. If break condition is occurred when normal data are transmitted on TXD, break signal is transmitted and internal serial data are also transmitted, but they are not outputted to external TXD pin. When Break condition is deleted, then they are transmitted to TXD pin.

Figure 9 below shows the Break Condition Block Diagram.

Time-out Condition:

When serial data is received from external UART, characters are stored in RX FIFO. When the number of characters in RX FIFO reaches the trigger level, interrupt is generated for the CPU to treat characters in RX FIFO. But when the number of characters in RX FIFO does not reach the trigger level and no more data arrives from external device, interrupt is not generated and therefore CPU cannot recognize it. IN16C1058 offers time-out function for this situation. Time-out function generates an interrupt and reports to CPU when the number of RX FIFO is less than trigger level and no more data receives for four character time.

Time-out interrupt is enabled when IER[2] is set to '1' and can be verified by ISR.

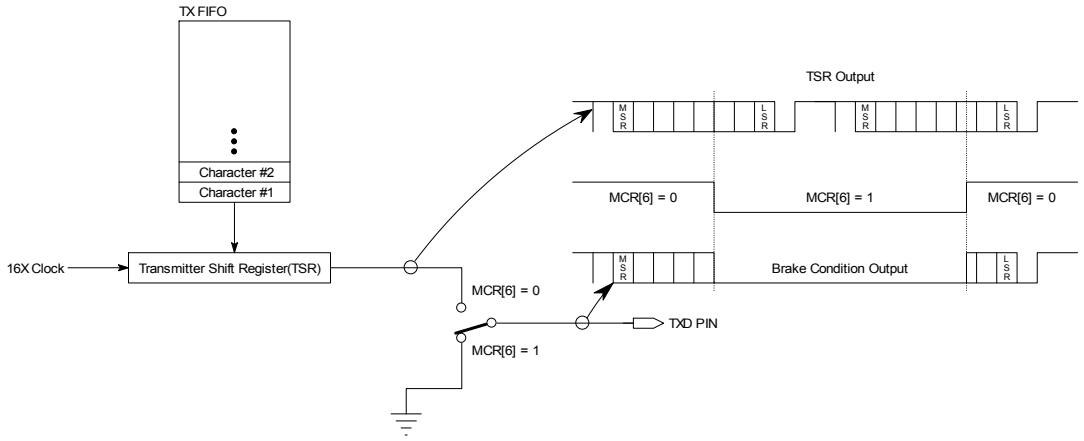


Figure 9: Break Condition Block Diagram

7. UART Register Descriptions

Each UART channel in the IN16C1058 has its own set of registers selected by address lines A2, A1, and A0 with a specific channel selected. The complete register set is shown on Table 8 and Table 9.

Table 8: Internal Registers Map

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

Table 9: Internal Registers Map…*continued*

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

Table 9: Internal Registers Map…*continued*

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

Table 9: Internal Registers Description

Addr. A[2:0]	Reg.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Page 0 Registers									
0h	THR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0h	RBR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1h	IER	0/NCTS Interrupt Enable	0/NRTS Interrupt Enable	0/Xoff Interrupt Enable	0/Sleep Mode Enable	Modem Status Interrupt Enable	Receive Line Status Interrupt Enable	THR Empty Interrupt Enable	Receive Data Available Interrupt Enable
2h	ISR	FCR[0]/ 256-TX FIFO Empty	FCR[0]/ 256-RX FIFO Full	Interrupt Priority Bit 5	Interrupt Priority Bit 4	Interrupt Priority Bit 3	Interrupt Priority Bit 2	Interrupt Priority Bit 1	Interrupt Priority Bit 0
2h	FCR	RX Trigger Level (MSB)	RX Trigger Level (LSB)	0/TX Trigger Level (MSB)	0/TX Trigger Level (LSB)	DMA Mode Select	TX FIFO Reset	RX FIFO Reset	FIFO Enable
3h	LCR	Divisor Enable	Set TX Brake	Set Parity	Parity Type Select	Parity Enable	Stop Bits	Word Length Bit 1	Word Length Bit 0
4h	MCR	Clock Select	Page 2 Select/Xoff Re-Transmit Access Enable	0/Xon Any	0/Loop Back	OUT2/ INTx Enable	OUT1/ Xoff Re-Transmit Enable	NRTS	NDTR
5h	LSR	RX FIFO Data Error	THR & TSR Empty	THR Empty	Receive Break	Framing Error	Parity Error	Overrun Error	Receive Data Ready
6h	MSR	NDCD	NRI	NDSR	NCTS	ΔNDCD	ΔNRI	ΔNDSR	ΔNCTS
7h	SCR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Page 1 Registers									
0h	DLL	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
1h	DLM	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
Page 2 Registers									
1h	GICR	CH7 Interrupt Mask	CH6 Interrupt Mask	CH5 Interrupt Mask	CH4 Interrupt Mask	CH3 Interrupt Mask	CH2 Interrupt Mask	CH1 Interrupt Mask	CH0 Interrupt Mask
2h	GISR	CH 7 Interrupt Status	CH 6 Interrupt Status	CH 5 Interrupt Status	CH 4 Interrupt Status	CH 3 Interrupt Status	CH 2 Interrupt Status	CH 1 Interrupt Status	CH 0 Interrupt Status
5h	TCR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
6h	RCR	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
7h	FSR	0	0	TX HW Flow Control Status	TX SW Flow Control Status	0	0	RX HW Flow Control Status	RX SW Flow Control Status

Table 9: Internal Registers Description…*continued*

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

7.1 Transmit Holding Register (THR, Page 0)

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

7.2 Receive Buffer Register (RBR, Page 0)

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

7.3 Interrupt Enable Register (IER, Page 0)

IER enables each of the seven types of Interrupt, namely receive data ready, transmit empty, line status, modem status, Xoff received, 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. Interrupt is enabled by setting appropriate bits. Table 10 shows IER bit settings.

Table 10: Interrupt Enable Register Description

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

7.4 Interrupt Status Register (ISR, Page 0)

The UART provides multiple levels of prioritized interrupts to minimize software work load. ISR provides the source of interrupt in a prioritized manner.
Table 11 shows ISR[7:0] bit settings.

Table 11: Interrupt Status Register Description

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

Table 11: Interrupt Status Register Description…*continued*

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

7.5 FIFO Control Register (FCR, Page 0)

FCR is used for enabling the FIFOs, clearing the FIFOs, setting transmit/receive FIFO trigger level, and selecting the DMA modes. Table 12 shows FCR bit settings.

Table 12: FIFO Control Register Description

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

7.6 Line Control Register (LCR, Page 0)

LCR controls the asynchronous data communication format. The word length, the number of stop bits, and the parity type are selected by writing the appropriate bits to the LCR. Table 13 shows LCR bit settings.

Table 13: Line Control Register Description

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

7.7 Modem Control Register (MCR, Page 0)

MCR controls the interface with the modem, data set, or peripheral device that is emulating the modem. Table 14 shows MCR bit settings.

