Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

XR20M1280

I2C/SPI UART WITH 128-BYTE FIFO AND INTEGRATED LEVEL SHIFTERS

MAY 2011 ***REV. 1.0.0***

GENERAL DESCRIPTION

The XR20M1280[1] (M1280) is a single-channel I^{2}C/SPI Universal Asynchronous Receiver and Transmitter (UART) with integrated level shifters and 128 bytes of transmit and receive FIFOs.

For flexibility in a mixed voltage environment, the M1280 has 4 VCC pins. There is a VCC pin for the core, a VCC pin for the UART signals, a VCC pin for the CPU interface signals and a VCC pin for the GPIO signals. The VCC pins for the UART, GPIO and I^{2}C/SPI interface signals allow for the M1280 to interface with devices operating at different voltage levels eliminating the need for external voltage level shifters. The VCC pin for the core voltage helps lower the overall power consumption of applications that use slower data rates.

The Auto RS-485 Half-Duplex Direction control feature simplifies both the hardware and software for half-duplex RS-485 applications. In addition, the Multidrop mode with Auto Address detection and Address Byte Control features increase the performance by simplifying the software routines.

The Independent TX/RX Baud Rate Generator feature allows the transmitter and receiver to operate at different baud rates. In addition, the Fractional Baud Rate Generator feature provides flexibility for crystal/clock frequencies for generating standard and non-standard baud rates.

The M1280 has programmable transmit and receive FIFO trigger levels, automatic hardware and software flow control, and data rates of up to 24 Mbps. Power consumption of the M1280 can be minimized by enabling the sleep mode.

The M1280 has a 16550 compatible register set that provide users with operating status and control, receiver error indications, and modem serial interface controls. An internal loopback capability allows onboard diagnostics. The M1280 has a selectable I^{2}C/SPI bus interface.

NOTE: *[1] Covered by U.S. Patent #5,649,122.*

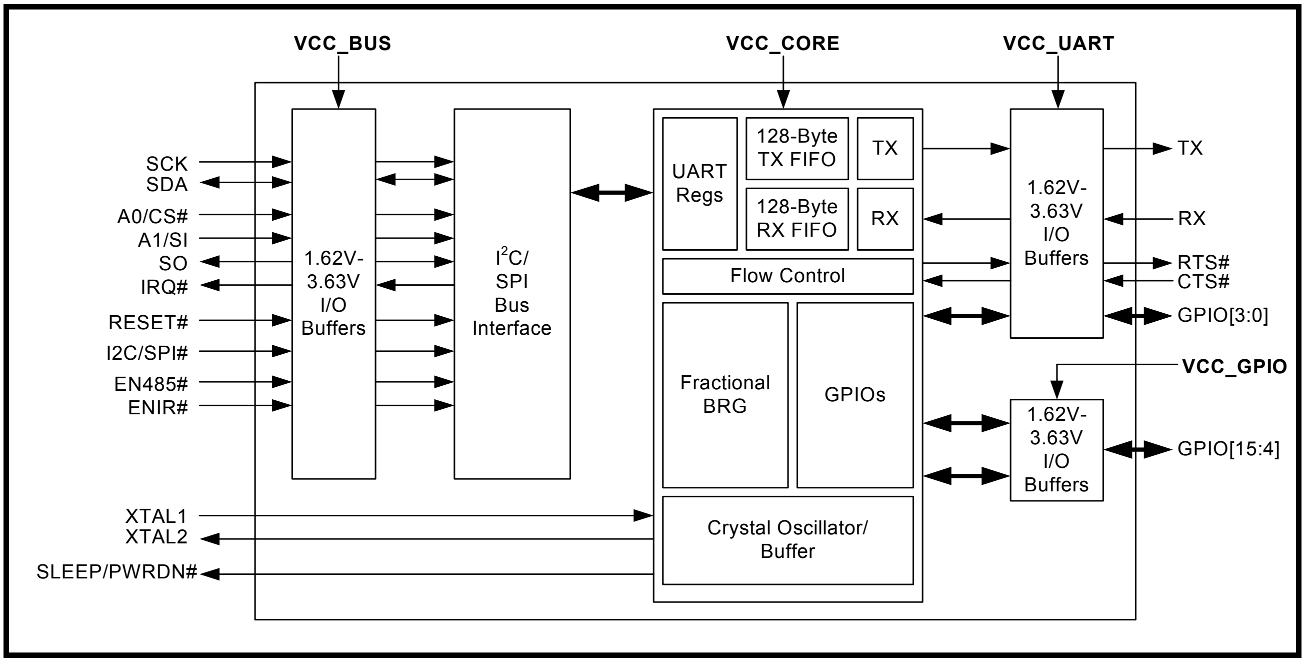
FEATURES

- Integrated Level Shifters on CPU interface, UART and GPIO signals
- Selectable I^{2}C/SPI bus interface
- 26MHz maximum SPI clock
- 24Mbps maximum UART data rate
- Up to 16 GPIOs
- 128-Bytes TX and RX FIFOs
- Programmable TX/RX trigger levels
- TX/RX FIFO Level Counters
- Independent TX/RX Baud Rate Generator
- Fractional Baud Rate Generator
- Auto RTS/CTS Hardware Flow Control
- Auto XON/XOFF Software Flow Control
- Auto RS-485 Half-Duplex Direction Control
- Multidrop mode w/ Auto Address Detect (RX)
- Multidrop mode w/ Address Byte Control (TX)
- Sleep Mode with Automatic Wake-up
- Infrared (IrDA 1.0 and 1.1) mode
- 1.62V to 3.63V supply operation
- 5V tolerant inputs
- Crystal oscillator or external clock input

APPLICATIONS

- Personal Digital Assistants (PDA)
- Cellular Phones/Data Devices
- Battery-Operated Devices
- Global Positioning System (GPS)
- Bluetooth

FIGURE 1. XR20M1280 BLOCK DIAGRAM

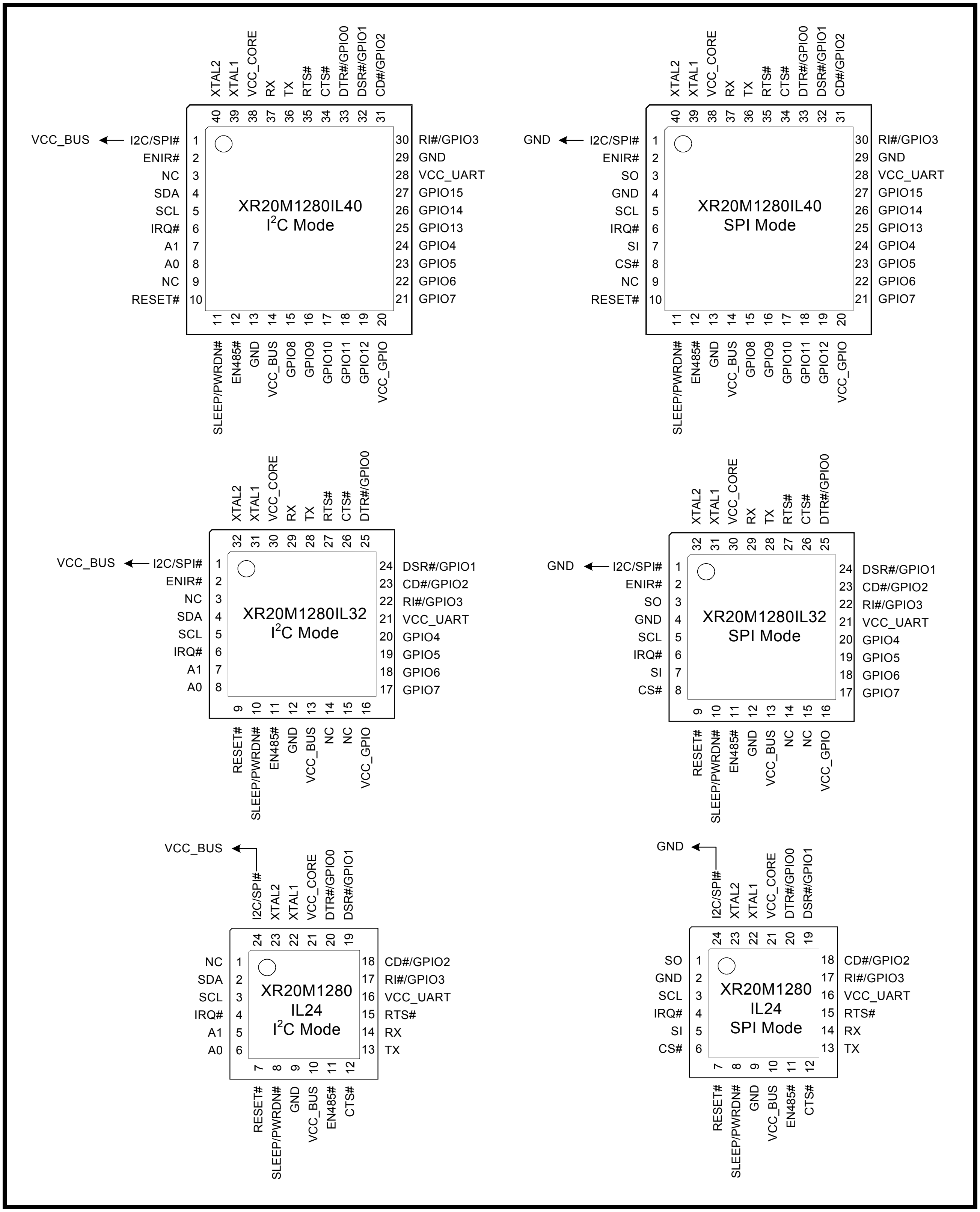


ORDERING INFORMATION

PART NUMBER	PACKAGE	NUMBER OF GPIOS	OPERATING TEMPERATURE RANGE	DEVICE STATUS
XR20M1280IL24-F	QFN-24	4	-40°C to +85°C	Active
XR20M1280IL24TR-F	QFN-24	4	-40°C to +85°C	Active
XR20M1280IL32-F	QFN-32	8	-40°C to +85°C	Active
XR20M1280IL32TR-F	QFN-32	8	-40°C to +85°C	Active
XR20M1280IL40-F	QFN-40	16	-40°C to +85°C	Active
XR20M1280IL40TR-F	QFN-40	16	-40°C to +85°C	Active

