



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



GENERAL DESCRIPTION

The XR17V258¹ (V258) is a single chip 8-channel 66MHz PCI (Peripheral Component Interconnect) UART (Universal Asynchronous Receiver and Transmitter) solution, optimized for higher performance and lower power. The V258 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 of products in a 144-pin LQFP package.

The V258 consists of eight 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 all 8-channels to speed up interrupt parsing. The V258 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 compliance
- PCI power management rev. 1.1 compliance
- EEPROM interface for PCI configuration
- 3.3V supply with 5V tolerant non-PCI (serial) inputs
- Data read/write burst operation
- Global interrupt register for all eight 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
- Eight 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
 - Programmable data rate with prescaler

FIGURE 1. BLOCK DIAGRAM OF THE XR17V258

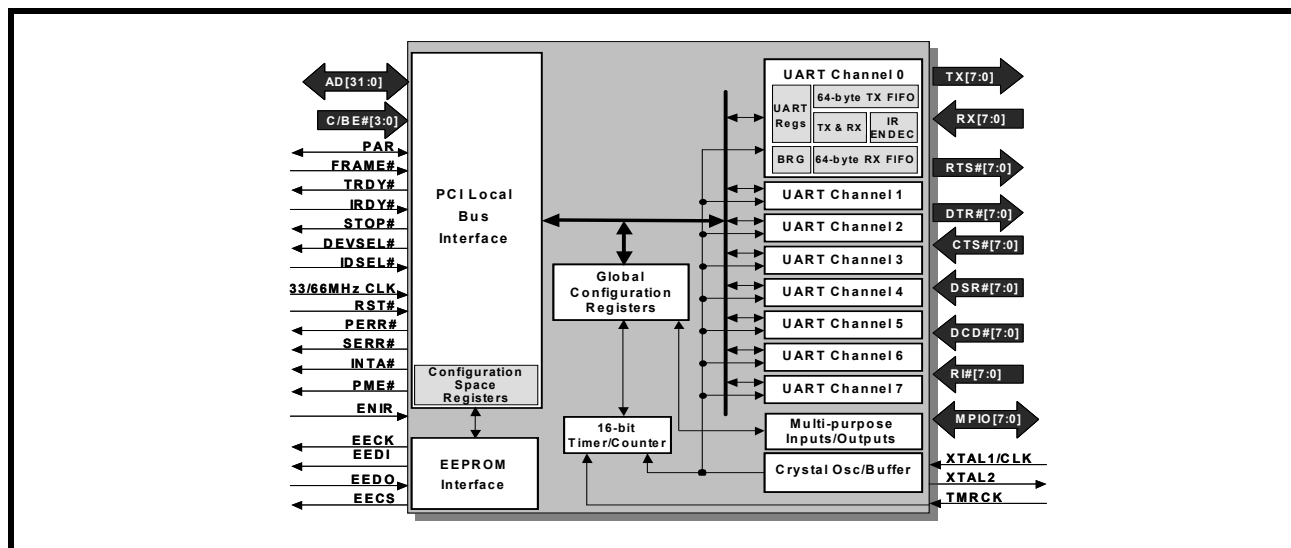
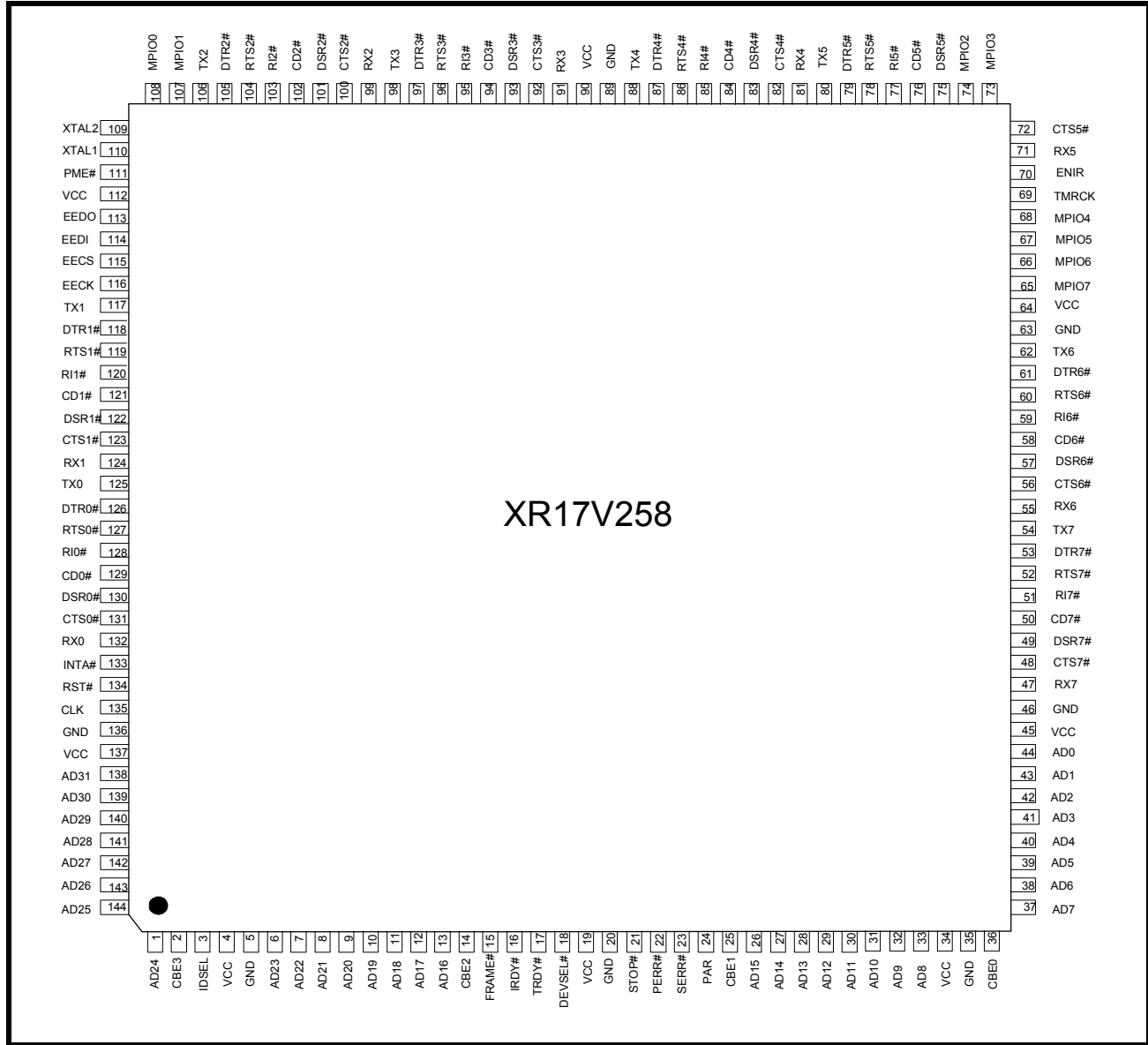