Table 14: Modem Control Register Description

Bit	Symbol	Description
7	MCR[7]	Clock Prescaler Select. 0 : Divide by 1 clock input (default). 1 : Divide by 4 clock input.
6	MCR[6]	Page 2 Select/Xoff Re-Transmit Access Enable 0 : Enable access to page 0 register when LCR[7] is '0' (default). 1 : Enable access to page 2 register and Xoff re-transmit bit when LCR[7] is '0'.
5	MCR[5]	Xon Any Enable. 0 : Disable Xon any (default). 1 : Enable Xon any.
4	MCR[4]	Internal Loop Back Enable. 0 : Disable loop back mode (default). 1 : Enable internal loop back mode. In this mode the MCR[3:0] signals are looped back into MSR[7:4] and TXD output is looped back to RXD input internally.
3	MCR[3]	OUT2/Interrupt Output Enable. 0 : INTx outputs disabled (default). During loop back mode, OUT2 output '0' and it controls MSR[7] to '1'. 1 : INTx outputs enabled. During loop back mode, OUT2 output '1' and it controls MSR[7] to '0'. OUT2 is not available as an output pin on the IN16C1058.
2	MCR[2]	OUT1/Xoff Re-transmit Enable. 0 : Xoff re-transmit disable when MCR[6] is '0'. During loop back mode, OUT1 output to '0' and it controls MSR[6] to '1'. 1 : Xoff re-transmit enable when MCR[6] is '1'. During loop back mode, OUT1 output to '1' and it controls MSR[6] to '0'. OUT1 is not available as an output pin on the IN16C1058. Xoff re-transmit is operated with XRCR, refer to XRCR.
1	MCR[1]	NRTS Output. 0 : Force NRTS output to '1'. During loop back mode, controls MSR[4] to '1'. 1 : Force NRTS output to '0'. During loop back mode, controls MSR[4] to '0'.
0	MCR[0]	NDTR Output. 0 : Force NDTR output to '1'. During loop back mode, controls MSR[5] to '1'. 1 : Force NDTR output to '0'. During loop back mode, controls MSR[5] to '0'.

7.8 Line Status Register (LSR, Page 0)

LSR provides the status of data transfers between the UART and the CPU. When LSR is read, LSR[4:2] reflect the error bits (BI, FE, PE) of the character at the top of the RX FIFO. The errors in a character are identified by reading LSR and then reading RBR. Reading LSR does not cause an increment of the RX FIFO read pointer. The RX FIFO read pointer is incremented by reading the RBR. Table 15 shows LSR bit settings.

Table 15: Line Status Register Description

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

7.9 Modem Status Register (MSR, Page 0)

MSR provides the current status of control signals from modem or auxiliary devices. MSR[3:0] are set to '1' when input from modem changes and cleared to '0' as soon as CPU reads MSR. Table 16 shows MSR bit settings.

Table 16: Modem Status Register Description

Bit	Symbol	Description
7	MSR[7]	nDCD 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]	nRI Input Status. Complement of Ring Indicator (nRI) input. In loop back mode this bit is equivalent to OUT1 in the MCR.
5	MSR[5]	nDSR 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]	nCTS 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]	Delta nDCD Input Status. 0 : No change on nDCD input (default). 1 : Indicates that the nDCD input state has changed.
2	MSR[2]	Delta nRI Input Status. 0 : No change on nRI input (default). 1 : Indicates that the nRI input state changed from '0' to '1'.
1	MSR[1]	Delta nDSR Input Status. 0 : No change on nDSR input (deault). 1 : Indicates that the nDSR input state has changed.
0	MSR[0]	Delta nCTS Input Status. 0 : No change on nCTS input (deault). 1 : Indicates that the nCTS input state has changed.

7.10 Scratch Pad Register (SPR, Page 0)

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

7.11 Divisor Latches (DLL, DLM, Page 1)

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

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

7.12 Global Interrupt Control Register (GICR, Page 2)

GICR is a register that internal eight 16C1050 UARTs share to use. It is used when determining whether each interrupt generated at eight 16C1050 UARTs are transmitted to global interrupts or not. Table 17 shows the GICR bit settings.

Table 17: Global Interrupt Control Register Description

Bit	Symbol	Description
7	GICR[7]	Interrupt Mask for 8th UART channel 0 : Interrupt Masking. Global interrupt is not generated even when the value of GISR[7] is '1'. 1 : Interrupt Non-masking. Global interrupt is generated when the value of GISR[7] is '1'.
6	GICR[6]	Interrupt Mask for 7th UART channel 0 : Interrupt Masking. Global interrupt is not generated even when the value of GISR[6] is '1'. 1 : Interrupt Non-masking. Global interrupt is generated when the value of GISR[6] is '1'.
5	GICR[5]	Interrupt Mask for 6th UART channel 0 : Interrupt Masking. Global interrupt is not generated even when the value of GISR[5] is '1'. 1 : Interrupt Non-masking. Global interrupt is generated when the value of GISR[5] is '1'.
4	GICR[4]	Interrupt Mask for 5th UART channel 0 : Interrupt Masking. Global interrupt is not generated even when the value of GISR[4] is '1'. 1 : Interrupt Non-masking. Global interrupt is generated when the value of GISR[4] is '1'.
3	GICR[3]	Interrupt Mask for 4th UART channel 0 : Interrupt Masking. Global interrupt is not generated even when the value of GISR[3] is '1'. 1 : Interrupt Non-masking. Global interrupt is generated when the value of GISR[3] is '1'.
2	GICR[2]	Interrupt Mask for 3rd UART channel 0 : Interrupt Masking. Global interrupt is not generated even when the value of GISR[2] is '1'. 1 : Interrupt Non-masking. Global interrupt is generated when the value of GISR[2] is '1'.
1	GICR[1]	Interrupt Mask for 2nd UART channel 0 : Interrupt Masking. Global interrupt is not generated even when the value of GISR[1] is '1'. 1 : Interrupt Non-masking. Global interrupt is generated when the value of GISR[1] is '1'.
0	GICR[0]	Interrupt Mask for 1st UART channel 0 : Interrupt Masking. Global interrupt is not generated even when the value of GISR[0] is '1'.

	1 : Interrupt Non-masking. Global interrupt is generated when the value of GISR[0] is '1'.

7.13 Global Interrupt Status Register (GISR, Page 2)

GISR is a register that internal eight 16C1050 UARTs share to use. It is used to verify the generation status of each interrupt of eight 16C1050 UARTs when global interrupt function is enabled. Table 18 shows GISR bit settings.

Table 18: Global Interrupt Status Register Description

Bit	Symbol	Description
7	GISR[7]	8th UART Interrupt Status. 0 : Interrupt of 8th UART channel was not generated. 1 : Interrupt of 8th UART channel was generated.
6	GISR[6]	7th UART Interrupt Status. 0 : Interrupt of 7th UART channel was not generated. 1 : Interrupt of 7th UART channel was generated.
	GISR[5]	6th UART Interrupt Status. 0 : Interrupt of 6th UART channel was not generated. 1 : Interrupt of 6th UART channel was generated.
4	GISR[4]	5th UART Interrupt Status. 0 : Interrupt of 5th UART channel was not generated. 1 : Interrupt of 5th UART channel was generated.
3	GISR[3]	4th UART Interrupt Status. 0 : Interrupt of 4th UART channel was not generated. 1 : Interrupt of 4th UART channel was generated.
2	GISR[2]	3rd UART Interrupt Status. 0 : Interrupt of 3rd UART channel was not generated. 1 : Interrupt of 3rd UART channel was generated.
1	GISR[1]	2nd UART Interrupt Status. 0 : Interrupt of 2nd UART channel was not generated. 1 : Interrupt of 2nd UART channel was generated.
0	GISR[0]	1st UART Interrupt Status. 0 : Interrupt of 1st UART channel was not generated. 1 : Interrupt of 1st UART channel was generated.

