GENERAL DESCRIPTION

The XR17V252[1] (V252) is a single chip 2-channel 66MHz PCI UART (Universal Asynchronous Receiver and Transmitter) solution, optimized for higher performance and lower power. The V252 device with its fifth generation register set is designed to meet the high bandwidth and power management requirements for multi-serial communication ports for system administration and management. The 32-bit 66MHz PCI interface is compliant with PCI 3.0 and PCI power management revision 1.1 specifications. The device provides an upgrade path for Exar's 33MHz 5V and Universal PCI UART family.

The V252 consists of two independent UART channels, each with set of configuration and enhanced registers, 64 bytes of Transmit (TX) and Receive (RX) FIFOs, and a fractional Baud Rate Generator (BRG). A global interrupt source register provides a complete interrupt status indication for both channels to speed up interrupt parsing. The V252 device operates at 33/66MHz and features fully programmable TX and RX FIFO level triggers, automatic hardware and software flow control, and automatic RS-485 half duplex direction control output for software and hardware design simplification.

NOTE 1: *Covered by U.S. Patents #5,649,122 and #5,949,787*

APPLICATIONS

- Remote Access Servers
- Storage Network Management
- Factory Automation and Process Control
- Instrumentation
- Multi-port RS-232/RS-422/RS-485 Cards
- Point-of-Sale Systems

FEATURES

- High performance 32-bit 66MHz PCI UART
- PCI 3.0 compliant
- PCI power management rev. 1.1 compliant
- EEPROM interface for PCI configuration
- 3.3V supply with 5V tolerant non-PCI (serial) inputs
- Data read/write burst operation
- Global interrupt register for both UART channels
- Up to 8 Mbps serial data rate
- Eight multi-purpose inputs/outputs
- A 16-bit general purpose timer/counter
- Sleep mode with wake-up Indicator
- Two independent UART channels controlled with
  - 16C550 compatible register Set
  - 64-byte TX and RX FIFOs with level counters and programmable trigger levels
  - Fractional baud rate generator
  - Automatic RTS/CTS or DTR/DSR hardware flow control with programmable hysteresis
  - Automatic Xon/Xoff software flow control
  - RS-485 half duplex direction control output with selectable turn-around delay
- Infrared (IrDA 1.0) data encoder/decoder

FIGURE 1. BLOCK DIAGRAM OF THE XR17V252

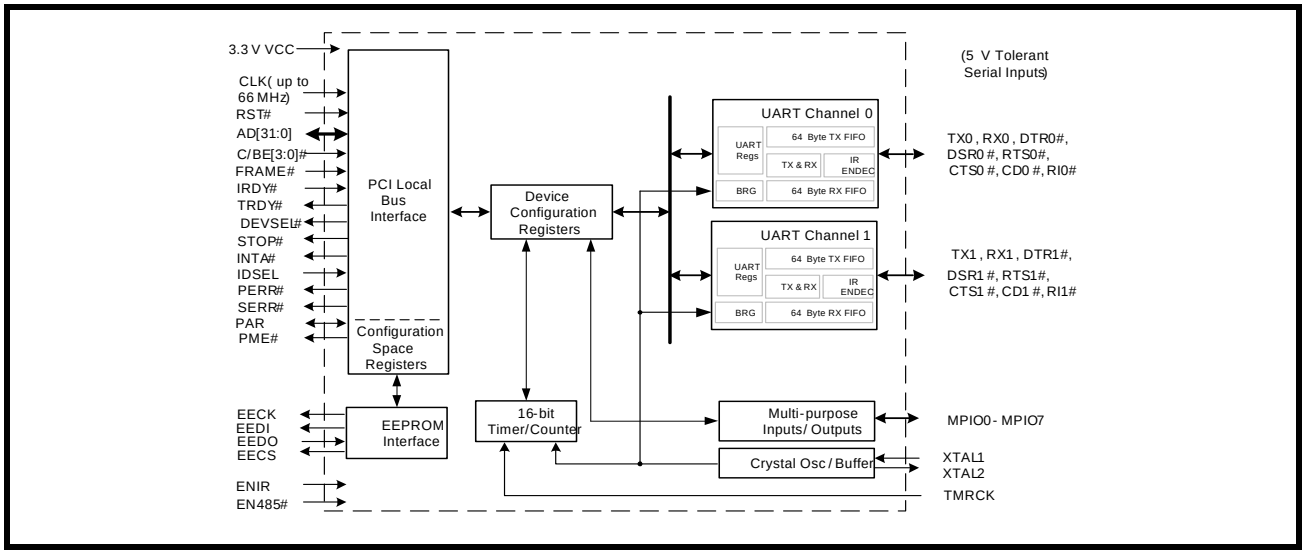
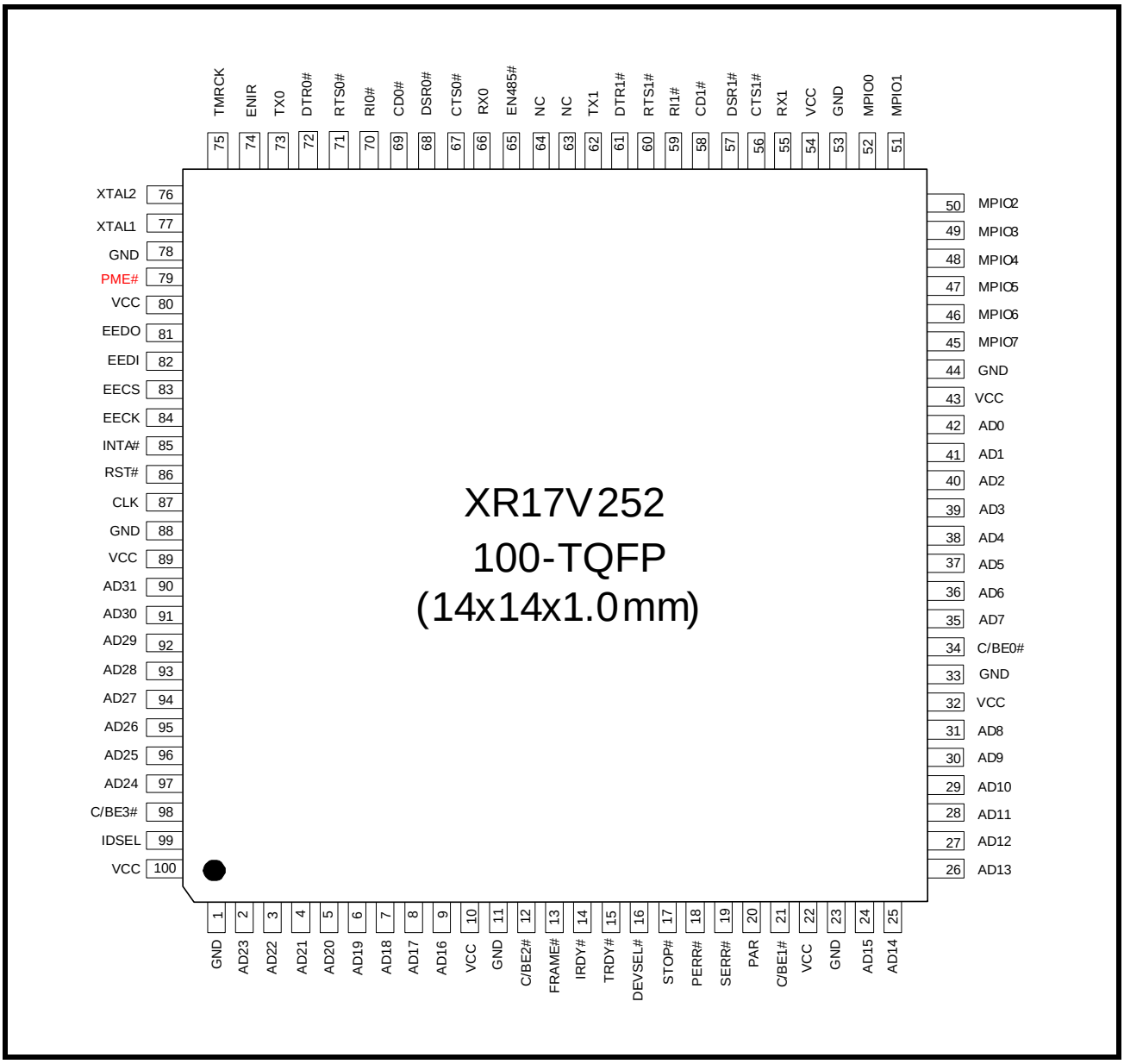


FIGURE 2. PIN OUT OF THE XR17V252



ORDERING INFORMATION

PART NUMBER	PACKAGE	OPERATING TEMPERATURE RANGE	DEVICE STATUS
XR17V252IM	100-Lead TQFP	-40°C to +85°C	Active

PIN DESCRIPTIONS

NAME	PIN #	TYPE	DESCRIPTION
PCI LOCAL BUS INTERFACE			
RST#	86	I	PCI Bus reset input (active low). It resets the PCI local bus configuration space registers, device configuration registers and UART channel registers to the default condition, see **Table 21**.
CLK	87	I	PCI Bus clock input of up to 66.67MHz.
AD31-AD24, AD23-AD16, AD15-AD8, AD7-AD0	90-97, 2-9, 24-31, 35-42	I/O	Address data lines [31:0] (bidirectional).
FRAME#	13	I	Bus transaction cycle frame (active low). It indicates the beginning and duration of an access.
C/BE3#-C/BE0#	98, 12, 21, 34	I	Bus Command/Byte Enable [3:0] (active low). This line is multiplexed for bus Command during the address phase and Byte Enables during the data phase.
IRDY#	14	I	Initiator Ready (active low). During a write, it indicates that valid data is present on data bus. During a read, it indicates the master is ready to accept data.
TRDY#	15	O	Target Ready (active low).
STOP#	17	O	Target request to stop current transaction (active low).
IDSEL	99	I	Initialization device select (active high).
DEVSEL#	16	O	Device select to the XR17V252 (active low).
INTA#	85	OD	Device interrupt from XR17V252 (open drain, active low).
PAR	20	I/O	Parity is even across AD[31:0] and C/BE[3:0]#. (bidirectional, active high).
PERR#	18	O	Data Parity error indicator, except for Special Cycle transactions (active low). Optional in bus target application.
SERR#	19	OD	System error indicator, Address parity or Data parity during Special Cycle transactions (open drain, active low). Optional in bus target application.
MODEM OR SERIAL I/O INTERFACE			
TX0	73	O	UART channel 0 Transmit Data or infrared transmit data. Normal TXD output idles HIGH while infrared TXD output idles LOW.
RX0	66	I	UART channel 0 Receive Data or infrared receive data. Normal RXD input idles HIGH while infrared RXD input idles LOW. In the infrared mode, the polarity of the incoming RXD signal can be selected via FCTR bit-4. If this bit is a logic 0, a LOW on the RXD input is considered a mark and if this bit is a logic 1, a HIGH on the RXD input is considered a space.
RTS0#	71	O	UART channel 0 Request to Send or general purpose output (active low). If this output is not used, leave it unconnected.
CTS0#	67	I	UART channel 0 Clear to Send or general purpose input (active low). This input should be connected to VCC when not used.
DTR0#	72	O	UART channel 0 Data Terminal Ready or general purpose output (active low). If this output is not used, leave it unconnected.

PIN DESCRIPTIONS

NAME	PIN #	TYPE	DESCRIPTION
DSR0#	68	I	UART channel 0 Data Set Ready or general purpose input (active low). This input should be connected to VCC when not used.
CD0#	69	I	UART channel 0 Carrier Detect or general purpose input (active low). This input should be connected to VCC when not used.
RI0#	70	I	UART channel 0 Ring Indicator or general purpose input (active low). This input should be connected to VCC when not used.
TX1	62	O	UART channel 1 Transmit Data or infrared transmit data. Normal TXD output idles HIGH while infrared TXD output idles LOW.
RX1	55	I	UART channel 1 Receive Data or infrared receive data. Normal RXD input idles HIGH while infrared RXD input idles LOW. In the infrared mode, the polarity of the incoming RXD signal can be selected via FCTR bit-4. If this bit is a logic 0, a LOW on the RXD input is considered a mark and if this bit is a logic 1, a HIGH on the RXD input is considered a space.
RTS1#	60	O	UART channel 1 Request to Send or general purpose output (active low). If this output is not used, leave it unconnected.
CTS1#	56	I	UART channel 1 Clear to Send or general purpose input (active low). This input should be connected to VCC when not used.
DTR1#	61	O	UART channel 1 Data Terminal Ready or general purpose output (active low). If this output is not used, leave it unconnected.
DSR1#	57	I	UART channel 1 Data Set Ready or general purpose input (active low). This input should be connected to VCC when not used.
CD1#	58	I	UART channel 1 Carrier Detect or general purpose input (active low). This input should be connected to VCC when not used.
RI1#	59	I	UART channel 1 Ring Indicator or general purpose input (active low). This input should be connected to VCC when not used.
ANCILLARY SIGNALS			
MPIO0-MPIO7	52-45	I/O	Multi-purpose inputs/outputs 0-7. The function of these pin are defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT
EECK	84	O	Serial clock to EEPROM. An internal clock of CLK divide by 256 is used for reading the vendor and sub-vendor ID during power up or reset. However, it can be manually clocked thru the Configuration Register REGB.
EECS	83	O	Chip select to a EEPROM device like 93C46. It is manually selectable thru the Configuration Register REGB. Requires a pull-up 4.7K ohm resister for external sensing of EEPROM during power up. See DAN112 for further details.
EEDI	82	O	Write data to EEPROM device. It is manually accessible thru the Configuration Register REGB. The V252 auto-configuration register interface logic uses the 16-bit format.
EEDO	81	I	Read data from EEPROM device. It is manually accessible thru the Configuration Register REGB.
XTAL1	77	I	Crystal or external clock input.
XTAL2	76	O	Crystal or buffered clock output.
TMRCK	75	I	16-bit timer/counter external clock input.

PIN DESCRIPTIONS

NAME	PIN #	TYPE	DESCRIPTION
ENIR	74	I	Global Infrared mode enable (active high). This pin is sampled during power up, following a hardware reset (RST#) or soft-reset (register RESET). It can be used to start up both UARTs in the infrared mode. The sampled logic state is transferred to MCR bit-6 in the UART. Software can override this pin thereafter and enable or disable it.
EN485#	65	I	Global AutoRS485 half-duplex direction control enable (active low). During power up or reset, this pin is sampled and if it is a logic high, both UARTs are set for Auto RS485 Mode. Also, the Auto RS485 bit, FCTR[5], is set in both channels. Software can override this pin thereafter and enable or disable it.
PME#	79	OD	Power Management Event signal. While in $D3_{hot}$ state, if the PME_Enable bit in the Power Management Control/Status Register is set, the V252 asserts the PME# upon receiving a new character or upon change of state of modem inputs on any channel.
VCC	54, 80, 10, 22, 32, 43, 89, 100	PWR	Power supply for the UART core logic and PCI bus I/O - 3.3V only. The V252 is PCI 3.0 signalling compliant at 3.3V operation. The non-PCI inputs (except XTAL1) are 5V tolerant. This includes all the serial (modem) inputs.
GND	1, 11, 23, 33, 44, 53, 78, 88	PWR	Power supply common, ground.
NC	63, 64		No Connection.

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

FUNCTIONAL DESCRIPTION

The XR17V252 (V252) consists of two enhanced 16550 UARTs with a conventional PCI interface and a non-volatile memory interface for PCI plug-and-play auto-configuration. The PCI local bus is a synchronous timing bus where all bus transactions are associated with the bus clock. The V252 supports 66MHz clock and 32-bit wide read and write data transfer operations including data burst mode through the PCI interface. Read and write data operations may be in byte, word or double-word (DWORD) format. The device consists of three sets of registers:

- PCI local bus configuration registers for PCI auto configuration
- 32-bit global device configuration registers for overall control and monitoring of the 2 UART channels.
- A combination set of the 16C550 compatible registers and enhanced registers in each of the individual UART channel, for control, status, and byte wide data transfer

Each UART channel has 64-byte FIFOs, automatic RTS/CTS or DTR/DSR hardware flow control with hysteresis control, automatic Xon/Xoff software flow control, programmable transmit and receive FIFO trigger level, FIFO level counters, infrared encoder and decoder (IrDA ver. 1.0), and a programmable fractional baud rate generator with a prescaler of 1X or 4X, and data rate up to 6.25 Mbps at 8X sampling clock.The XR17V252 is available in a 100-pin TQFP (14x14x1.0mm) industrial grade package.

PCI LOCAL BUS INTERFACE

This is the host interface and it meets the PCI local bus specification revision 3.0. The PCI local bus operations are synchronous, where each transaction is associated to the bus clock. The V252 can operate with the bus clock of up to a 66.67MHz. Data transfers operation can be formatted in 8-bit, 16-bit, 24-bit or 32-bit wide. With 32-bit data operations, it pushes the data transfer rate on the bus up to 264 MByte/sec. This increases the overall system's communication performance up to 32 times better than the 8-bit ISA bus. See PCI local bus specification revision 3.0 for bus operation details.

PCI LOCAL BUS CONFIGURATION SPACE REGISTERS

A set of PCI local bus configuration space register is provided. These registers provide the PCI local bus operating system with the card's vendor ID, device ID, sub-vendor ID, product model number, and resources and capabilities. The PCI local bus operating system collects this data from all the cards on the bus during the auto configuration phase that follows immediately after a power up or system reset/reboot. After it has sorted out all devices on the bus, it defines and download the operating conditions to the cards. One of the definitions is the base address loaded into the Base Address Register (BAR) where the card will be operating in the PCI local bus memory space. All this is described in more detail in **"Section 1.1, PCI LOCAL BUS CONFIGURATION SPACE REGISTERS" on page 7**.

POWER MANAGEMENT REGISTERS

This set of registers is a continuation of the Configuration Space and provides status and control of Power Management functions of the V252. The Power Management Capabilities (PMC) register and the Power Management Control/Status Register (PMCSR) are implemented. **"Section 1.2, Power Management Registers" on page 9** describes these registers and details how Power Management is implemented in the device.

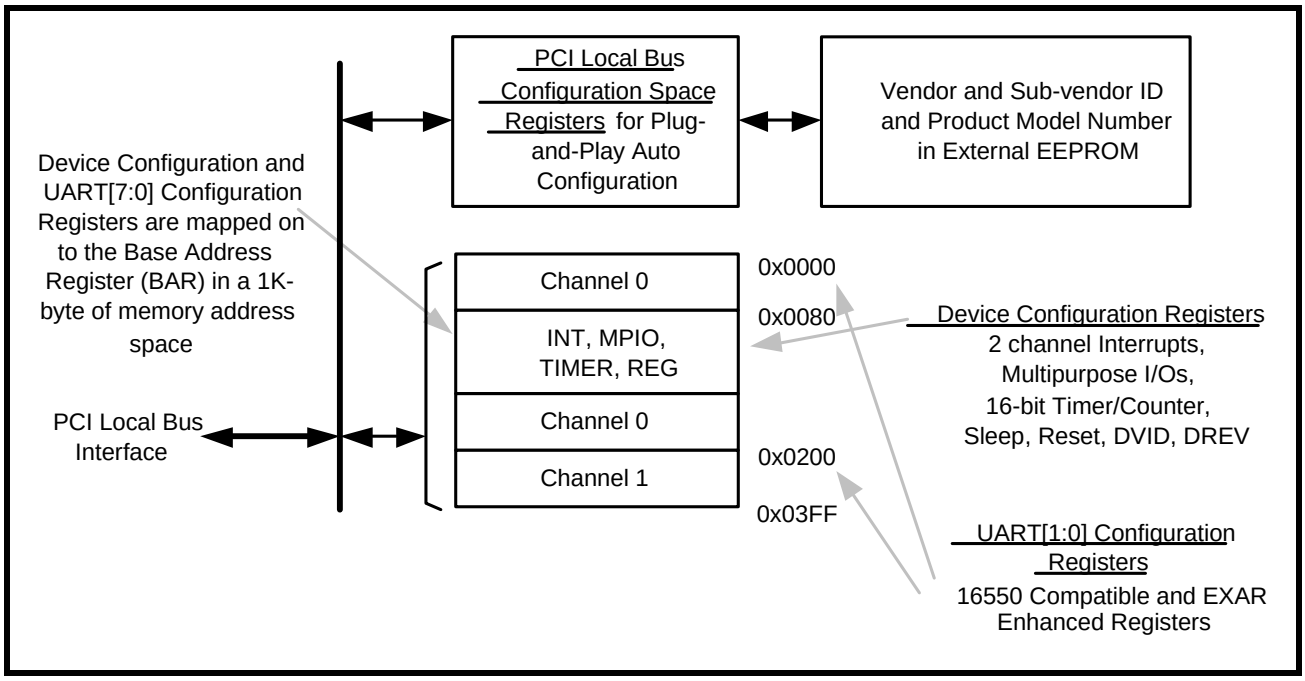
EEPROM INTERFACE

An external 93C46 EEPROM is used to store 8 words of information. Details of this information can be found in **"Section 1.4, EEPROM Interface" on page 12**. This information is only used with the plug-and-play auto configuration of the PCI local bus. These data provide automatic hardware installation onto the PCI bus. The EEPROM interface consists of 4 signals, EEDI, EEDO, EECS, and EECK. The EEPROM is not needed when auto configuration is not required in the application. However, if your design requires non-volatile memory for other purpose, it is possible to store and retrieve data on the EEPROM through a special PCI device configuration register. See application note DAN112 for its programming details.

1.0 XR17V252 INTERNAL REGISTERS

The XR17V252 UART has three different sets of registers as shown in Figure 3. The ***PCI Local Bus Configuration Space Registers*** are for plug-and-play auto-configuration when connecting the device to a the PCI bus. This auto-configuration feature makes installation very easy into a PCI system and it is part of the PCI local bus specification. The second register set is the ***Device Configuration Registers*** that are also accessible directly from the PCI bus for programming general operating conditions of the device and monitoring the status of various functions common to both channels. These functions include both channel UARTs' interrupt control and status, 16-bit general purpose timer control and status, multipurpose inputs/ outputs control and status, sleep mode, soft-reset, and device identification and revision. And lastly, each UART channel has its own set of internal ***UART Configuration Registers*** for its own operation control and status reporting. All 4 sets of channel registers are embedded inside the device configuration registers space, which provides faster access. The second and third set of registers are mapped into 1K of the PCI bus memory address space. The following paragraphs describe all 3 sets of registers in detail.

FIGURE 3. THE XR17V252 REGISTER SETS



1.1 PCI LOCAL BUS CONFIGURATION SPACE REGISTERS

The PCI local bus configuration space registers are responsible for setting up the device's operating environment in the PCI local bus. The pre-defined operating parameters of the device is read by the PCI bus plug-and-play auto-configuration manager in the operating system. After the PCI bus has collected all data from every device/card on the bus, it defines and downloads the memory mapping information to each device/ card about their individual operation memory address location and conditions. The operating memory mapped address location is downloaded into the Base Address Register (BAR) register, located at an address offset of 0x10 in the configuration space. Custom modification of certain registers is possible by using an external 93C46 EEPROM. The EEPROM contains the device vendor and sub-vendor data, along with 6 other words of information (see **"Section 1.4, EEPROM Interface" on page 12**) required by the auto-configuration setup.

TABLE 1: PCI LOCAL BUS CONFIGURATION SPACE REGISTERS

ADDRESS OFFSET	BITS	TYPE	DESCRIPTION	RESET VALUE (HEX OR BINARY)
0x00	31:16	EWR	Device ID (Exar device ID number)	0x0252
	15:0	EWR	Vendor ID (Exar) specified by PCISIG	0x13A8
0x04	31 30 29:28	RWC RWC RO	Parity error detected. Cleared by writing a logic 1. System error detected. Cleared by writing a logic 1. Unused	0b 0b 00b
	27	RO	Target Abort.	0b
	26:25	RO	DEVSEL# timing.	00b
	24	RO	Unemployments bus master error reporting bit	0b
	23	RO	Fast back to back transactions are supported	1b
	22	RO	Reserved Status bit	0b
	21	RO	66MHz capable	1b
	20	RO	Capabilities List	1b
	19:16	RO	Reserved Status bits	0000b
	15:9,7, 5,4,3,2	RO	Command bits (reserved)	0x0000
	8	RWR	SERR# driver enable. logic 1=enable driver and 0=disable driver	0b
	6	RWR	Parity error enable. logic 1=respond to parity error and 0=ignore	0b
	1	RWR	Command controls a device's response to mem space accesses: 0=disable mem space accesses, 1=enable mem space accesses	0b
	0	RO	Device's response to I/O space accesses is disabled. (0 = disable I/O space accesses)	0b
0x08	31:8	EWR	Class Code (Default is 'Simple 550 Communication Controller')	0x070002
	7:0	RO	Revision ID (Exar device revision number)	Current Rev. value
0x0C	31:24	RO	BIST (Built-in Self Test)	0x00
	23:16	RO	Header Type (a single function device with one BAR)	0x00
	15:8	RO	Unimplemented Latency Timer (needed only for bus master)	0x00
	7:0	RO	Unimplemented Cache Line Size	0x00
0x10	31:10	RWR	Memory Base Address Register (BAR)	0x00
	9:0	RO	Claims a 1K address space for the memory mapped UARTs	0x000
0x14	31:0	RO	Unimplemented Base Address Register (returns zeros)	0x00000000
0x18h	31:0	RO	Unimplemented Base Address Register (returns zeros)	0x00000000
0x1C	31:0	RO	Unimplemented Base Address Register (returns zeros)	0x00000000
0x20	31:0	RO	Unimplemented Base Address Register (returns zeros)	0x00000000

TABLE 1: PCI LOCAL BUS CONFIGURATION SPACE REGISTERS

ADDRESS OFFSET	BITS	TYPE	DESCRIPTION	RESET VALUE (HEX OR BINARY)
0x24	31:0	RO	Unimplemented Base Address Register (returns zeros)	0x00000000
0x28	31:0	RO	Reserved	0x00000000
0x2C	31:16	EWR	Subsystem ID (write from external EEPROM by customer)	0x0000
	15:0	EWR	Subsystem Vendor ID (write from external EEPROM by customer)	0x0000
0x30	31:0	RO	Expansion ROM Base Address (Unimplemented)	0x00000000
0x34	31:8	RO	Reserved (returns zeros)	0x000000
	7:0	RO	Capability Pointer (Implemented for Power Management)	0x40
0x38	31:0	RO	Reserved (returns zeros)	0x00000000
0x3C	31:24	RO	Unimplemented MAXLAT	0x00
	23:16	RO	Unimplemented MINGNT	0x00
	15:8	RO	Interrupt Pin, use INTA#.	0x01
	7:0	RWR	Interrupt Line.	0xXX

NOTE: EWR=Read/Write from external EEPROM. RWR=Read/Write from AD[31:0]. RO= Read Only. RWC=Read/Write-Clear.