NOTE: **TR = Tape and reel, F = Green / RoHS**

Figure 2. Pin Out Assignments



PIN DESCRIPTIONS

Pin Description

NAME	QFN-24 PIN#	QFN-32 PIN#	QFN-40 PIN#	TYPE	DESCRIPTION
I2C (SPI) INTERFACE					
I2C/SPI#	24	1	1	I	I²C-bus or SPI interface select. I²C-bus interface is selected if this pin is HIGH. SPI interface is selected if this pin is LOW
SDA (GND)	2	4	4	I/O	I²C-bus data input/output (open-drain). If SPI configuration is selected, then this pin should be tied LOW.
SCL	3	5	5	I	I²C-bus or SPI serial input clock. When the I²C-bus interface is selected, the serial clock idles HIGH. When the SPI interface is selected, the serial clock idles LOW.
IRQ#	4	6	6	OD	Interrupt output (open-drain, active LOW).
A0 (CS#)	6	8	8	I	I²C-bus device address select A0 or SPI chip select. If I²C-bus configuration is selected, this pin along with the A1 pin allows user to change the device's base address. If SPI configuration is selected, this pin is the SPI chip select pin (Schmitt-trigger, active LOW).
A1 (SI)	5	7	7	I	I²C-bus device address select A1 or SPI data input pin. If I²C-bus configuration is selected, this pin along with A0 pin allows user to change the device's base address. If SPI configuration is selected, this pin is the SPI data input pin.
SO (NC)	1	3	3	O	SPI data output pin. If I2C-bus configuration is selected, this pin must be left unconnected.
RESET#	7	9	10	I	Reset (active LOW) - A longer than 40 ns LOW pulse on this pin will reset the internal registers and all outputs. The UART transmitter output will be idle and the receiver input will be ignored.
MODEM I/O and GPIOs					
TX	13	28	36	O	UART Transmit Data or infrared encoder data. Standard transmit and receive interface is enabled when MCR[6] = 0. In this mode, the TX signal will be a logic 1 during reset or idle (no data). Infrared IrDA transmit and receive interface is enabled when MCR[6] = 1. In the Infrared mode, the inactive state (no data) for the Infrared encoder/decoder interface is a logic 0. If it is not used, leave it unconnected.
RX	14	29	37	I	UART Receive Data or infrared receive data. Normal receive data input must idle at logic 1 condition. The infrared receiver idles at logic 0. This input should be connected to VCC when not used.
RTS#	15	27	35	O	UART Request-to-Send (active low) or general purpose output. This output must be asserted prior to using auto RTS flow control, see EFR[6], MCR[1] and IER[6].
CTS#	12	26	34	I	UART Clear-to-Send (active low) or general purpose input. It can be used for auto CTS flow control, see EFR[7], MSR[4] and IER[7]. This input should be connected to VCC when not used.
GPIO0/DTR#	20	25	33	I/O	General purpose I/O or UART Data-Terminal-Ready (active low).
GPIO1/DSR#	19	24	32	I/O	General purpose I/O or UART Data-Set-Ready (active low).

Pin Description

NAME	QFN-24 PIN#	QFN-32 PIN#	QFN-40 PIN#	TYPE	DESCRIPTION
GPIO2/CD#	18	23	31	I/O	General purpose I/O or UART Carrier-Detect (active low).
GPIO3/RI#	17	22	30	I/O	General purpose I/O or UART Ring-Indicator (active low).
GPIO4 GPIO5 GPIO6 GPIO7	- - - -	20 19 18 17	24 23 22 21	I/O I/O I/O I/O	General purpose I/Os.
GPIO8 GPIO9 GPIO10 GPIO11 GPIO12 GPIO13 GPIO14 GPIO15	- - - - - - - -	- - - - - - - -	15 16 17 18 19 25 26 27	I/O I/O I/O I/O I/O I/O I/O I/O	General purpose I/Os.
ANCILLARY SIGNALS					
XTAL1	22	31	39	I	Crystal or external clock input. Note: This input is not 5V tolerant.
XTAL2	23	32	40	O	Crystal or buffered clock output.
EN485#	11	11	12	I	Enable Auto RS-485 Half-Duplex Mode. This pin is sampled upon power-up. If this pin is HIGH, then the RTS# output can be used for Auto RTS Hardware Flow Control or as a general purpose output. If this pin is LOW, then the RTS# output is the Auto RS-485 Half-Duplex direction control pin.
ENIR#	-	2	2	I	Enable IR Mode. This pin is sampled upon power-up. If this pin is HIGH, then the TX output and RX input will behave as the UART transmit data output and UART receive data input. If this pin is LOW, then the TX output and RX input will behave as the infrared encoder data output and the infrared receive data input.
SLEEP/ PWRDN#	8	10	11	I/O	Sleep / Power Down pin. This pin powers up as the SLEEP input. The SLEEP input can force the UART to enter into the sleep mode after the next byte transmitted or received without meeting any of the sleep mode conditions. This pin can also be configured as an output pin which can be used to indicate to the CPU that the UART has entered the sleep mode. This output can also be used to power down other devices.
VCC_CORE	21	30	38	Pwr	1.62V to 3.63V VCC for the core. This supply voltage is used for the core logic including the crystal oscillator circuit.
VCC_BUS	10	13	14	Pwr	1.62V to 3.63V VCC for bus interface signals. This supply voltage pin will determine the I/O levels of the CPU bus interface signals.
VCC_UART	16	21	28	Pwr	1.62V to 3.63V VCC for the UART signals. This supply voltage pin will determine the I/O levels of the UART I/O signals including GPIO[3:0].

Pin Description

NAME	QFN-24 PIN#	QFN-32 PIN#	QFN-40 PIN#	TYPE	DESCRIPTION
VCC_GPIO	-	16	20	Pwr	1.62V to 3.63V VCC for the GPIO signals. This supply voltage pin will determine the I/O levels of the GPIO[15:4] signals.
GND	9	12	13, 29	Pwr	Power supply common, ground.
GND	Center Pad	Center Pad	Center Pad	Pwr	The center pad on the backside of the QFN package is metallic and should be connected to GND on the PCB. The thermal pad size on the PCB should be the approximate size of this center pad and should be solder mask defined. The solder mask opening should be at least 0.0025" inwards from the edge of the PCB thermal pad.

Pin type: I=Input, O=Output, I/O= Input/output, OD=Output Open Drain.

1.0 FUNCTIONAL DESCRIPTIONS

1.1 CPU Interface

The XR20M1280 can operate with either an I^{2}C-bus interface or an SPI interface. The CPU interface is selected via the I2C/SPI# input pin.

1.1.1 I^{2}C-bus Interface

The I^{2}C-bus interface is compliant with the Standard-mode and Fast-mode I^{2}C-bus specifications. The I^{2}C-bus interface consists of two lines: serial data (SDA) and serial clock (SCL). In the Standard-mode, the serial clock and serial data can go up to 100 kbps and in the Fast-mode, the serial clock and serial data can go up to 400 kbps. The first byte sent by an I^{2}C-bus master contains a start bit (SDA transition from HIGH to LOW when SCL is HIGH), 7-bit slave address and whether it is a read or write transaction. The next byte is the sub-address that contains the address of the register to access. The XR20M1280 responds to each write with an acknowledge (SDA driven LOW by XR20M1280 for one clock cycle when SCL is HIGH). If the TX FIFO is full, the XR20M1280 will respond with a negative acknowledge (SDA driven HIGH by XR20M1280 for one clock cycle when SCL is HIGH) when the CPU tries to write to the TX FIFO. The last byte sent by an I^{2}C-bus master contains a stop bit (SDA transition from LOW to HIGH when SCL is HIGH). See **Figures 3 - 5** below. For complete details, see the I^{2}C-bus specifications.

FIGURE 3. I^{2}C START AND STOP CONDITIONS

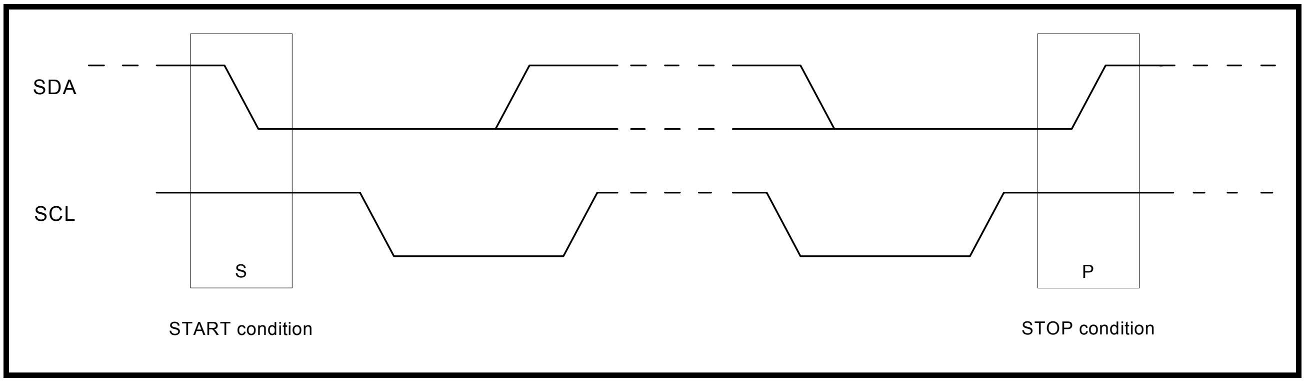
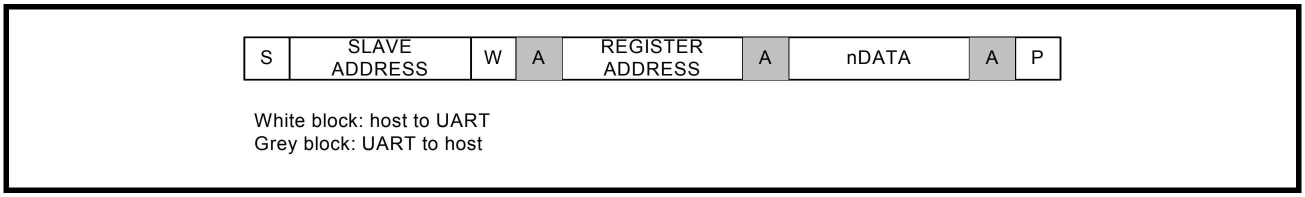