7.14 Transmit FIFO Count Register (TCR, Page 2)

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

7.15 Receive FIFO Count Register (RCR, Page 2)

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

7.16 Flow Control Status Register (FSR, Page 2)

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

Table 19: Flow Control Status Register Description

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

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

7.17 Page Select Register (PSR, Page 3)

If BFh is written in LCR, registers in Page3 and Page4 can be accessed. PSR is used to determine which page to use. Table 20 shows PSR bit settings.

Table 20: Page Select Register Description

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

7.18 Auto Toggle Control Register (ATR, Page 3)

ATR controls the signals for controlling input/output signals when using Line Interface as RS422 or RS485, so eliminates additional glue logic outside. Table 21 shows ATR bit settings.

Table 21: Auto Toggle Control Register Description

Bit	Symbol	Description
7	ATR[7]	RXEN Polarity Select. 0 : Asserted output of RXEN is '0'. 1 : Asserted output of RXEN is '1'. (default)
6	ATR[6]	RXEN Control Mode Select. Only when ATR[1:0] is '11'; 0 : RXEN is outputted as same as ATR[7], irrespective of TXD signal. (default) 1 : RXEN is outputted as same as ATR[7] when TXD signal is not transmitting. And outputted as complement of ATR[7] when TXD signal is transmitting.
5	ATR[5]	TXEN Polarity Select. 0 : Asserted output of TXEN is '0'. (default) 1 : Asserted output of TXEN is '1'.
4	ATR[4]	TXEN Control Mode Select. 0 : TXEN is outputted as same as ATR[5], irrespective of TXD signal. (default) 1 : TXEN is outputted as complement of ATR[5] when TXD signal is not transmitting, and outputted as same as ATR[5] when TXD signal is transmitting..
3:2	ATR[3:2]	Not used, always '00'.
1:0	ATR[1:0]	Auto Toggle Enable. 00 : Auto toggle is disabled (default). nRTS_TXEN, nDTR_TXEN pin operate as nRTS, nDTR. And each of nTXRDY_TXEN, nRXRDY_RXEN operates as nTXRDY, nRXRDY. 01 : nRTS_TXEN pin operates as TXEN. nDTR_TXEN pin operates as nDTR. And each of nTXRDY_TXEN, nRXRDY_RXEN operates as nTXRDY, nRXRDY. 10 : nDTR_TXEN pin operates as TXEN. nRTS_TXEN operates as nRTS. And each of nTXRDY_TXEN, nRXRDY_RXEN operates as nTXRDY, nRXRDY. 11 : nTXRDY_TXEN, nRXRDY_RXEN pin operates as TXEN, RXEN. nRTS_TXEN, nDTR_TXEN operates as nRTS, nDTR.

7.19 Enhanced Feature Register (EFR, Page 3)

EFR enables or disables the enhanced features of the UART. Table 22 shows EFR bit settings.

Table 22: Enhanced Feature Register Description

Bit	Symbol	Description
7	EFR[7]	Auto-CTS Flow Control Enable. 0 : Auto-CTS flow control is disabled (default). 1 : Auto-CTS flow control is enabled. Transmission stops when NCTS pin is inputted '1'. Transmission resumes when NCTS pin is inputted '0'.
6	EFR[6]	Auto-RTS Flow Control Enable. 0 : Auto-RTS flow control is disabled (default). 1 : Auto-RTS flow control is enabled. The NRTS pin outputs '1' when data in RX FIFO fill above the FUR. NRTS pin outputs '0' when data in RX FIFO fall below the FLR.
5	EFR[5]	Special Character Detect. 0 : Special character detect disabled (default). 1 : Special character detect enabled. The UART compares each incoming character with data in Xoff2 register. If a match occurs, the received data is transferred to RX FIFO and ISR[4] is set to '1' to indicate that a special character has been detected.
4	EFR[4]	Enhanced Function Bits Enable. 0 : Disables enhanced functions and writing to IER[7:4], FCR[5:4], MCR[7:5]. 1 : Enables enhanced function IER[7:4], FCR[5:4], and MCR[7:5] can be modified, i.e., this bit is therefore a write enable.
3:0	EFR[3:0]	Software Flow Control Select. 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 4 "Software flow control options (EFR[3:0])" on page 15.

7.23 Additional Feature Register (AFR, Page 4)

AFR enables or disables the 256-byte FIFO mode and controls the global interrupt. Table 23 shows AFR bit settings.

Table 23: Additional Feature Register Description

Bit	Symbol	Description
7:6	AFR[7:6]	Not used, always '00'.
5	AFR[5]	Global Interrupt Polarity Select 0 : GINT pin outputs '0' when interrupt is generated (default). 1 : GINT pin outputs '1' when interrupt is generated.
4	AFR[4]	Global Interrupt Enable 0 : INT0/GINT pin is selected to INT0 (default). 1 : INT0/GINT pin is selected to GINT.
3:1	AFR[3:1]	Not used, always '000'.
0	AFR[0]	256-byte FIFO Enable. 0 : 256-byte FIFO mode is disabled and this means IN16C1058 operates as Non FIFO mode or 64-byte FIFO mode (default). 1 : 256-byte FIFO mode is enabled and ISR[7:6] operates as 256-TX FIFO Empty and 256-RX FIFO Full.

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

XRCR operates only when Software flow control is enabled by EFR[3:0] and Xoff Re-transmit function of MCR[2] is also enabled. And it determines the period of retransmission of Xoff character. Table 24 shows XRCR bit settings.

Table 24: Xoff Re-transmit Count Register Description

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

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

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

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

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

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

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

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

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

Table 25: IN16C1058 Reset Conditions

Registers	Reset State
	Page 0
RBR	[7:0] = 'XXXX_XXXX'
IER	[7:0] = '0000_0000'
FCR	[7:0] = '0000_0000'
ISR	[7:0] = '0000_0001'
LCR	[7:0] = '0000_0000'
MCR	[7:0] = '0000_0000'
LSR	[7:0] = '0110_0000'
MSR	[7:4] = '0000' [3:0] = Logic levels of the inputs inverted
SPR	[7:0] = '0000_0000'
	Page 1
DLL	[7:0] = '1111_1111'
DLM	[7:0] = '1111_1111'
	Page 2
GICR	[7:0] = '0000_0000'
GISR	[7:0] = '0000_0000'
TCR	[7:0] = '0000_0000'
RCR	[7:0] = '0000_0000'
FSR	[7:0] = '0000_0000'
	Page 3
PSR	[7:0] = '0000_0000'
ATR	[7:0] = '0000_0000'
EFR	[7:0] = '0000_0000'
XON1	[7:0] = '0000_0000'
XON2	[7:0] = '0000_0000'
XOFF1	[7:0] = '0000_0000'
XOFF2	[7:0] = '0000_0000'
	Page 4
AFR	[7:0] = '0000_0000'
XRCR	[7:0] = '0000_0000'
TTR	[7:0] = '1000_0000'
RTR	[7:0] = '1000_0000'
FUR	[7:0] = '0000_0000'
FLR	[7:0] = '0000_0000'
Output Signals	**Reset State**
TXD, NRTS, NDTR	Logic 1
nTXRDY	Logic 0
nRXRDY	Logic 1
INT	Tri-State Condition = INTSEL is open or low state Logic 0 = INTSEL is high state