1.2 Power Management Registers

The Power Management Registers are implemented in 2 DWORDs starting at address offset 0x40 of the PCI local bus configuration space. The bit definitions of these registers are shown in Table 2 below. The V252 complies with Revision 1.1 of the PCI Power Management Interface Specification.

TABLE 2: POWER MANAGEMENT REGISTERS

ADDRESS OFFSET	BITS	TYPE	DESCRIPTION	RESET VALUE (HEX OR BINARY)
0x40	31:16	See Below	Power Management Capabilities (PMC)	See Below
	31:27	RO	PME Support (PME# can be asserted from $D3_{hot}$ only)	01000b
	26:20	RO	Reserved or Not Supported	0000000b
	19	RO	PME Clock (PCI clock is required for PME# generation)	1b
	18:16	RO	Version	010b
	15:8	RO	Next Item Pointer	0x00
	7:0	RO	Capability ID	0x01
0x44	31:24	RO	Unimplemented Data Register	0x00
	23:16	RO	Unimplemented Bridge Support Extensions	0x00
	15:0	See Below	Power Management Control/Status Register (PMCSR)	See Below
	15	RWC	PME_Status	0b

TABLE 2: POWER MANAGEMENT REGISTERS

ADDRESS OFFSET	BITS	TYPE	DESCRIPTION	RESET VALUE (HEX OR BINARY)
	14:9	RO	Reserved	00000b
	8	RWR	PME_Enable	0b
	7:2	RO	Reserved	000000b
	1:0	RWR	PowerState	00b

NOTE: RWR=Read/Write from AD[31:0]. RO= Read Only. RWC=Read/Write-Clear.

1.2.1 Power States and Power State Transitions of the V252

The XR17V252 supports ***D0, D3$_{hot}$*** and ***D3$_{cold}$*** power states and is capable of generating the PME# signal from the ***D3$_{hot}$*** state. The following paragraphs describe these power states and **Figure 4** shows the power state transitions of the V252.

D0 STATE

The XR17V252 must be placed in the ***D0*** state before being used in a system. The ***D0*** state represents two states - ***D0*** Uninitalized and ***D0*** Active. Upon entering ***D0*** from power up or transition from ***D3$_{hot}$***, the V252 will be in the ***D0*** Uninitialized state. Once initialized by the system software, the V252 will enter the ***D0*** Active state. In the ***D0*** Active state, the V252 is fully functional and will respond to all PCI bus transactions as well as issue interrupts (INTA#). The system software can program the V252 to enter the ***D3$_{hot}$*** state from the ***D0*** state.

D3$_{HOT}$ STATE

The V252 enters the ***D3$_{hot}$*** state when the system software programs the V252 from ***D0*** to ***D3$_{hot}$***. In this state, the V252 will not be fully functional. The V252 will respond only to PCI configuration space accesses, if a PCI clock is provided and will not respond to PCI memory accesses nor will it issue interrupts. However, the V252 will continue to receive data and the automatic software and hardware flow control, if enabled, will continue to function normally. While in the ***D3$_{hot}$*** state, the V252 asserts the PME# (Power Management Event) signal, if enabled by setting PME_Enable bit, upon one of the following events:

- RX pin of any of the channels goes LOW (START bit detected), or
- Any of the delta bits of modem inputs (MSR register bits [3:0]) is set in any of the 2 channels (see **page 48**)

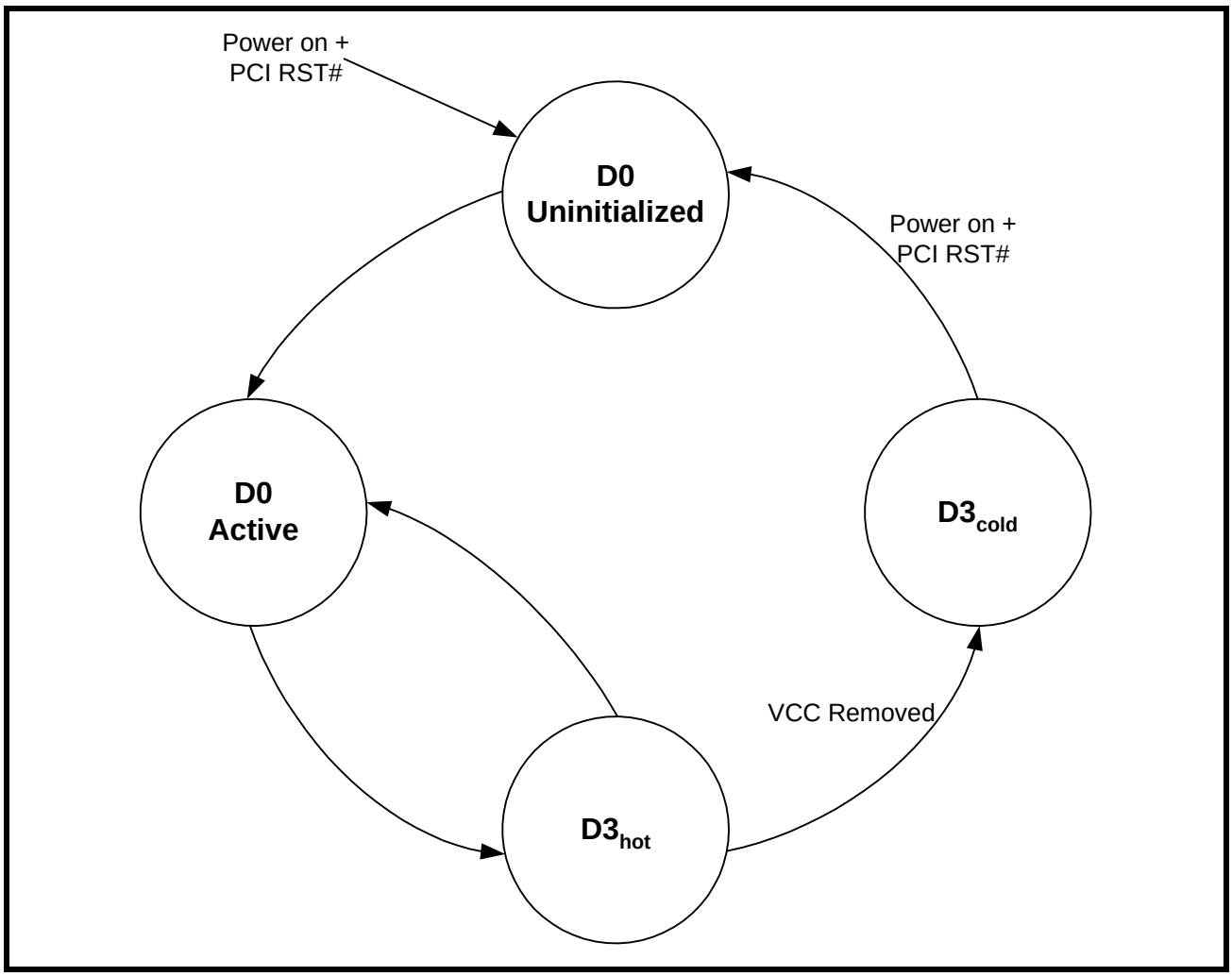
The V252 also sets the PME_Status bit when such an event occurs, regardless of whether the PME_Enable bit is set or not. The system software can reset the PME_Status bit by writing a '1' to it. When the system software programs the V252 from ***D3$_{hot}$*** to ***D0***, typically in response to the PME# signal, the V252 enters the ***D0*** Active state and will retain all the values of its internal registers. The V252 will keep its PCI signal drivers disabled for the duration of the ***D3$_{hot}$*** to ***D0*** Uninitialized state transition. The V252 saves the PME context (configuration registers and functional state information) in the ***D3$_{hot}$*** state.

Note: The V252 has a sleep mode which keeps the power consumption to a minimum (see Sleep Mode description on **page 20**). This is independent of the power state the V252 is in. The user can optionally place the V252 in sleep mode (via the software driver) in the Active ***D0*** state anytime or specifically when the system software commands the V252 to enter the ***D3$_{hot}$*** state. The crystal oscillator shuts down when the conditions given in Sleep Mode section on **page 20** are satisfied, and re-starts when one of the events as described in the same section occurs. Upon re-starting, the oscillator may take a long time to settle. This time may be more than 20ms which is the maximum wait time guaranteed by the system software before resuming normal PCI bus transactions in the Active ***D0*** state. Therefore, there may be data errors if the V252 is commanded to transmit data before the oscillator is ready. ***It is recommended not to use sleep mode while in the D3$_{hot}$ state for this reason.***

D3$_{COLD}$ STATE

The V252 enters the state when power is removed from the device. All context is lost in this state and the V252 does not support PME# in this state. When power is restored, PCI **RST#** must be asserted and the V252 will return to the ***D0*** Uninitialized state with a full PCI 3.0 compliant power-on reset sequence. The V252 will set all its registers and outputs to the power-on defaults just as at initial power up. The system software must then fully initialize and re-configure the V252 to place it in the ***D0*** Active state.

FIGURE 4. POWER STATE TRANSITIONS OF THE XR17V252



1.3 Special Read/Write Register to store User Information

This 32-bit register can be used to store user information and is writable only via the EEPROM. This is implemented at an offset of 0x48 in the PCI Configuration Space immediately following the Power Management Registers. This register can be used to store application-specific information which may be used by the device driver to initialize the device appropriately.

TABLE 3: SPECIAL READ/WRITE REGISTER

ADDRESS OFFSET	BITS	TYPE	DESCRIPTION	RESET VALUE (HEX)
0x48	31:0	EWR	User Information Writable only through EEPROM	0x00000000

NOTE: *EWR=Read/Write from external EEPROM.*

1.4 EEPROM Interface

The V252 provides an interface to an Electrically Erasable Programmable Read Only Memory (EEPROM). The EEPROM must be a 93C46-like device, with its memory configured as 16-bit words. This interface is provided in order to program the registers in the PCI Configuration Space of the PCI UART during power-up. The following table gives the mapping of the EEPROM memory to the registers in the V252's PCI Configuration Space. When the PCI RST# is negated, the V252 will download the data from the EEPROM, if it detects a HIGH on the EECS pin. The V252 takes a maximum of 2^{16} PCI clocks from the rising edge of the PCI RST# signal to read the EEPROM data. For more details on the EEPROM interface, please refer to the application note DAN112 on Exar's website.

TABLE 4: EEPROM ADDRESS DEFINITIONS

EEPROM MEMORY ADDRESS	EEPROM DATA [D15:D0]	V252'S PCI CONFIGURATION SPACE ADDRESS (WORD OFFSET)	DEFAULT VALUES
0x00	Vendor ID	0x00	0x13A8
0x01	Device ID	0x02	0x0252
0x02	Class Code*	0x08	0x0200
0x03	Class Code (Continued)	0x0A	0x0700
0x04	Subsystem Vendor ID	0x2C	0x0000
0x05	Subsystem ID	0x2E	0x0000
0x06	Special Register (Lower Word)	0x48	0x0000
0x07	Special Register (Upper Word)	0x4A	0x0000

*NOTE: * Only the upper 8 bits in this word in EEPROM location are used and the lower 8 bits are ignored. The lower byte at PCI Config space 0x08 is Device Revision and is read-only.*

1.5 Device Internal Register Sets

The ***Device Configuration Registers*** and the two individual ***UART Configuration Registers*** of the V252 occupy 2K of PCI bus memory address space. These addresses are offset onto the basic memory address, a value loaded into the Memory Base Address Register (BAR) in the PCI local bus configuration register set. The UART Configuration Registers are mapped into 4 address blocks where each UART channel occupies 512 bytes memory space for its own registers that include the 16550 compatible registers. The Device Configuration Registers are embedded inside the UART channel zero's address space between 0x0080 to 0x0093. All these registers can be accessed in 8, 16, 24 or 32 bits width depending on the starting address given by the host at beginning of the bus cycle. Transmit and receive data may be loaded or unloaded in 8, 16, 24 or 32 bits format in special locations given in the **Table 5** below. Every time a read or write operation is made to the transmit or receive register, its FIFO data pointer is automatically bumped to the next sequential data location either in byte, word or dword. One special case applies to the receive data unloading when reading the receive data together with its LSR register content. The host must read them in 16 or 32 bits format in order to maintain integrity of the data byte with its associated error flags. These special registers are further discussed in **"Section 3.1, FIFO DATA LOADING AND UNLOADING IN 32-BIT FORMAT" on page 25**.

TABLE 5: XR17V252 UART AND DEVICE CONFIGURATION REGISTERS

OFFSET ADDRESS	MEMORY SPACE	READ/WRITE	DATA WIDTH	COMMENT
0x000 - 0x00F	UART channel 0 Regs	(Table 13 & Table 14)	8/16/24/32	First 8 regs are 16550 compatible
0x010 - 0x07F	Reserved			
0x080 - 0x093	DEVICE CONFIGURATION REGISTERS	(Table 6)	8/16/24/32	
0x094 - 0x0FF	Reserved			
0x100	UART 0 – Read FIFO	Read-Only	8/16/24/32	64 bytes of RX FIFO data
0x100	UART 0 – Write FIFO	Write-Only	8/16/24/32	64 bytes of TX FIFO data
0x140 - 0x17F	Reserved			
0x180 - 0x1FF	UART 0 – Read FIFO with errors	Read-Only	16/32	64 bytes of RX FIFO data + LSR
0x200 - 0x20F	UART channel 1 Regs	(Table 13 & Table 14)	8/16//24/32	First 8 regs are 16550 compatible
0x210 - 0x2FF	Reserved			d
0x300	UART 1 – Read FIFO	Read-Only	8/16/24/32	64 bytes of RX FIFO data
0x300	UART 1 – Write FIFO	Write-Only	8/16/24/32	64 bytes of TX FIFO data
0x340 - 0x37F	Reserved			
0x380 - 0x3FF	UART 1 – Read FIFO with errors	Read-Only	16/32	64 bytes of RX FIFO data + LSR

1.6 Device Configuration Registers

The Device Configuration Registers provide easy programming of general operating parameters to the V252 and for monitoring the status of various functions. These registers control or report on all 4 channel UARTs functions that include interrupt control and status, 16-bit general purpose timer control and status, multipurpose inputs/outputs control and status, sleep mode control, soft-reset control, and device identification and revision, and others. Tables 6 and 7 below show these registers in BYTE and DWORD alignment. Each of these registers is described in detail in the following paragraphs.

TABLE 6: DEVICE CONFIGURATION REGISTERS SHOWN IN BYTE ALIGNMENT

ADDRESS [A7:A0]	REGISTER	READ/WRITE COMMENT	RESET STATE
Ox080	INT0 [7:0]	Read-only Interrupt [7:0]	Bits [7:0] = 0x00
Ox081	INT1 [15:8]	Read-only	Bits [7:0] = 0x00
Ox082	INT2 [23:16]	Read-only	Bits [7:0] = 0x00
Ox083	INT3 [31:24]	Read-only	Bits [7:0] = 0x00
Ox084	TIMERCNTL	Read/Write Timer Control	Bits [7:0] = 0x00
Ox085	TIMER	Reserved	Bits [7:0] = 0x00
Ox086	TIMERLSB	Read/Write Timer LSB	Bits [7:0]= 0x00
Ox087	TIMERMSB	Read/Write Timer MSB	Bits [7:0]= 0x00
Ox088	8XMODE	Read/Write	Bits [7:0] = 0x00
Ox089	REGA	Reserved	Bits [7:0] = 0x00
Ox08A	RESET	Write-only Self clear bits after executing Reset	Bits [7:0] = 0x00
Ox08B	SLEEP	Read/Write Sleep mode	Bits [7:0]= 0x00
Ox08C	DREV	Read-only Device revision	Bits [7:0] = Current Rev.
Ox08D	DVID	Read-only Device identification	Bits [7:0] = 0x38
Ox08E	REGB	Write-only	Bits [7:0] = 0x00
Ox08F	MPIOINT	Read/Write MPIO interrupt mask	Bits [7:0] = 0x00
Ox090	MPIOLVL	Read/Write MPIO level control	Bits [7:0] = 0x00
Ox091	MPIO3T	Read/Write MPIO output control	Bits [7:0] = 0x00
Ox092	MPIOINV	Read/Write MPIO input polarity select	Bits [7:0] = 0x00
Ox093	MPIOSEL	Read/Write MPIO select	Bits [7:0] = 0xFF

TABLE 7: DEVICE CONFIGURATION REGISTERS SHOWN IN DWORD ALIGNMENT

ADDRESS	REGISTER	BYTE 3 [31:24]	BYTE 2 [23:16]	BYTE 1 [15:8]	BYTE 0 [7:0]
0x080-083	INTERRUPT (read-only)	INT3	INT2[	INT1	INT0
0x084-087	TIMER (read/write)	TIMERMSB	TIMERLSB	TIMER (reserved)	TIMERCNTL
0x088-08B	ANCILLARY1 (read/write)	SLEEP	RESET	REGA	8XMODE
0x08C-08F	ANCILLARY2 (read-only)	MPIOINT	REGB	DVID	DREV
0x090-093	MPIO (read/write)	MPIOSEL	MPIOINV	MPIO3T	MPIOLVL

1.6.1 The Global Interrupt Register

The XR17V252 has a 32-bit wide register [INT0, INT1, INT2 and INT3] to provide interrupt information and supports two interrupt schemes. The first scheme uses bits 0 to 1 of an 8-bit indicator (INT0) representing channels 0 to 1 of the XR17V252, respectively. This permits the interrupt routine to quickly vector and serve that UART channel and determine the source(s) in each individual routines. INT0 bit-0 represents the interrupt status for UART channel 0 when its transmitter, receiver, line status, or modem port status requires service. INT0 bit-1 provides interrupt status for channel 1 and bits 2 to 7 are reserved and remain at a logic 0.

The second scheme provides detail about the source of the interrupts for each UART channel. All the interrupts are encoded into a 3-bit code per channel. This 3-bit code represents 7 interrupts corresponding to individual UART's transmitter, receiver, line status, modem port status. INT1 and INT2 registers provide the 6-bit interrupt status for both channels. Bits 8, 9 and 10 represents channel 0 and bits 11,12 and 13 represents channel 1. Bits 14 to 31 are reserved and remain at logic zero. Both channels interrupt status are available with a single DWORD read operation. This feature allows the host to quickly vector and serve the interrupts, reducing service interval, hence, reducing host bandwidth requirements. **Figure 5** shows the 4-byte interrupt register and its make up.

GLOBAL INTERRUPT REGISTER (DWORD) [default 0x00-00-00-00]

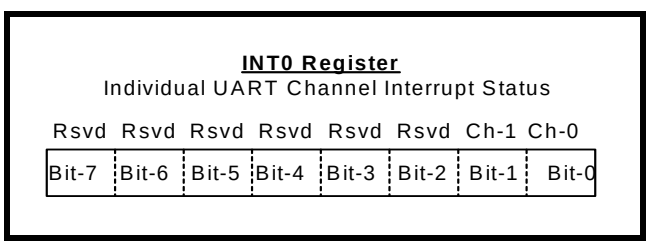
INT3 [31:24]	INT2 [23:16]	INT1 [15:8]	INT0 [7:0]

A special interrupt condition is generated by the V252 when it wakes up from sleep mode. This special interrupt is cleared by reading the INT0 register. If there are not any other interrupts pending, the value read from INT0 would be 0x00.

INT0 [7:0] Channel Interrupt Indicator

Each bit gives an indication of the channel that has requested for service. Bit-0 represents channel 0 and bit-1 indicates channel 1. Logic 1 indicates that a channel has requested for service. Bits 2 to 7 are reserved and remain at logic zero The interrupt bit clears after reading the appropriate register of the interrupting channel register, see Interrupt Clearing section.

The INT0 register provides status for each channel



Registers INT3, INT2 and INT1 [32:8]

Twenty four bit encoded interrupt indicator. Each channel's interrupt is encoded into 3 bits for receive, transmit, and status. Bit [10:8] represent channel 0 and channel 1 with bits [13:11]. The 3 bit encoding and their priority order are shown below in **Table 8**. The Timer and MPIO interrupts are for the device and therefore they exist within channel 0 (bits [10:8]) only..

FIGURE 5. THE GLOBAL INTERRUPT REGISTER, INT0, INT1, INT2 AND INT3

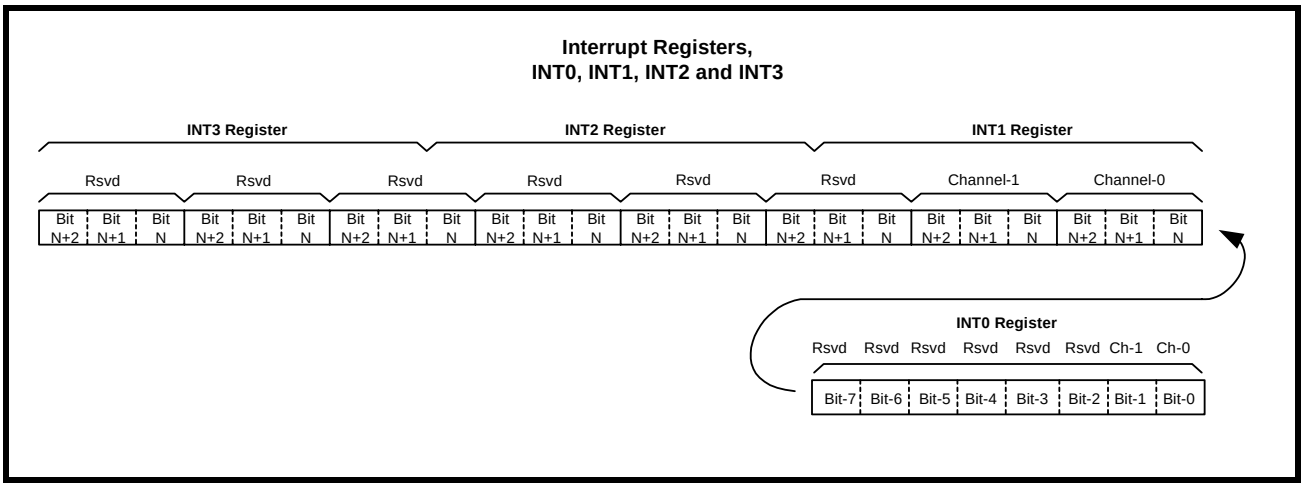


TABLE 8: UART CHANNEL [1:0] INTERRUPT SOURCE ENCODING

PRIORITY	BIT[N+2]	BIT[N+1]	BIT[N]	INTERRUPT SOURCE(S)
x	0	0	0	None or wake-up indicator
1	0	0	1	RXRDY and RX Line Status (logic OR of LSR[4:1])
2	0	1	0	RXRDY Time-out
3	0	1	1	TXRDY, THR or TSR (auto RS485 mode) empty
4	1	0	0	MSR, RTS/CTS or DTR/DSR delta or Xoff/Xon det. or special char. detected
5	1	0	1	Reserved.
6	1	1	0	MPIO pin(s). Available only in channel 0, reserved in channel 1.
7	1	1	1	TIMER Time-out. Available only in channel 0, reserved channel 1.

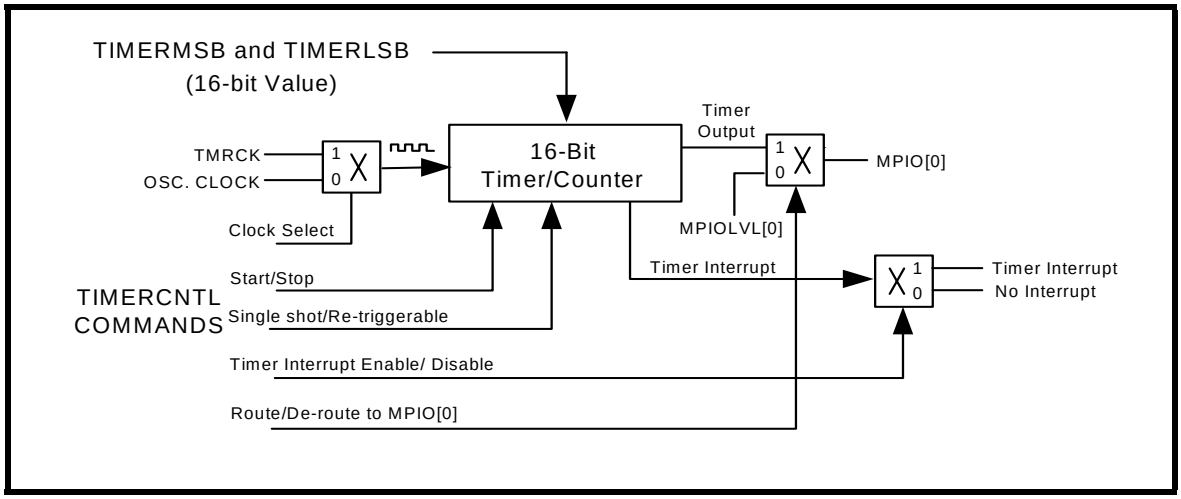
TABLE 9: UART CHANNEL [1:0] INTERRUPT CLEARING

RXRDY is cleared by reading data in the RX FIFO until it falls below the trigger level.
RXRDY Time-out is cleared by reading data until the RX FIFO is empty.
RX Line Status interrupt clears after reading the LSR register.
TXRDY interrupt clears after reading ISR register that is in the UART channel register set.
Modem Status Register interrupt clears after reading MSR register that is in the UART channel register set.
RTS/CTS or DTR/DSR delta interrupt clears after reading MSR register that is in the UART channel register set.
Xoff/Xon interrupt clears after reading the ISR register that is in the UART channel register set.
Special character detect interrupt is cleared by a read to ISR or after the next character is received.
TIMER Time-out interrupt clears after reading the TIMERCNTL register that is in the Device Configuration register set.
MPIO interrupt clears after reading the MPIOLVL register that is in the Device Configuration register set.