FIGURE 2. PIN OUT OF THE DEVICE



ORDERING INFORMATION

PART NUMBER	PACKAGE	OPERATING TEMPERATURE RANGE	DEVICE STATUS
XR17V258IV	144-Lead LQFP	-40°C to +85°C	Active



PIN DESCRIPTIONS

NAME	PIN #	TYPE	DESCRIPTION
PCI LOCAL BUS INTERFACE			
RST#	134	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.
CLK	135	I	PCI bus clock input of up to 66.67MHz.
AD31-AD25, AD24, AD23-AD16, AD15-AD8, AD7-AD0	138-144, 1, 6-13, 26-33, 37-44	IO	Address data lines [31:0] (bidirectional).
FRAME#	15	I	Bus transaction cycle frame (active LOW). It indicates the beginning and duration of an access.
C/BE0#- C/BE3#	36,25,14,2	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#	16	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#	17	O	Target ready (active LOW).
STOP#	21	O	Target request to stop current transaction (active LOW).
IDSEL	3	I	Initialization device select (active HIGH).
DEVSEL#	18	O	Device select to the XR17V258 (active LOW).
INTA#	133	OD	Device interrupt from XR17V258 (open drain, active LOW).
PME#	111	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 V258 asserts the PME# upon receiving a new character or upon change of state of modem inputs on any channel.
PAR	24	IO	Parity is even across AD[31:0] and C/BE[3:0]# (bidirectional, active HIGH).
PERR#	22	O	Data parity error indicator, except for special cycle transactions (active LOW). Optional in bus target application.
SERR#	23	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	125	O	UART channel 0 Transmit Data or infrared transmit data.
RX0	132	I	UART channel 0 Receive Data or infrared receive data. Normal RXD input idles at HIGH condition. The infrared pulses can be inverted internally prior to decoding by setting FCTR bit [4].
RTS0#	127	O	UART channel 0 Request to Send or general purpose output (active LOW).
CTS0#	131	I	UART channel 0 Clear to Send or general purpose input (active LOW).