8. Option Register Descriptions

IN16C1058 can be used as normal Octal-UART with Normal mode or expand up to 32 ports with SystemBase MIO mode. Option Register Set is provided to efficiently manage these ports in MIO mode. These Option Registers contain the control information to manage serial ports and handles the interrupts from 8 channels as vectors so that device drivers can quickly access and resolve them. It is possible to immediately check which channel the interrupt occurred in through Interrupt Poll Register and nullify the interrupts by each channel through Interrupt Mask Register.

8.1 Option Registers Map

In MIO mode, IN16C1058s are connected with daisy chain and up to 4 devices can be connected. The daisy chain connection is composed of the first 8 ports, the second 8 ports, the third 8 ports and the last 8 ports. Option Registers can be accessed depending on where the device is located of the four places. For instance, if you were to access IN16C1058's DIR in second panel, first access address 05h through MIO Bus and then you can access DIR1 to get the device information of Port9 ~ Port16. If you were to access IN16C1058's IPR in third panel, access address 11h through MIO Bus and then you can access IPR2 to get the device information of Port17 ~ Port24.

By providing these Option Registers, IN16C1058 stores information of each serial port, allows creation of software drivers for given communication specifications and provide users with various information. Also, it is possible to form fast Interrupt Service Routine by handling Interrupts from UART as Vectors.

In the Option Register Map on below chart, the same register set is providing 0 ~ 3. This address map shows the addresses that can be accessed through MIO Bus.

MIO Bus can be expanded up to 32 ports by 8 ports. Register Set is designed as below to process basic unit information of the 8 ports. In a 32-port MultiPort Application, as it is unknown where the IN16C1058 panel is going to be placed of the 4 places, the order of placement is analyzed with daisy chain and the access address for each panel is selected. For instance, for the first panel, corresponding option register set are DIR0, IIR0, IMR0 and IPR0 and only MIO Bus Access commands that correspond to these are responded. For the fourth panel, corresponding option register set are DIR3, IIR3, IMR3 and IPR3 and only MIO Bus Access commands that correspond to these are responded.

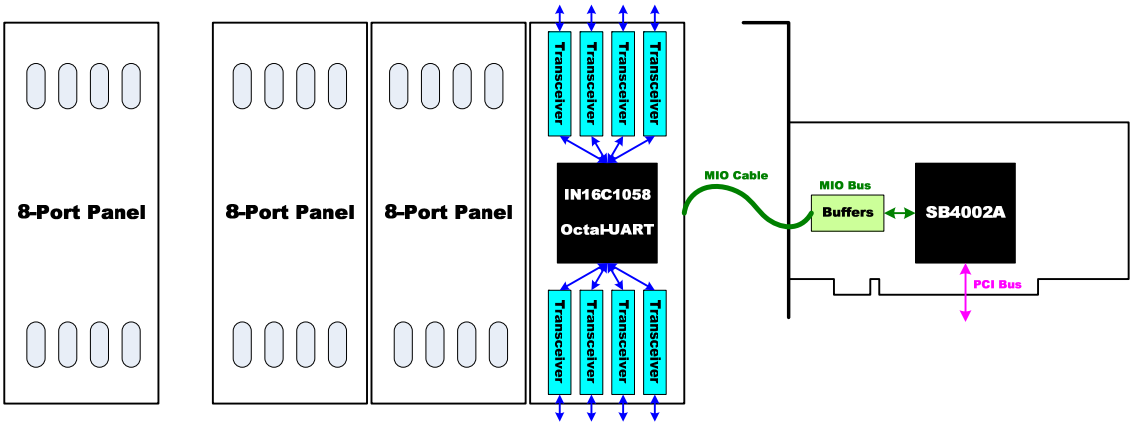


Table 26: Option Registers Map

ADDR[4:0]	R/W	Descriptions
04h	R/W	DIR0 (Device Information Register for Port1 ~ Port8)
05h	R/W	DIR1 (Device Information Register for Port9 ~ Port16)
06h	R/W	DIR2 (Device Information Register for Port17 ~ Port24)
07h	R/W	DIR3 (Device Information Register for Port25 ~ Port32)
08h	R/W	IIR0 (Interface Information Register for Port1 ~ Port8)
09h	R/W	IIR1 (Interface Information Register for Port9 ~ Port16)
0Ah	R/W	IIR2 (Interface Information Register for Port17 ~ Port24)
0Bh	R/W	IIR3 (Interface Information Register for Port25 ~ Port32)
0Ch	R/W	IMR0 (Interrupt Mask Register for Port1 ~ Port8)
0Dh	R/W	IMR1 (Interrupt Mask Register for Port9 ~ Port16)
0Eh	R/W	IMR2 (Interrupt Mask Register for Port17 ~ Port24)
0Fh	R/W	IMR3 (Interrupt Mask Register for Port25 ~ Port32)
10h	RO	IPR0 (Interrupt Poll Register for Port1 ~ Port8)
11h	RO	IPR1 (Interrupt Poll Register for Port9 ~ Port16)
12h	RO	IPR2 (Interrupt Poll Register for Port17 ~ Port24)
13h	RO	IPR3 (Interrupt Poll Register for Port25 ~ Port32)

8.2 Device Information Register

Table 27: Device Information Register

Bit	Name	Descriptions
7	U[3:0]	Shows the type of UART. 0h : Use 16C550 UART Core 1h : Use 16C1050 UART Core Others: Not Defined
6		
5		
4		
3	O[3:0]	Shows the UART's operating frequency. (Oscillator/Crystal) 0h: Use 1.8432MHz UART Clock 1h: Use 3.6864MHz UART Clock 2h: Use 7.3728MHz UART Clock 3h: Use 14.7456MHz UART Clock 4h: Use 29.4912MHz UART Clock 5h: Use 58.9854MHz UART Clock Others: Not Defined
2		
1		
0		

8.3 Interface Information Register

Table 28: Interface Information Register

Bit	Name	Descriptions
7	0b	Hardwired to 0
6	0b	
5	I[1:0]	Type of Serial Port Interface 0h: RS232 Interface 1h: RS422 Interface 2h: RS485 Interface 4h: Unknown
4		
3	TRXEN[1:0]	In RS422/485 communication, set the signal line used as TX/RX Enable signal 0h: RTS 1h: DTR 2h: Exclusive signal line (TXEN/RXEN) 3h: Not Defined
2		
1	0b	Hardwired to 0
0	0b	

8.4 Interrupt Mask Register

In Normal Mode, UART's internal registers can be accessed through nCS and UART's GICR (Global Interrupt Control Register) is used for configuring 8 UART channel's interrupt mask. GICR is the same register as IPR in this case.
IMR access Option Register through nOPT control signal in MIO mode and GICR access UART internal Register through nCS or nUART signal. GICR and IMR are same region and IN16C1058 provides different methods of accessing them.