1.6.2 General Purpose 16-bit Timer/Counter [TIMERMSB, TIMELSB, TIMER, TIMECNTL] (DEFAULT 0xXX-XX-00-00)

A 16-bit down-count timer for general purpose timer or counter. Its clock source may be selected from internal crystal oscillator or externally on pin TMRCK. The timer can be set to be a single-shot for a one-time event or re-triggerable for a periodic signal. An interrupt may be generated when the timer times out and will show up as a Channel 0 interrupt (see **Table 8**). It is controlled through 4 configuration registers [TIMERCNTL, TIMER, TIMELSB, TIMERMSB]. The TIMERCNTL register provides the Timer commands such as start/stop, as shown in **Table 10** below. The time-out output of the Timer can also be optionally routed to the MPIO[0] pin. The block diagram of the Timer/Counter circuit is shown below:

FIGURE 6. TIMER/COUNTER CIRCUIT.

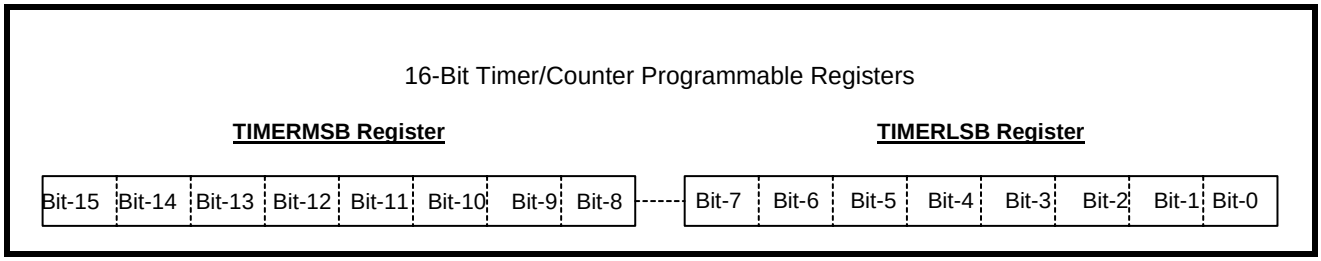


TIMERMSB [31:24] and TIMERLSB [23:16]

The concatentaion of the 8-bit registers TIMERMSB and TIMERLSB forms a 16-bit value which decides the time-out period of the Timer, per the following equation:

Timer output frequency = Timer input clock / 16-bit Timer value

The least-significant bit of the timer is being bit [0] of the TIMERLSB with most-significant-bit being bit [7] in TIMERMSB. Notice that these registers do not hold the current counter value when read. Default value is zero (timer disabled) upon powerup and reset. The 'Reset Timer' command does not have any effect on this register.



TIMER [15:8] Reserved

TIMERCNTL [7:0] Register

The bits [3:0] of this register are used to issue commands. The commands are self-clearing, so reading this register does not show the last written command. Reading this register returns a value of 0x01 when the Timer interrupt is enabled and there is a pending Timer interrupt. It returns a value of 0x00 at all other times. The default settings of the Timer, upon power-up, a hardware reset or upon the issue of a 'Timer Reset' command are:

- Timer Interrupt Disabled
- Re-triggerable mode selected
- Internal crystal oscillator outputs selcted as clock source
- Timer output not routed to MPIO[0]
- Timer stopped

TABLE 10: TIMER CONTROL REGISTERS

TIMERCNTL [7:4]	Reserved
TIMERCNTL [3:0]	These bits are used to invoke a series of commands that control the function of the Timer/Counter. The commands 1100 to 1111 are reserved. 0001: Enable Timer Interrupt 0010: Disable Timer Interrupt 0011: Select One-shot mode 0100: Select Re-triggerable mode 0101: Select Internal Crystal Oscillator output as clock input for the Timer 0110: Select External Clock input through the TMRCK pin for the Timer 0111: Route Timer output to MPIO[0] pin 1000: De-route Timer output from MPIO[0] 1001: Start Timer 1010: Stop Timer 1011: Reset Timer

TIMER OPERATION

The following paragraphs describe the operation of the 16-bit Timer/Counter. The following conventions will be used in this discussion:

- 'N' is the 16-bit value programmed in the TIMER MSB, LSB registers
- P +Q = N, where 'P' and 'Q' are approximately half of 'N'.
- If N is even, P = Q = N/2.
- If N is odd, P = (N – 1)/2 and Q = (N + 1)/2.
- 'N' can take any value from 0x0002 to 0xFFFF.

Timer Operation in One-Shot Mode:

In the one-shot mode, the Timer output will stay HIGH when started (default state) and will continue to stay HIGH until it times out (reaches the terminal count of 'N' clocks), at which time it will become LOW and stay LOW. If the Timer is re-started before the Timer times out, the counter is reset and the Timer will wait for another time-out period before setting its output LOW (See **Figure 7**). If the Timer times out, re-starting the Timer does not have any effect and a 'Stop Timer' command needs to be issued first which will set the Timer output to its default HIGH state. The Timer must be programmed while it is stopped since the following operations are blocked after the Timer has been started:

- Any write to TIMER MSB, LSB registers
- Issue of any command other than 'Start Timer', 'Stop Timer' and 'Reset Timer'

Timer Operation in Re-triggerable Mode:

In the re-triggerable mode, when the Timer is started, the Timer output will stay HIGH until it reaches half of the terminal count N (= P clocks) and toggle LOW and stay LOW for a similar amount of time (Q clocks). The above step will keep repeating until the Timer is stopped at which time the output will become HIGH (default state). See **Figure 7**. Also, after the Timer is started, re-starting the Timer does not have any effect in re-triggerable mode. The Timer must be programmed while it is stopped since the following operations are blocked when the Timer is running:

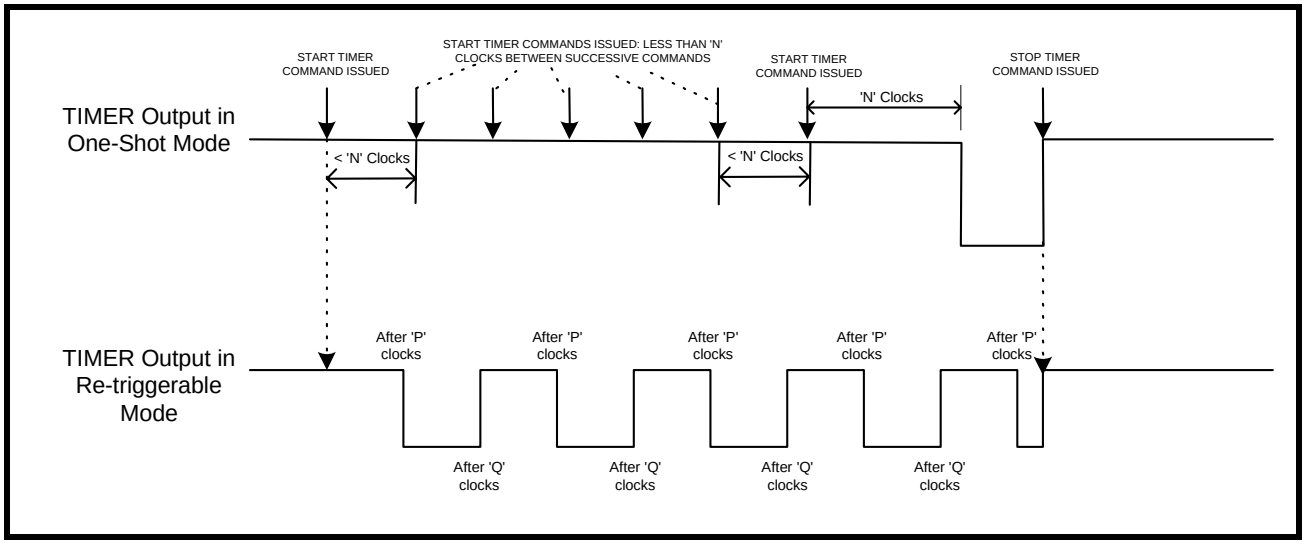
- Any write to TIMER MSB, LSB registers
- Issue of any command other than 'Stop Timer' and 'Reset Timer' ('Start Timer' is not allowed)

Routing the Timer Output to MPIO[0] Pin:

MPIO[0] pin is by default (on power up or reset, for example) an input. However, whenever the Timer output is routed to MPIO[0] pin,

- MPIO[0] will be automatically selected as an output
- MPIO[0] will become HIGH (the default state of Timer output)
- All MPIO control registers (MPIOLVL, MPIOSEL etc) lose control over MPIO[0] and get the control back only when the Timer output is de-routed from MPIO[0].

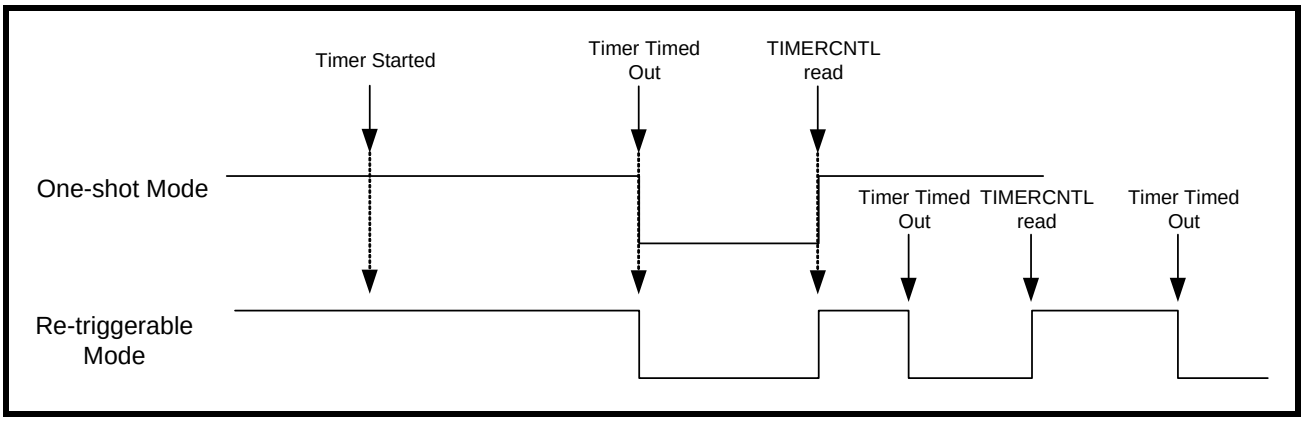
FIGURE 7. TIMER OUTPUT IN ONE-SHOT AND RE-TRIGGERABLE MODES



Timer Interrupt

In the one-shot mode, the Timer will issue an interrupt upon timing out which is 'N' clocks after the Timer is started. In the re-triggerable mode, the Timer will keep issuing an interrupt every 'N' clocks which is on every rising edge of the Timer output. The Timer interrupt can be cleared by reading the TIMERCNTL register or when a Timer Reset command is issued which brings the Timer back to its default settings. The TIMERCNTL will read a value of 0x01 when the Timer interrupt is enabled and there is a pending interrupt. It reads a value of 0x00 at all other times. Stopping the Timer does not clear the interrupt and neither does subsequent re-starting.

FIGURE 8. INTERRUPT OUTPUT (ACTIVE LOW) IN ONE-SHOT AND RE-TRIGGERABLE MODES



1.6.3 8XMODE [7:0] (default 0x00)

Each bit selects 8X or 16X sampling rate for that UART channel, bit-0 is channel 0. Logic 0 (default) selects normal 16X sampling with logic one selects 8X sampling rate. Transmit and receive data rates will double by selecting 8X.

8XMODE Register
Individual UART Channel 8X Clock Mode Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
Rsvd	Rsvd	Rsvd	Rsvd	Rsvd	Rsvd	Ch-1	Ch-0

1.6.4 REGA [15:8] Reserved

1.6.5 RESET [23:16] - (default 0x00)

Bits 0 to 1 of the Reset register [RESET] provides the software with the ability to reset the UART(s) when there is a need. Each bit is self-resetting after it is written a logic 1 to perform a reset to that channel. All registers in that channel will be reset to the default condition, see **Table 21** for details. Bit-0 =1 resets UART channel 0 while bit-1=1 resets channel 1.

RESET Register
Individual UART Channel Reset Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
Rsvd	Rsvd	Rsvd	Rsvd	Rsvd	Rsvd	Ch-1	Ch-0

1.6.6 SLEEP [31:24] (default 0x00)

SLEEP Register
Individual UART Channel Sleep Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
Rsvd	Rsvd	Rsvd	Rsvd	Rsvd	Rsvd	Ch-1	Ch-0

The 8-bit Sleep register enables each UART separately to enter Sleep mode. Sleep mode reduces power consumption when the system needs to put the UART(s) to idle. The UART enters sleep mode when the following conditions are satisfied after the sleep mode is enabled (LOW (default) is to disable and logic HIGH is to enable sleep mode):

- There is no pending interrupt
- RX pin is idling at a HIGH in normal mode or a LOW in infrared mode
- The modem inputs (CTS#, DSR#, CD# and RI#) are steady at either HIGH or LOW (MSR bits [3:0] = 0000)

When both UART channels are put to sleep, the on-chip oscillator shuts off to further conserve power. In this case, the V252 is awakened by any of the following events occurring at any of the 2 UART channels:

- A receive data start bit transition (HIGH to LOW in normal mode or from LOW to HIGH in infrared mode)
- A data byte is loaded into the transmitter
- A change of logic state on any of the modem inputs, i.e. any of the delta bits (MSR bits[7:4]) is set

The V252 is ready after 32 crystal clocks to ensure full functionality. Therefore, if the V252 is awakened by a receive data start bit transition, that character (and the subsequent few characters) may not be received correctly. Also, a special interrupt is generated with an indication of no pending interrupt. The V252 will return to sleep mode automatically after all interrupting conditions have been serviced and cleared. It will stay in the sleep mode of operation until it is disabled by resetting the SLEEP register bits.

1.6.7 Device Identification and Revision

There are two internal registers that provide device identification and revision, DVID and DREV registers. The 8-bit content in the DVID register provides device identification. A return value of 0x42 from this register indicates the device is a XR17V252. The DREV register returns an 8-bit value of 0x01 for revision A with 0x02 equals to revision B and so on. This information is very useful to the software driver for identifying which device it is communicating with and to keep up with revision changes.

DVID [15:8]

Device identification for the type of UART. The Device ID of the XR17V252 is 0x42.

DREV [7:0]

Revision number of the XR17V252. A 0x01 represents "revision-A" with 0x02 for rev-B and so on.

REGB [23:16] (default 0x00)

REGB register provides a control for simultaneous write to both UARTs configuration register or individually. This is very useful for device initialization in the power up and reset routines. Also, the register provides a facility to interface to the non-volatile memory device such as a 93C46 EEPROM. In embedded applications, the user can use this facility to store proprietary data in an external EEPROM.

1.6.8 REGB Register

REGB[16](Read/Write)	LOW (default) write to each UART configuration registers individually.
	HIGH enables simultaneous write to both UARTs configuration register.
REGB[19:17]	Reserved
REGB[20] (Write-Only)	Control the EECK, clock, output (pin 116) on the EEPROM interface.
REGB[21] (Write-Only)	Control the EECS, chips select, output (pin 115) to the EEPROM device.
REGB[22] (Write-Only)	EEDI (pin 114) data input. Write data to the EEPROM device.
REGB[23] (Read-Only)	EEDO (pin 113) data output. Read data from the EEPROM device.

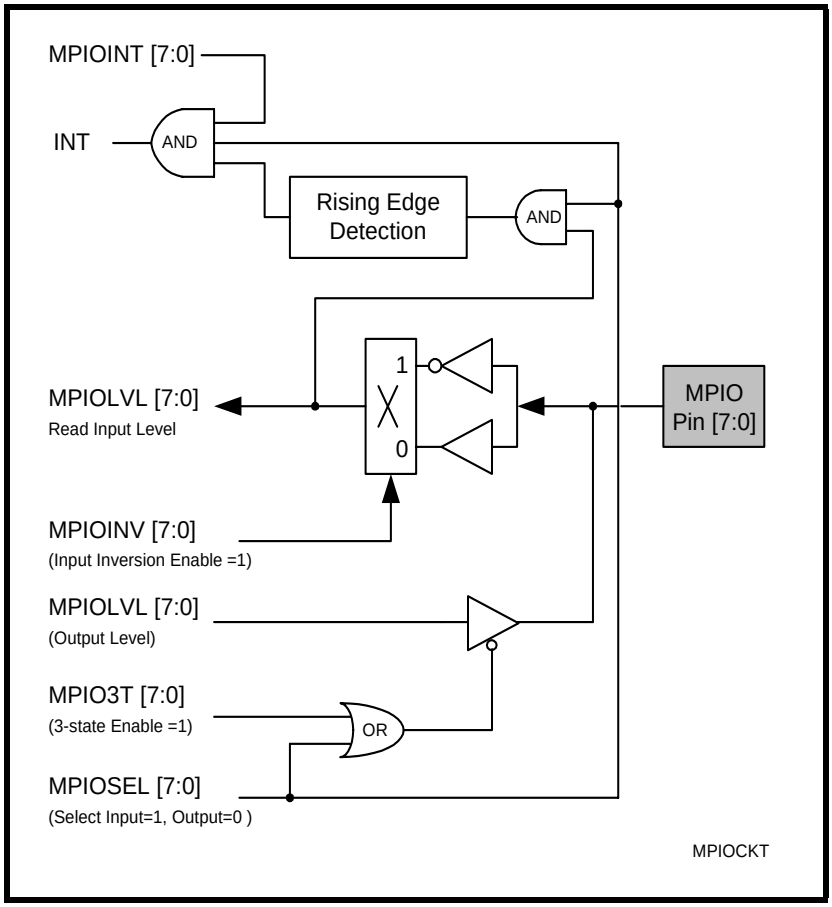
1.6.9 Multi-Purpose Inputs and Outputs

The V252 provides 8 multi-purpose inputs/outputs MPIO[7:0] for general use. Each pin can be programmed to be an input or output function. The input logic state can be set for normal or inverted level, and optionally set to generate an interrupt. The outputs can be set to be normal HIGH or LOW state, or 3-state. Their functions and definitions are programmed through 5 registers: MPIOINT, MPIOLVL, MPIO3T, MPIOINV and MPIOSEL. If all 8 pins are set for inputs, all 8 interrupts would be Or'ed together. The Or'ed interrupt is reported in the channel 0 UART interrupt status, see Interrupt Status Register. The pins may also be programmed to be outputs and to the 3-state condition for signal sharing. The MPIO[0] pin can be programmed to show the Timer output. When it is programmed to be the Timer output, all the above 5 registers lose control over the MPIO[0] pin. For details on Timer output, please see **"Section 1.6.2, General Purpose 16-bit Timer/Counter [TIMERMSB, TIMELSB, TIMER, TIMECNTL] (default 0xXX-XX-00-00)" on page 17**.

1.6.10 MPIO REGISTER

Bit [7] represents MPIO7 pin and bit [0] represents MPIO0 pin. There are 5 registers that select, control and monitor the 8 multipurpose inputs and outputs. **Figure 9** shows the internal circuitry.

FIGURE 9. MULTIPURPOSE INPUT/OUTPUT INTERNAL CIRCUIT



MPIOINT [7:0] (default 0x00)

Enable multipurpose input pin interrupt. If the pin is selected by MPIOSEL as input then bit [0] enables input pin 0 for interrupt, and bit [7] enables input pin 7. No interrupt is enable if the pin is selected to be an output. The interrupt is edge sensing and determined by MPIOINV and MPIOLVL registers. The MPIO interrupt clears after a read to register MPIOLVL. The combination of MPIOLVL and MPIOINV determines the interrupt being active LOW or active high, it's level trigger. Logic LOW (default) disables the pin's interrupt and logic HIGH enables it.

MPIOINT Register
Multipurpose Input/Output Interrupt Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
MPIO7	MPIO6	MPIO5	MPIO4	MPIO3	MPIO2	MPIO1	MPIO0

MPIOLVL [7:0] (default 0x00)

Output pin level control and input level status. The status of the input pin(s) is read on this register and output pins are controlled on this register. A logic 0 (default) sets the output to LOW and a logic 1 sets the output pin to HIGH. The MPIO interrupt will clear upon reading this register.

MPIOLVL Register
Multipurpose Output Level Control

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
MPIO7	MPIO6	MPIO5	MPIO4	MPIO3	MPIO2	MPIO1	MPIO0

MPIO3T [7:0] (default 0x00)

Output pin tri-state control. A logic 0 (default) sets the output to active level per register MPIOBIT settling, a logic 1 sets the output pin to tri-state.

MPIO3T Register
Multipurpose Output 3-state Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
MPIO7	MPIO6	MPIO5	MPIO4	MPIO3	MPIO2	MPIO1	MPIO0

MPIOINV [7:0] (default 0x00)

Input inversion control. A logic 0 (default) does not invert the input pin logic. A logic 1 inverts the input logic level.

MPIOINV Register
Multipurpose Input Signal Inversion Enable

Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
MPIO7	MPIO6	MPIO5	MPIO4	MPIO3	MPIO2	MPIO1	MPIO0

MPIOSEL [7:0] (default 0xFF)

Multipurpose input/output pin select. This register defines the functions of the pins. A logic 1 (default) defines the pin for input and a logic 0 for output.

MPIOSEL Register
Multipurpose Input/Output Selection

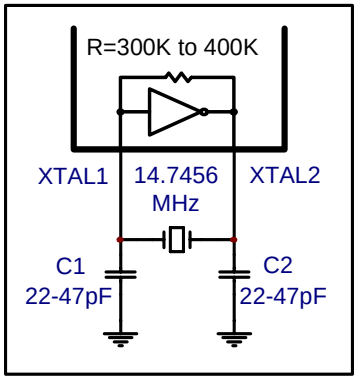
Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
MPIO7	MPIO6	MPIO5	MPIO4	MPIO3	MPIO2	MPIO1	MPIO0

2.0 CRYSTAL OSCILLATOR / BUFFER

The V252 includes an on-chip oscillator (XTAL1 and XTAL2). The crystal oscillator provides the system clock to the Baud Rate Generators (BRG) in each of the 2 UARTs, the 16-bit general purpose timer/counter and internal logics. XTAL1 is the input to the oscillator or external clock buffer input with XTAL2 pin being the output. See the Programmable Baud Rate Generator in the UART section on **page 28** for programming details.

The on-chip oscillator is designed to use an industry standard microprocessor crystal (parallel resonant with 10-22 pF capacitance load, 100ppm) connected externally between the XTAL1 and XTAL2 pins (see **Figure 10**). Alternatively, an external clock can be connected to the XTAL1 pin to clock the internal 2 baud rate generators for standard or custom rates. Typically, the oscillator connections are shown in **Figure 10**. For further reading on oscillator circuit please see application note DAN108 on EXAR's web site.

FIGURE 10. TYPICAL CRYSTAL CONNECTIONS



3.0 TRANSMIT AND RECEIVE DATA

There are two methods to load transmit data and unload receive data from each UART channel. First, there is a transmit data register and receive data register for each UART channel as shown in **Table 5** set to ease programming. These registers support 8, 16, 24 and 32 bits wide format. In the 32-bit format, it increases the data transfer rate on the PCI bus. Additionally, a special register location provides receive data byte with its associated error flags. This is a 16-bit or 32-bit read operation where the Line Status Register (LSR) content in the UART channel register is paired along with the data byte. This operation further facilitates data unloading with the error flags without having to read the LSR register separately. Furthermore, the XR17V252 supports PCI burst mode for read/write operation of up to 64 bytes of data.

The second method is through each UART channel's transmit holding register (THR) and receive holding register (RHR). The THR and RHR registers are 16550 compatible so their access is limited to 8-bit format. The software driver must separately read the LSR content for the associated error flags before reading the data byte.