PIN DESCRIPTIONS

NAME	PIN #	TYPE	DESCRIPTION
DTR0#	126	O	UART channel 0 Data Terminal Ready or general purpose output (active LOW).
DSR0#	130	I	UART channel 0 Data Set Ready or general purpose input (active LOW).
CD0#	129	I	UART channel 0 Carrier Detect or general purpose input (active LOW).
RI0#	128	I	UART channel 0 Ring Indicator or general purpose input (active LOW).
TX1	117	O	UART channel 1 Transmit Data or infrared transmit data.
RX1	124	I	UART channel 1 Receive Data or infrared receive data. Normal RXD input idles at HIGH condition. The infrared pulses can be inverted prior to decoding by setting FCTR bit [4].
RTS1#	119	O	UART channel 1 Request to Send or general purpose output (active LOW).
CTS1#	123	I	UART channel 1 Clear to Send or general purpose input (active LOW).
DTR1#	118	O	UART channel 1 Data Terminal Ready or general purpose output (active LOW).
DSR1#	122	I	UART channel 1 Data Set Ready or general purpose input (active LOW).
CD1#	121	I	UART channel 1 Carrier Detect or general purpose input (active LOW).
RI1#	120	I	UART channel 1 Ring Indicator or general purpose input (active LOW).
TX2	106	O	UART channel 2 Transmit Data or infrared transmit data.
RX2	99	I	UART channel 2 Receive Data or infrared receive data. Normal RXD input idles at HIGH condition. The infrared pulses can be inverted prior to decoding by setting FCTR bit [4].
RTS2#	104	O	UART channel 2 Request to Send or general purpose output (active LOW).
CTS2#	100	I	UART channel 2 Clear to Send or general purpose input (active LOW).
DTR2#	105	O	UART channel 2 Data Terminal Ready or general purpose output (active LOW).
DSR2#	101	I	UART channel 2 Data Set Ready or general purpose input (active LOW).
CD2#	102	I	UART channel 2 Carrier Detect or general purpose input (active LOW).
RI2#	103	I	UART channel 2 Ring Indicator or general purpose input (active LOW).
TX3	98	O	UART channel 3 Transmit Data or infrared transmit data.
RX3	91	I	UART channel 3 Receive Data or infrared receive data. Normal RXD input idles at HIGH condition. The infrared pulses can be inverted prior to decoding by setting FCTR bit [4].
RTS3#	96	O	UART channel 3 Request to Send or general purpose output (active LOW).
CTS3#	92	I	UART channel 3 Clear to Send or general purpose input (active LOW).
DTR3#	97	O	UART channel 3 Data Terminal Ready or general purpose output (active LOW).
DSR3#	93	I	UART channel 3 Data Set Ready or general purpose input (active LOW).
CD3#	94	I	UART channel 3 Carrier Detect or general purpose input (active LOW).