Table 29: Interrupt Mask Register

Bit	Name	Descriptions
7	M7	1h: Enables Port8 Interrupt. 0h : Disables Port8 Interrupt. (default)
6	M6	1h: Enables Port7 Interrupt. 0h : Disables Port7 Interrupt. (default)
5	M5	1h: Enables Port6 Interrupt. 0h : Disables Port6 Interrupt. (default)
4	M4	1h: Enables Port5 Interrupt. 0h : Disables Port5 Interrupt. (default)
3	M3	1h: Enables Port4 Interrupt. 0h : Disables Port4 Interrupt. (default)
2	M2	1h: Enables Port3 Interrupt. 0h : Disables Port3 Interrupt. (default)
1	M1	1h: Enables Port2 Interrupt. 0h : Disables Port2 Interrupt. (default)
0	M0	1h: Enables Port1 Interrupt. 0h : Disables Port1 Interrupt. (default)

8.5 Interrupt Poll Register

In Normal Mode, UART's internal registers can be accessed through nCS and UART's GISR (Global Interrupt Status Register) is used for checking 8 UART channel's interrupt status. GISR is the same register as IPR in this case.

IPR access Option Register through nOPT control signal in MIO mode and GISR access UART internal Register through nCS or nUART signal. GISR and IMR are same region and IN16C1058 provides different methods of accessing them.

Table 30: Interrupt Poll Register

Bit	Name	Descriptions
7	P7	1h: Interrupt not occurred in Port8. 0h : Interrupt occurred in Port8.
6	P6	1h: Interrupt not occurred in Port7. 0h : Interrupt occurred in Port7.
5	P5	1h: Interrupt not occurred in Port6. 0h : Interrupt occurred in Port6.
4	P4	1h: Interrupt not occurred in Port5. 0h : Interrupt occurred in Port5.
3	P3	1h: Interrupt not occurred in Port4. 0h : Interrupt occurred in Port4.
2	P2	1h: Interrupt not occurred in Port3. 0h : Interrupt occurred in Port3.
1	P1	1h: Interrupt not occurred in Port2. 0h : Interrupt occurred in Port2.
0	P0	1h: Interrupt not occurred in Port1. 0h : Interrupt occurred in Port1.

9. Programmer's Guide

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

Table 31: Register Programming Guide

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

Table 31: Register Programming Guide…*continued*

Command	Action
	Set FUR to Upper Threshold Value Set FLR to Lower Threshold Value Set PSR to A4h Set LCR to temp
Set TX FIFO / RX FIFO Interrupt Trigger Level for 64-byte FIFO Mode	1) Set FCR to '0000_xxx1' → Set RTR to 8, set TTR to 8 2) Set FCR to '0101_xxx1' → Set RTR to 16, set TTR to 16 3) Set FCR to '1010_xxx1' → Set RTR to 56, set TTR to 32 4) Set FCR to '1111_xxx1' → Set RTR to 60, set TTR to 56
Set TX FIFO / RX FIFO Interrupt Trigger Level for 256-byte FIFO Mode	Set FCR to 'xxxx_xxx1' Read LCR, then save in temp Set LCR to BFh Set PSR to A5h Set AFR to 01h Set TTR to TX FIFO Trigger Level Value Set RTR to RX FIFO Trigger Level Value Set PSR to A4h Set LCR to temp
Read Flow Control Status	Read LCR, then save in temp1 Read MCR, then save in temp2 Set LCR to ('0111_1111' **AND** temp1) Set MCR to ('0100_0000' **OR** temp2) Read FSR, then save in temp3 Pass temp3 back to host Set MCR to temp2 Set LCR to temp1
Read TX FIFO / RX FIFO Count Value	Read LCR, then save in temp1 Read MCR, then save in temp2 Set LCR to ('0111_1111' **AND** temp1) Set MCR to ('0100_0000' **OR** temp2) Read TCR, then save in temp3 Read RCR, then save in temp4 Pass temp3 back to host Pass temp4 back to host Set MCR to temp2 Set LCR to temp1

Table 31: Register Programming Guide…*continued*

Command	Action
Read 256-byte TX FIFO Empty Status / RX FIFO Full Status	Set FCR to 'xxxx_xxx1' Read LCR, then save in temp1 Set LCR to BFh Set PSR to A5h Set AFR to 01h Set PSR to A4h Set LCR to temp1 Read ISR, then save in temp2 Pass temp2 back to host
Enable Xoff Re-transmit	Read LCR, then save in temp1 Set LCR to not BFh Read MCR, then save in temp2 Set MCR to ('0100_0000' **OR** temp2) Set MCR to ('0100_0100' **OR** temp2) Set MCR to ('1011_1111' **AND** temp2) Set MCR to temp2 Set LCR to temp1
Disable Xoff Re-transmit	Read LCR, then save in temp1 Set LCR to not BFh Read MCR, then save in temp2 Set MCR to ('0100_0000' **OR** temp2) Set MCR to ('1011_1011' **AND** temp2) Set MCR to temp2 Set LCR to temp1

Set Prescaler Value to Divide-by-1 or 4	Read LCR, then save in temp1
	Set LCR to BFh
	Read EFR, then save in temp2
	Set EFR to ('0001_0000' **OR** temp2)
	Set LCR to 00h
	Read MCR, then save in temp3
	
	if Divide-by-1 = OK then
	Set MCR to ('0111_1111' **AND** temp3)
	else
	Set MCR to ('1000_0000' **OR** temp3)
	
	Set LCR to BFh
	Set EFR to temp2
	Set LCR to temp1

Table 32: IN16C1058 Programming Guide

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

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

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

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

Table 32: IN16C1058 Programming Guide…*continued*

Command	Action

	Else
	For (Cnt = 0; Cnt < temp3; Cnt++)
	5. Read Data — Read TX_Data from TX_User_Buffer
	6. Output TX — Set THR to TX_Data
	
	Return from Interrupt Service Routine

Serial Input Process 1. RX Interrupt is generated and Jumped to Interrupt Service Routine	
2. Read ISR	Read ISR, then save in temp1

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

4. Read RX FIFO Count	Read RCR, then save in temp3
	Set MCR to temp2
	If temp1[6] = 1b then
	For (Cnt = 0; Cnt <= 255; Cnt++)
	5. Read RX Data — Read RBR, save in RX_User_Buffer
	Else
	For (Cnt = 0; Cnt < temp3; Cnt++)
	5. Read Data — Read RBR, save in RX_User_Buffer
	
	Return from Interrupt Service Routine

10. Electrical Characteristics

10.1 Absolute Maximum Ratings

Table 32: Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
V_{DD}	DC Supply Voltage	-0.5	7.0	V
V_{IN}	Input Voltage	-0.5	V_{DD} + 0.5	V
V_{OUT}	Output Voltage Range	0	V_{DD} + 0.5	V
T_{STG}	Storage Temperature	-65	150	℃