3.1 FIFO DATA LOADING AND UNLOADING IN 32-BIT FORMAT

The XR17V252 supports PCI Burst Read and PCI Burst Write transactions anywhere in the mapped memory region (except reserved areas). In addition, to utilize this feature fully, the device provides a separate memory location (apart from the individual channel's register set) where the RX and the TX FIFO can be read from/ written to, as shown in **Table 5**. The following is an extract from the table showing the burstable memory locations:

Channel 0:

RX FIFO	:	0x100 - 0x13F (64 bytes)
TX FIFO	:	0x100 - 0x13F (64 bytes)
RX FIFO + status	:	0x180 - 0x1FF (64 bytes data + 64 bytes status)

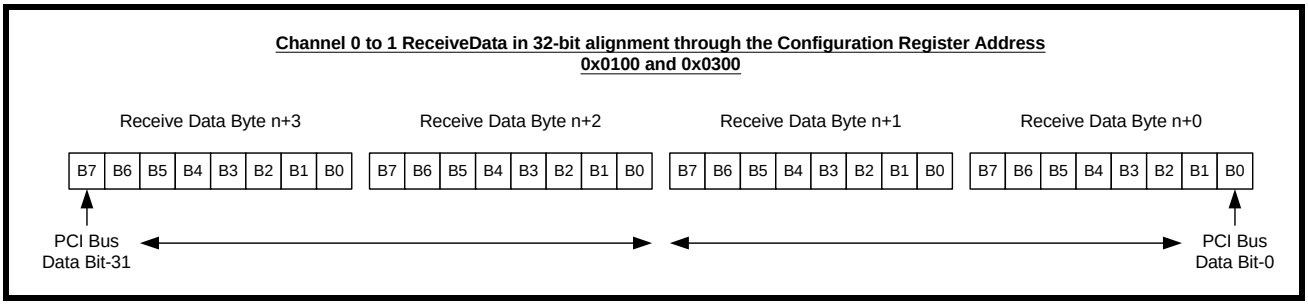
Channel 1:

RX FIFO	:	0x300 - 0x33F (64 bytes)
TX FIFO	:	0x300 - 0x33F (64 bytes)
RX FIFO + status	:	0x380 - 0x3FF (64 bytes data + 64 bytes status)

3.1.1 Normal Rx FIFO Data Unloading at locations 0x100 (channel 0) and 0x300 (channel 1)

The RX FIFO data (up to the maximum 64 bytes) can be read out in a single burst 32-bit read operation (maximum 16 DWORD reads) at memory locations 0x100 (channel 0) and 0x300 (channel 1). This operation is at least 16 times faster than reading the data in 64 separate 8-bit memory reads of RHR register (0x000 for channel 0 and 0x200 for channel 1).

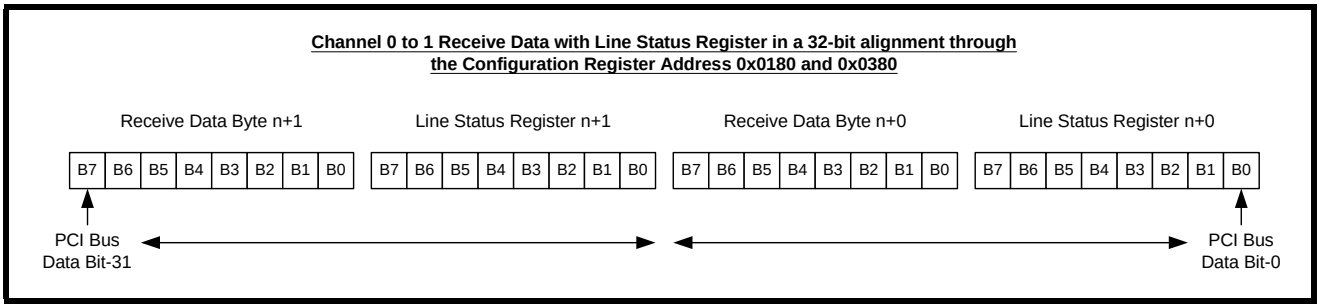
READ RX FIFO, WITH NO ERRORS	BYTE 3	BYTE 2	BYTE 1	BYTE 0
Read n+0 to n+3	FIFO Data n+3	FIFO Data n+2	FIFO Data n+1	FIFO Data n+0
Read n+4 to n+7	FIFO Data n+7	FIFO Data n+6	FIFO Data n+5	FIFO Data n+4
Etc.				



3.1.2 Special Rx FIFO Data Unloading at locations 0x180 (channel 0) and 0x380 (channel 1)

The XR17D152 also provides the same RX FIFO data along with the LSR status information of each byte side-by-side, at locations 0x180 (channel 0) and 0x380 (channel 1). The entire RX data along with the status can be downloaded in a single PCI Burst Read operation of 32 DWORD reads. The Status and Data bytes must be read in 16 or 32 bits format to maintain data integrity. The following tables show this clearly.

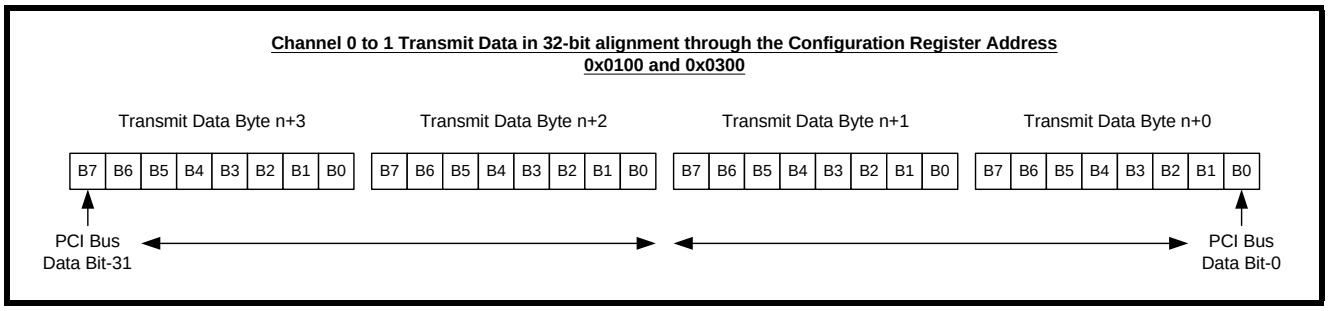
READ RX FIFO, WITH LSR ERRORS	BYTE 3	BYTE 2	BYTE 1	BYTE 0
Read n+0 to n+1	FIFO Data n+1	LSR n+1	FIFO Data n+0	LSR n+0
Read n+2 to n+3	FIFO Data n+3	LSR n+3	FIFO Data n+2	LSR n+2
Etc				



3.1.3 Tx FIFO Data Loading at locations 0x100 (channel 0) and 0x300 (channel 1)

The TX FIFO data (up to the maximum 64 bytes) can be loaded in a single burst 32-bit write operation (maximum 16 DWORD writes) at memory locations 0x100 (channel 0) and 0x300 (channel 1).

WRITE TX FIFO	BYTE 3	BYTE 2	BYTE 1	BYTE 0
Write n+0 to n+3	FIFO Data n+3	FIFO Data n+2	FIFO Data n+1	FIFO Data n+0
Write n+4 to n+7	FIFO Data n+7	FIFO Data n+6	FIFO Data n+5	FIFO Data n+4
Etc.				



3.2 FIFO DATA LOADING AND UNLOADING THROUGH THE UART CHANNEL REGISTERS, THR AND RHR IN 8-BIT FORMAT

The THR and RHR register address for channel 0 to channel 1 is shown in **Table 11** below. The THR and RHR for each channel 0 tand 1 are located sequentially at address 0x0000 and 0x0200. Transmit data byte is loaded to the THR when writing to that address and receive data is unloaded from the RHR register when reading that address. Both THR and RHR registers are 16C550 compatible in 8-bit format, so each bus operation can only write or read in bytes.

TABLE 11: TRANSMIT AND RECEIVE DATA REGISTER IN BYTE FORMAT, 16C550 COMPATIBLE

THR and RHR Address Locations For CH0 to CH1 (16C550 Compatible)

			Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
CH0	0x000	Write THR	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
CH0	0x000	Read RHR	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
CH1	0x200	Write THR	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
CH1	0x200	Read RHR	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0

4.0 UART

There are 2 UARTs channel [1:0] in the V252. Each has its own 64-byte of transmit and receive FIFO, a set of 16550 compatible control and status registers, and a baud rate generator for individual channel data rate setting. Eight additional registers per UART were added for the EXAR enhanced features.

4.1 Programmable Baud Rate Generator with Fractional Divisor

Each UART has its own Baud Rate Generator (BRG) with a prescaler for the transmitter and receiver. The prescaler is controlled by a software bit in the MCR register. The MCR register bit [7] sets the prescaler 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 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 a random value upon power up. Therefore, the BRG must be programmed during initialization to the operating data rate. The DLL and DLM registers provide the integer part of the divisor and the DLD register provides the fractional part of the divisor. Only the four lower bits of the DLD are implemented and they are used to select a value from 0 (for setting 0000) to 0.9375 or 15/16 (for setting 1111). Programming the Baud Rate Generator Registers DLL, DLM and DLD provides the capability for selecting the operating data rate. **Table 12** shows the standard data rates available with a 14.7456 MHz 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 12**. At 8X sampling rate, these data rates would double. Also, when using 8X sampling mode, please note that the bit-time will have a jitter (+/- 1/16) whenever the DLD is an odd number. When using a non-standard data rate crystal or external clock, the divisor value can be calculated with the following equation(s):

Required Divisor (decimal) = (XTAL1 clock frequency / prescaler) / (serial data rate x 16), **WITH 8XMODE [7:0] IS 0**

Required Divisor (decimal) = (XTAL1 clock frequency / prescaler / (serial data rate x 8), **WITH 8XMODE [7:0] IS 1**

The closest divisor that is obtainable in the V252 can be calculated using the following formula:

ROUND((Required Divisor - TRUNC (Required Divisor))*16)/16 + TRUNC (Required Divisor), where

DLM = TRUNC(Required Divisor) >> 8
DLL = TRUNC (Required Divisor) & 0xFF
DLD = ROUND ((Required Divisor-TRUNC(Required Divisor))*16)

In the formulas above, please note that:

TRUNC (N) = Integer Part of N. For example, TRUNC (5.6) = 5.

ROUND (N) = N rounded towards the closest integer. For example, ROUND (7.3) = 7 and ROUND (9.9) = 10.

FIGURE 11. BAUD RATE GENERATOR

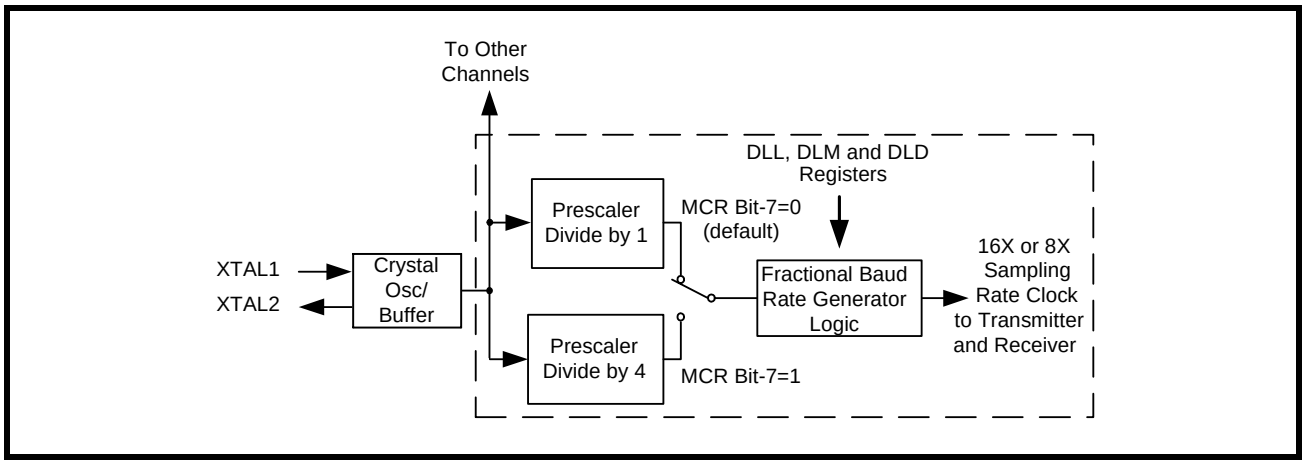


TABLE 12: TYPICAL DATA RATES WITH A 24 MHZ CRYSTAL OR EXTERNAL CLOCK AT 16X SAMPLING

REQUIRED OUTPUT DATA RATE	DIVISOR FOR 16x Clock (Decimal)	DIVISOR OBTAINABLE IN V252	DLM PROGRAM VALUE (HEX)	DLL PROGRAM VALUE (HEX)	DLD PROGRAM VALUE (HEX))	DATA ERROR RATE (%)
400	3750	3750	E	A6	0	0
2400	625	625	2	71	0	0
4800	312.5	312 8/16	1	38	8	0
9600	156.25	156 4/16	0	9C	4	0
10000	150	150	0	96	0	0
19200	78.125	78 2/16	0	4E	2	0
25000	60	60	0	3C	0	0
28800	52.0833	52 1/16	0	34	1	0.04
38400	39.0625	39 1/16	0	27	1	0
50000	30	30	0	1E	0	0
57600	26.0417	26 1/16	0	1A	1	0.08
75000	20	20	0	14	0	0
100000	15	15	0	F	0	0
115200	13.0208	13	0	D	0	0.16
153600	9.7656	9 12/16	0	9	C	0.16
200000	7.5	7 8/16	0	7	8	0
225000	6.6667	6 11/16	0	6	B	0.31
230400	6.5104	6 8/16	0	6	8	0.16
250000	6	6	0	6	0	0
300000	5	5	0	5	0	0
400000	3.75	3 12/16	0	3	C	0
460800	3.2552	3 4/16	0	3	4	0.16
500000	3	3	0	3	0	0
750000	2	2	0	2	0	0
921600	1.6276	1 10/16	0	1	A	0.16
1000000	1.5	1 8/16	0	1	8	0

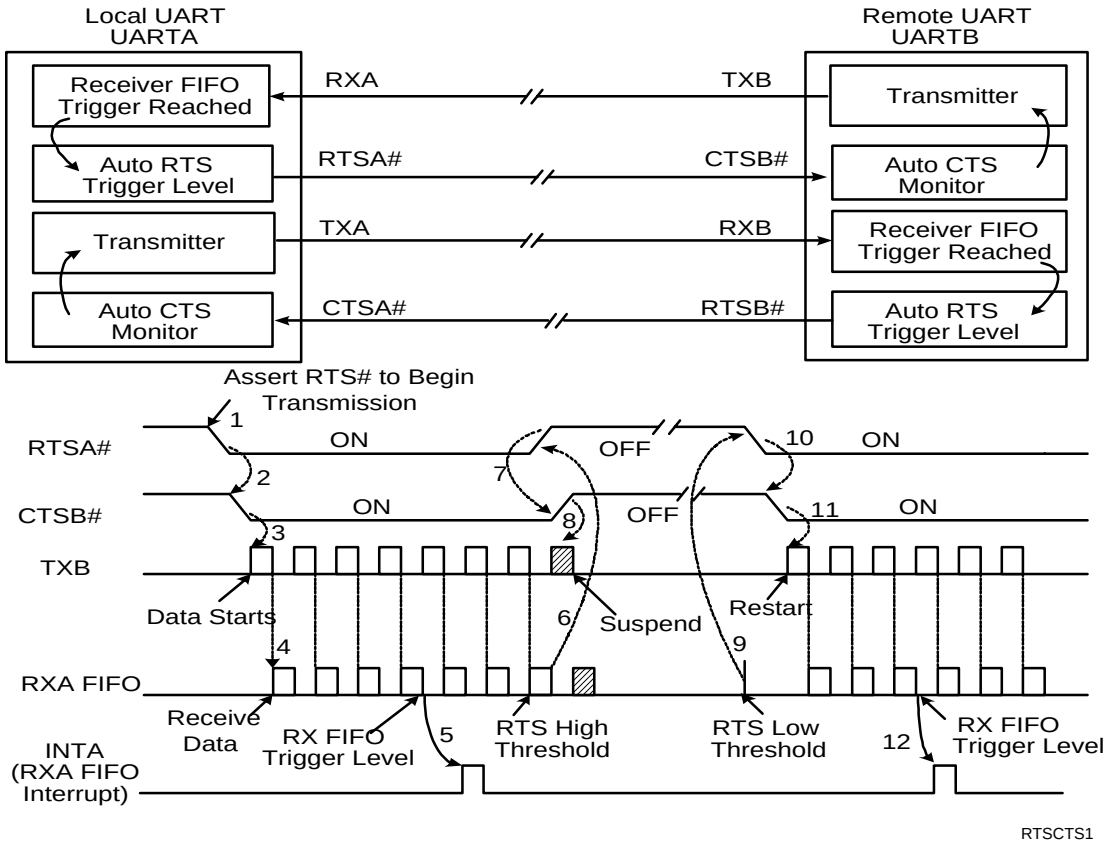
4.2 Automatic Hardware (RTS/CTS or DTR/DSR) Flow Control Operation

Automatic hardware or RTS/DTR and CTS/DSR flow control is used to prevent data overrun to the local receiver FIFO and remote receiver FIFO. The RTS#/DTR# output pin is used to request remote unit to suspend/restart data transmission while the CTS#/DSR# input pin is monitored to suspend/restart local transmitter. The auto RTS/DTR and auto CTS/DSR flow control features are individually selected to fit specific application requirement and enabled through EFR bit[6:7] and MCR bit [2] for either RTS/CTS or DTR/DSR control signals. The auto RTS/DTR function must be started by asserting RTS/DTR# output pin (MCR bit [0] or bit [1] to logic 1) after it is enabled. **Figure 12** below explains how it works.

Two interrupts associated with RTS/DTR and CTS/DSR flow control have been added to give indication when RTS/DTR# pin or CTS/DSR# pin is de-asserted during operation. The RTS/DTR and CTS/DSR interrupts must be first enabled by EFR bit [4], and then enabled individually by IER bits [7:6], and chosen with MCR bit [2].

Automatic hardware flow control is selected by setting bits [7 (CTS): 6 (RTS)] of the EFR register to logic 1. If CTS# pin transitions from LOW to HIGH indicating a flow control request, ISR bit [5] will be set to logic 1, (if enabled via IER bit [7:6]), and the UART will suspend TX transmissions as soon as the stop bit of the character in process is shifted out. Transmission is resumed after the CTS# input returns to LOW, indicating more data may be sent.

FIGURE 12. AUTO RTS/DTR AND CTS/DSR 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.

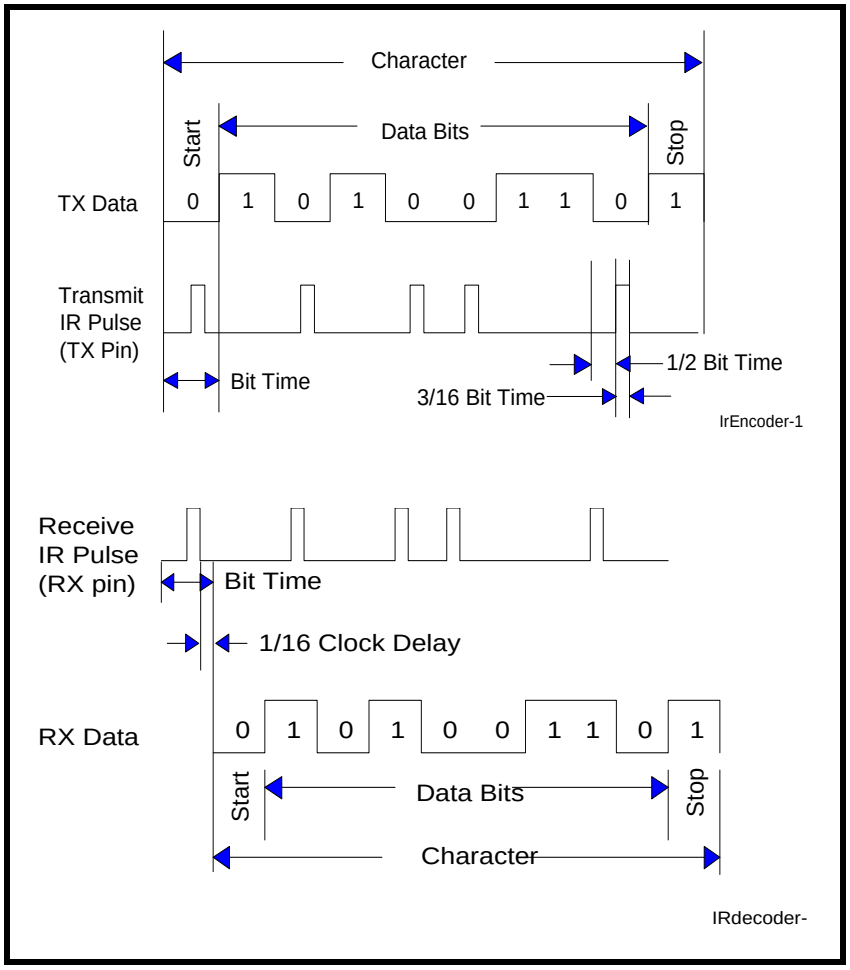
4.3 Infrared Mode

Each UART in the V252 includes the infrared encoder and decoder compatible to the IrDA (Infrared Data Association) version 1.0. The input pin ENIR conveniently activates both UART channels to start up in the infrared mode. This global control pin enables the MCR bit [6] function in every UART channel register. After power up or a reset, the software can overwrite MCR bit [6] if so desired. ENIR and MCR bit [6] also disable its receiver while the transmitter is sending data. This prevents the echoed data from going to the receiver. The global activation ENIR pin prevents the infrared emitter from turning on and drawing large amount of current while the system is starting up. When the infrared feature is enabled, the transmit data outputs, TX[7:0], would idle at logic 0 level. Likewise, the RX [7:0] inputs assume an idle level of logic 0.

The infrared encoder sends out a 3/16 of a bit wide pulse for each “LOW” bit in the transmit data stream. This signal encoding reduces the on-time of the infrared LED, hence reduces the power consumption. See **Figure 13** below.

The infrared decoder receives the input pulse from the infrared sensing diode on RX pin. Each time the decoder senses a light pulse, it returns a LOW to the data bit stream. The RX input signal may be inverted prior delivered to the input of the decoder via internal register setting. This option supports active LOW instead of normal active HIGH pulse from some infrared modules on the market.

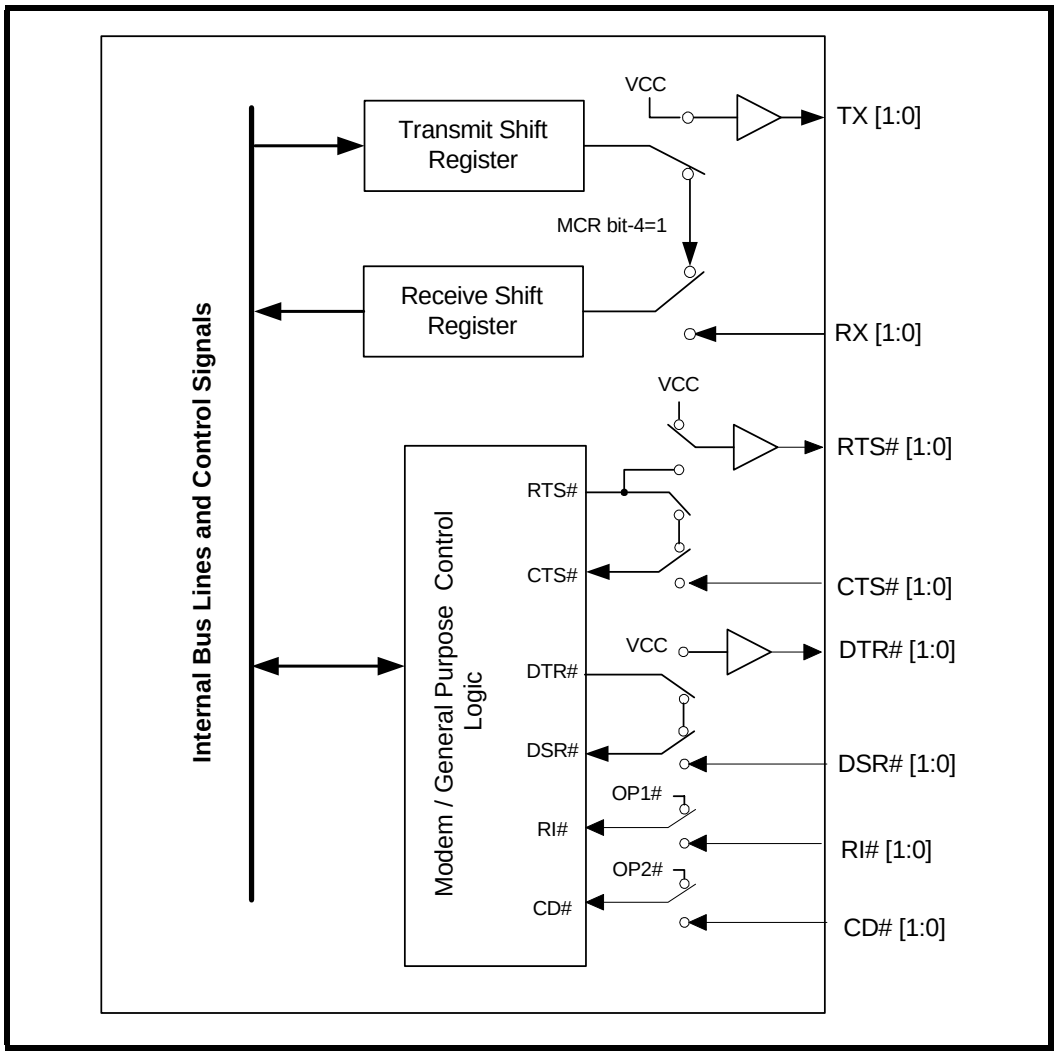
FIGURE 13. INFRARED TRANSMIT DATA ENCODING AND RECEIVE DATA DECODING



4.4 Internal Loopback

Each UART channel provides an internal loopback capability for system diagnostic. The internal loopback mode is enabled by setting MCR register bit [4] to logic 1. All regular UART functions operate normally. Figure 14 shows how the modem port signals are re-configured. Transmit data from the transmit shift register output is internally routed to the receive shift register input allowing the system to receive the same data that it was sending. The TX pin is held at HIGH or mark condition while RTS# and DTR# are de-asserted, and CTS#, DSR# CD# and RI# inputs are ignored.

FIGURE 14. INTERNAL LOOP BACK



4.5 UART CHANNEL CONFIGURATION REGISTERS AND ADDRESS DECODING

The 2 sets of UART configuration registers are decoded using address lines A9 to A11 as shown below. Address lines A0 to A3 select the 16 registers in each channel. The first 8 registers are 16550 compatible with EXAR enhanced feature registers located on the upper 8 addresses.