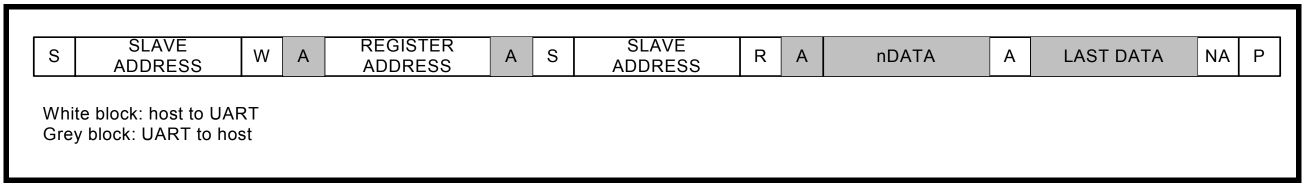
FIGURE 4. MASTER WRITES TO SLAVE (XR20M1280)



S	SLAVE ADDRESS	W	A	REGISTER ADDRESS	A	nDATA	A	P

White block: host to UART
Grey block: UART to host

FIGURE 5. MASTER READS FROM SLAVE (XR20M1280)



S	SLAVE ADDRESS	W	A	REGISTER ADDRESS	A	S	SLAVE ADDRESS	R	A	nDATA	A	LAST DATA	NA	P

White block: host to UART
Grey block: UART to host

1.1.2 I^{2}C-bus Addressing

There could be many devices on the I^{2}C-bus. To distinguish itself from the other devices on the I^{2}C-bus, there are eight possible slave addresses that can be selected for the XR20M1280 using the A1 and A0 address lines. Table 1 below shows the different addresses that can be selected. Note that there are two different ways to select each I2C address.

TABLE 1: XR20M1280 I^{2}C ADDRESS MAP

A1	A0	I^{2}C ADDRESS
VCC	VCC	0x60 (0110 000X)
VCC	GND	0x62 (0110 001X)
VCC	SCL	0x64 (0110 010X)
VCC	SDA	0x66 (0110 011X)
GND	VCC	0x68 (0110 100X)
GND	GND	0x6A (0110 101X)
GND	SCL	0x6C (0110 110X)
GND	SDA	0x6E (0110 111X)
SCL	VCC	0x60 (0110 000X)
SCL	GND	0x62 (0110 001X)
SCL	SCL	0x64 (0110 010X)
SCL	SDA	0x66 (0110 011X)
SDA	VCC	0x68 (0110 100X)
SDA	GND	0x6A (0110 101X)
SDA	SCL	0x6C (0110 110X)
SDA	SDA	0x6E (0110 111X)

An I^{2}C sub-address is sent by the I^{2}C master following the slave address. The sub-address contains the UART register address being accessed. A read or write transaction is determined by bit-0 of the slave address. If bit-0 is’0’, then it is a write transaction. If bit-0 is ’1’, then it is a read transaction. If bit-0 is a logic 1, then it is a read transaction. Table 2 below lists the functions of the bits in the I^{2}C sub-address.

TABLE 2: I^{2}C SUB-ADDRESS (REGISTER ADDRESS)

BIT	FUNCTION
7:6	Reserved
5:3	UART Internal Register Address A2:A0
2:1	UART Channel Select ’00’ = UART Channel, other values are reserved
0	Reserved

After the last read or write transaction, the I^{2}C-bus master will set the SCL signal back to its idle state (HIGH).

1.1.3 SPI Bus Interface

The SPI interface consists of four lines: serial clock (SCL), chip select (CS#), slave output (SO) and slave input (SI). The serial clock, slave output and slave input can be as fast as 26 Mbps. To access the device in the SPI mode, the CS# signal for the XR20M1280 is asserted by the SPI master, then the SPI master starts toggling the SCL signal with the appropriate transaction information. The first bit sent by the SPI master includes whether it is a read or write transaction and the UART register being accessed. See **Table 3** below.

TABLE 3: SPI FIRST BYTE FORMAT

BIT	FUNCTION
7	Read/Write# Logic 1 = Read Logic 0 = Write
6	Reserved
5:3	UART Internal Register Address A2:A0
2:1	UART Channel Select ’00’ = UART Channel, other values are reserved
0	Reserved

FIGURE 6. SPI WRITE

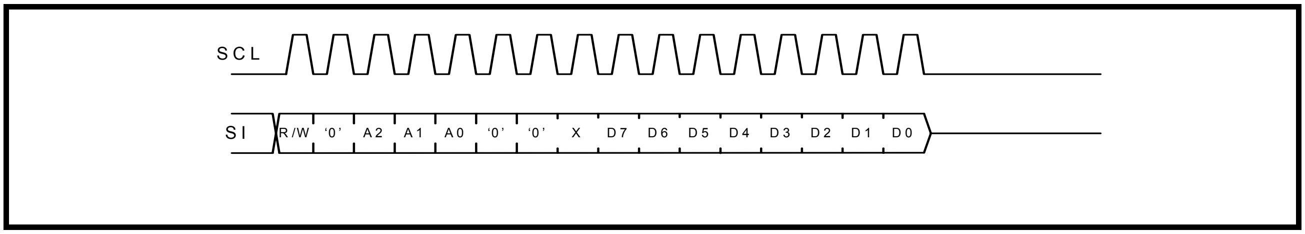
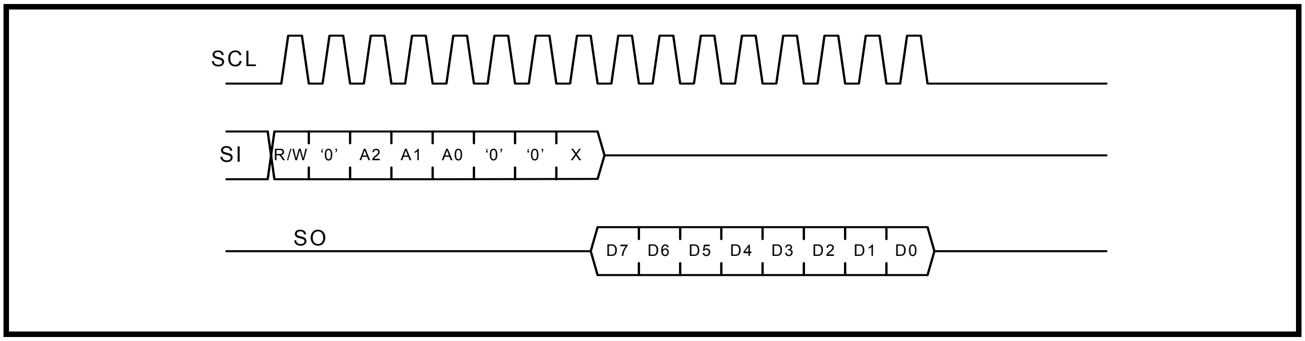


FIGURE 7. SPI READ



The 128 byte TX FIFO can be loaded with data or 128 byte RX FIFO data can be unloaded in one SPI write or read sequence.

FIGURE 8. SPI FIFO WRITE

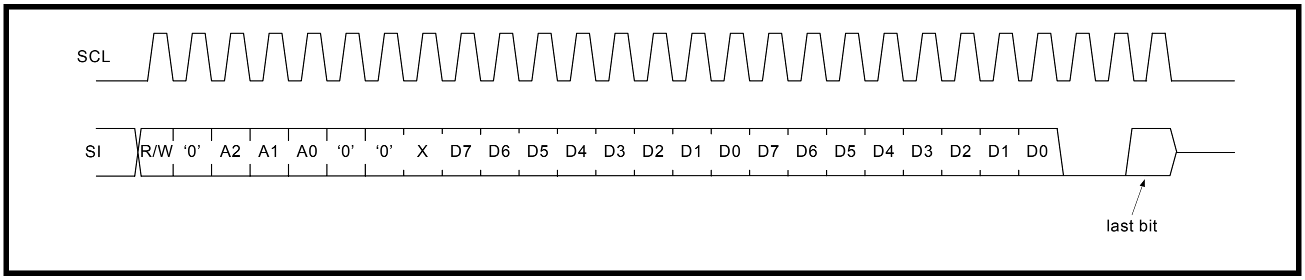
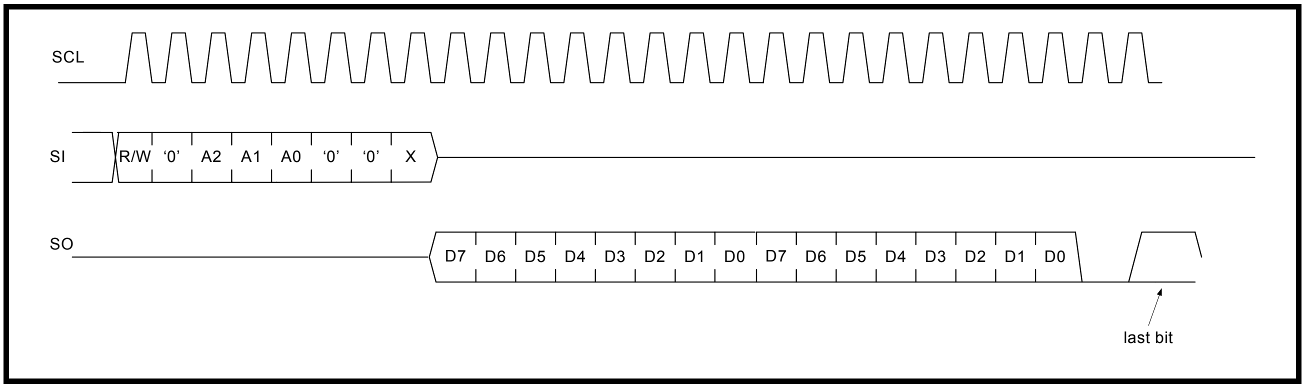