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

10.2 Power Consumption

Table 33: IN16C1058 Power Consumption

Power Consumption	Minimum	Typical	Maximum	Unit
IN16C1058-TQ	-	3.270	3.597	W

10.3 DC Electrical Characteristics

Table 34: DC Electrical Characteristics

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

10.4 AC Electrical Characteristics

Table 31: AC Electrical Characteristics

Symbol	Parameter	Min	Max	Unit
t_{rd}	Pulse duration, nIOR low	24		ns
t_{csr}	Set up time, nCS valid before nIOR low †	10		ns
t_{ar}	Set up time, A2~A0 valid before nIOR low †	10		ns
t_{ra}	Hold time, A2~A0 valid after nIOR high †	2		ns
t_{rcs}	Hold time, nCS valid after nIOR high †	0		ns
t_{frc}	Delay time, $t_{ar}+t_{rd}+t_{rc}$ ‡	54		ns
t_{rc}	Delay time, nIOR high to nIOR or nIOW low	20		ns
t_{wr}	Pulse duration, nIOW ↓	24		ns
t_{csw}	Setup time, nCS valid before nIOW ↓	10		ns
t_{aw}	Setup time, A7~A0 valid before nIOW ↓	10		ns
t_{ds}	Setup time, D7~D0 valid before nIOW ↑	15		ns
t_{wa}	Hold time, A7~A0 valid after nIOW ↑	2		ns
t_{wcs}	Hold time, nCS valid after nIOW ↑	2		ns
t_{dh}	Hold time, D7~D0 valid after nIOW ↑	5		ns
t_{fwc}	Delay time, $t_{aw}+t_{wr}+t_{wc}$	54		ns
t_{wc}	Delay time, nIOW ↑ to nIOW or nIOR ↓	20		ns
t_{rvd}	Enable time, nIOR ↓ to D7~D0 valid		24	ns
t_{hz}	Disable time, nIOR to D7~D0 released	4		ns
t_{irs}	Delay time, INT ↓ to TXDx ↓ at start	8	24	RCLK
t_{sti}	Delay time, TXDx ↓ at start to INT ↑	8	8	RCLK
t_{si}	Delay time, nIOW high or low (WR THR) to INT ↑	16	32	RCLK
t_{sxa}	Delay time, TXDx ↓ at start to nTXRDY ↓		8	RCLK
t_{hr}	Propagation delay time, nIOW(WR THR) ↓ to INT ↓		12	ns
t_{ir}	Propagation delay time, nIOR(RD IIR)↑ to INT ↓		12	ns
t_{wxi}	Propagation delay time, nIOW(WR THR) ↓ to nTXRDY ↑		10	ns
t_{sint}	Delay time, stop bit to INT ↑ or stop bit to nRXRDY or read RBR to set interrupt	1		RCLK
t_{rint}	Propagation delay time, Read RBR/LSR to INT ↓/LSR interrupt ↓		12	ns
t_{rint}	Propagation delay time, nIOR RCLK↓ to nRXRDY ↑		12	ns
t_{mdo}	Propagation delay time, nIOW(WR MCR)↑ to nRTSx, nDTRx↑		12	ns
t_{sim}	Propagation delay time, modem input nCTSx, nDSRx, and nDCDx ↓↑ to INT ↑		12	ns
t_{rim}	Propagation delay time, nIOR(RD MSR) ↑ to interrupt ↓		3	ns

t_{sim}	Propagation delay time, nRIx ↑ to INT ↓	12	ns

† The internal address strobe is always in active state.