**PIN DESCRIPTIONS**

NAME	PIN #	TYPE	DESCRIPTION
RI3#	95	I	UART channel 3 Ring Indicator or general purpose input (active LOW).
TX4	88	O	UART channel 4 Transmit Data or infrared transmit data.
RX4	81	I	UART channel 4 Receive Data or infrared receive data. Normal RXD input idles at HIGH condition. The infrared pulses can be inverted prior to decoding by setting FCTR bit [4].
RTS4#	86	O	UART channel 4 Request to Send or general purpose output (active LOW).
CTS4#	82	I	UART channel 4 Clear to Send or general purpose input (active LOW).
DTR4#	87	O	UART channel 4 Data Terminal Ready or general purpose output (active LOW).
DSR4#	83	I	UART channel 4 Data Set Ready or general purpose input (active LOW).
CD4#	84	I	UART channel 4 Carrier Detect or general purpose input (active LOW).
RI4#	85	I	UART channel 4 Ring Indicator or general purpose input (active LOW).
TX5	80	O	UART channel 5 Transmit Data or infrared transmit data.
RX5	71	I	UART channel 5 Receive Data or infrared receive data. Normal RXD input idles at HIGH condition. The infrared pulses can be inverted prior to decoding by setting FCTR bit [4].
RTS5#	78	O	UART channel 5 Request to Send or general purpose output (active LOW).
CTS5#	72	I	UART channel 5 Clear to Send or general purpose input (active LOW).
DTR5#	79	O	UART channel 5 Data Terminal Ready or general purpose output (active LOW).
DSR5#	75	I	UART channel 5 Data Set Ready or general purpose input (active LOW).
CD5#	76	I	UART channel 5 Carrier Detect or general purpose input (active LOW).
RI5#	77	I	UART channel 5 Ring Indicator or general purpose input (active LOW).
TX6	62	O	UART channel 6 Transmit Data or infrared transmit data.
RX6	55	I	UART channel 6 Receive Data or infrared receive data. Normal RXD input idles at HIGH condition. The infrared pulses can be inverted prior to decoding by setting FCTR bit [4].
RTS6#	60	O	UART channel 6 Request to Send or general purpose output (active LOW).
CTS6#	56	I	UART channel 6 Clear to Send or general purpose input (active LOW).
DTR6#	61	O	UART channel 6 Data Terminal Ready or general purpose output (active LOW).
DSR6#	57	I	UART channel 6 Data Set Ready or general purpose input (active LOW).
CD6#	58	I	UART channel 6 Carrier Detect or general purpose input (active LOW).
RI6#	59	I	UART channel 6 Ring Indicator or general purpose input (active LOW).
TX7	54	O	UART channel 7 Transmit Data or infrared transmit data.

PIN DESCRIPTIONS

NAME	PIN #	TYPE	DESCRIPTION
RX7	47	I	UART channel 7 Receive Data or infrared receive data. Normal RXD input idles at HIGH condition. The infrared pulses can be inverted prior to decoding by setting FCTR bit [4].
RTS7#	52	O	UART channel 7 Request to Send or general purpose output (active LOW).
CTS7#	48	I	UART channel 7 Clear to Send or general purpose input (active LOW).
DTR7#	53	O	UART channel 7 Data Terminal Ready or general purpose output (active LOW).
DSR7#	49	I	UART channel 7 Data Set Ready or general purpose input (active LOW).
CD7#	50	I	UART channel 7 Carrier Detect or general purpose input (active LOW).
RI7#	51	I	UART channel 7 Ring Indicator or general purpose input (active LOW).
ANCILLARY SIGNALS			
MPIO0	108	I/O	Multi-purpose input/output 0. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT
MPIO1	107	I/O	Multi-purpose input/output 1. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO2	74	I/O	Multi-purpose input/output 2. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO3	73	I/O	Multi-purpose input/output 3. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO4	68	I/O	Multi-purpose input/output 4. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO5	67	I/O	Multi-purpose input/output 5. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO6	66	I/O	Multi-purpose input/output 6. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
MPIO7	65	I/O	Multi-purpose input/output 7. The function of this pin is defined thru the Configuration Register MPIOSEL, MPIOLVL, MPIOINV, MPIO3T and MPIOINT.
EECK	116	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	115	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 resistor for external sensing of EEPROM during power up. See DAN112 for further details.
EEDI	114	O	Write data to EEPROM device. It is manually accessible thru the Configuration Register REGB.
EEDO	113	I	Read data from EEPROM device. It is manually accessible thru the Configuration Register REGB.
XTAL1	110	I	Crystal or external clock input.
XTAL2	109	O	Crystal or buffered clock output.