FIGURE 9. SPI FIFO READ

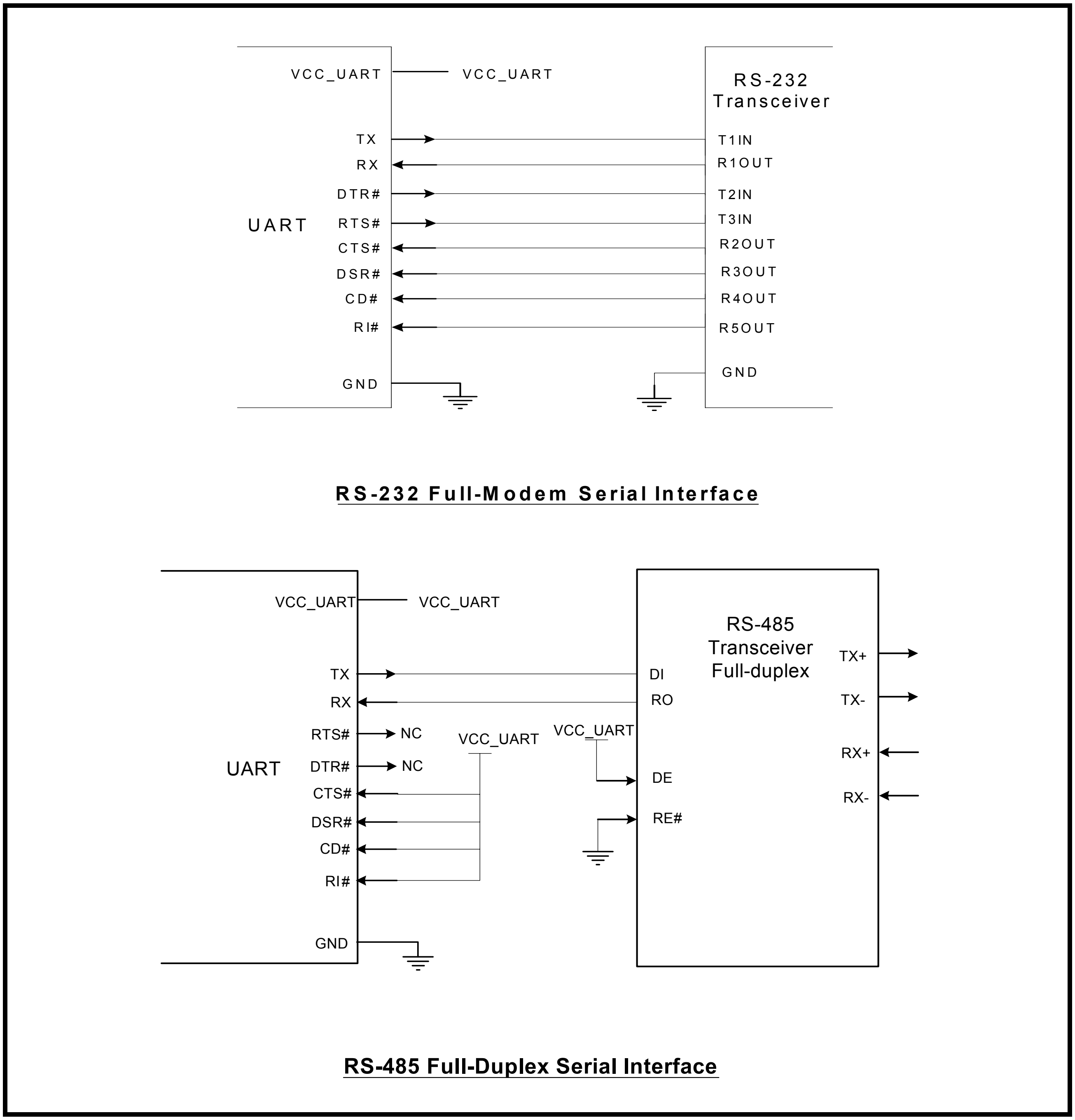


After the last read or write transaction, the SPI master will set the SCL signal back to its idle state (LOW).

1.2 Serial Interface

The M1280 is typically used with RS-232, RS-485 and IR transceivers. The following figure shows typical connections from the UART to the different transceivers. For more information on RS-232 and RS-485/422 transceivers, go to www.exar.com or send an e-mail to uarttechsupport@exar.com.

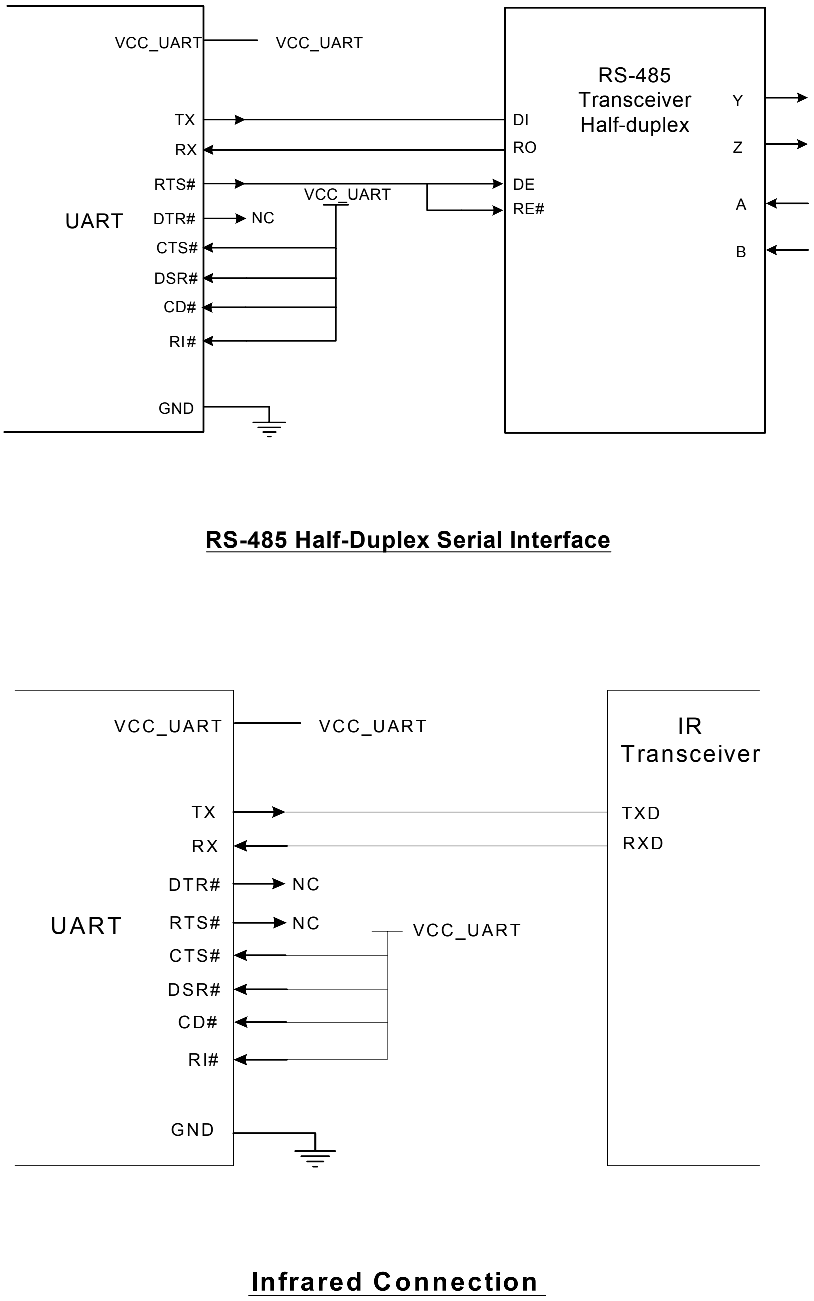
FIGURE 10. XR20M1280 TYPICAL SERIAL INTERFACE CONNECTIONS



RS-232 Full-Modem Serial Interface

RS-485 Full-Duplex Serial Interface

FIGURE 11. XR20M1280 TYPICAL SERIAL INTERFACE CONNECTIONS



1.3 Device Reset

The RESET# input resets the internal registers and the serial interface outputs to their default state (see **Table 21**). An active low pulse of longer than 40 ns duration will be required to activate the reset function in the device. Following a power-on reset or an external reset, the M1280 is software compatible with previous generation of UARTs.

1.4 5-Volt Tolerant Inputs

The M1280 can accept and withstand 5V signals on the inputs without any damage. But note that if the supply voltage for the M1280 is at the lower end of the supply voltage range (ie. 1.8V), its V_{OH} may not be high enough to meet the requirements of the V_{IH} of a CPU or a serial transceiver that is operating at 5V. Caution: XTAL1 is not 5 volt tolerant.

1.5 Internal Registers

The M1280 has a set of 16550 compatible registers for controlling, monitoring and data loading and unloading. These registers function as data holding registers (THR/RHR), interrupt status and control registers (ISR/IER), a FIFO control register (FCR), receive line status and control registers (LSR/LCR), modem status and control registers (MSR/MCR), programmable data rate (clock) divisor registers (DLL/DLM/DLD), and a user accessible scratchpad register (SPR).

Beyond the general 16C550 features and capabilities, the M1280 offers enhanced feature registers (EFR, Xon1/Xoff1, Xon2/Xoff2, DLD, FCTR, EMSR, FC and TRIG, SFR, SHR, GPIOINT, GPIO3T, GPIOINV, GPIOSEL) that provide automatic RTS and CTS hardware flow control, automatic Xon/Xoff software flow control, 9-bit (Multidrop) mode, auto RS-485 half duplex control, different baud rate for TX and RX and fractional baud rate generator. All the register functions are discussed in full detail later in **"Section 2.0, UART INTERNAL REGISTERS" on page 27**.

1.6 IRQ# Ouput

The IRQ# interrupt output changes according to the operating mode and enhanced features setup. **Table 4 and 5** summarize the operating behavior for the transmitter and receiver. Also see **Figure 33** through **35**.