A11	A10	A9	UART CHANNEL SELECTION
0	0	0	0
0	0	1	1

TABLE 13: UART CHANNEL CONFIGURATION REGISTERS.

ADDRESS	REGISTER	READ/WRITE	COMMENTS
A3 A2 A1 A0			
16550 COMPATIBLE			
0 0 0 0	RHR - Receive Holding Reg THR - Transmit Holding Register	Read-only Write-only	LCR[7] = 0
0 0 0 0	DLL - Divisor LSB	Read/Write	LCR[7] = 1
0 0 0 1	DLM - Divisor MSB	Read/Write	LCR[7] = 1
0 0 1 0	DLD - Divisor Fractional Part	Read/Write	LCR[7] = 1
0 0 0 1	IER - Interrupt Enable Reg	Read/Write	LCR[7] = 0
0 0 1 0	ISR - Interrupt Status Reg FCR - FIFO Control Reg	Read-only Write-only	LCR[7] = 0
0 0 1 1	LCR - Line Control Reg	Read/Write	
0 1 0 0	MCR - Modem Control Reg	Read/Write	
0 1 0 1	LSR - Line Status Reg reserved	Read-only Write-only	
0 1 1 0	MSR - Modem Status Reg - Auto RS485 Delay	Read-only Write-only	
0 1 1 1	SPR - Scratch Pad Reg	Read/Write	
ENHANCED REGISTER			
1 0 0 0	FCTR	Read/Write	
1 0 0 1	EFR - Enhanced Function Reg	Read/Write	
1 0 1 0	TXCNT - Transmit FIFO Level Counter TXTRG - Transmit FIFO Trigger Level	Read-only Write-only	
1 0 1 1	RXCNT - Receive FIFO Level Counter RXTRG - Receive FIFO Trigger Level	Read-only Write-only	
1 1 0 0	Xoff-1 - Xoff Character 1 Xchar	Write-only Read-only	Xon,Xoff Rcvd. Flags
1 1 0 1	Xoff-2 - Xoff Character 2 reserved	Write-only Read-only	
1 1 1 0	Xon-1 - Xon Character 1 reserved	Write-only Read-only	
1 1 1 1	Xon-2 - Xon Character 2 reserved	Write-only Read-only	

TABLE 14: UART CHANNEL CONFIGURATION REGISTERS DESCRIPTION. SHADED BITS ARE ENABLED BY EFR BIT-4.

ADDRESS A3-A0	REG NAME	READ/ WRITE	BIT [7]	BIT [6]	BIT [5]	BIT [4]	BIT [3]	BIT [2]	BIT[1]	BIT [0]	COMMENT
0 0 0 0	RHR	R	BIT [7]	BIT [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	LCR[7]=0
0 0 0 0	THR	W	BIT [7]	BIT [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	LCR[7]=0
0 0 0 0	DLL	R/W	BIT [7]	BIT [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	LCR[7]=1
0 0 0 1	DLM	R/W	BIT [7]	BIT [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	LCR[7]=1
0 0 1 0	DLD	R/W	0	0	0	0	Bit [3]	Bit [2]	Bit [1]	Bit [0]	LCR[7]=1
0 0 0 1	IER	R/W	0/ CTS/ DSR# Int. Enable	0/ RTS/ DTR# Int. Enable	0/ Xon/ Xoff/Sp. Char. Int. Enable	0	Modem Status Int. Enable	RX Line Status Int. Enable	TX Empty Int. Enable	RX Data Int. Enable	LCR[7]=0
0 0 1 0	ISR	R	FIFOs Enable	FIFOs Enable	0/ Delta-Flow Cntl	0/ Xoff/special char	INT Source Bit [3]	INT Source Bit [2]	INT Source Bit [1]	INT Source Bit [0]	LCR[7]=0
0 0 1 0	FCR	W	RX FIFO Trigger	RX FIFO Trigger	0/ TX FIFO Trigger	0/ TX FIFO Trigger	DMA Mode	TX FIFO Reset	RX FIFO Reset	FIFOs Enable	LCR[7]=0
0 0 1 1	LCR	R/W	Divisor Enable	Set TX Break	Set Parity	Even Parity	Parity Enable	Stop Bits	Word Length Bit [1]	Word Length Bit [0]	
0 1 0 0	MCR	R/W	0/ BRG Prescaler	0/ IR Enable	0/ XonAny	Internal Loopback Enable	(OP2)[1] TX char Immediate	(OP1)[1] RTS/ DTR Flow Sel	RTS# Pin Control	DTR# Pin Control	
0 1 0 1	LSR	R/W	RX FIFO ERROR	TSR Empty	THR Empty	RX Break	RX Framing Error	RX Parity Error	RX Overrun	RX Data Ready	
0 1 1 0	MSR	R	CD	RI	DSR	CTS	Delta CD#	Delta RI#	Delta DSR#	Delta CTS#	
	MSR	W	RS485 DLY-3	RS485 DLY-2	RS485 DLY-1	RS485 DLY-0	Disable TX	Disable RX			
0 1 1 1	SPR	R/W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	User Data
1 0 0 0	FCTR	R/W	TRG Table Bit [1]	TRG Table Bit [0]	Auto RS485 Enable	Invert IR RX Input	RTS/ DTR Hyst Bit [3]	RTS/ DTR Hyst Bit [2]	RTS/ DTR Hyst Bit [1]	RTS/ DTR Hyst Bit [0]	

TABLE 14: UART CHANNEL CONFIGURATION REGISTERS DESCRIPTION. SHADED BITS ARE ENABLED BY EFR BIT-4.

ADDRESS A3-A0	REG NAME	READ/ WRITE	BIT [7]	BIT [6]	BIT [5]	BIT [4]	BIT [3]	BIT [2]	BIT[1]	BIT [0]	COMMENT
1 0 0 1	EFR	R/W	Auto CTS/ DSR Enable	Auto RTS/ DTR Enable	Special Char Select	Enable IER [7:5], ISR [5:4], FCR[5:4], MCR[7:5,2] MSR[7:4]	Soft-ware Flow Cntl Bit [3]	Soft-ware Flow Cntl Bit [2]	Soft-ware Flow Cntl Bit [1]	Soft-ware Flow Cntl Bit [0]	
1 0 1 0	TXCNT	R	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	
1 0 1 0	TXTRG	W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	
1 0 1 1	RXCNT	R	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	
1 0 1 1	RXTRG	W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit [2]	Bit [1]	Bit [0]	
1 1 0 0	XCHAR	R	0	0	0	0	TX Xon Indicator	TX Xoff Indicator	Xon Det. Indicator	Xoff Det. Indicator	Self clear after read
1 1 0 0	XOFF1	W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit-2	Bit [1]	Bit [0]	
1 1 0 1	XOFF2	W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit-2	Bit [1]	Bit [0]	
1 1 1 0	XON1	W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit-2	Bit [1]	Bit [0]	
1 1 1 1	XON2	W	Bit [7]	Bit [6]	Bit [5]	Bit [4]	Bit [3]	Bit-2	Bit [1]	Bit [0]	

NOTE: *MCR bits [3:2] (OP1 and OP2 outputs) are not available in the XR17V252. They are present for 16C550 compatibility during Internal loopback, see* **Figure 14***.*

4.6 Transmitter

The transmitter section comprises of a 64 bytes of FIFO, a byte-wide Transmit Holding Register (THR) and an 8-bit Transmit Shift Register (TSR). THR receives a data byte from the host (non-FIFO mode) or a data byte from the FIFO when the FIFO is enabled by FCR bit [0]. TSR shifts out every data bit with the 16X or 8X internal clock. A bit time is 16 or 8 clock periods. The transmitter sends the start bit followed by the number of data bits, inserts the proper parity bit if enable, and adds the stop bit(s). The status of the THR and TSR are reported in the Line Status Register (LSR bit [6:5]).

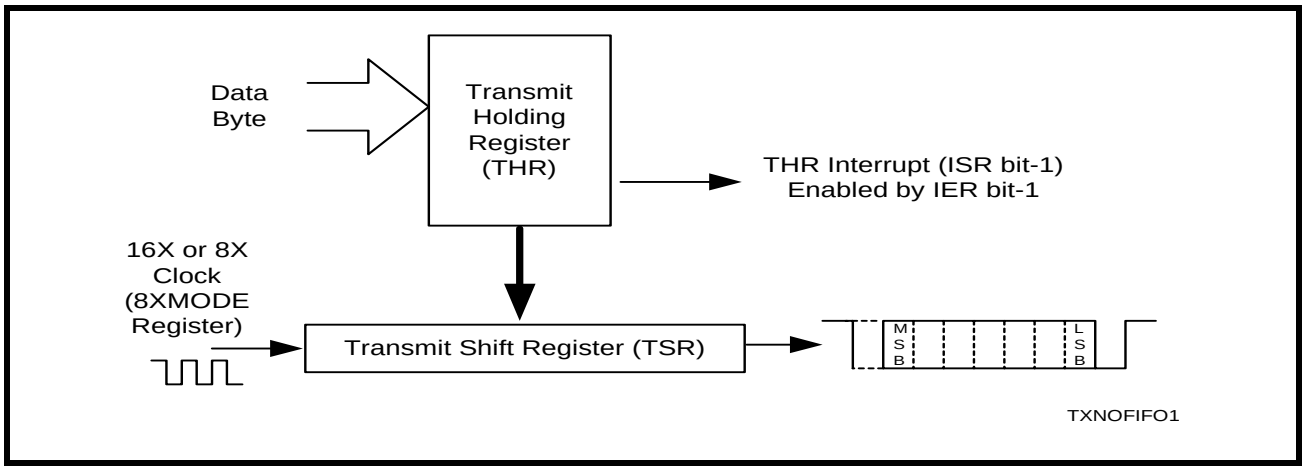
4.6.1 Transmit Holding Register (THR)

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 also the input register to the transmit FIFO of 64 bytes when FIFO operation is enabled by FCR bit[0]. A THR empty interrupt can be generated when it is enabled in IER bit [1].

4.6.2 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 15. TRANSMITTER OPERATION IN NON-FIFO MODE



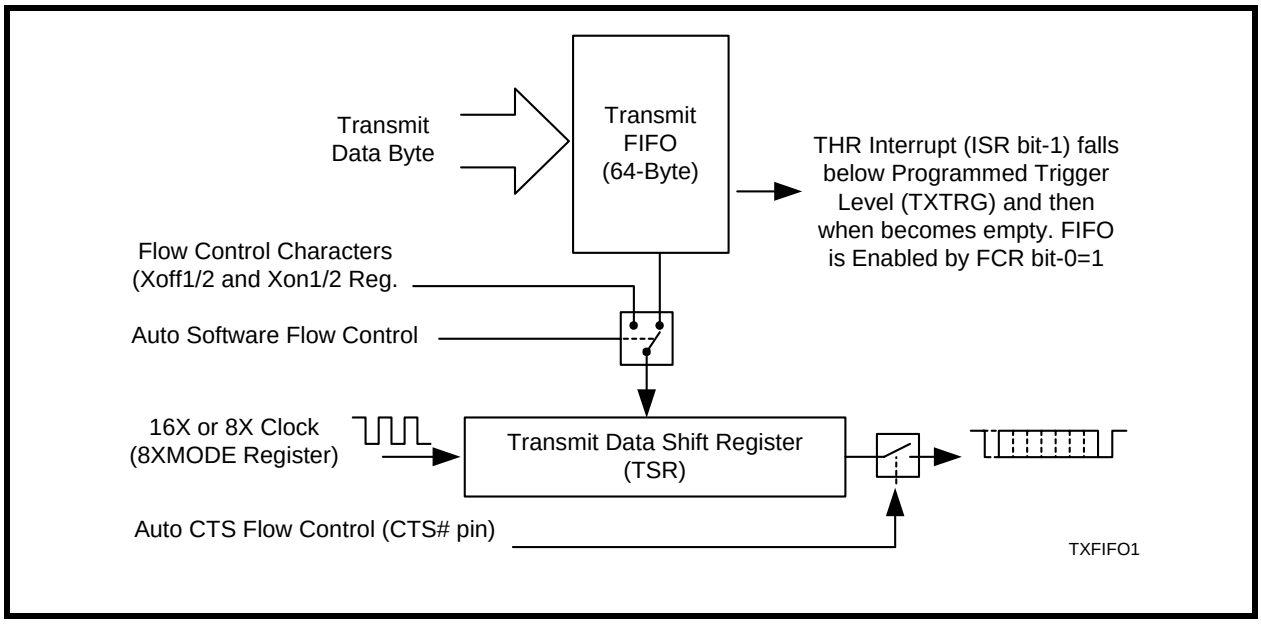
4.6.3 Transmitter Operation in FIFO Mode

The host may fill the transmit FIFO with up to 64 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 amount of data in the FIFO falls below its programmed trigger level (see TXTRG register). The transmit empty interrupt is enabled by IER bit [1]. The TSR flag (LSR bit [6]) is set when TSR becomes completely empty. Furthermore, with the RS485 half-duplex direction control enabled (FCTR bit [5]=1) the source of the transmit empty interrupt changes to TSR empty instead of THR empty. This is to ensure the RTS# output is not changed until the last stop bit of the last character is shifted out.

4.6.4 Auto RS485 Operation

The auto RS485 half-duplex direction control changes the behavior of the transmitter when enabled by FCTR bit [5]. It de-asserts RTS# or DTR# after a specified delay indicated in MSR[7:4] following the last stop bit of the last character that has been transmitted. This helps in turning around the transceiver to receive the remote station's response. 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. This delay prevents undesirable line signal disturbance that causes signal degradation. It also changes the transmitter empty interrupt to TSR empty instead of THR empty.

FIGURE 16. TRANSMITTER OPERATION IN FIFO AND FLOW CONTROL MODE

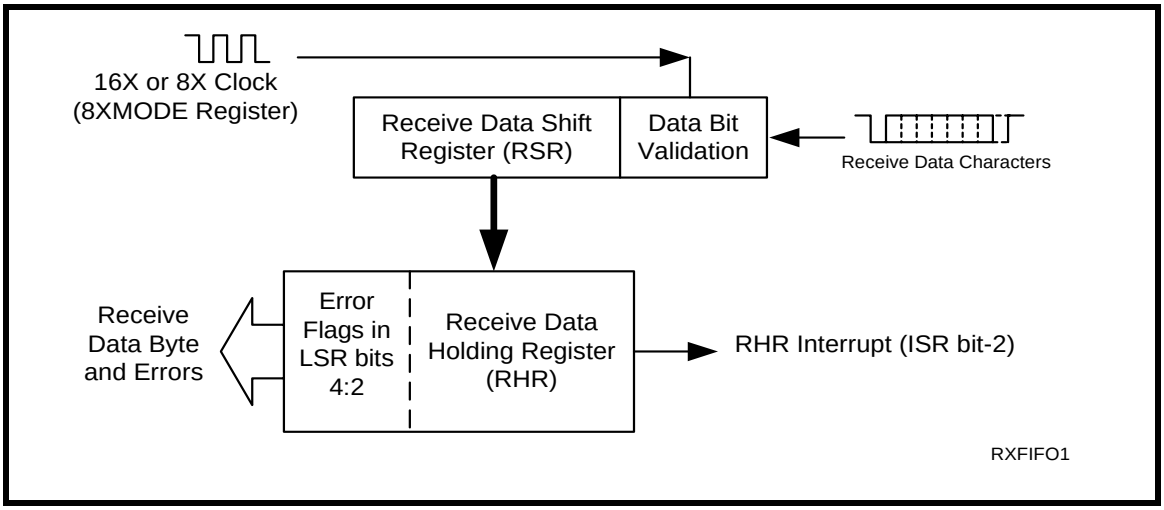


4.7 Receiver

The receiver section contains an 8-bit Receive Shift Register (RSR) and Receive Holding Register (RHR). The RSR uses the 16X or 8X clock 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 or 8X clock rate. After 8 or 4 clocks 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 [4:1]. Upon unloading the receive data byte from RHR, the receive FIFO pointer is bumped and the error flags 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 function when receive data does not reach the receive FIFO trigger level. This time-out delay is 4 word lengths as defined by LCR[1:0] plus 12 bits time. The RHR interrupt is enabled by IER bit [0].

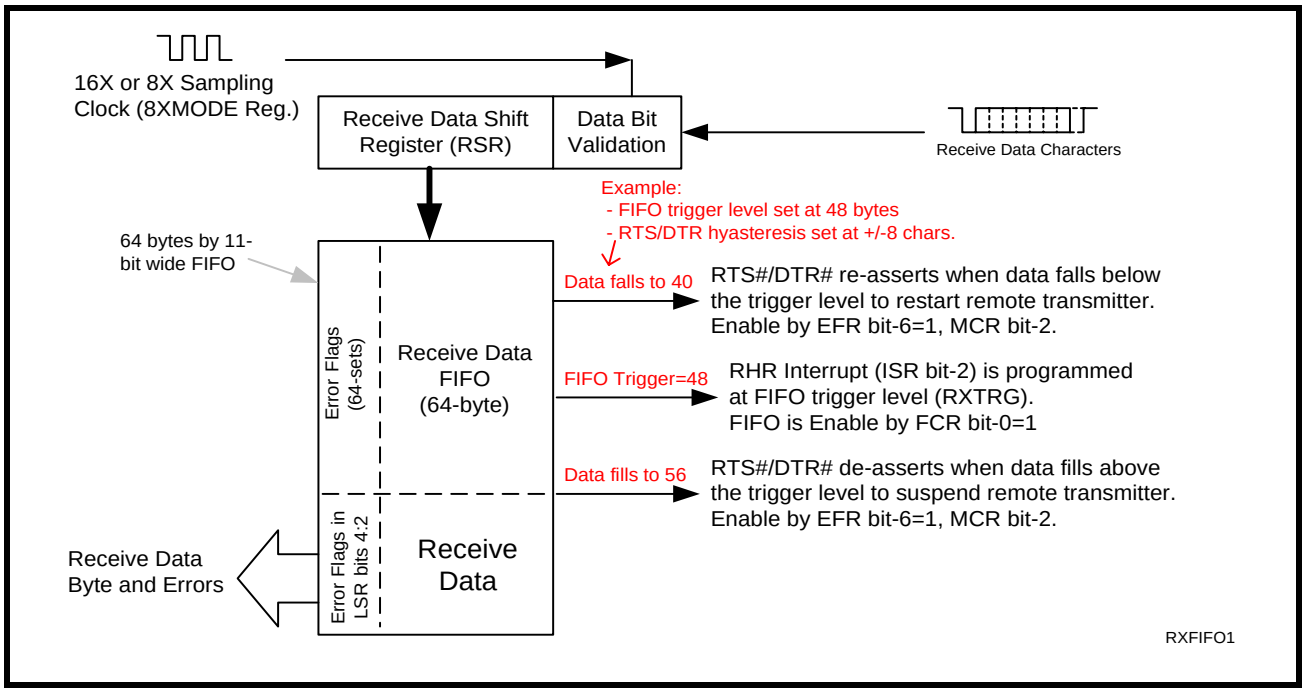
4.7.1 Receiver Operation in non-FIFO Mode

FIGURE 17. RECEIVER OPERATION IN NON-FIFO MODE



4.7.2 Receiver Operation with FIFO

FIGURE 18. RECEIVER OPERATION IN FIFO AND FLOW CONTROL MODE



5.0 UART CONFIGURATION REGISTERS

5.1 Receive Holding Register (RHR) - Read only

SEE"RECEIVER" ON PAGE 38.

5.2 Transmit Holding Register (THR) - Write only

SEE"TRANSMITTER" ON PAGE 36.

5.3 Baud Rate Generator Divisors (DLM, DLL and DLD)

The Baud Rate Generator (BRG) generates the data rate for the transmitter and receiver. The rate is programmed through registers DLM, DLL and DLD which are only accessible when LCR bit [7] is set to logic 1. Refer to **"Section 4.1, Programmable Baud Rate Generator with Fractional Divisor" on page 28** for more details.

5.4 Interrupt Enable Register (IER) - Read/Write

The Interrupt Enable Register (IER) masks the interrupts from receive data ready, transmit empty, line status and modem status registers. These interrupts are reported in the Interrupt Status Register (ISR) and also encoded in INT (INT0-INT3) register in the Device Configuration Registers.

5.4.1 IER versus Receive FIFO Interrupt Mode Operation

When the receive FIFO (FCR bit [0] = a logic 1) and receive interrupts (IER bit [0] = logic 1) are enabled, the RHR interrupts (see ISR bits [4:3]) status will reflect the following:

A. The receive data available interrupts are issued to the host when the FIFO has reached the programmed trigger level. It will be cleared when the FIFO drops below the programmed trigger level.

B. FIFO level will be reflected in the ISR register when the FIFO trigger level is reached. Both the ISR register status bit and the interrupt will be cleared when the FIFO drops below the trigger level.

C. The receive data ready bit (LSR bit [0]) is set as soon as a character is transferred from the shift register to the receive FIFO. It is reset when the FIFO is empty.

5.4.2 IER versus Receive/Transmit FIFO Polled Mode Operation

When FCR bit [0] equals a logic 1 for FIFO enable; resetting IER bits [3:0] enables the XR16V252 in the FIFO polled mode of operation. Since the receiver and transmitter have separate bits in the LSR either can be used in the polled mode by selecting respective transmit or receive control bit(s).

A. LSR BIT-0 indicates there is data in RHR (non-FIFO mode) or RX FIFO (FIFO mode).

B. LSR BIT-1 indicates an overrun error has occurred and that data in the FIFO may not be valid.

C. LSR BIT 2-4 provides the type of receive data errors encountered for the data byte in RHR, if any.

D. LSR BIT-5 indicates THR (non-FIFO mode) or TX FIFO (FIFO mode) is empty.

E. LSR BIT-6 indicates when both the transmit FIFO and TSR are empty.

F. LSR BIT-7 indicates a data error in at least one character in the RX FIFO.

IER[7]: CTS# Input Interrupt Enable (requires EFR bit [4]=1)

- Logic 0 = Disable the CTS# interrupt (default).
- Logic 1 = Enable the CTS# interrupt. The UART issues an interrupt when CTS# pin makes a transition from LOW to HIGH.

IER[6]: RTS# Output Interrupt Enable (requires EFR bit [4]=1)

- Logic 0 = Disable the RTS# interrupt (default).
- Logic 1 = Enable the RTS# interrupt. The UART issues an interrupt when RTS# pin makes a transition from LOW to HIGH.

IER[5]: Xoff Interrupt Enable (requires EFR bit [4]=1)

- Logic 0 = Disable the software flow control, receive Xoff interrupt (default).
- Logic 1 = Enable the software flow control, receive Xoff interrupt. See Software Flow Control section for details.

IER[4]: Reserved.

IER[3]: Modem Status Interrupt Enable

The Modem Status Register interrupt is issued whenever any of the delta bits of the MSR register (bits [3:0]) is set.

- Logic 0 = Disable the modem status register interrupt (default).
- Logic 1 = Enable the modem status register interrupt.

IER[2]: Receive Line Status Interrupt Enable

An Overrun error, Framing error, Parity error or detection of a Break character will result in an LSR interrupt. The V252 will issue an LSR interrupt immediately after receiving a character with an error. It will again re-issue the interrupt (if the first one has been cleared by reading the LSR register) when the character with the error is on the top of the FIFO, meaning the next one to be read out of the FIFO.

For example, let's consider an incoming data stream of 0x55, 0xAA, etc. and that the character 0xAA has a Parity error associated with it. Let's assume that the character 0x55 has not been read out of the FIFO yet. The V252 will issue an interrupt as soon as the stop bit of the character 0xAA is received. The LSR register will have only the FIFO error bit (bit [7]) set and none of the other error bits (bits [4:1]) will be set, since the byte on the top of the FIFO is 0x55 which does not have any errors associated with it. When this byte has been read out, the V252 will issue another LSR interrupt and this time the LSR register will show the Parity bit (bit [2]) set.

- Logic 0 = Disable the receiver line status interrupt (default).
- Logic 1 = Enable the receiver line status interrupt.

IER[1]: TX Ready Interrupt Enable

In non-FIFO mode, a TX interrupt is issued whenever the THR is empty. In the FIFO mode, an interrupt is issued twice: once when the number of bytes in the TX FIFO falls below the programmed trigger level and again when the TX FIFO becomes empty. When autoRS485 mode is enabled (FCTR bit [5] = 1), the second interrupt is delayed until the transmitter (both the TX FIFO and the TX Shift Register) is empty.

- Logic 0= Disable Transmit Ready Interrupt (default).
- Logic 1 = Enable Transmit Ready Interrupt.

IER[0]: RX Interrupt Enable

The receive data ready interrupt will be issued when RHR has a data character in the non-FIFO mode or when the receive FIFO has reached the programmed trigger level in the FIFO mode.

- Logic 0 = Disable the receive data ready interrupt (default).
- Logic 1 = Enable the receiver data ready interrupt.

5.5 Interrupt Status Register (ISR) - Read Only

The UART provides multiple levels of prioritized interrupts to minimize external software interaction. The Interrupt Status Register (ISR) provides the user with six interrupt status bits. Performing a read cycle on the ISR will give the user the current highest pending interrupt level to be serviced, others queue up for next service. No other interrupts are acknowledged until the pending interrupt is serviced. The Interrupt Source Table, **Table 15**, shows the data values (bit [5:0]) for the six prioritized interrupt levels and the interrupt sources associated with each of these interrupt levels.