**PIN DESCRIPTIONS**

NAME	PIN #	TYPE	DESCRIPTION
TMRCK	69	I	16-bit timer/counter external clock input.
ENIR	70	I	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 all 8 UARTs in the infrared mode. The sampled logic state is transferred to MCR bit [6] in the UART.
VCC	64, 90, 112, 4, 19, 34, 45, 137		Power supply for the UART core logic and PCI bus I/O - 3.3V only. The V258 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	5, 20, 35, 46, 63, 89, 136		Power supply common, ground.

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

FUNCTIONAL DESCRIPTION

The XR17V258 (V258) consists of eight 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 V258 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 8 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 8 Mbps at 8X sampling clock. The XR17V258 is available in a 144-pin LQFP (20x20x1.4mm) 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 V258 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 9](#).

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 V258. The Power Management Capabilities (PMC) register and the Power Management Control/Status Register (PMCSR) are implemented. ["Section 1.2, Power Management Registers" on page 11](#) describes these registers and details how Power Management is implemented in the device.

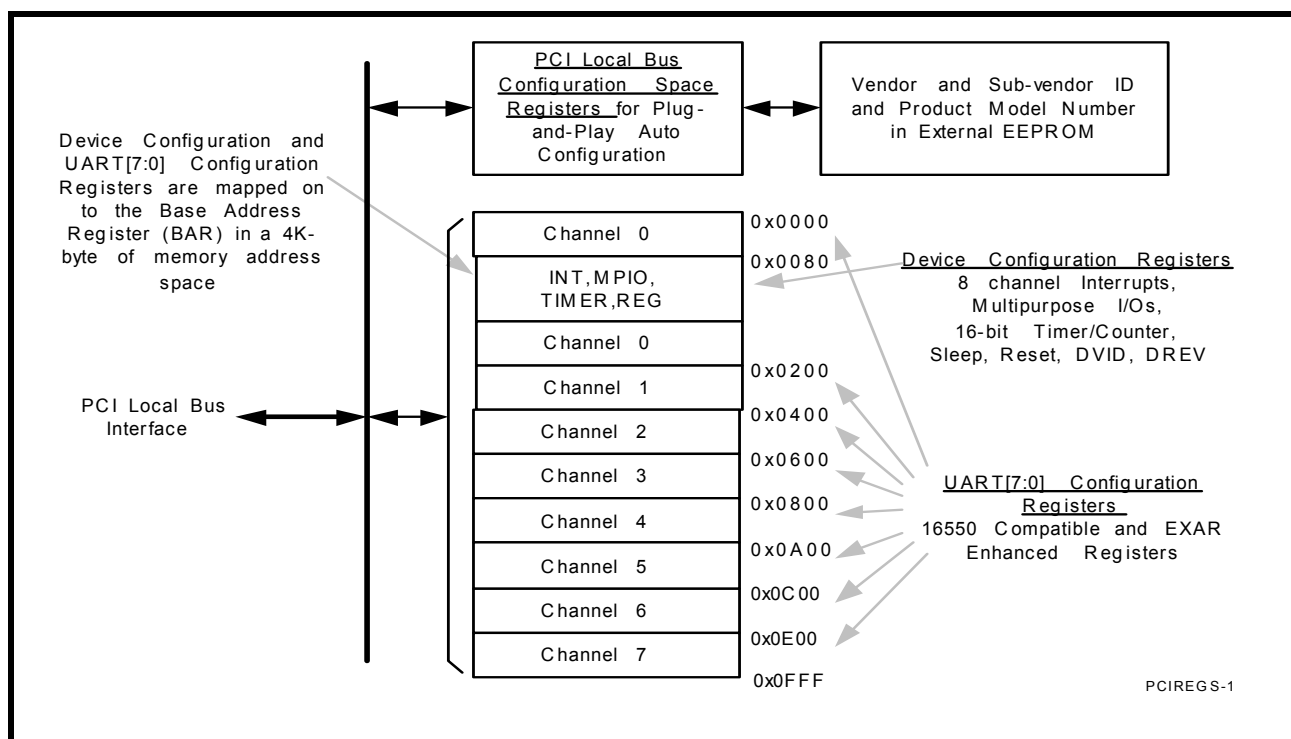
EEPROM INTERFACE

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 14](#). 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 XR17V258 INTERNAL REGISTERS

The XR17V258 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 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 all eight channels. These functions include all 8 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 8 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 4K of the PCI bus memory address space. The following paragraphs describe all 3 sets of registers in detail.