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

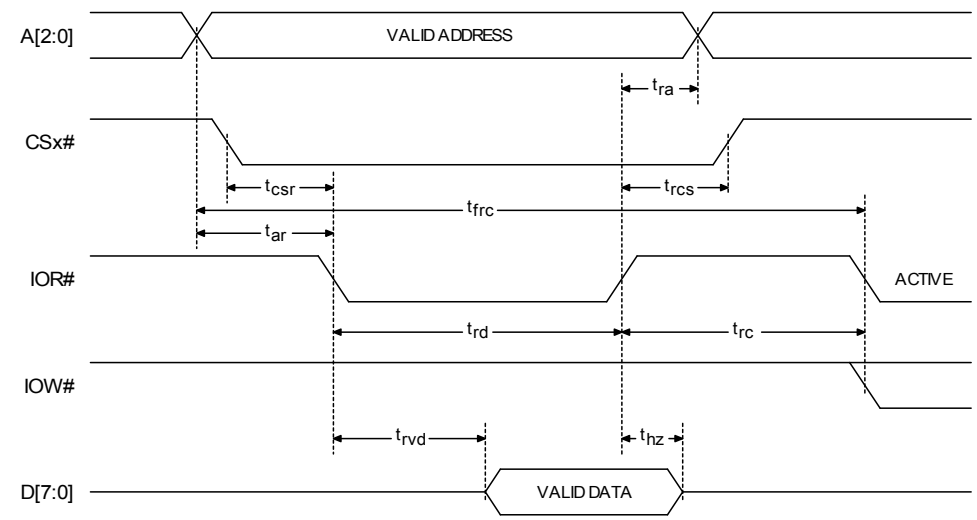


Figure 10: Read Cycle Timing

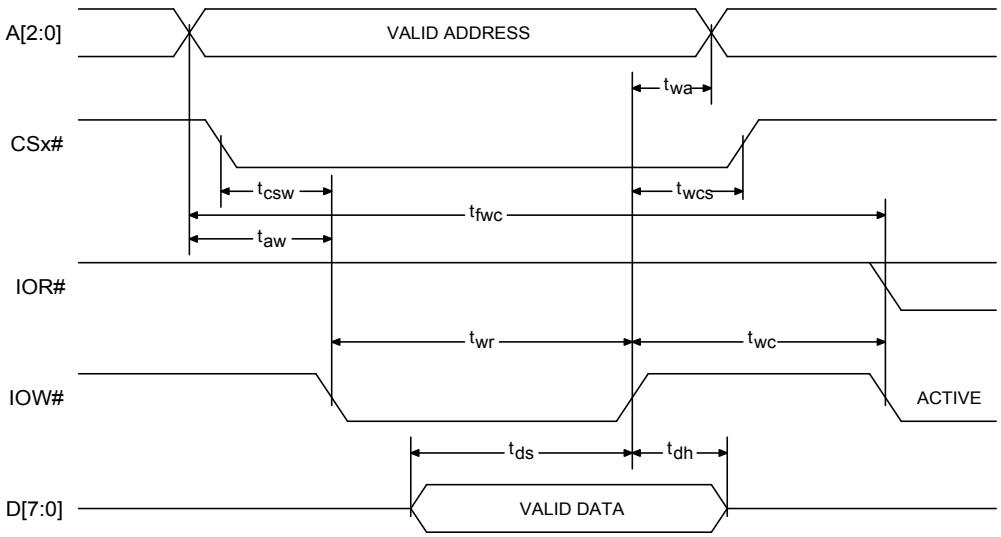


Figure 11: Write Cycle Timing

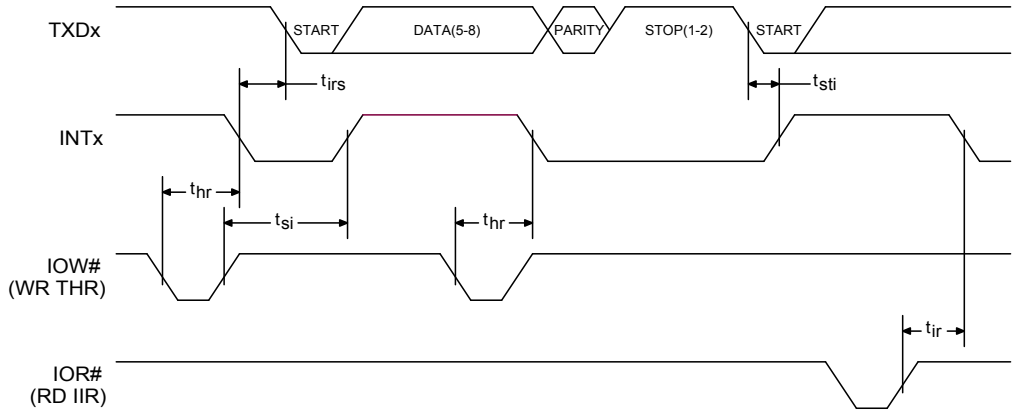


Figure 12: Transmitter Timing

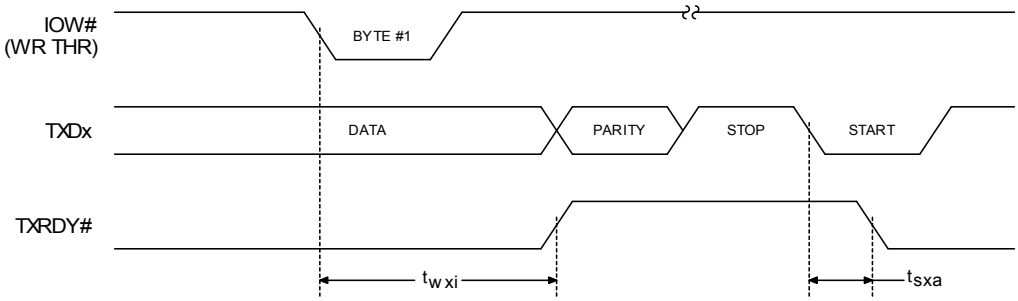


Figure 13: Transmitter Ready Mode 0 Timing

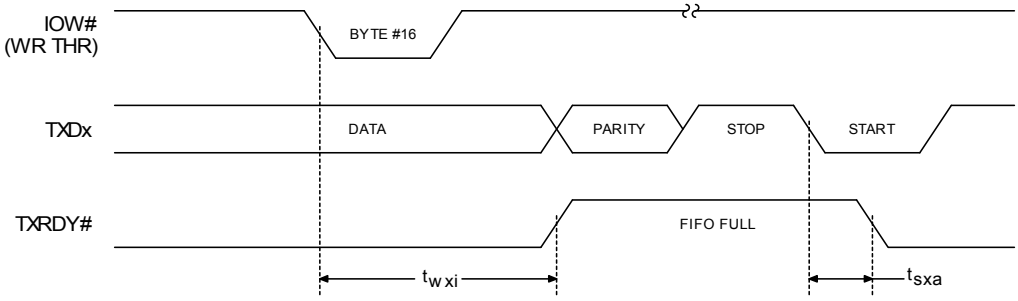


Figure 14: Transmitter Ready Mode 1 Timing

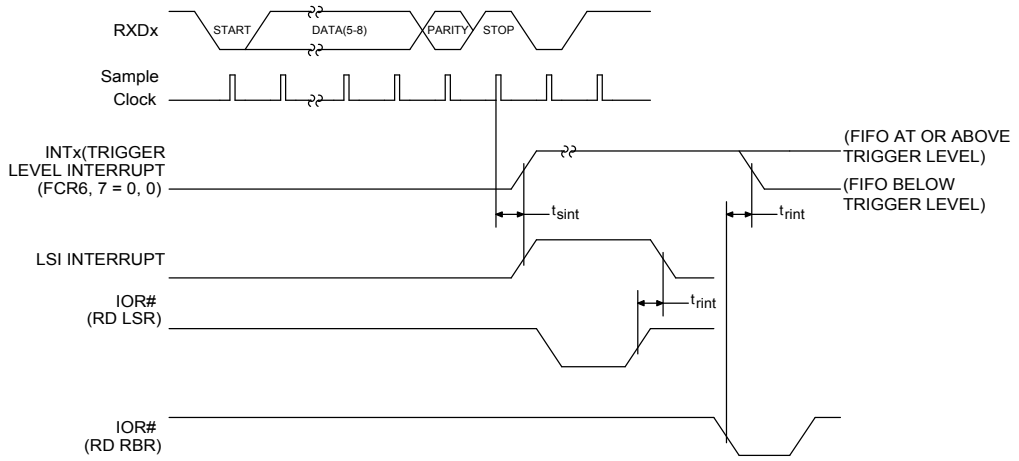


Figure 15: Receiver FIFO First Byte (Sets RBR) Timing

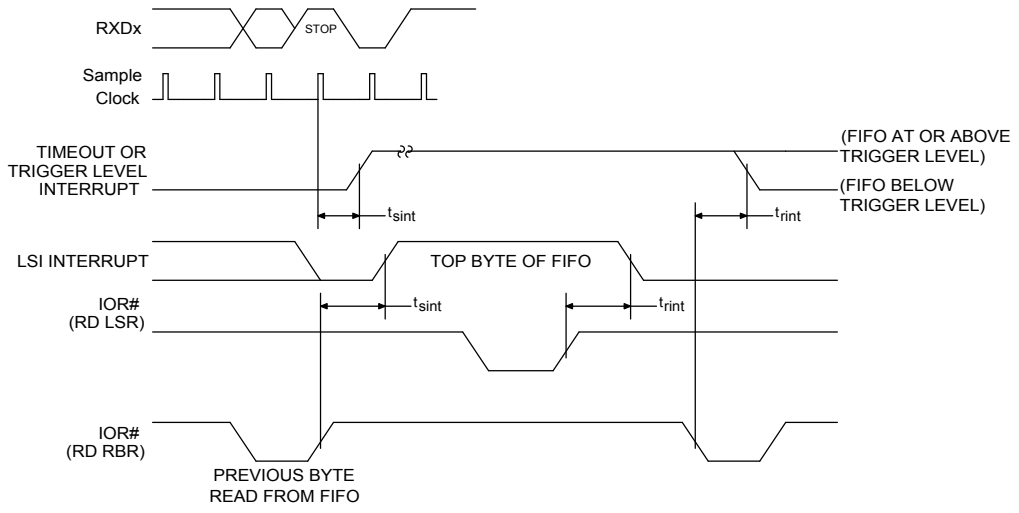