5.5.1 Interrupt Generation:

- LSR is by any of the LSR bits [4:1]. See IER bit [2] description above.
- RXRDY is by RX trigger level.
- RXRDY Time-out is by a 4-char plus 12 bits delay timer.
- TXRDY is by TX trigger level or TX FIFO empty (or transmitter empty in auto RS-485 control).
- MSR is by any of the MSR bits [3:0].
- Receive Xon/Xoff/Special character is by detection of a Xon, Xoff or Special character.
- CTS#/DSR# is when its transmitter toggles the input pin (from LOW to HIGH) during auto CTS/DSR flow control enabled by EFR bit [7] and selection on MCR bit [2].
- RTS#/DTR# is when its receiver toggles the output pin (from LOW to HIGH) during auto RTS/DTR flow control enabled by EFR bit [6] and selection on MCR bit [2].

5.5.2 Interrupt Clearing:

- LSR interrupt is cleared by a read to the LSR register.
- RXRDY interrupt is cleared by reading data until FIFO falls below the trigger level.
- RXRDY Time-out interrupt is cleared by reading RHR.
- TXRDY interrupt is cleared by a read to the ISR register or writing to THR.
- MSR interrupt is cleared by a read to the MSR register.
- Xon or Xoff interrupt is cleared by a read to ISR register.
- Special character interrupt is cleared by a read to ISR or after the next character is received.
- RTS#/DTR# and CTS#/DSR# status change interrupts are cleared by a read to the MSR register.

TABLE 15: INTERRUPT SOURCE AND PRIORITY LEVEL

PRIORITY LEVEL	ISR REGISTER STATUS BITS						SOURCE OF THE INTERRUPT
	BIT [5]	BIT [4]	BIT [3]	BIT [2]	BIT [1]	BIT [0]	
1	0	0	0	1	1	0	LSR (Receiver Line Status Register)
2	0	0	0	1	0	0	RXRDY (Received Data Ready)
3	0	0	1	1	0	0	RXRDY (Receive Data Time-out)
4	0	0	0	0	1	0	TXRDY (Transmitter Holding Register Empty)
5	0	0	0	0	0	0	MSR (Modem Status Register)
6	0	1	0	0	0	0	RXRDY (Received Xon/Xoff or Special character)
7	1	0	0	0	0	0	CTS#/DSR#, RTS#/DTR# change of state
X	0	0	0	0	0	1	None (default) or wake-up indicator

ISR[7:6]: FIFO Enable Status

These bits are set to a logic 0 when the FIFOs are disabled. They are set to a logic 1 when the FIFOs are enabled.

ISR[5:1]: Interrupt Status

These bits indicate the source for a pending interrupt at interrupt priority levels (See **Table 15**). See **"Section 5.5.1, Interrupt Generation:" on page 41** and **"Section 5.5.2, Interrupt Clearing:" on page 41** for details.

ISR[0]: Interrupt Status

- Logic 0 = An interrupt is pending and the ISR contents may be used as a pointer to the appropriate interrupt service routine.
- Logic 1 = No interrupt pending. (default condition)

5.6 FIFO Control Register (FCR) - Write Only

This register is used to enable the FIFOs, clear the FIFOs, set the transmit/receive FIFO trigger levels, and select the DMA mode. The DMA, and FIFO modes are defined as follows:

FCR[7:6]: Receive FIFO Trigger Select

(logic 0 = default, RX trigger level =1)

The FCTR bits [5:4] are associated with these 2 bits. These 2 bits are used to set the trigger level for the receive FIFO. The UART will issue a receive interrupt when the number of the characters in the FIFO crosses the trigger level. **Table 16** shows the complete selections. Note that the receiver and the transmitter cannot use different trigger tables. Whichever selection is made last applies to both the RX and TX side.

FCR[5:4]: Transmit FIFO Trigger Select (requires EFR bit [4]=1)

(logic 0 = default, TX trigger level = 1)

The FCTR bits [7:6] are associated with these 2 bits by selecting one of the four tables. The 4 user selectable trigger levels in 4 tables are supported for compatibility reasons. These 2 bits set the trigger level for the transmit FIFO interrupt. The UART will issue a transmit interrupt when the number of characters in the FIFO falls below the selected trigger level, or when it gets empty in case that the FIFO did not get filled over the trigger level on last re-load. **Table 16** below shows the selections.

FCR[3]: DMA Mode Select

This bit has no effect since TXRDY and RXRDY pins are not available in this device. It is provided for legacy software compatibility.

- Logic 0 = Set DMA to mode 0 (default).
- Logic 1 = Set DMA to mode 1.

FCR[2]: TX FIFO Reset

This bit is only active when FCR bit [0] is active.

- Logic 0= No transmit FIFO reset (default).
- Logic 1 = Reset the transmit FIFO pointers and FIFO level counter logic (the transmit shift register is not cleared or altered). This bit will return to a logic 0 after resetting the FIFO.

FCR[1]: RX FIFO Reset

This bit is only active when FCR bit [0] is active.

- Logic 0 = No receive FIFO reset (default).
- Logic 1 = Reset the receive FIFO pointers and FIFO level counter logic (the receive shift register is not cleared or altered). This bit will return to a logic 0 after resetting the FIFO.

FCR[0]: TX and RX FIFO Enable

- Logic 0 = Disable the transmit and receive FIFO (default).
- Logic 1 = Enable the transmit and receive FIFOs. This bit must be set to logic 1 when other FCR bits are written or they will not be programmed.

TABLE 16: TRANSMIT AND RECEIVE FIFO TRIGGER TABLE AND LEVEL SELECTION

TRIGGER TABLE	FCTR BIT [7]	FCTR BIT [6]	FCR BIT [7]	FCR BIT [6]	FCR BIT [5]	FCR BIT [4]	RECEIVE TRIGGER LEVEL	TRANSMIT TRIGGER LEVEL	COMPATIBILITY
Table-A	0	0			0	0		1 (default)	16C550, 16C2550, 16C2552, 16C554, 16C580, 16L580
			0	0			1 (default)		
			0	1			4		
			1	0			8		
			1	1			14		
Table-B	0	1			0	0		16	16C650A, 16L651
					0	1		8	
					1	0		24	
					1	1		30	
			0	0			8		
			0	1			16		
			1	0			24		
			1	1			28		
Table-C	1	0			0	0		8	16C654
					0	1		16	
					1	0		32	
					1	1		56	
			0	0			8		
			0	1			16		
			1	0			56		
			1	1			60		
Table-D	1	1	X	X	X	X	Programmable via RXTRG register	Programmable via TXTRG register	16L2752, 16L2750, 16C2852, 16C850, 16C854, 16C864

5.7 Line Control Register (LCR) - Read/Write

The Line Control Register is used to specify the asynchronous data communication format. The word or character length, the number of stop bits, and the parity are selected by writing the appropriate bits in this register.

LCR[7]: Baud Rate Divisors Enable

Baud rate generator divisor (DLL, DLM, DLD) enable.

- Logic 0 = Data registers are selected (default).
- Logic 1 = Divisor latch registers (DLL, DLM and DLD) are selected.

LCR[6]: Transmit Break Enable

When enabled the Break control bit causes a break condition to be transmitted (the TX output is forced to a “space’, LOW, state). This condition remains until disabled by setting LCR bit [6] to a logic 0.

- Logic 0 = No TX break condition. (default)
- Logic 1 = Forces the transmitter output (TX) to a “space”, logic 0, for alerting the remote receiver of a line break condition.

LCR[5]: TX and RX Parity Select

If the parity bit is enabled, LCR bit [5] selects the forced parity format.

- LCR bit [5] = logic 0, parity is not forced (default).
- LCR bit [5] = logic 1 and LCR bit [4] = logic 0, parity bit is forced to a logical 1for the transmit and receive data.
- LCR bit [5] = logic 1 and LCR bit [4] = logic 1, parity bit is forced to a logical 0 for the transmit and receive data.

TABLE 17: PARITY PROGRAMMING

LCR BIT [5]	LCR BIT [4]	LCR BIT [3]	PARITY SELECTION
X	X	0	No parity
0	0	1	Odd parity
0	1	1	Even parity
1	0	1	Force parity to mark, “1”
1	1	1	Forced parity to space, “0”

LCR[4]: TX and RX Parity Select

If the parity bit is enabled with LCR bit [3] set to a logic 1, LCR bit [4] selects the even or odd parity format.

- Logic 0 = ODD Parity is generated by forcing an odd number of logic 1’s in the transmitted character. The receiver must be programmed to check the same format (default).
- Logic 1 = EVEN Parity is generated by forcing an even the number of logic 1’s in the transmitted character. The receiver must be programmed to check the same format.

LCR[3]: TX and RX Parity Select

Parity or no parity can be selected via this bit. The parity bit is a simple way used in communications for data integrity check. See **Table 17** above for parity selection summary.

- Logic 0 = No parity.
- Logic 1 = A parity bit is generated during the transmission while the receiver checks for parity error of the data character received.

LCR[2]: TX and RX Stop-bit Length Select

The length of stop bit is specified by this bit in conjunction with the programmed word length.

BIT [2]	WORD LENGTH	STOP BIT LENGTH (BIT TIME(S))
0	5,6,7,8	1 (default)
1	5	1-1/2
1	6,7,8	2

LCR[1:0]: TX and RX Word Length Select

These two bits specify the word length to be transmitted or received.

BIT [1]	BIT [0]	WORD LENGTH
0	0	5 (default)
0	1	6
1	0	7
1	1	8

5.8 Modem Control Register (MCR) - Read/Write

The MCR register is used for controlling the modem interface signals or general purpose inputs/outputs.

MCR[7]: Clock Prescaler Select (requires EFR bit [4]=1)

- Logic 0 = Divide by one. The input clock from the crystal or external clock is fed directly to the Programmable Baud Rate Generator without further modification, i.e., divide by one (default).
- Logic 1 = Divide by four. The prescaler divides the input clock from the crystal or external clock by four and feeds it to the Programmable Baud Rate Generator, hence, data rates become one forth.

MCR[6]: Infrared Encoder/Decoder Enable (requires EFR bit [4]=1)

The state of this bit depends on the sampled logic level of pin ENIR during power up, following a hardware reset (rising edge of RST# input). Afterward user can override this bit for desired operation.

- Logic 0 = Enable the standard modem receive and transmit character interface.
- Logic 1 = Enable infrared IrDA receive and transmit inputs/outputs. While in this mode, the TX/RX output/input are routed to the infrared encoder/decoder. The data input and output levels will conform to the IrDA infrared interface requirement. As such, while in this mode the infrared TX output will be a LOW during idle data conditions. FCTR bit [4] may be selected to invert the RX input signal level going to the decoder for infrared modules that provide rather an inverted output.

MCR[5]: Xon-Any Enable (requires EFR bit [4]=1)

- Logic 0 = Disable Xon-Any function (default).
- Logic 1 = Enable Xon-Any function. In this mode any RX character received will enable Xon, resume data transmission.

MCR[4]: Internal Loopback Enable

- Logic 1 = Disable loopback mode (default).
- Logic 1 = Enable local loopback mode, see loopback section and **Figure 14**.

MCR[3]: Send Char Immediate (OP2 in Local Loopback Mode)

This bit is used to transmit a character immediately irrespective of the bytes currently in the transmit FIFO. The data byte must be loaded into the transmit holding register (THR) immediately following the write to this bit (to set it to a '1'). In other words, no other register must be accessed between setting this bit and writing to the THR. The loaded byte will be transmitted ahead of all the bytes in the TX FIFO, immediately after the character currently being shifted out of the transmit shift register is sent out. The existing line parameters (parity, stop bits) will be used when composing the character. This bit is self clearing, therefore, must be set before sending a custom character each time. Please note that the Transmitter must be enabled for this function (MSR[3] = 0). Also, if software flow control is enabled, the software flow control characters (Xon, Xoff) have higher priority and will get shifted out before the custom byte is transmitted.

- Logic 0 = Send Char Immediate disabled (default).
- Logic 1 = Send Char Immediate enabled.

In Local Loopback Mode (MCR[4] = 1), this bit acts as the legacy OP2 output and controls the CD bit in the MSR register as shown in **Figure 14**. Please make sure that this bit is a '0' when exiting the Local Loopback Mode.

MCR[2]: DTR# or RTS# for Auto Flow Control (OP1 in Local Loopback Mode)

DTR# or RTS# auto hardware flow control select. This bit is in effect only when auto RTS/DTR is enabled by EFR bit [6]. DTR# selection is associated with DSR# and RTS# is with CTS#.

- Logic 0 = Uses RTS# and CTS# pins for auto hardware flow control.
- Logic 1 = Uses DTR# and DSR# pins for auto hardware flow control.

In Local Loopback mode (MCR[4] = 1), this bit acts as the legacy OP1 output and controls the RI bit in the MSR register, as shown in **Figure 14**.

MCR[1]: RTS# Output

The RTS# pin may be used for automatic hardware flow control by enabled by EFR bit [6] and MCR bit [2]=0. If the modem interface is not used, this output may be used for general purpose.

- Logic 0 = Force RTS# output to a HIGH (default).
- Logic 1= Force RTS# output to LOW.

MCR[0]: DTR# Output

The DTR# pin may be used for automatic hardware flow control enabled by EFR bit [6] and MCR bit [2]=1. If the modem interface is not used, this output may be used for general purpose.

- Logic 0 = Force DTR# output to a HIGH (default).
- Logic 1 = Force DTR# output to a LOW.

5.9 Line Status Register (LSR) - Read Only

This register provides the status of data transfers between the UART and the host. If IER bit [2] is set to a logic 1, an LSR interrupt will be generated immediately when any character in the RX FIFO has an error (parity, framing, overrun, break).

LSR[7]: Receive FIFO Data Error Flag

- Logic 0 = No FIFO error (default).
- Logic 1 = An indicator for the sum of all error bits in the RX FIFO. At least one parity error, framing error or break indication is in the FIFO data. This bit clears when there are no more errors in the FIFO.

LSR[6]: Transmitter Empty Flag

This bit is the Transmitter Empty indicator. This bit is set to a logic 1 whenever both the transmit FIFO (or THR, in non-FIFO mode) and the transmit shift register (TSR) are both empty. It is set to logic 0 whenever either the TX FIFO or TSR contains a data character.

LSR[5]: Transmit FIFO Empty Flag

This bit is the Transmit FIFO Empty indicator. This bit indicates that the transmitter is ready to accept a new character for transmission. This bit is set to a logic HIGH when the last data byte is transferred from the transmit FIFO to the transmit shift register. The bit is reset to logic 0 as soon as a data byte is loaded into the transmit FIFO. In the non-FIFO mode this bit is set when the transmit holding register (THR) is empty; it is cleared when at a byte is written to the THR.

LSR[4]: Receive Break Flag

- Logic 0 = No break condition (default).
- Logic 1 = The receiver received a break signal (RX was LOW for one character frame time). In the FIFO mode, only one break character is loaded into the FIFO. The break indication remains until the RX input returns to the idle condition, “mark” or HIGH.

LSR[3]: Receive Data Framing Error Flag

- Logic 1 = No framing error (default).
- Logic 1 = Framing error. The receive character did not have a valid stop bit(s). This error is associated with the character available for reading in RHR.

LSR[2]: Receive Data Parity Error Flag

- Logic 0 = No parity error (default).
- Logic 1 = Parity error. The receive character in RHR (top of the FIFO) does not have correct parity information and is suspect. This error is associated with the character available for reading in RHR.

LSR[1]: Receiver Overrun Flag

- Logic 0 = No overrun error (default).
- Logic 1 = Overrun error. A data overrun error condition occurred in the receive shift register. This happens when additional data arrives while the FIFO is full. In this case the previous data in the receive shift register is overwritten. Note that under this condition the data byte in the receive shift register is not transferred into the FIFO, therefore the data in the FIFO is not corrupted by the error.

LSR[0]: Receive Data Ready Indicator

- Logic 0 = No data in receive holding register or FIFO (default).
- Logic 1 = Data has been received and is saved in the receive holding register or FIFO.

5.10 Modem Status Register (MSR) - Read Only

This register provides the current state of the modem interface signals, or other peripheral device that the UART is connected. Lower four bits of this register are used to indicate the changed information. These bits are set to a logic 1 whenever a signal from the modem changes state. These bits may be used as general purpose inputs/outputs when they are not used with modem signals.

MSR[7]: CD Input Status

Normally this bit is the complement of the CD# input. In the loopback mode this bit is equivalent to bit [3] in the MCR register. The CD# input may be used as a general purpose input when the modem interface is not used.

MSR[6]: RI Input Status

Normally this bit is the complement of the RI# input. In the loopback mode this bit is equivalent to bit [2] in the MCR register. The RI# input may be used as a general purpose input when the modem interface is not used.

MSR[5]: DSR Input Status

DSR# pin may function as automatic hardware flow control signal input if it is enabled and selected by Auto CTS/DSR bit (EFR bit [6]=1) and RTS/DTR flow control select bit (MCR bit [2]=1). Auto CTS/DSR flow control allows starting and stopping of local data transmissions based on the modem DSR# signal. A HIGH on the DSR# pin will stop UART transmitter as soon as the current character has finished transmission, and a LOW will resume data transmission. Normally MSR bit [5] is the complement of the DSR# input. However in the loopback mode, this bit is equivalent to the DTR# bit in the MCR register. The DSR# input may be used as a general purpose input when the modem interface is not used.

MSR[4]: CTS Input Status

CTS# pin may function as automatic hardware flow control signal input if it is enabled and selected by Auto CTS/DSR bit (EFR bit [6]=1) and RTS/DTR flow control select bit (MCR bit [2]=0). Auto CTS/DSR flow control allows starting and stopping of local data transmissions based on the modem CTS# signal. A HIGH on the CTS# pin will stop UART transmitter as soon as the current character has finished transmission, and a LOW will resume data transmission. Normally MSR bit [4] is the complement of the CTS# input. However in the loopback mode, this bit is equivalent to the RTS# bit in the MCR register. The CTS# input may be used as a general purpose input when the modem interface is not used.

MSR[3]: Delta CD# Input Flag

- Logic 0 = No change on CD# input (default).
- Logic 1 = Indicates that the CD# input has changed state since the last time it was monitored. A modem status interrupt will be generated if MSR interrupt is enabled (IER bit [3]).

MSR[2]: Delta RI# Input Flag

- Logic 0 = No change on RI# input (default).
- Logic 1 = The RI# input has changed from a LOW to a HIGH, ending of the ringing signal. A modem status interrupt will be generated if MSR interrupt is enabled (IER bit [3]).

MSR[1]: Delta DSR# Input Flag

- Logic 0 = No change on DSR# input (default).
- Logic 1 = The DSR# input has changed state since the last time it was monitored. A modem status interrupt will be generated if MSR interrupt is enabled (IER bit [3]).

MSR[0]: Delta CTS# Input Flag

- Logic 0 = No change on CTS# input (default).
- Logic 1 = The CTS# input has changed state since the last time it was monitored. A modem status interrupt will be generated if MSR interrupt is enabled (IER bit [3]).

5.11 Modem Status Register (MSR) - Write Only

The upper four bits [7:4] of this register set the delay in number of bits time for the auto RS-485 turn around from transmit to receive.

MSR [7:4]: Auto RS485 Turn-Around Delay (requires EFR bit [4]=1)

When Auto RS485 feature is enabled (FCTR bit [5]=1) and RTS# output is connected to the enable input of a RS-485 transceiver. These 4 bits select from 0 to 15 bit-time delay after the end of the last stop-bit of the last transmitted character. This delay controls when to change the state of RTS# output. This delay is very useful in long-cable networks. **Table 18** shows the selection. The bits are enabled by EFR bit [4].

TABLE 18: AUTO RS485 HALF-DUPLEX DIRECTION CONTROL DELAY FROM TRANSMIT-TO-RECEIVE

MSR[7]	MSR[6]	MSR[5]	MSR[4]	DELAY IN DATA BIT(S) TIME
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
9	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

MSR [3]: Transmitter Disable

This bit can be used to disable the transmitter by halting the Transmit Shift Register (TSR). When this bit is set to a ’1’, the bytes already in the FIFO will not be sent out. Also, any more data loaded into the FIFO will stay in the FIFO and will not be sent out. When this bit is set to a ’0’, the bytes currently in the TX FIFO will be sent out. Please note that setting this bit to a ’1’ stops any character from going out. Also, this bit must be a ’0’ for Send Char Immediate function (see MCR[3]).

- Logic 0 = Enable Transmitter (default).
- Logic 1 = Disable Transmitter.

MSR [2]: Receiver Disable

This bit can be used to disable the receiver by halting the Receive Shift Register (RSR). When this bit is set to a logic 1, the receiver will operate in one of the following ways:

- If a character is being received at the time of setting this bit, that character will be correctly received. No more characters will be received.
- If the receiver is idle at the time of setting this bit, no characters will be received.

The receiver can be enabled and will start receiving characters by resetting this bit to a logic 0. The receiver will operate in one of the following ways:

- If the receiver is idle (RX pin is HIGH) at the time of setting this bit, the next character will be received normally. It is recommended that the receiver be idle when resetting this bit to a logic 0.
- If the receiver is not idle (RX pin is toggling) at the time of setting this bit, the RX FIFO will be filled with unknown data.

Any data that is in the RX FIFO can be read out at any time whether the receiver is disabled or not.

- Logic 0 = Enable Receiver (default).
- Logic 1 = Disable Receiver.

MSR [1:0]: Reserved

5.12 SCRATCH PAD REGISTER (SPR) - Read/Write

This is a 8-bit general purpose register for the user to store temporary data. The content of this register is preserved during sleep mode but becomes 0xFF (default) after a reset or a power off-on cycle.

5.13 FEATURE CONTROL REGISTER (FCTR) - Read/Write

FCTR[7:6]: TX and RX FIFO Trigger Table Select

These 2 bits select the transmit and receive FIFO trigger level table A, B, C or D. When table A, B, or C is selected the auto RTS flow control trigger level is set to "next FIFO trigger level" for compatibility to ST16C550 and ST16C650 series. RTS/DTR# triggers on the next level of the RX FIFO trigger level, in another word, one FIFO level above and one FIFO level below. See **Table 16** for complete selection with FCR bit [5:4] and FCTR bits [7:6], i.e. if Table C is used on the receiver with RX FIFO trigger level set to 56 bytes, RTS/DTR# output will de-assert at 60 and re-assert at 16.

FCTR[5]: Auto RS485 Enable

Auto RS485 half duplex control enable/disable.

- Logic 0 = Standard ST16C550 mode. Transmitter generates an interrupt when transmit holding register (THR) becomes empty. Transmit Shift Register (TSR) may still be shifting data bit out.
- Logic 1 = Enable Auto RS485 half duplex direction control. RTS# output changes from HIGH to LOW when finished sending the last stop bit of the last character out of the TSR register. It changes from LOW to HIGH when a data byte is loaded into the THR or transmit FIFO. The change to HIGH occurs prior sending the start-bit. It also changes the transmitter interrupt from transmit holding to transmit shift register (TSR) empty.

FCTR[4]: Infrared RX Input Logic Select

- Logic 0 = Select RX input as active HIGH encoded IrDA data, normal (default).
- Logic 1 = Select RX input as active LOW encoded IrDA data, inverted.

FCTR [3:0] - Auto RTS/DTR Flow Control Hysteresis Select

These bits select the auto RTS/DTR flow control hysteresis and only valid when TX and RX Trigger Table-D is selected (FCTR bit [7:6] are set to logic 1). The RTS/DTR hysteresis is referenced to the RX FIFO trigger level. After reset, these bits are set to logic 0 selecting the next FIFO trigger level for hardware flow control. **Table 19** below shows the 16 selectable hysteresis levels.

TABLE 19: 16 SELECTABLE HYSTERESIS LEVELS WHEN TRIGGER TABLE-D IS SELECTED

FCTR BIT [3]	FCTR BIT [2]	FCTR BIT [1]	FCTR BIT [0]	RTS/DTR HYSTERESIS (CHARACTERS)
0	0	0	0	0
0	0	0	1	+/- 4
0	0	1	0	+/- 6
0	0	1	1	+/- 8
0	1	0	0	+/- 8
0	1	0	1	+/- 16
0	1	1	0	+/- 24
0	1	1	1	+/- 32
1	1	0	0	+/- 12
1	1	0	1	+/- 20
1	1	1	0	+/- 28
1	1	1	1	+/- 36
1	0	0	0	+/- 40
1	0	0	1	+/- 44
1	0	1	0	+/- 48
1	0	1	1	+/- 52

5.14 Enhanced Feature Register (EFR) - Read/Write

Enhanced features are enabled or disabled using this register. Bits [3:0] provide single or dual consecutive character software flow control selection (see **Table 20**). When the Xon1 and Xon2 and Xoff1 and Xoff2 modes are selected, the double 8-bit words are concatenated into two sequential characters. Caution: note that whenever changing the TX or RX flow control bits, always reset all bits back to logic 0 (disable) before programming a new setting.

EFR[7]: Auto CTS Flow Control Enable

Automatic CTS or DSR Flow Control.

- Logic 0 = Automatic CTS/DSR flow control is disabled (default).
- Logic 1 = Enable Automatic CTS/DSR flow control. Transmission stops when CTS/DSR# pin de-asserts (HIGH). Transmission resumes when CTS/DSR# pin is asserted (LOW). The selection for CTS# or DSR# is through MCR bit [2].

EFR[6]: Auto RTS or DTR Flow Control Enable

RTS#/DTR# output may be used for hardware flow control by setting EFR bit [6] to logic 1. When Auto RTS/DTR is selected, an interrupt will be generated when the receive FIFO is filled to the programmed trigger level and RTS/DTR# will de-assert (HIGH) at the next upper trigger or selected hysteresis level. RTS/DTR# will re-assert (LOW) when FIFO data falls below the next lower trigger or selected hysteresis level (see FCTR bits 4-7). The RTS# or DTR# output must be asserted (LOW) before the auto RTS/DTR can take effect. The selection for RTS# or DTR# is through MCR bit [2]. RTS/DTR# pin will function as a general purpose output when hardware flow control is disabled.