FIGURE 3. THE XR17V258 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 14) 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)	0x0258
	15:0	EWR	Vendor ID (Exar) specified by PCISIG	0x13A8
0x04	31	RWC	Parity error detected. Cleared by writing a logic 1.	0b
	30	RWC	System error detected. Cleared by writing a logic 1.	0b
	29:28	RO	Unused	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	31:12	RWR	Memory Base Address Register (BAR)	0x00
	11:0	RO	Claims a 4K 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 V258 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
	14:9	RO	Reserved	00000b

TABLE 2: POWER MANAGEMENT REGISTERS

ADDRESS OFFSET	BITS	TYPE	DESCRIPTION	RESET VALUE (HEX OR BINARY)
	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 V258

The XR17V258 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 V258.

D0 STATE

The XR17V258 must be placed in the **D0** state before being used in a system. The **D0** state represents two states - **D0** Uninitialized and **D0** Active. Upon entering **D0** from power up or transition from **D3_{hot}**, the V258 will be in the **D0** Uninitialized state. Once initialized by the system software, the V258 will enter the **D0** Active state. In the **D0** Active state, the V258 is fully functional and will respond to all PCI bus transactions as well as issue interrupts (INTA#). The system software can program the V258 to enter the **D3_{hot}** state from the **D0** state.

D3_{HOT} STATE

The V258 enters the **D3_{hot}** state when the system software programs the V258 from **D0** to **D3_{hot}**, via PMCSR bits [1:0]. In this state, the V258 will not be fully functional. The V258 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 V258 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 V258 asserts the PME# (Power Management Event) signal, if enabled by setting PME_Enable bit (bit-8 of PMCSR), 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 8 channels (see [page 52](#))