TABLE 4: IRQ# PIN OPERATION FOR TRANSMITTER

	Auto RS485 Mode	FCR BIT-0 = 0 (FIFO DISABLED)	FCR BIT-0 = 1 (FIFO ENABLED)

IRQ# Pin	NO	HIGH = One byte in THR
		LOW = THR empty
			HIGH = FIFO above trigger level
			LOW = FIFO below trigger level or FIFO empty

IRQ# Pin	YES	HIGH = One byte in THR
		LOW = THR empty
			HIGH = FIFO above trigger level
			LOW = FIFO below trigger level or FIFO empty

TABLE 5: IRQ# PIN OPERATION FOR RECEIVER

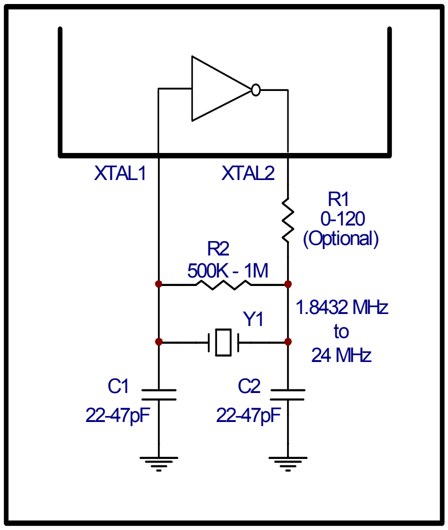
	FCR BIT-0 = 0 (FIFO DISABLED)	FCR BIT-0 = 1 (FIFO ENABLED)

IRQ# Pin	HIGH = One byte in THR
	LOW = RHR empty
		HIGH = FIFO above trigger level
		LOW = FIFO above trigger level or RX Data Timeout

1.7 Crystal Oscillator or External Clock Input

The M1280 includes an on-chip oscillator to produce a clock for the baud rate generators in the device when a crystal is connected between XTAL1 and XTAL2 as shown below. The CPU data bus does not require this clock for bus operation. The crystal oscillator provides a system clock to the Baud Rate Generators (BRGs) in the UART. XTAL1 is the input to the oscillator or external clock buffer input with XTAL2 pin being the output. For programming details, see **"Section 1.8, Programmable Baud Rate Generator with Fractional Divisor" on page 15**.

FIGURE 12. TYPICAL CRYSTAL CONNECTIONS



The on-chip oscillator is designed to use an industry standard microprocessor crystal (parallel resonant, fundamental frequency with 10-22 pF capacitance load, ESR of 20-120 ohms and 100ppm frequency tolerance) connected externally between the XTAL1 and XTAL2 pins. Typical oscillator connections are shown in **Figure 12**. Alternatively, an external clock can be connected to the XTAL1 pin to clock the internal baud rate generator for standard or custom rates. For further reading on oscillator circuit, see application note DAN108 on EXAR's web site.

1.8 Programmable Baud Rate Generator with Fractional Divisor

The M1280 has independent Baud Rate Generators (BRGs) with prescalers for the transmitter and receiver. The prescalers are controlled by a software bit in the MCR register. The MCR register bit-7 sets the prescalers to divide the input crystal or external clock by 1 or 4. The output of the prescaler clocks to the BRG. The BRG further divides this clock by a programmable divisor between 1 and (2^{16} - 0.0625) in increments of 0.0625 (1/16) to obtain a 16X or 8X or 4X sampling clock of the serial data rate. The sampling clock is used by the transmitter for data bit shifting and receiver for data sampling.

The BRG divisor (DLL, DLM, and DLD registers) defaults to the value of ’1’ (DLL = 0x01, DLM = 0x00 and DLD = 0x00) during power-on reset. The DLL and DLM registers provide the integer part of the divisor and the DLD register provides the fractional part of the divisor. The four lower bits of the DLD are used to select a value from 0 (for setting 0000) to 0.9375 or 15/16 (for setting 1111). The divisor values can be calculated with the following equations:

Divisor = (XTAL1 clock frequency / prescaler) / (serial data rate * 16), with 16X mode, **DLD[5:4] = ’00’**
Divisor = (XTAL1 clock frequency / prescaler / (serial data rate * 8), with 8X mode, **DLD[5:4] = ’01’**
Divisor = (XTAL1 clock frequency / prescaler / (serial data rate * 4), with 4X mode, **DLD[5:4] = ’10’**

The BRG divisors can be calculated using the following formulas:

Integer Divisor = TRUNC (Divisor)
Fractional Divisor = Divisor - Integer Divisor

DLM = Integer Divisor / 256
DLL = Integer Divisor & 256
DLD = TRUNC(Fractional Divisor * 16)

In the formulas above, please note that TRUNC (N) = Integer Part of N. For example, TRUNC (5.6) = 5.

1.8.1 Fractional BRG Example

For example, if the crystal clock is 24MHz, prescaler is 1, and the sampling mode is 16X, the divisor for a baud rate of 38400bps would be:

Divisor = (24000000 / 1) / (38400 * 16) = 39.0625

Integer Divisor = TRUNC (39.0625) = 39

Fractional Divisor = 39.0625 - 39 = 0.0625

DLM = 39 / 256 = 0 = 0x00

DLL = 39 & 256 = 39 = 0x27

DLD = 0.0625 * 16 = 1 = 0x1

Table 6 shows the standard data rates available with a 24MHz crystal or external clock at 16X clock rate. If the pre-scaler is used (MCR bit-7 = 1), the output data rate will be 4 times less than that shown in Table 6. At 8X sampling rate, these data rates would double. And at 4X sampling rate, they would quadruple. Also, when using 8X sampling mode, please note that the bit-time will have a jitter (+/- 1/16) whenever the DLD is non-zero and is an odd number.

1.8.2 Independent TX/RX BRG

The XR20M1280 has two independent sets of TX and RX baud rate generator. See Figure 13. TX and RX can use different baud rates by setting DLD, DLL and DLM register. For example, TX can transmit data to the remote UART at 9600 bps while RX receives data from remote UART at 921.6 Kbps. For the baud rate setting, **See ”Section 3.15, Baud Rate Generator Registers (DLL, DLM and DLD) - Read/Write” on page 44.**

Figure 13. Baud Rate Generator

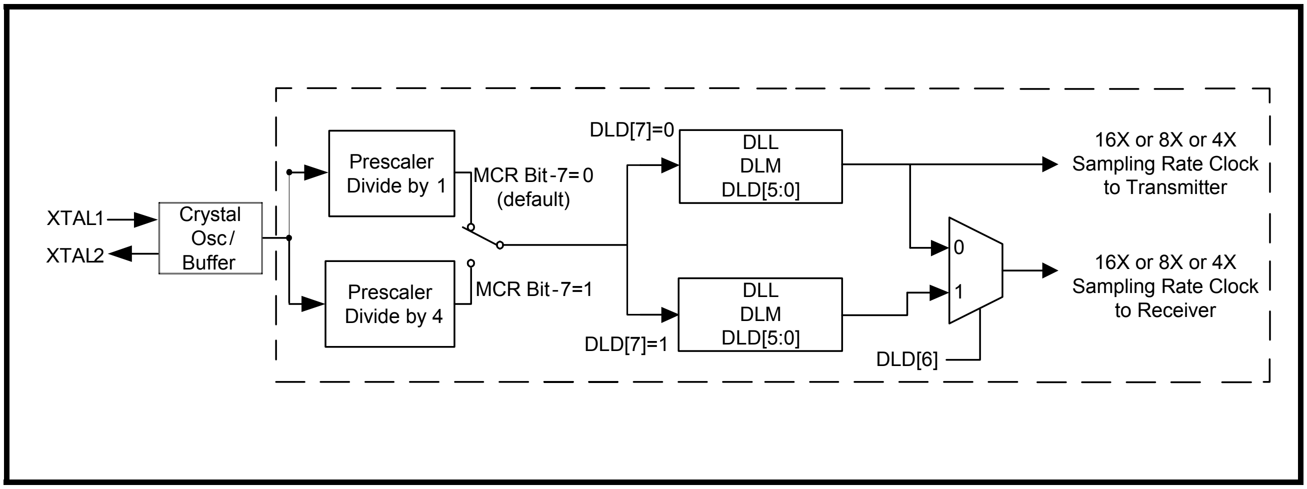


Table 6: Typical data rates with a 24 MHz crystal or external clock at 16X Sampling

Required Output Data Rate	Divisor for 16x Clock (Decimal)	DLM Program Value (HEX)	DLL Program Value (HEX)	DLD Program Value (HEX)	Data Error Rate (%)
400	3750	E	A6	0	0
2400	625	2	71	0	0
4800	312.5	1	38	8	0
9600	156.25	0	9C	4	0
10000	150	0	96	0	0
19200	78.125	0	4E	2	0
25000	60	0	3C	0	0
28800	52.0833	0	34	1	0.04
38400	39.0625	0	27	1	0
50000	30	0	1E	0	0
57600	26.0417	0	1A	0	0.08
75000	20	0	14	0	0
100000	15	0	F	0	0
115200	13.0208	0	D	0	0.16
153600	9.7656	0	9	C	0.16
200000	7.5	0	7	8	0
225000	6.6667	0	6	A	0.31
230400	6.5104	0	6	8	0.16
250000	6	0	6	0	0
300000	5	0	5	0	0
400000	3.75	0	3	C	0
460800	3.2552	0	3	4	0.16
500000	3	0	3	0	0
750000	2	0	2	0	0
921600	1.6276	0	1	A	0.16
1000000	1.5	0	1	8	0

1.9 Transmitter

The transmitter section comprises of an 8-bit Transmit Shift Register (TSR) and 128 bytes of FIFO which includes a byte-wide Transmit Holding Register (THR). TSR shifts out every data bit with the 16X/8X/4X internal clock. A bit time is 16/8/4 clock periods. The transmitter sends the start-bit followed by the number of data bits, inserts the proper parity-bit if enabled, and adds the stop-bit(s). The status of the FIFO and TSR are reported in the Line Status Register (LSR bit-5 and bit-6).

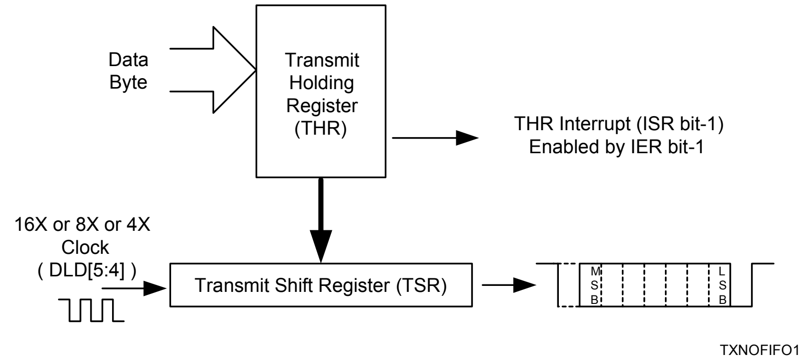
1.9.1 Transmit Holding Register (THR) - Write Only

The transmit holding register is an 8-bit register providing a data interface to the host processor. The host writes transmit data byte to the THR to be converted into a serial data stream including start-bit, data bits, parity-bit and stop-bit(s). The least-significant-bit (Bit-0) becomes first data bit to go out. The THR is the input register to the transmit FIFO of 128 bytes when FIFO operation is enabled by FCR bit-0. Every time a write operation is made to the THR, the FIFO data pointer is automatically bumped to the next sequential data location.