- Logic 0 = Automatic RTS/DTR flow control is disabled (default).
- Logic 1 = Enable Automatic RTS/DTR flow control.

EFR[5]: Special Character Detect Enable

- Logic 0 = Special Character Detect Disabled (default).
- Logic 1 = Special Character Detect Enabled. The UART compares each incoming receive character with data in Xoff-2 register. If a match exists, the received data will be transferred to FIFO and ISR bit [4] will be set to indicate detection of the special character. bit [0] corresponds with the LSB bit for the receive character. If flow control is set for comparing Xon1, Xoff1 (EFR [1:0]=10) then flow control and special character work normally. However, if flow control is set for comparing Xon2, Xoff2 (EFR[1:0]=01) then flow control works normally, but Xoff2 will not go to the FIFO, and will generate an Xoff interrupt and a special character interrupt.

EFR[4]: Enhanced Function Bits Enable

Enhanced function control bit. This bit enables the enhanced functions in IER bits [7:5], ISR bits [5:4], FCR bits [5:4], MCR bits [7:5,3:2] and MSR [7:2] bits to be modified. After modifying any enhanced bits, EFR bit [4] can be set to a logic 0 to latch the new values. This feature prevents legacy software from altering or overwriting the enhanced functions once set. Normally, it is recommended to leave it enabled, HIGH.

- Logic 0 = modification disable/latch enhanced features. IER bits [7:5], ISR bits [5:4], FCR bits [5:4], MCR bits [7:5, 3:2] and MSR [7:2] bits are saved to retain the user settings. After a reset, all these bits are set to a logic 0 to be compatible with ST16C550 mode (default).
- Logic 1 = Enables the enhanced functions. When this bit is set to a logic 1 all enhanced features are enabled.

EFR[3:0]: Software Flow Control Select

Combinations of software flow control can be selected by programming these bits, as shown in Table 20 below.

TABLE 20: SOFTWARE FLOW CONTROL FUNCTIONS

EFR BIT [3]	EFR BIT [2]	EFR BIT [1]	EFR BIT [0]	TRANSMIT AND RECEIVE SOFTWARE FLOW CONTROL
0	0	0	0	No TX and RX flow control (default and reset)
0	0	X	X	No transmit flow control
1	0	X	X	Transmit Xon1, Xoff1
0	1	X	X	Transmit Xon2, Xoff2
1	1	X	X	Transmit Xon1 and Xon2, Xoff1 and 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
1	0	1	1	Transmit Xon1, Xoff1 Receiver compares Xon1 or Xon2, Xoff1 or Xoff2
0	1	1	1	Transmit Xon2, Xoff2 Receiver compares Xon1 or Xon2, Xoff1 or Xoff2
1	1	1	1	Transmit Xon1 and Xon2, Xoff1 and Xoff2 Receiver compares Xon1 and Xon2, Xoff1 and Xoff2
0	0	1	1	No transmit flow control Receiver compares Xon1 and Xon2, Xoff1 and Xoff2

5.15 TXCNT[7:0]: Transmit FIFO Level Counter - Read Only

Transmit FIFO level byte count from 0x00 (LOW) to 0x40 (64). This 8-bit register gives an indication of the number of characters in the transmit FIFO. The FIFO level Byte count register is read only. The user can take advantage of the FIFO level byte counter for faster data loading to the transmit FIFO, which reduces CPU bandwidth requirements.

5.16 TXTRG [7:0]: Transmit FIFO Trigger Level - Write Only

An 8-bit value written to this register sets the TX FIFO trigger level from 0x00 (zero) to 0x40 (64). The TX FIFO trigger level generates an interrupt whenever the data level in the transmit FIFO falls below this preset trigger level.

5.17 RXCNT[7:0]: Receive FIFO Level Counter - Read Only

Receive FIFO level byte count from 0x00 (zero) to 0x40 (64). It gives an indication of the number of characters in the receive FIFO. The FIFO level byte count register is read only. The user can take advantage of the FIFO level byte counter for faster data unloading from the receiver FIFO, which reduces CPU bandwidth requirements.

5.18 RXTRG[7:0]: Receive FIFO Trigger Level - Write Only

An 8-bit value written to this register, sets the RX FIFO trigger level from 0x00 (zero) to 0x40 (64). The RX FIFO trigger level generates an interrupt whenever the receive FIFO level rises to this preset trigger level.

5.19 XOFF1, XOFF2, XON1 AND XON2 REGISTERS, WRITE ONLY

These registers are used to program the Xoff1, Xoff2, Xon1 and Xon2 control characters respectively.

5.20 XCHAR REGISTER, READ ONLY

This register gives the status of the last sent control character (Xon or Xoff) and the last received control character (Xon or Xoff). This register will be reset to 0x00 if, at anytime, the Software Flow Control is disabled.

XCHAR [7:4]: Reserved

XCHAR [3]: Transmit Xon Indicator

If the last transmitted control character was a Xon character or characters (Xon1, Xon2), this bit will be set to a logic 1. This bit will clear after the read.

XCHAR [2]: Transmit Xoff Indicator

If the last transmitted control character was a Xoff character or characters (Xoff1, Xoff2), this bit will be set to a logic 1. This bit will clear after the read.

XCHAR [1]: Xon Detect Indicator

If the last received control character was a Xon character or characters (Xon1, Xon2), this bit will be set to a logic 1. This bit will clear after the read.

XCHAR [0]: Xoff Detect Indicator

If the last received control character was a Xoff character or characters (Xoff1, Xoff2), this bit will be set to a logic 1. This bit will clear after the read.

TABLE 21: UART RESET CONDITIONS

REGISTERS	RESET STATE
DLL	Bits [7:0] = 0x01
DLM	Bits [7:0] = 0x00
DLD	Bits [7:0] = 0x00
RHR	Bits [7:0] = 0xXX
THR	Bits [7:0] = 0xXX
IER	Bits [7:0] = 0x00
FCR	Bits [7:0] = 0x00
ISR	Bits [7:0] = 0x01
LCR	Bits [7:0] = 0x00
MCR	Bits [7:0] = 0x00
LSR	Bits [7:0] = 0x60
MSR	Bits [3:0] = logic 0 Bits [7:4] = logic levels of the inputs
SPR	Bits [7:0] = 0xFF
FCTR	Bits [7:0] = 0x00
EFR	Bits [7:0] = 0x00
TFCNT	Bits [7:0] = 0x00
TFTRG	Bits [7:0] = 0x00
RFCNT	Bits [7:0] = 0x00
RFTRG	Bits [7:0] = 0x00
XCHAR	Bits [7:0] = 0x00
XON1	Bits [7:0] = 0x00
XON2	Bits [7:0] = 0x00
XOFF1	Bits [7:0] = 0x00
XOFF2	Bits [7:0] = 0x00

I/O SIGNALS	RESET STATE
TX[ch-1:0]	HIGH
IRTX[ch-1:0]	LOW
RTS#[ch-1:0]	HIGH
DTR#[ch-1:0]	HIGH
EECK	LOW
EECS	LOW
EEDI	LOW

ABSOLUTE MAXIMUM RATINGS

Power Supply Range	4 Volts
Voltage at Any Pin	-0.5 to 4V
Operating Temperature	-40^{o} to $+85^{o}$ C
Storage Temperature	-65^{o} to $+150^{o}$ C
Package Dissipation	500 mW
Thermal Resistance (14x14x1.0mm 100-TQFP)	theta-ja = 42, theta-jc = 8

ELECTRICAL CHARACTERISTICS

DC ELECTRICAL CHARACTERISTICS

TA=-40^{O} TO +85^{O}C (INDUSTRIAL GRADE) SUPPLY VOLTAGE, VCC = 3.0 - 3.6V

SYMBOL	PARAMETER	MIN	MAX	UNITS	CONDITION	NOTES
V_{IL}	Input Low Voltage	-0.5	0.3VCC	V	For PCI bus inputs	
		-0.5	0.8	V	For Non-PCI bus inputs	
V_{IH}	Input High Voltage	0.5VCC	VCC + 0.5	V	For PCI bus inputs	
		2.0	6.0	V	For Non-PCI bus inputs	5V tolerant inputs
		2.0	VCC + 0.5		For external clock (XTAL1) input only	Not 5V tolerant
V_{OL}	Output Low Voltage		0.1VCC	V	I_{OL} = 1.5mA	PCI bus outputs
			0.4	V	I_{OL} = 6mA	Non-PCI bus outputs
V_{OH}	Output High Voltage	0.9VCC		V	I_{OH} = -0.5mA	PCI bus outputs
		2.4		V	I_{OH} = -2mA	Non-PCI bus outputs
I_{IL}	Input Leakage Current		±10	µA	$0 < V_{in} < VCC$	
I_{CL}	Input Clock Leakage		±10	µA		
C_{IN}	Input Pin Capacitance		10	pF		
C_{CLK}	CLK Pin Capacitance	5	12	pF		
C_{IDSEL}	IDSEL Pin Capacitance		8	pF		
I_{Off}	PME# input leakage	-	1	µA	$V_o \leq 3.6V$, VCC off or floating	
I_{CC}	Power Supply Current		4	mA	PCI bus CLK and Ext. Clock = 2MHz, all inputs at VCC or GND and all outputs are unloaded.	
I_{SLEEP}	Sleep Current		1	mA	Both UARTs asleep. AD[31:0] at GND, all inputs at VCC or GND.	

AC ELECTRICAL CHARACTERISTICS

TA=-40^{O} TO+85^{O}C (INDUSTRIAL GRADE) VCC = 3.0 - 3.6V

SYMBOL	PARAMETER	MIN	MAX	UNITS	NOTES
XTAL1	UART Crystal Oscillator		24	MHz	On-chip osc.
ECLK	External Clock		64	MHz	
$I_{OH(AC, min)}$	Switching Current High, Min	-12VCC		mA	V_{out} = 0.3VCC
$I_{OH(AC, max)}$	Switching Current High, Max		-32VCC	mA	V_{out} = 0.7VCC
$I_{OL(AC, min)}$	Switching Current Low, Min	16VCC		mA	V_{out} = 0.6VCC
$I_{OL(AC, max)}$	Switching Current Low, Max		38VCC	mA	V_{out} = 0.18VCC
I_{CH}	High Clamp Current	25+(Vin-VCC-1)/0.015		mA	VCC+4 > Vin ≥ VCC+1
I_{CL}	Low Clamp Current	-25+(Vin+1)/0.015		mA	-3 < Vin ≤ -1
$Slew_R$	Output Rise Slew Rate	1	4	V/ns	0.3VCC to 0.6VCC
$Slew_F$	Output Fall Slew Rate	1	4	V/ns	0.6VCC to 0.3VCC
T_{cyc}	CLK Cycle Time	15		ns	PCI bus Clock, CLK up to 66.67MHz
T_{high}	CLK High Time	6		ns	
T_{low}	CLK Low Time	6		ns	
	CLK Slew Rate	1.5	4	V/ns	
T_{val}	CLK to Signal Valid Delay	2	6	ns	Bused and point to point signals
T_{on}	Float to Active Delay	2		ns	
T_{off}	Active to Float Delay		14	ns	
T_{su}	Input Setup Time to CLK - bused signals	3		ns	
T_{su} (ptp)	Input Setup Time to CLK - point to point signals	5		ns	
T_h	Input Hold Time from CLK	0		ns	
T_{rst}	RST# Active Time After Power Stable	1		ms	
$T_{rst-clk\#}$	RST# Active Time After CLK Stable	100		us	
$T_{rst-off}$	Reset Active to output float delay		40	ns	
T_{rhfa}	RST# HIGH to first Configu-ration access	2^{25}		clocks	
T_{rhff}	RST# HIGH to first FRAME# assertion	5		clocks	
	RST# Slew Rate	50		mV/ns	

FIGURE 19. PCI BUS CONFIGURATION SPACE REGISTERS READ AND WRITE OPERATION

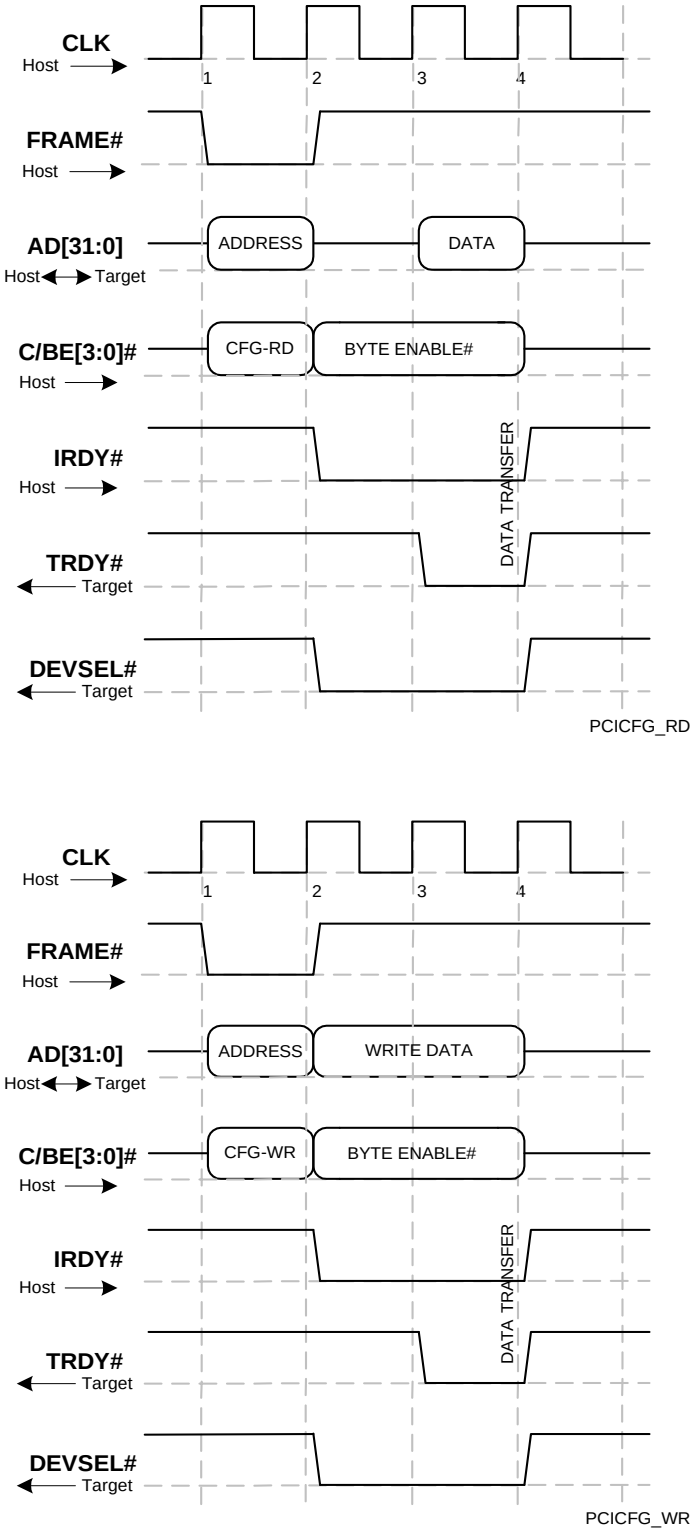
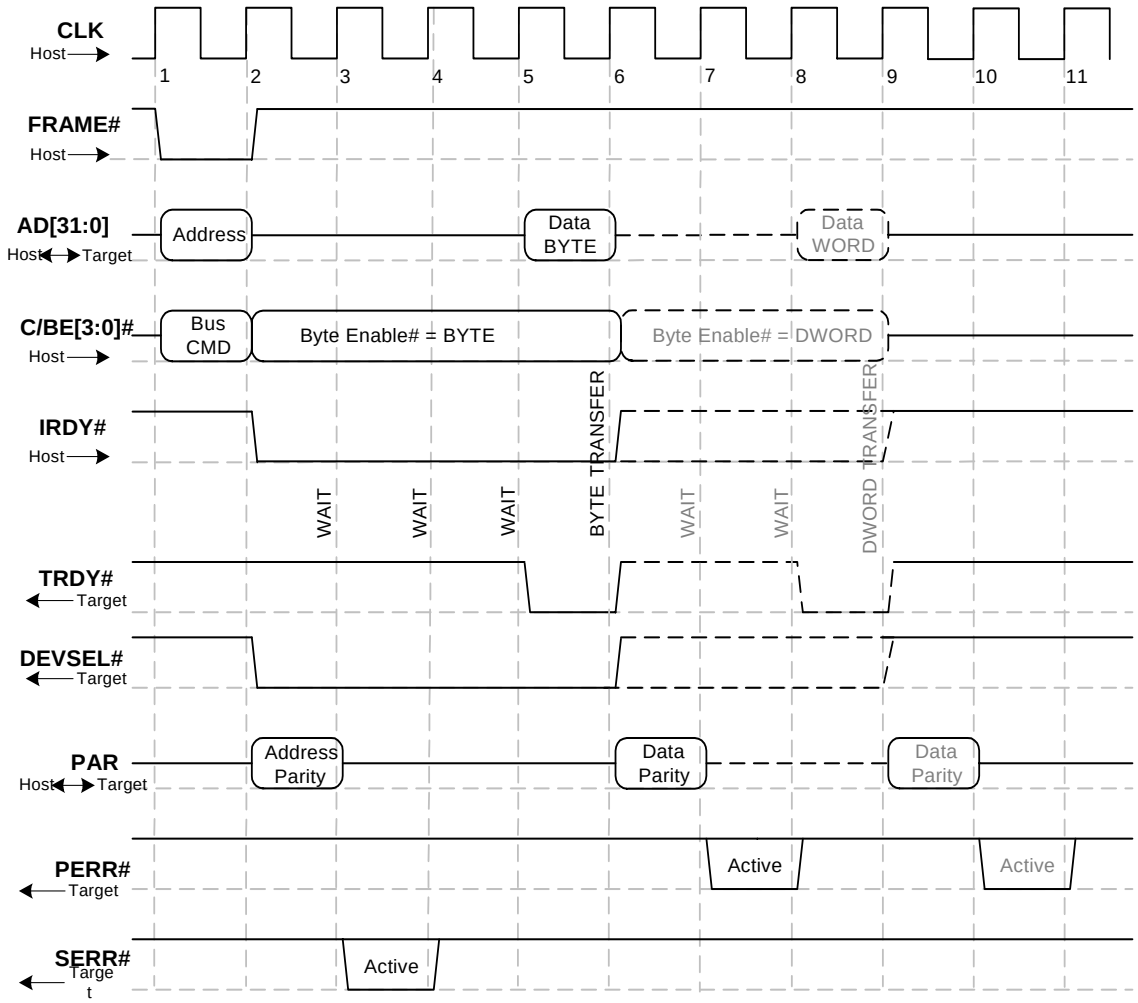


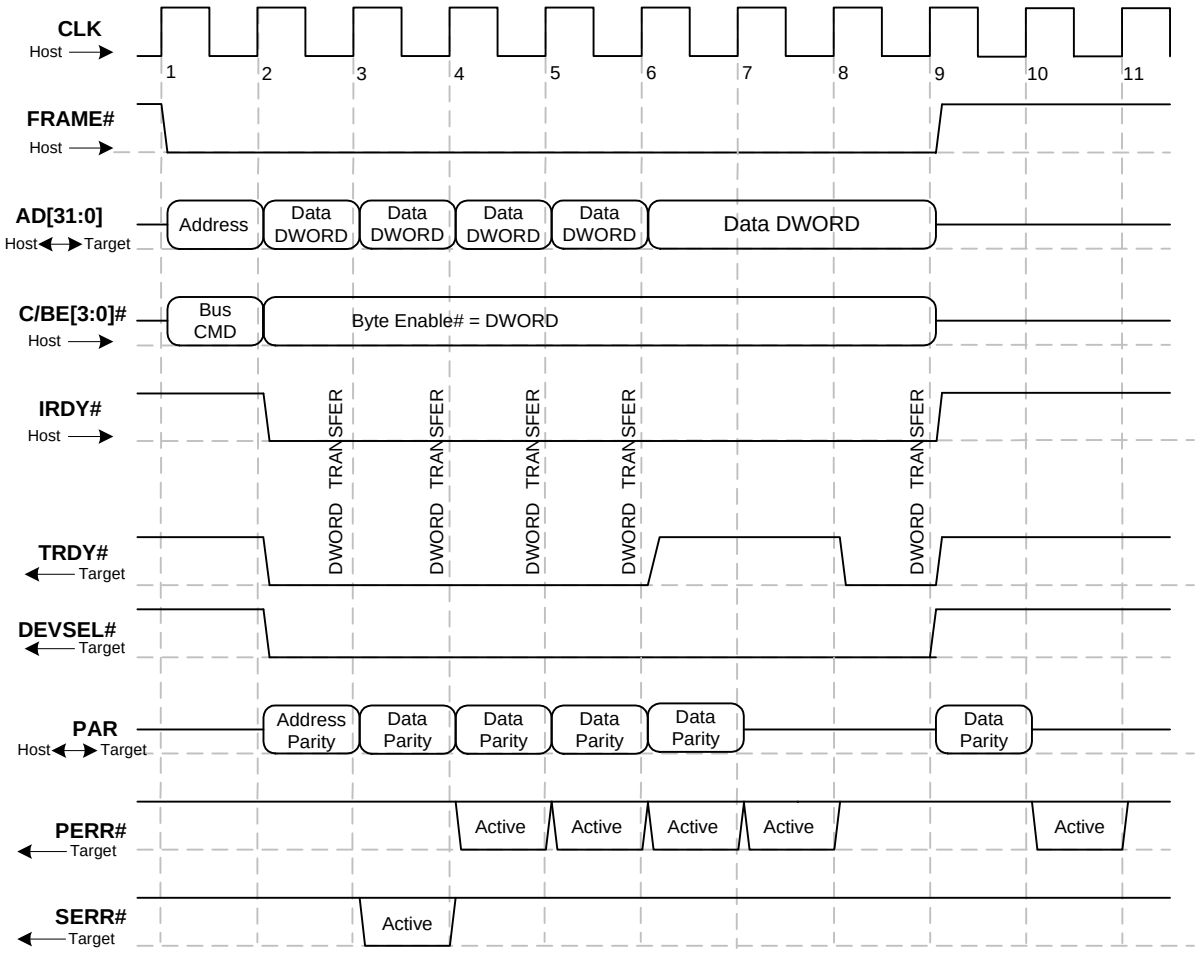
FIGURE 20. DEVICE CONFIGURATION AND UART REGISTERS READ OPERATION FOR A BYTE OR DWORD



Note: PERR# and SERR are optional in a bus target application.
Even Parity is on AD[31:0], C/BE[3:0]#, and PAR

PCI_RD1

FIGURE 21. DEVICE CONFIGURATION REGISTERS, UART REGISTERS AND TRANSMIT DATA BURST WRITE OPERATION



Note: PERR# and SERR are optional in a bus target application.
Even Parity is on AD[31:0], C/BE[3:0]#, and PAR

PCI BWR

FIGURE 22. DEVICE CONFIGURATION REGISTERS, UART REGISTERS AND RECEIVE DATA BURST READ OPERATION

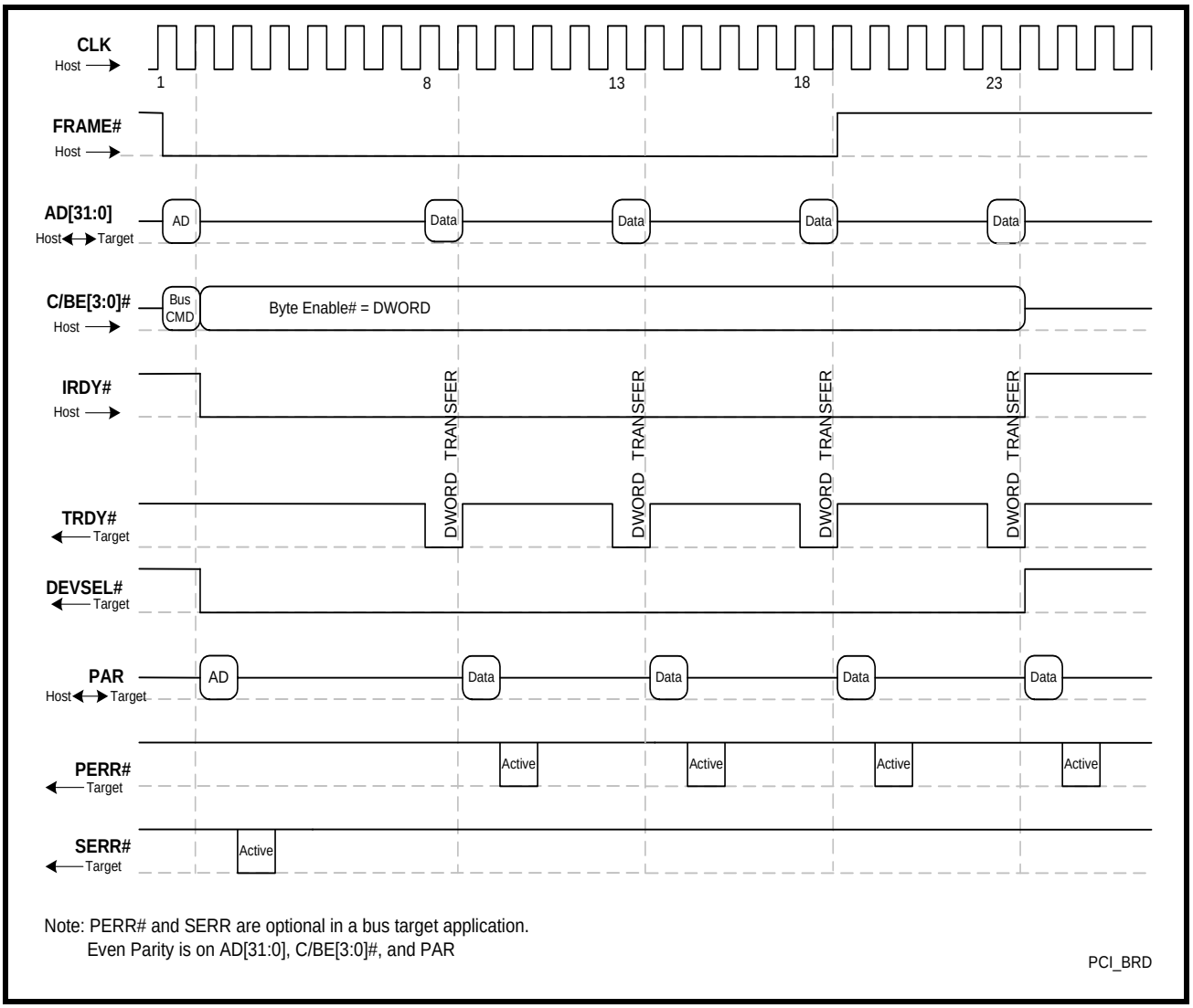


FIGURE 23. 3.3V PCI BUS CLOCK (DC TO 66MHZ)