Figure 16: Receiver FIFO After First Byte (After RBR Set) Timing

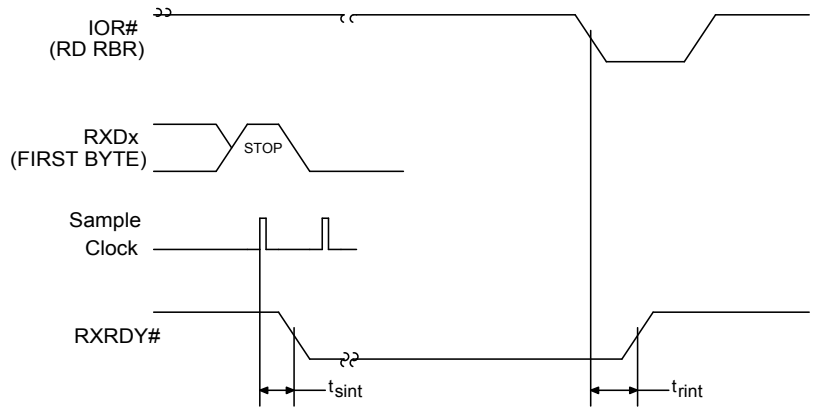


Figure 17: Receiver Ready Mode 0 Timing

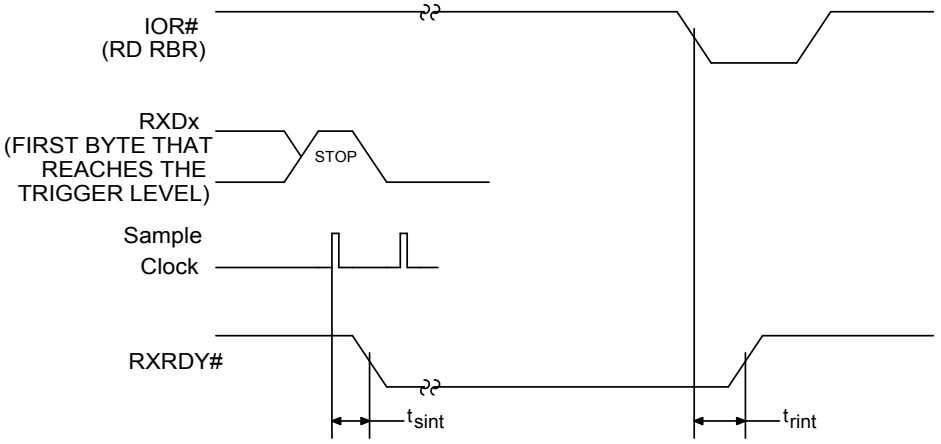


Figure 18: Receiver Ready Mode 1 Timing

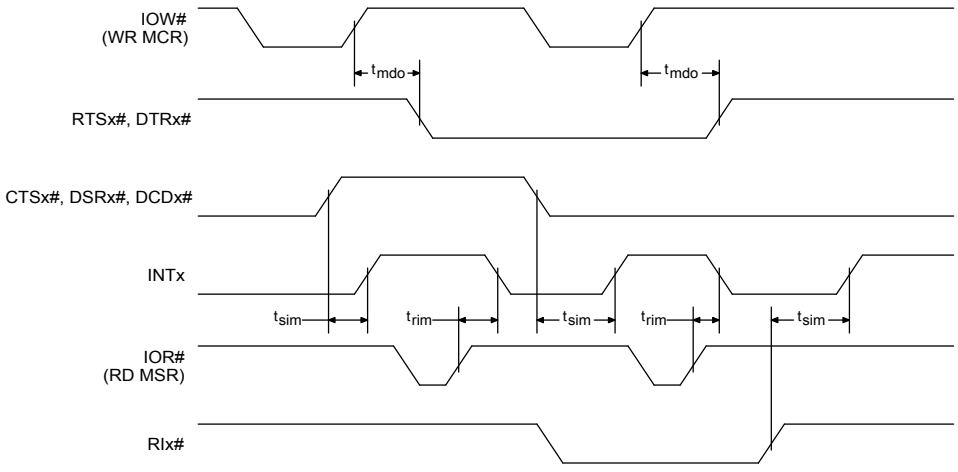


Figure 19: Modem Control Timing

11.Package Outline

128-Pin TQFP: Thin Plastic Octal Flat Package; Body 20 x 20 x 1.2 mm

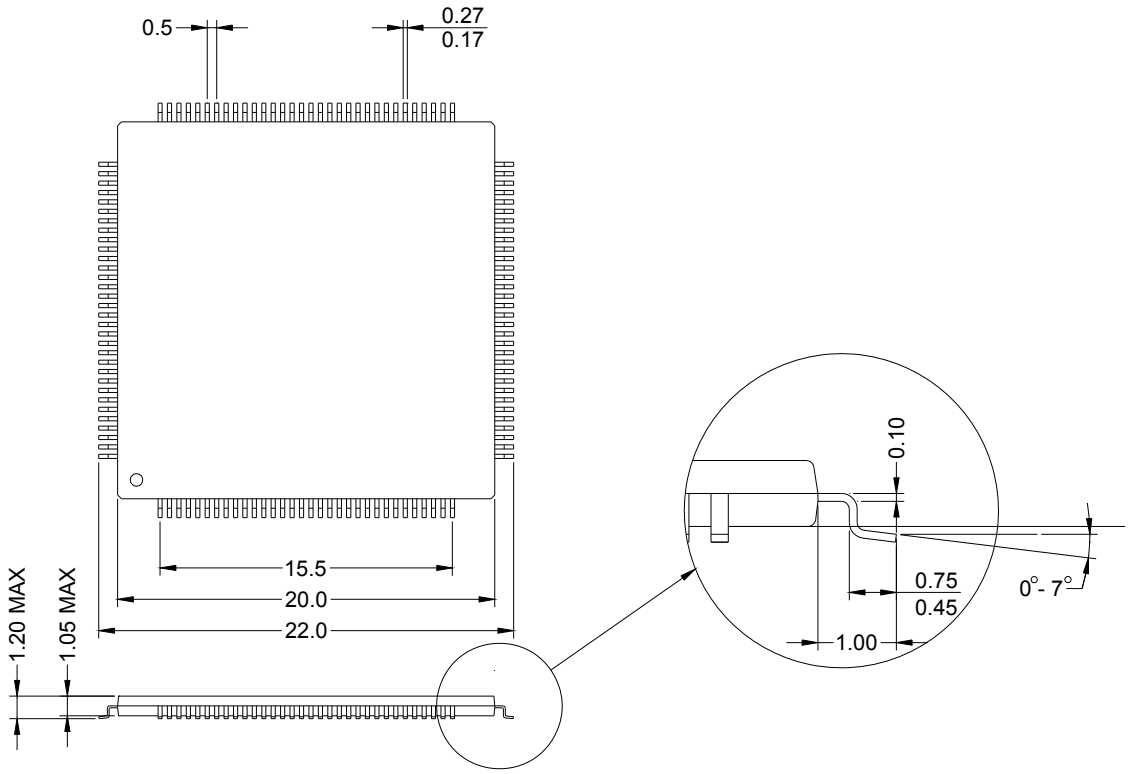


Figure 20: IN16C1058 128-Pin TQFP Outline (Mechanical Drawing)

Note :

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