1.9.2 Transmitter Operation in non-FIFO Mode

The host loads transmit data to THR one character at a time. The THR empty flag (LSR bit-5) is set when the data byte is transferred to TSR. THR flag can generate a transmit empty interrupt (ISR bit-1) when it is enabled by IER bit-1. The TSR flag (LSR bit-6) is set when TSR becomes completely empty.

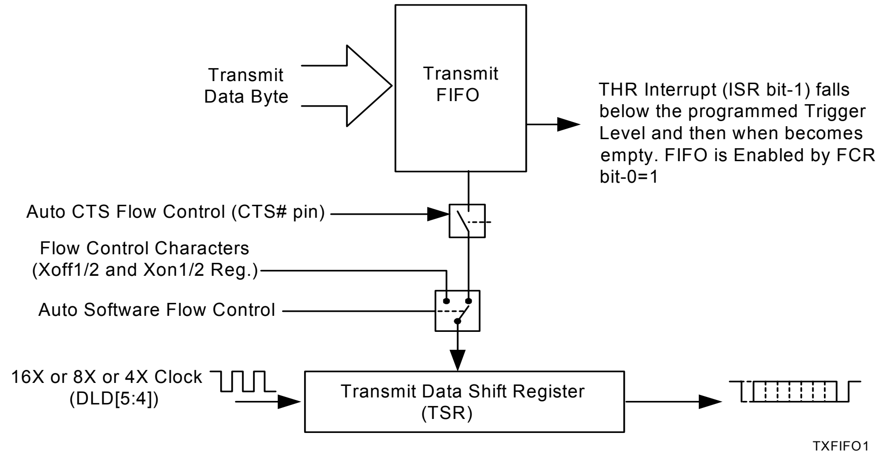
FIGURE 14. TRANSMITTER OPERATION IN NON-FIFO MODE



1.9.3 Transmitter Operation in FIFO Mode

The host may fill the transmit FIFO with up to 128 bytes of transmit data. The THR empty flag (LSR bit-5) is set whenever the FIFO is empty. The THR empty flag can generate a transmit empty interrupt (ISR bit-1) when the FIFO becomes empty. The transmit empty interrupt is enabled by IER bit-1. The TSR flag (LSR bit-6) is set when TSR/FIFO becomes empty.

FIGURE 15. TRANSMITTER OPERATION IN FIFO AND FLOW CONTROL MODE


Transmit Data Byte
Transmit FIFO
THR Interrupt (ISR bit-1) falls below the programmed Trigger Level and then when becomes empty. FIFO is Enabled by FCR bit-0=1
Auto CTS Flow Control (CTS# pin)
Flow Control Characters (Xoff1/2 and Xon1/2 Reg.)
Auto Software Flow Control
16X or 8X or 4X Clock (DLD[5:4])
Transmit Data Shift Register (TSR)
TXFIFO1

1.10 Receiver

The receiver section contains an 8-bit Receive Shift Register (RSR) and 128 bytes of FIFO which includes a byte-wide Receive Holding Register (RHR). The RSR uses the 16X/8X/4X clock (DLD[5:4]) for timing. It verifies and validates every bit on the incoming character in the middle of each data bit. On the falling edge of a start or false start bit, an internal receiver counter starts counting at the 16X/8X/4X clock rate. After 8 clocks (or 4 if 8X or 2 if 4X) the start bit period should be at the center of the start bit. At this time the start bit is sampled and if it is still a logic 0 it is validated. Evaluating the start bit in this manner prevents the receiver from assembling a false character. The rest of the data bits and stop bits are sampled and validated in this same manner to prevent false framing. If there were any error(s), they are reported in the LSR register bits 2-4. Upon unloading the receive data byte from RHR, the receive FIFO pointer is bumped and the error tags are immediately updated to reflect the status of the data byte in RHR register. RHR can generate a receive data ready interrupt upon receiving a character or delay until it reaches the FIFO trigger level. Furthermore, data delivery to the host is guaranteed by a receive data ready time-out interrupt when data is not received for 4 word lengths as defined by LCR[1:0] plus 12 bits time. This is equivalent to 3.7-4.6 character times. The RHR interrupt is enabled by IER bit-0. See Figure 16 and Figure 17 below.

1.10.1 Receive Holding Register (RHR) - Read-Only

The Receive Holding Register is an 8-bit register that holds a receive data byte from the Receive Shift Register. It provides the receive data interface to the host processor. The RHR register is part of the receive FIFO of 128 bytes by 11-bits wide, the 3 extra bits are for the 3 error tags to be reported in LSR register. When the FIFO is enabled by FCR bit-0, the RHR contains the first data character received by the FIFO. After the RHR is read, the next character byte is loaded into the RHR and the errors associated with the current data byte are immediately updated in the LSR bits 2-4.

FIGURE 16. RECEIVER OPERATION IN NON-FIFO MODE

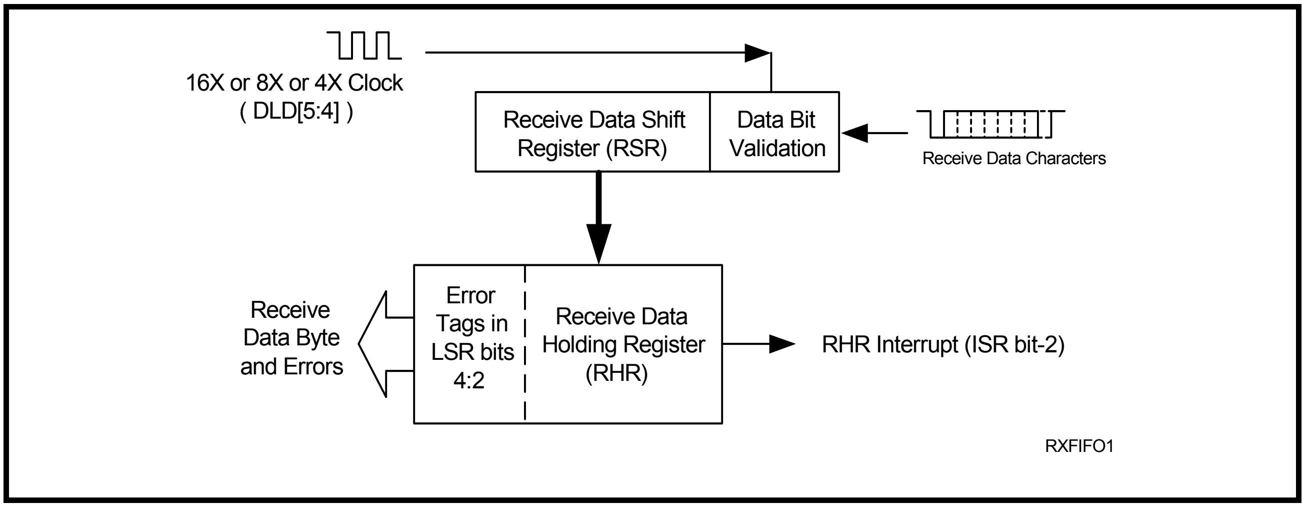
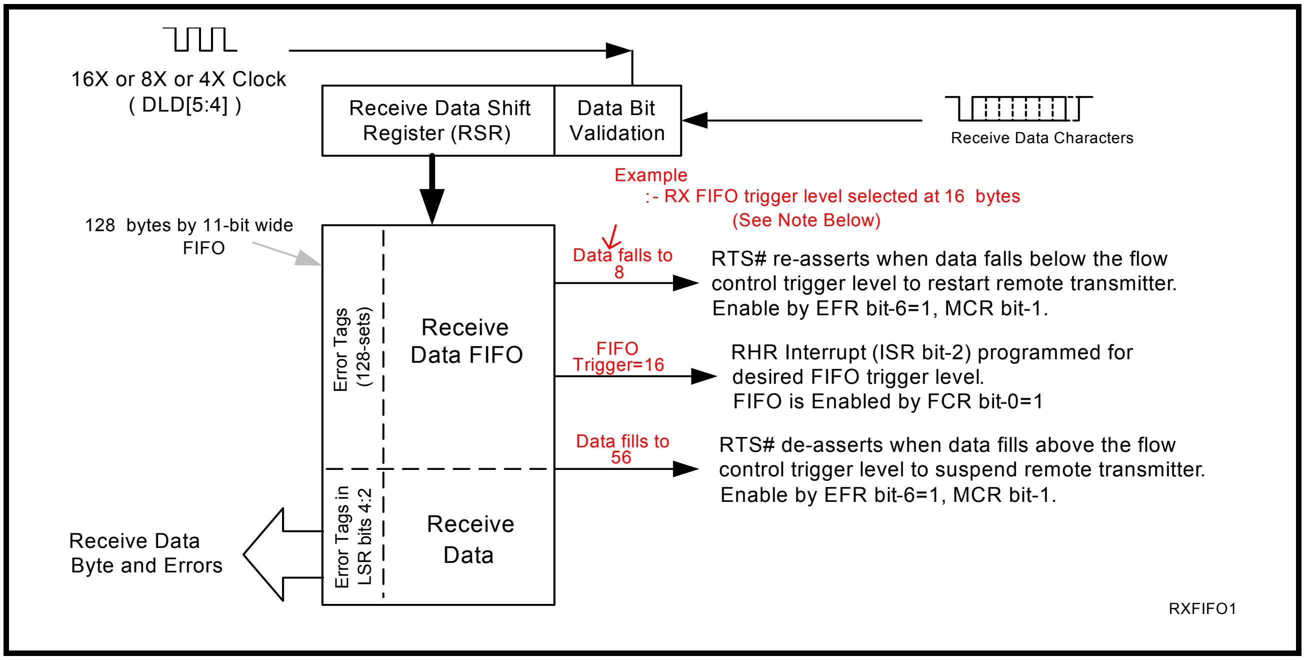