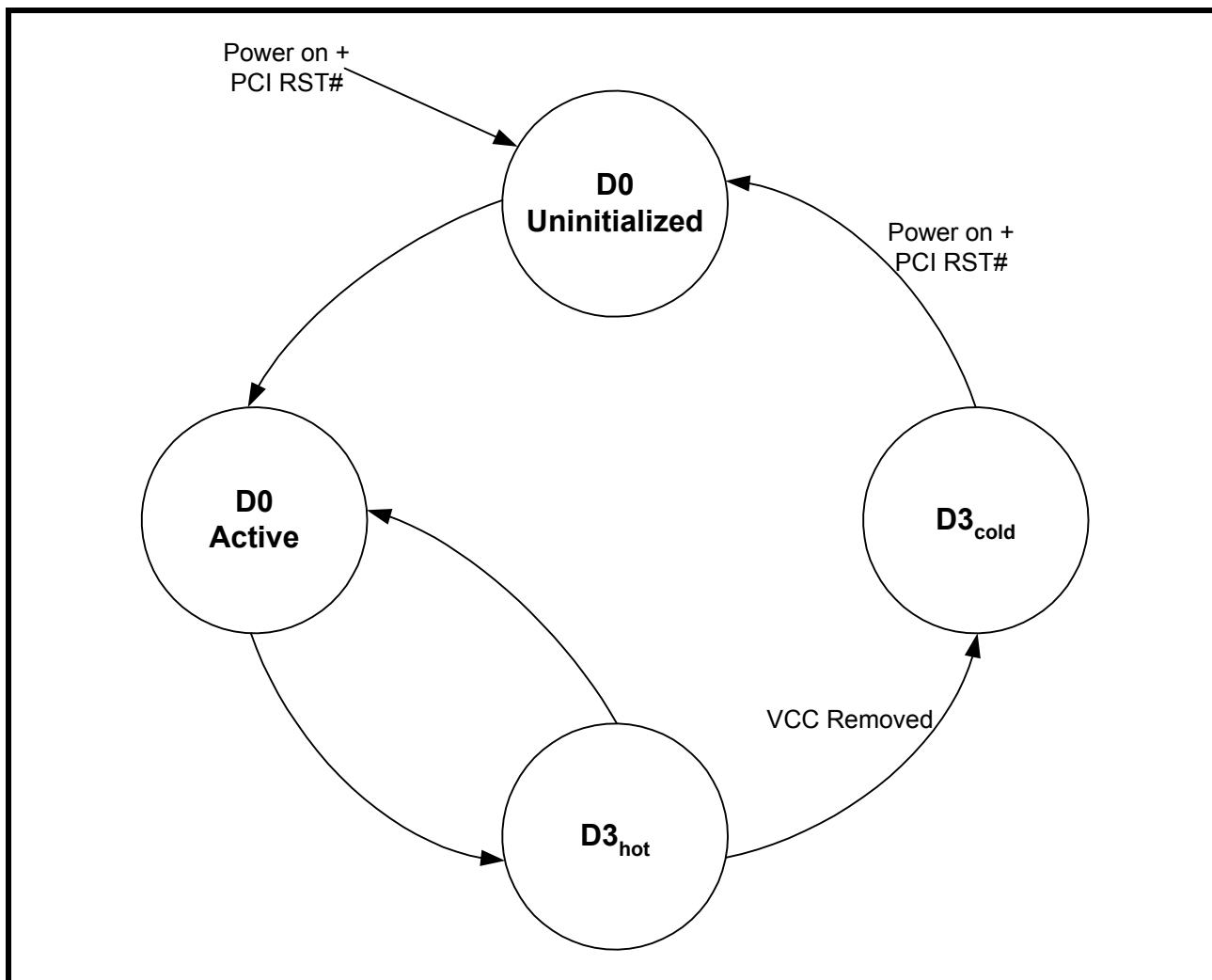
The V258 also sets the PME_Status bit when such an event occurs, regardless of whether the PME_Enable bit is set or not. The system software can reset the PME_Status bit by writing a '1' to it. When the system software programs the V258 from **D3_{hot}** to **D0**, typically in response to the PME# signal, the V258 enters the **D0** Active state and will retain all the values of its internal registers. The V258 will keep its PCI signal drivers disabled for the duration of the **D3_{hot}** to **D0** Uninitialized state transition. The V258 saves the PME context (configuration registers and functional state information) in the **D3_{hot}** state.

Note: The V258 has a sleep mode which keeps the power consumption to a minimum (see Sleep Mode description on [page 24](#)). This is independent of the power state the V258 is in. The user can optionally place the V258 in sleep mode (via the software driver) in the Active **D0** state anytime or specifically when the system software commands the V258 to enter the **D3_{hot}** state. The crystal oscillator shuts down when the conditions given in Sleep Mode section on [page 24](#) 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 V258 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 V258 enters the state when power is removed from the device. All context is lost in this state and the V258 does not support PME# in this state. When power is restored, PCI RST# must be asserted and the V258 will return to the **D0** Uninitialized state with a full PCI 3.0 compliant power-on reset sequence. The V258 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 V258 to place it in the **D0** Active state.

FIGURE 4. POWER STATE TRANSITIONS OF THE XR17V258



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 V258 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 V258's PCI Configuration Space. When the PCI RST# is negated, the V258 will download the data from the EEPROM, if it detects a HIGH on the EECS pin. The V258 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]	V258's PCI CONFIGURATION SPACE ADDRESS (WORD OFFSET)	DEFAULT VALUES
0x00	Vendor ID	0x00	0x13A8
0x01	Device ID	0x02	0x0258
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 eight individual **UART Configuration Registers** of the V258 occupy 4K 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 8 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 the 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 28.