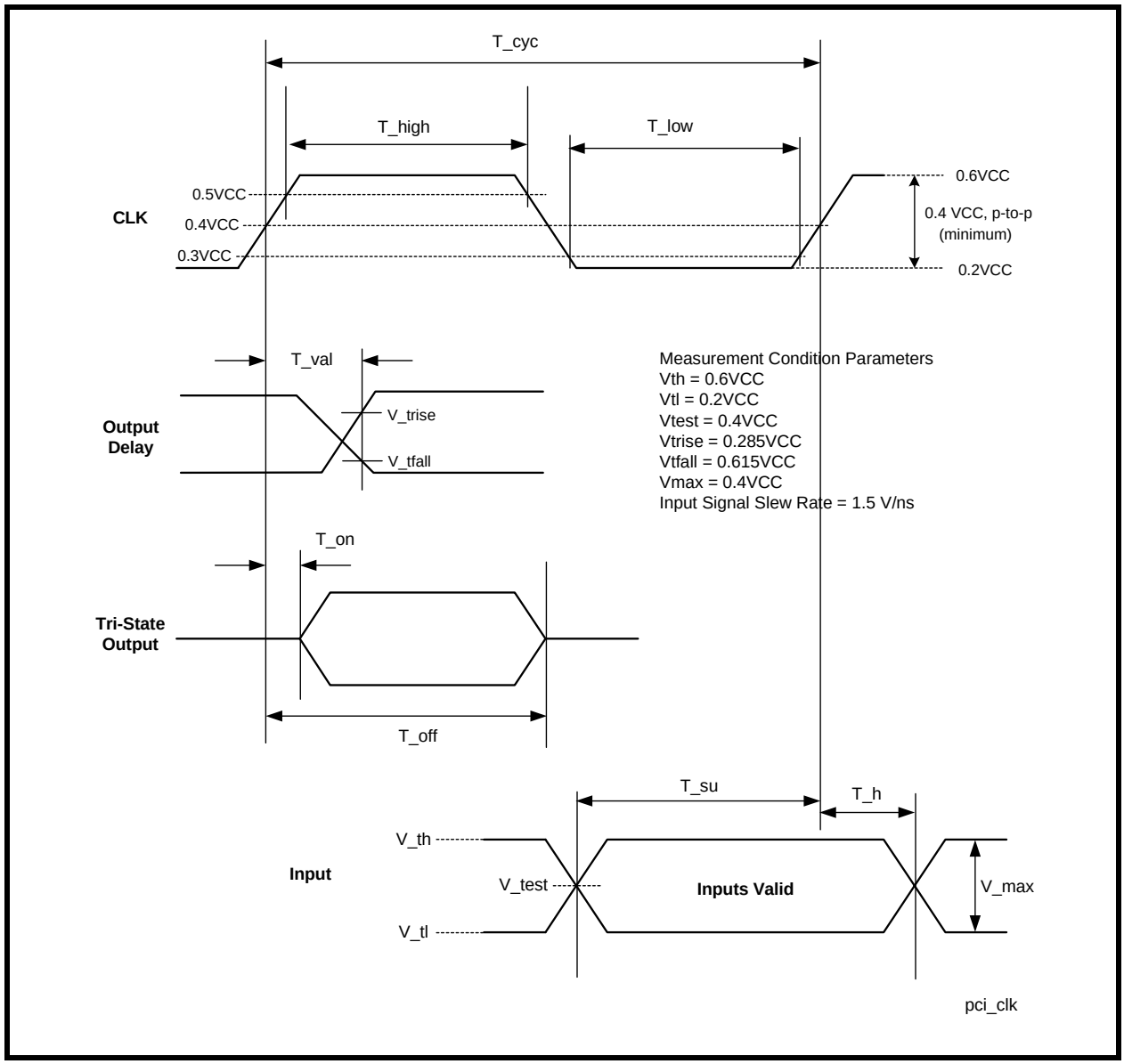


FIGURE 24. TRANSMIT DATA INTERRUPT AT TRIGGER LEVEL

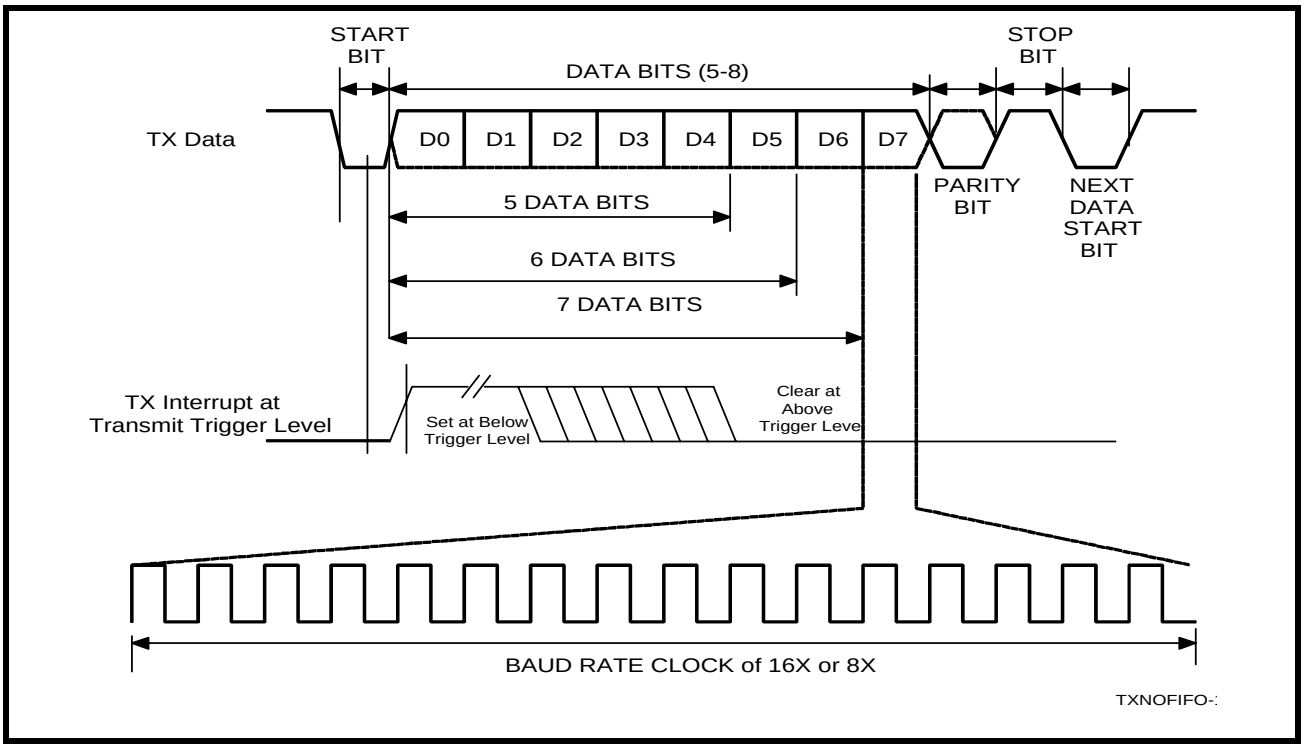
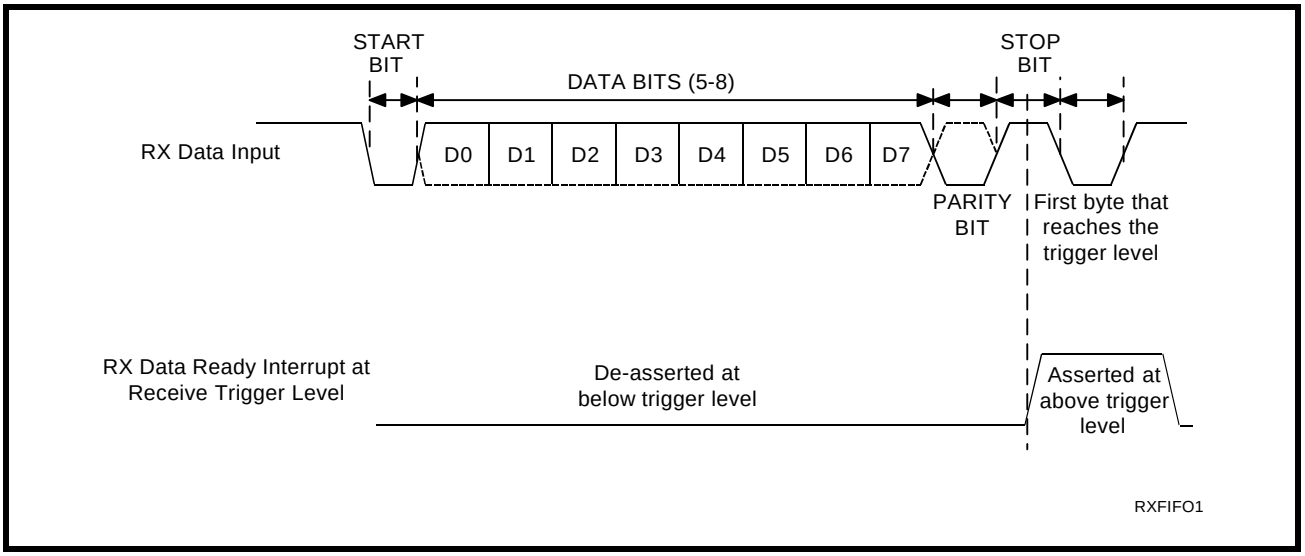


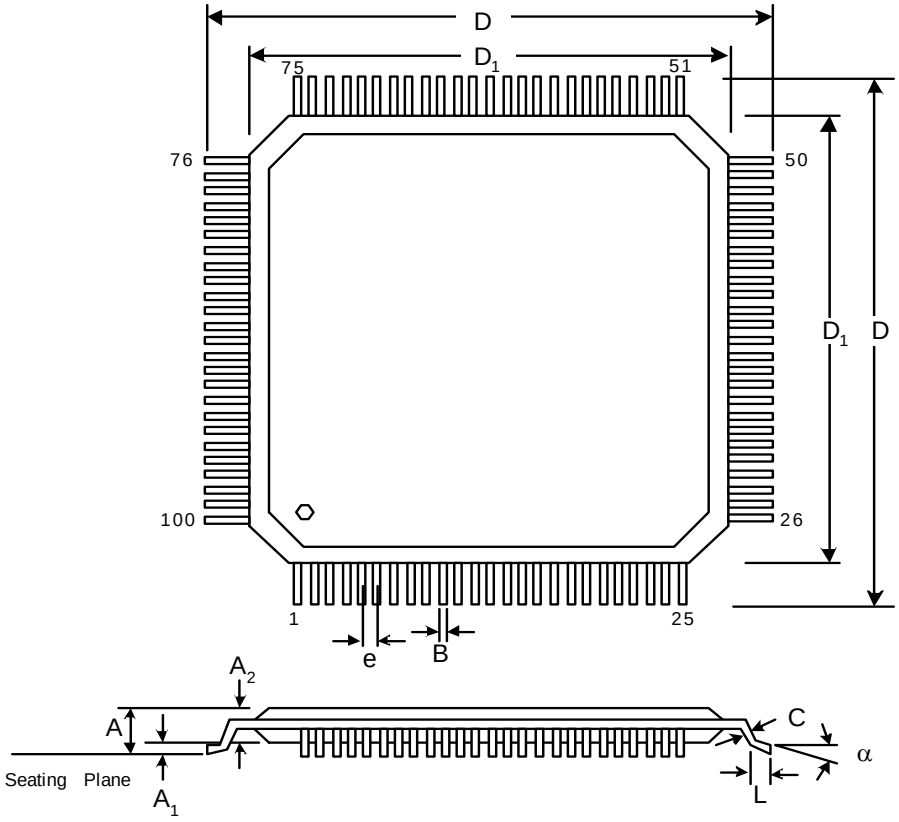
FIGURE 25. RECEIVE DATA READY INTERRUPT AT TRIGGER LEVEL



PACKAGE DIMENSIONS

100 LEAD THIN QUAD FLAT PACK
(14 x 14 x 1.0 mm, TQFP)

Rev.1.00



SYMBOL	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.039	0.047	1.00	1.20
A1	0.002	0.006	0.05	0.15
A2	0.037	0.041	0.95	1.05
B	0.007	0.011	0.17	0.27
C	0.004	0.008	0.09	0.20
D	0.622	0.638	15.80	16.20
D1	0.547	0.555	13.90	14.10
e	0.020 BSC		0.50 BSC	
L	0.018	0.030	0.45	0.75
α	0^o	7^o	0^o	7^o

Note: The control dimension is in millimeter.

REVISION HISTORY

DATE	REVISION	DESCRIPTION
February 2006	1.0.0	Initial Datasheet Release.
March 2006	1.0.1	Corrected maximum external clock frequency in AC Electrical Characteristics.
July 2008	1.0.2	Corrected description of Xon/Xoff/Special character interrupt.

NOTICE

EXAR Corporation reserves the right to make changes to the products contained in this publication in order to improve design, performance or reliability. EXAR Corporation assumes no responsibility for the use of any circuits described herein, conveys no license under any patent or other right, and makes no representation that the circuits are free of patent infringement. Charts and schedules contained here in are only for illustration purposes and may vary depending upon a user's specific application. While the information in this publication has been carefully checked; no responsibility, however, is assumed for inaccuracies.

EXAR Corporation does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless EXAR Corporation receives, in writing, assurances to its satisfaction that: (a) the risk of injury or damage has been minimized; (b) the user assumes all such risks; (c) potential liability of EXAR Corporation is adequately protected under the circumstances.

Copyright 2008 EXAR Corporation

Datasheet July 2008.

Send your UART technical inquiry with technical details to hotline: uarttechsupport@exar.com.

Reproduction, in part or whole, without the prior written consent of EXAR Corporation is prohibited.

TABLE OF CONTENTS

GENERAL DESCRIPTION 1
APPLICATIONS 1
FEATURES 1
FIGURE 1. BLOCK DIAGRAM OF THE XR17V252 1
FIGURE 2. PIN OUT OF THE XR17V252 2
ORDERING INFORMATION 2
PIN DESCRIPTIONS 3
PCI LOCAL BUS INTERFACE 3
MODEM OR SERIAL I/O INTERFACE 3
ANCILLARY SIGNALS 4
FUNCTIONAL DESCRIPTION 6
PCI Local Bus Interface 6
PCI Local Bus Configuration Space Registers 6
Power Management Registers 6
EEPROM Interface 6
1.0 XR17V252 INTERNAL REGISTERS 7
FIGURE 3. THE XR17V252 REGISTER SETS 7
1.1 PCI LOCAL BUS CONFIGURATION SPACE REGISTERS 7
TABLE 1: PCI LOCAL BUS CONFIGURATION SPACE REGISTERS 8
1.2 POWER MANAGEMENT REGISTERS 9
TABLE 2: POWER MANAGEMENT REGISTERS 9
1.2.1 POWER STATES AND POWER STATE TRANSITIONS OF THE V252 10
D0 State 10
D3hot State 10
D3cold State 11
FIGURE 4. POWER STATE TRANSITIONS OF THE XR17V252 11
1.3 SPECIAL READ/WRITE REGISTER TO STORE USER INFORMATION 11
TABLE 3: SPECIAL READ/WRITE REGISTER 11
1.4 EEPROM INTERFACE 12
TABLE 4: EEPROM ADDRESS DEFINITIONS 12
1.5 DEVICE INTERNAL REGISTER SETS 12
TABLE 5: XR17V252 UART AND DEVICE CONFIGURATION REGISTERS 13
1.6 DEVICE CONFIGURATION REGISTERS 14
TABLE 6: DEVICE CONFIGURATION REGISTERS SHOWN IN BYTE ALIGNMENT 14
TABLE 7: DEVICE CONFIGURATION REGISTERS SHOWN IN DWORD ALIGNMENT 15
1.6.1 THE GLOBAL INTERRUPT REGISTER 15
FIGURE 5. THE GLOBAL INTERRUPT REGISTER, INT0, INT1, INT2 AND INT3 16
TABLE 8: UART CHANNEL [1:0] INTERRUPT SOURCE ENCODING 16
TABLE 9: UART CHANNEL [1:0] INTERRUPT CLEARING 16
1.6.2 GENERAL PURPOSE 16-BIT TIMER/COUNTER [TIMERMSB, TIMELSB, TIMER, TIMECNTL] (DEFAULT 0XXX-XX-00-00) 17
FIGURE 6. TIMER/COUNTER CIRCUIT 17
TABLE 10: TIMER CONTROL REGISTERS 18
TIMER OPERATION 18
FIGURE 7. TIMER OUTPUT IN ONE-SHOT AND RE-TRIGGERABLE MODES 19
FIGURE 8. INTERRUPT OUTPUT (ACTIVE LOW) IN ONE-SHOT AND RE-TRIGGERABLE MODES 20
1.6.3 8XMODE [7:0] (DEFAULT 0X00) 20
1.6.4 REGA [15:8] RESERVED 20
1.6.5 RESET [23:16] - (DEFAULT 0X00) 20
1.6.6 SLEEP [31:24] (DEFAULT 0X00) 20
1.6.7 DEVICE IDENTIFICATION AND REVISION 21
1.6.8 REGB REGISTER 21
1.6.9 MULTI-PURPOSE INPUTS AND OUTPUTS 22
1.6.10 MPIO REGISTER 22
FIGURE 9. MULTIPURPOSE INPUT/OUTPUT INTERNAL CIRCUIT 22
2.0 CRYSTAL OSCILLATOR / BUFFER 24
FIGURE 10. TYPICAL CRYSTAL CONNECTIONS 24
3.0 TRANSMIT AND RECEIVE DATA 25
3.1 FIFO DATA LOADING AND UNLOADING IN 32-BIT FORMAT 25

3.1.1 NORMAL RX FIFO DATA UNLOADING AT LOCATIONS 0X100 (CHANNEL 0) AND 0X300 (CHANNEL 1) 26
3.1.2 SPECIAL RX FIFO DATA UNLOADING AT LOCATIONS 0X180 (CHANNEL 0) AND 0X380 (CHANNEL 1) 26
3.1.3 TX FIFO DATA LOADING AT LOCATIONS 0X100 (CHANNEL 0) AND 0X300 (CHANNEL 1) 27
3.2 FIFO DATA LOADING AND UNLOADING THROUGH THE UART CHANNEL REGISTERS, THR AND RHR IN 8-BIT FORMAT 27
TABLE 11: TRANSMIT AND RECEIVE DATA REGISTER IN BYTE FORMAT, 16C550 COMPATIBLE 27
4.0 UART 28
4.1 PROGRAMMABLE BAUD RATE GENERATOR WITH FRACTIONAL DIVISOR 28
FIGURE 11. BAUD RATE GENERATOR 29
TABLE 12: TYPICAL DATA RATES WITH A 24 MHZ CRYSTAL OR EXTERNAL CLOCK AT 16X SAMPLING 29
4.2 AUTOMATIC HARDWARE (RTS/CTS OR DTR/DSR) FLOW CONTROL OPERATION 30
FIGURE 12. AUTO RTS/DTR AND CTS/DSR FLOW CONTROL OPERATION 31
4.3 INFRARED MODE 32
FIGURE 13. INFRARED TRANSMIT DATA ENCODING AND RECEIVE DATA DECODING 32
4.4 INTERNAL LOOPBACK 33
FIGURE 14. INTERNAL LOOP BACK 33
4.5 UART CHANNEL CONFIGURATION REGISTERS AND ADDRESS DECODING 33
TABLE 13: UART CHANNEL CONFIGURATION REGISTERS. 34
TABLE 14: UART CHANNEL CONFIGURATION REGISTERS DESCRIPTION. SHADED BITS ARE ENABLED BY EFR BIT-4. 35
4.6 TRANSMITTER 36
4.6.1 TRANSMIT HOLDING REGISTER (THR) 36
4.6.2 TRANSMITTER OPERATION IN NON-FIFO MODE 36
FIGURE 15. TRANSMITTER OPERATION IN NON-FIFO MODE 37
4.6.3 TRANSMITTER OPERATION IN FIFO MODE 37
4.6.4 AUTO RS485 OPERATION 37
FIGURE 16. TRANSMITTER OPERATION IN FIFO AND FLOW CONTROL MODE 38
4.7 RECEIVER 38
4.7.1 RECEIVER OPERATION IN NON-FIFO MODE 38
FIGURE 17. RECEIVER OPERATION IN NON-FIFO MODE 38
4.7.2 RECEIVER OPERATION WITH FIFO 39
FIGURE 18. RECEIVER OPERATION IN FIFO AND FLOW CONTROL MODE 39
5.0 UART CONFIGURATION REGISTERS 39
5.1 RECEIVE HOLDING REGISTER (RHR) - READ ONLY 39
5.2 TRANSMIT HOLDING REGISTER (THR) - WRITE ONLY 39
5.3 BAUD RATE GENERATOR DIVISORS (DLM, DLL AND DLD) 39
5.4 INTERRUPT ENABLE REGISTER (IER) - READ/WRITE 39
5.4.1 IER VERSUS RECEIVE FIFO INTERRUPT MODE OPERATION 39
5.4.2 IER VERSUS RECEIVE/TRANSMIT FIFO POLLED MODE OPERATION 40
5.5 INTERRUPT STATUS REGISTER (ISR) - READ ONLY 41
5.5.1 INTERRUPT GENERATION: 41
5.5.2 INTERRUPT CLEARING: 41
TABLE 15: INTERRUPT SOURCE AND PRIORITY LEVEL 42
5.6 FIFO CONTROL REGISTER (FCR) - WRITE ONLY 42
TABLE 16: TRANSMIT AND RECEIVE FIFO TRIGGER TABLE AND LEVEL SELECTION 44
5.7 LINE CONTROL REGISTER (LCR) - READ/WRITE 44
TABLE 17: PARITY PROGRAMMING 45
5.8 MODEM CONTROL REGISTER (MCR) - READ/WRITE 46
5.9 LINE STATUS REGISTER (LSR) - READ ONLY 47
5.10 MODEM STATUS REGISTER (MSR) - READ ONLY 48
5.11 MODEM STATUS REGISTER (MSR) - WRITE ONLY 49
TABLE 18: AUTO RS485 HALF-DUPLEX DIRECTION CONTROL DELAY FROM TRANSMIT-TO-RECEIVE 50
5.12 SCRATCH PAD REGISTER (SPR) - READ/WRITE 51
5.13 FEATURE CONTROL REGISTER (FCTR) - READ/WRITE 51
TABLE 19: 16 SELECTABLE HYSTERESIS LEVELS WHEN TRIGGER TABLE-D IS SELECTED 52
5.14 ENHANCED FEATURE REGISTER (EFR) - READ/WRITE 52
TABLE 20: SOFTWARE FLOW CONTROL FUNCTIONS 54
5.15 TXCNT[7:0]: TRANSMIT FIFO LEVEL COUNTER - READ ONLY 54
5.16 TXTRG [7:0]: TRANSMIT FIFO TRIGGER LEVEL - WRITE ONLY 54
5.17 RXCNT[7:0]: RECEIVE FIFO LEVEL COUNTER - READ ONLY 54
5.18 RXTRG[7:0]: RECEIVE FIFO TRIGGER LEVEL - WRITE ONLY 54
5.19 XOFF1, XOFF2, XON1 AND XON2 REGISTERS, WRITE ONLY 54
5.20 XCHAR REGISTER, READ ONLY 55
TABLE 21: UART RESET CONDITIONS 56

ABSOLUTE MAXIMUM RATINGS 57
ELECTRICAL CHARACTERISTICS 57
DC ELECTRICAL CHARACTERISTICS 57
TA=-40o to +85oC (industrial grade) Supply Voltage, VCC = 3.0 - 3.6V 57
AC ELECTRICAL CHARACTERISTICS 58
TA=-40o to+85oC (industrial grade) VCC = 3.0 - 3.6V 58
FIGURE 19. PCI BUS CONFIGURATION SPACE REGISTERS READ AND WRITE OPERATION 59
FIGURE 20. DEVICE CONFIGURATION AND UART REGISTERS READ OPERATION FOR A BYTE OR DWORD 60
FIGURE 21. DEVICE CONFIGURATION REGISTERS, UART REGISTERS AND TRANSMIT DATA BURST WRITE OPERATION 61
FIGURE 22. DEVICE CONFIGURATION REGISTERS, UART REGISTERS AND RECEIVE DATA BURST READ OPERATION 62
FIGURE 23. 3.3V PCI BUS CLOCK (DC TO 66MHZ) 63
FIGURE 24. TRANSMIT DATA INTERRUPT AT TRIGGER LEVEL 64
FIGURE 25. RECEIVE DATA READY INTERRUPT AT TRIGGER LEVEL 64
PACKAGE DIMENSIONS 65
REVISION HISTORY 66
TABLE OF CONTENTS I