16X or 8X or 4X Clock
(DLD[5:4])
Receive Data Shift Register (RSR)
Data Bit Validation
Receive Data Characters
Receive Data Byte and Errors
Error Tags in LSR bits 4:2
Receive Data Holding Register (RHR)
RHR Interrupt (ISR bit-2)
RXFIFO1

FIGURE 17. RECEIVER OPERATION IN FIFO AND AUTO RTS FLOW CONTROL MODE


16X or 8X or 4X Clock
(DLD[5:4])
Receive Data Shift Register (RSR)
Data Bit Validation
Receive Data Characters
Example
:- RX FIFO trigger level selected at 16 bytes
(See Note Below)
128 bytes by 11-bit wide FIFO
Error Tags (128-sets)
Receive Data FIFO
Data falls to 8
RTS# re-asserts when data falls below the flow control trigger level to restart remote transmitter. Enable by EFR bit-6=1, MCR bit-1.
FIFO Trigger=16
RHR Interrupt (ISR bit-2) programmed for desired FIFO trigger level. FIFO is Enabled by FCR bit-0=1
Data fills to 56
RTS# de-asserts when data fills above the flow control trigger level to suspend remote transmitter. Enable by EFR bit-6=1, MCR bit-1.
Error Tags in LSR bits 4:2
Receive Data
Receive Data Byte and Errors
RXFIFO1

1.11 Auto RTS (Hardware) Flow Control

Automatic RTS hardware flow control is used to prevent data overrun to the local receiver FIFO. The RTS# output is used to request remote unit to suspend/resume data transmission. The auto RTS flow control features is enabled to fit specific application requirement (see **Figure 18**):

- Enable auto RTS flow control using EFR bit-6.
- The auto RTS function must be started by asserting RTS# output pin (MCR bit-1 to logic 1 after it is enabled).

If using the Auto RTS interrupt:

- Enable RTS interrupt through IER bit-6 (after setting EFR bit-4). The UART issues an interrupt when the RTS# pin makes a transition from low to high: ISR bit-5 will be set to logic 1.

1.12 Auto RTS Hysteresis

With the Auto RTS function enabled, an interrupt is generated when the receive FIFO reaches the selected RX trigger level. The RTS# pin will not be forced HIGH (RTS off) until the receive FIFO reaches one trigger level above the selected trigger level in the trigger table (**Table 13**). The RTS# pin will return LOW after the RX FIFO is unloaded to one level below the selected trigger level. Under the above described conditions, the M1280 will continue to accept data until the receive FIFO gets full. The Auto RTS function is initiated when the RTS# output pin is asserted LOW (RTS On). **Table 7** below explains this when Trigger Table-C (**Table 13**) is selected.

TABLE 7: AUTO RTS (HARDWARE) FLOW CONTROL

RX TRIGGER LEVEL	IRQ# PIN ACTIVATION	RTS# DE-ASSERTED (HIGH) (CHARACTERS IN RX FIFO)	RTS# ASSERTED (LOW) (CHARACTERS IN RX FIFO)
8	8	16	0
16	16	56	8
56	56	60	16
60	60	60	56

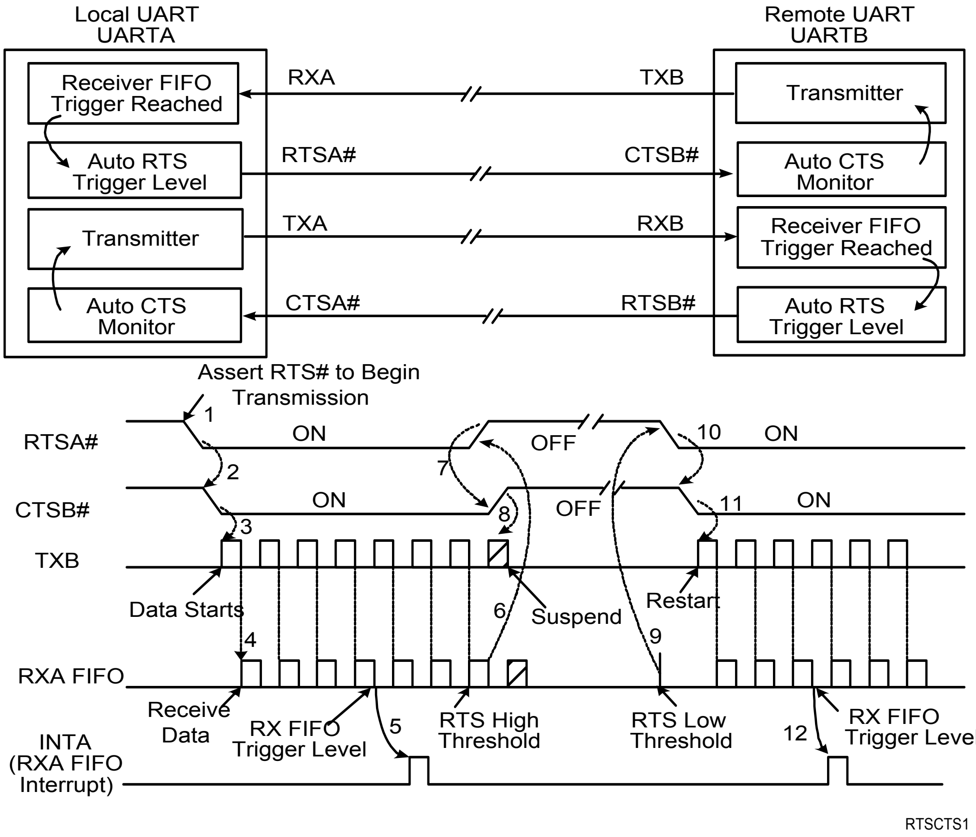
1.13 Auto CTS Flow Control

Automatic CTS flow control is used to prevent data overrun to the remote receiver FIFO. The CTS# input is monitored to suspend/restart the local transmitter. The auto CTS flow control feature is selected to fit specific application requirement (see Figure 18):

- Enable auto CTS flow control using EFR bit-7.

If needed, the CTS interrupt can be enabled through IER bit-7 (after setting EFR bit-4). The UART issues an interrupt when the CTS# pin is de-asserted (HIGH): ISR bit-5 will be set to 1, and UART will suspend transmission as soon as the stop bit of the character in process is shifted out. Transmission is resumed after the CTS# input is re-asserted (LOW), indicating more data may be sent.

FIGURE 18. AUTO RTS AND CTS FLOW CONTROL OPERATION



The local UART (UARTA) starts data transfer by asserting RTSA# (1). RTSA# is normally connected to CTSB# (2) of remote UART (UARTB). CTSB# allows its transmitter to send data (3). TXB data arrives and fills UARTA receive FIFO (4). When RXA data fills up to its receive FIFO trigger level, UARTA activates its RXA data ready interrupt (5) and continues to receive and put data into its FIFO. If interrupt service latency is long and data is not being unloaded, UARTA monitors its receive data fill level to match the upper threshold of RTS delay and de-assert RTSA# (6). CTSB# follows (7) and request UARTB transmitter to suspend data transfer. UARTB stops or finishes sending the data bits in its transmit shift register (8). When receive FIFO data in UARTA is unloaded to match the lower threshold of RTS delay (9), UARTA re-asserts RTSA# (10), CTSB# recognizes the change (11) and restarts its transmitter and data flow again until next receive FIFO trigger (12). This same event applies to the reverse direction when UARTA sends data to UARTB with RTSB# and CTSA# controlling the data flow.

1.14 Auto Xon/Xoff (Software) Flow Control

When software flow control is enabled (**See Table 20**), the M1280 compares one or two sequential receive data characters with the programmed Xon or Xoff-1,2 character value(s). If receive character(s) (RX) match the programmed values, the M1280 will halt transmission (TX) as soon as the current character has completed transmission. When a match occurs, the Xoff (if enabled via IER bit-5) flag will be set and the interrupt output pin will be activated. Following a suspension due to a match of the Xoff character, the M1280 will monitor the receive data stream for a match to the Xon-1,2 character. If a match is found, the M1280 will resume operation and clear the flags (ISR bit-4).

Reset initially sets the contents of the Xon/Xoff 8-bit flow control registers to a logic 0. Following reset the user can write any Xon/Xoff value desired for software flow control. Different conditions can be set to detect Xon/Xoff characters (**See Table 20**) and suspend/resume transmissions. When double 8-bit Xon/Xoff characters are selected, the M1280 compares two consecutive receive characters with two software flow control 8-bit values (Xon1, Xon2, Xoff1, Xoff2) and controls TX transmissions accordingly. Under the above described flow control mechanisms, flow control characters are not placed in the RX FIFO.

In the event that the receive buffer is overfilling and flow control needs to be executed, the M1280 automatically sends an Xoff message (when enabled) via the serial TX output to the remote modem. The M1280 sends the Xoff-1,2 characters two-character-times (= time taken to send two characters at the programmed baud rate) after the receive FIFO crosses the programmed trigger level. To clear this condition, the M1280 will transmit the programmed Xon-1,2 characters as soon as receive FIFO is less than one trigger level below the programmed trigger level. **Table 8** below explains this when Trigger Table-C is selected.

TABLE 8: AUTO XON/XOFF (SOFTWARE) FLOW CONTROL

RX TRIGGER LEVEL	IRQ# PIN ACTIVATION	XOFF CHARACTER(S) SENT (CHARACTERS IN RX FIFO)	XON CHARACTER(S) SENT (CHARACTERS IN RX FIFO)
8	8	8*	0
16	16	16*	8
56	56	56*	16
60	60	60*	56

* *After the trigger level is reached, an xoff character is sent after a short span of time (= time required to send 2* characters); for example, after 2.083ms has elapsed for 9600 baud and 10-bit word length setting.

1.15 Special Character Detect

A special character detect feature is provided to detect an 8-bit character when bit-5 is set in the Enhanced Feature Register (EFR). When this character (Xoff2) is detected, it will be placed in the FIFO along with normal incoming RX data.

The M1280 compares each incoming receive character with Xoff-2 data. If a match exists, the received data will be transferred to the RX FIFO and ISR bit-4 will be set to indicate detection of special character. Although the Internal Register Table shows Xon, Xoff Registers with eight bits of character information, the actual number of bits is dependent on the programmed word length. Line Control Register (LCR) bits 0-1 defines the number of character bits, i.e., either 5 bits, 6 bits, 7 bits, or 8 bits. The word length selected by LCR bits 0-1 also determines the number of bits that will be used for the special character comparison. Bit-0 in the Xon, Xoff Registers corresponds with the LSB bit for the receive character.

1.16 Auto RS485 Half-Duplex Control Operation

The auto RS485 half-duplex direction control feature can be enabled by FCTR bit [3]. The RTS# pin becomes the half-duplex control output when this feature has been enabled. The RTS# pin is typically connected to both the Driver Enable (DE) and Receiver Enable (RE) of an RS-485 transceiver. When the Transmitter is idle, the RTS# pin is de-asserted so that the RS-485 driver is disabled and the RS-485 receiver is enabled. When data is loaded into the TX FIFO, the RTS# pin is asserted to enable the RS-485 driver and disable the RS-485 receiver. This changes the transmitter empty interrupt to TSR empty instead of THR empty.

1.16.1 RS-485 Setup Time

By default, the RTS# pin is asserted immediately before there is data on the TX output pin. For faster baud rates, it may be possible that data is lost due to a long start-up time for an RS-485 transceiver. The M1280 can delay the data from 0-15 bit times to allow the RS-485 transceiver to start up (**See ”Section , SHR[7:4]: RS-485 Setup Delay” on page 40.**).

1.16.2 RS-485 Turn-Around Delay

At the end of sending data, the RTS# pin is de-asserted immediately after the TX pin goes idle. The RTS# pin can be programmed to delay the RTS# from being asserted from 0-15 bit times (**See ”Section , SHR[3:0]: RS-485 Turn-Around Delay / Auto RTS Hysteresis” on page 40.**). The delay optimizes the time needed for the last transmission to reach the farthest station on a long cable network before switching off the line driver.

1.17 Normal Multidrop (9-bit) Mode - Receiver

Normal multidrop mode is enabled when SFR[6] = 1 (requires EFR[4] = 1) and EFR[5] = 0 (Special Character Detect disabled). The receiver is set to Force Parity 0 (LCR[5:3] = ’111’) in order to detect address bytes.

With the receiver initially disabled, it ignores all the data bytes (parity bit = 0) until an address byte is received (parity bit = 1). This address byte will cause the UART to set the parity error. The UART will generate an LSR interrupt and place the address byte in the RX FIFO. The software then examines the byte and enables the receiver if the address matches its slave address, otherwise, it does not enable the receiver.

If the receiver has been enabled, the receiver will receive the subsequent data. If an address byte is received, it will generate an LSR interrupt. The software again examines the byte and if the address matches its slave address, it does not have to do anything. If the address does not match its slave address, then the receiver should be disabled.

1.17.1 Auto Address Detection - Receiver

Auto address detection mode is enabled when SFR[6] = 1 (requires EFR[4] = 1) and EFR bit-5 = 1. The desired slave address will need to be written into the XOFF2 register. The receiver will try to detect an address byte that matches the porgrammed character in the XOFF2 register. If the received byte is a data byte or an address byte that does not match the programmed character in the XOFF2 register, the receiver will discard these data. Upon receiving an address byte that matches the XOFF2 character, the receiver will be automatically enabled if not already enabled, and the address character is pushed into the RX FIFO along with the parity bit (in place of the parity error bit). The receiver also generates an LSR interrupt. The receiver will then receive the subsequent data. If another address byte is received and this address does not match the programmed XOFF2 character, then the receiver will automatically be disabled and the address byte is ignored. If the address byte matches XOFF2, the receiver will put this byte in the RX FIFO along with the parity bit in the parity error bit.

1.18 Multidrop (9-bit) Mode - Transmitter

This feature simplifies sending an address byte (9th bit = 1) and improves the efficiency of the transmit data routine for transmitting 9-bit data. In previous generation UARTs, the only way to send an address byte is by changing the parity to Forced 1 parity, load the address byte in the THR, wait for the byte to be transmitted, change the parity back to Forced 0 parity, then load data into the TX FIFO. In the XR20M1280, there’s no waiting required and no changing parity. The transmit routine can set SFR[7]=1, then write the address byte into the TX FIFO followed immediately by the data bytes. SFR[7] is self-clearing, therefore, if multiple address bytes need to be transmitted, then SFR[7] will need to be set prior to each address byte written into the TX FIFO. During initialization, the parity must be set to Force Parity 0 (LCR[5:3] = ’111’).

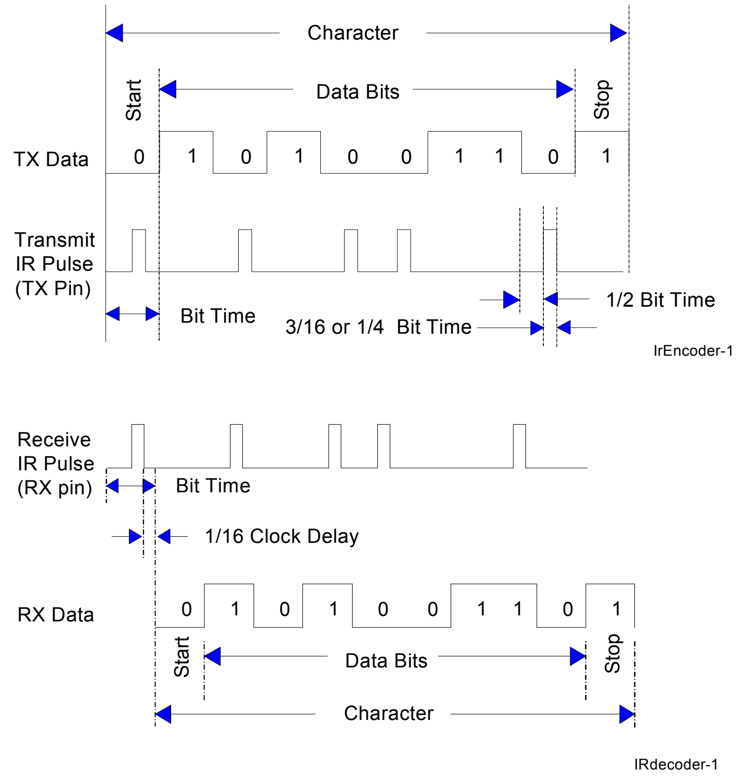
1.19 Infrared Mode

The M1280 UART includes the infrared encoder and decoder compatible to the IrDA (Infrared Data Association) version 1.0 and 1.1. The IrDA 1.0 standard that stipulates the infrared encoder sends out a 3/16 of a bit wide HIGH-pulse for each “0” bit in the transmit data stream with a data rate up to 115.2 Kbps. For the IrDA 1.1 standard, the infrared encoder sends out a 1/4 of a bit time wide HIGH-pulse for each "0" bit in the transmit data stream with a data rate up to 1.152 Mbps. This signal encoding reduces the on-time of the infrared LED, hence reduces the power consumption. See **Figure 19** below.

The infrared encoder and decoder are enabled by setting MCR register bit-6 to a ‘1’. With this bit enabled, the infrared encoder and decoder is compatible to the IrDA 1.0 standard. For the infrared encoder and decoder to be compatible to the IrDA 1.1 standard, SFR bit-3 will also need to be set to a ’1’ when EFR bit-4 is set to ’1’. Likewise, the RX input assumes an idle level of logic zero from a reset and power up, see **Figure 19**.

Typically, the wireless infrared decoder receives the input pulse from the infrared sensing diode on the RX pin. Each time it senses a light pulse, it returns a logic 1 to the data bit stream.

FIGURE 19. INFRARED TRANSMIT DATA ENCODING AND RECEIVE DATA DECODING



1.20 Sleep Mode with Auto Wake-Up

The M1280 supports low voltage system designs, hence, a sleep mode with auto wake-up feature is included to reduce its power consumption when the chip is not actively used.

1.20.1 Sleep mode - IER bit-4

All of these conditions must be satisfied for the M1280 to enter sleep mode:

- no interrupts pending (ISR bit-0 = 1)
- sleep mode is enabled (IER bit-4 = 1)
- modem inputs are not toggling (MSR bits 0-3 = 0)
- RX input pin is idling HIGH in normal mode or LOW in infrared mode
- divisor is non-zero
- TX and RX FIFOs are empty

The M1280 stops its crystal oscillator to conserve power in the sleep mode. User can check the XTAL2 pin for no clock output as an indication that the device has entered the sleep mode.

The M1280 resumes normal operation by any of the following:

- a receive data start bit transition (HIGH to LOW)
- a data byte is loaded to the transmitter, THR or FIFO
- a change of logic state on any of the modem or general purpose serial inputs: CTS#, DSR#, CD#, RI#

If the M1280 is awakened by any one of the above conditions, it will return to the sleep mode automatically after all interrupting conditions have been serviced and cleared. If the M1280 is awakened by the modem inputs, a read to the MSR is required to reset the modem inputs. In any case, the sleep mode will not be entered while an interrupt is pending from any channel. The M1280 will stay in the sleep mode of operation until it is disabled by setting IER bit-4 to a logic 0.

A word of caution: owing to the starting up delay of the crystal oscillator after waking up from sleep mode, the first few receive characters may be lost. Also, make sure the RX pin is idling HIGH or “marking” condition during sleep mode. This may not occur when the external interface transceivers (RS-232, RS-485 or another type) are also put to sleep mode and cannot maintain the “marking” condition. To avoid this, the system design engineer can use a 47k ohm pull-up resistor on each of the RX input.

1.20.2 Sleep Mode - SLEEP pin

The M1280 has a new pin called the SLEEP pin that can be used instead of setting IER bit-4=1. The M1280 will enter the sleep mode when:

- the current byte in the TSR has completely shifted out
- the current byte in the RSR has been completely received

Under this condition, there could be data in the TX and RX FIFOs. Any data that is the TX and RX FIFOs when the SLEEP pin is asserted will not be affected. The only data that will be lost is any data that is still being received on the RX pin. The M1280 will only wake up after the SLEEP pin has been de-asserted.

1.20.3 Wake-up Interrupt

The M1280 has the wake up interrupt. By setting the FCR bit-3, wake up interrupt is enabled or disabled. The default status of wake up interrupt is disabled. Please **See ”Section 3.5, FIFO Control Register (FCR) - Write-Only” on page 34.**