TABLE 5: XR17V258 UART AND DEVICE CONFIGURATION REGISTERS

OFFSET ADDRESS	MEMORY SPACE	READ/WRITE	DATA WIDTH	COMMENT
0x000 - 0x00F	UART channel 0 Regs	(Table 14 & Table 15)	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 14 & Table 15)	8/16/24/32	First 8 regs are 16550 compatible
0x210 - 0x2FF	Reserved			
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
0x400 - 0x40F	UART channel 2 Regs	(Table 14 & Table 15)	8/16/24/32	First 8 regs are 16550 compatible
0x410 - 0x4FF	Reserved			
0x500	UART 2 – Read FIFO	Read-Only	8/16/24/32	64 bytes of RX FIFO data
0x500	UART 2 – Write FIFO	Write-Only	8/16/24/32	64 bytes of TX FIFO data
0x540 - 0x57F	Reserved			
0x580 - 0x5FF	UART 2 – Read FIFO with errors	Read-Only	16/32	64 bytes of RX FIFO data + LSR
0x600 - 0x60F	UART channel 3 Regs	(Table 14 & Table 15)	8/16/24/32	First 8 regs are 16550 compatible
0x610 - 0x6FF	Reserved			
0x700	UART 3 – Read FIFO	Read-Only	8/16/24/32	64 bytes of RX FIFO data
0x700	UART 3 – Write FIFO	Write-Only	8/16/24/32	64 bytes of TX FIFO data
0x740 - 0x77F	Reserved			
0x780 - 0x7FF	UART 3 – Read FIFO with errors	Read-Only	16/32	64 bytes of RX FIFO data + LSR

TABLE 5: XR17V258 UART AND DEVICE CONFIGURATION REGISTERS

OFFSET ADDRESS	MEMORY SPACE	READ/WRITE	DATA WIDTH	COMMENT
0x800 - 0x80F	UART channel 4 Regs	(Table 14 & Table 15)	8/16/24/32	First 8 regs are 16550 compatible
0x810 - 0x8FF	Reserved			
0x900	UART 4 – Read FIFO	Read-Only	8/16/24/32	64 bytes of RX FIFO data
0x900	UART 4 – Write FIFO	Write-Only	8/16/24/32	64 bytes of TX FIFO data
0x940 - 0x97F	Reserved			
0x980 - 0x9FF	UART 4 – Read FIFO with errors	Read-Only	16/32	64 bytes of RX FIFO data + LSR
0xA00 - 0xA0F	UART channel 5 Regs	(Table 14 & Table 15)	8/16/24/32	First 8 regs are 16550 compatible
0xA10 - 0xAFF	Reserved			
0xB00	UART 5 – Read FIFO	Read-Only	8/16/24/32	64 bytes of RX FIFO data
0xB00	UART 5 – Write FIFO	Write-Only	8/16/24/32	64 bytes of TX FIFO data
0xB40 - 0xB7F	Reserved			
0xB80 - 0xBFF	UART 5 – Read FIFO with errors	Read-Only	16/32	64 bytes of RX FIFO data + LSR
0xC00 - 0xC0F	UART channel 6 Regs	(Table 14 & Table 15)	8/16/24/32	First 8 regs are 16550 compatible
0xC10 - 0xCFF	Reserved			
0xD00	UART 6 – Read FIFO	Read-Only	8/16/24/32	64 bytes of RX FIFO data
0xD00	UART 6 – Write FIFO	Write-Only	8/16/24/32	64 bytes of TX FIFO data
0xD40 - 0xD7F	Reserved			
0xD80 - 0xDFF	UART 6 – Read FIFO with errors	Read-Only	16/32	64 bytes of RX FIFO data + LSR
0xE00 - 0xE0F	UART channel 7 Regs	(Table 14 & Table 15)	8/16/24/32	First 8 regs are 16550 compatible
0xE10 - 0xEFF	Reserved			
0xF00	UART 7 – Read FIFO	Read-Only	8/16/24/32	64 bytes of RX FIFO data
0xF00	UART 7 – Write FIFO	Write-Only	8/16/24/32	64 bytes of TX FIFO data
0xF40 - 0xF7F	Reserved			
0xF80 - 0xFFFF	UART 7 – 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 V258 and for monitoring the status of various functions. These registers control or report on all 8 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
0x080	INT0 [7:0]	Read-only Interrupt [7:0]	Bits [7:0] = 0x00
0x081	INT1 [15:8]	Read-only	Bits [7:0] = 0x00
0x082	INT2 [23:16]	Read-only	Bits [7:0] = 0x00
0x083	INT3 [31:24]	Read-only	Bits [7:0] = 0x00
0x084	TIMERCNTL	Read/Write Timer Control	Bits [7:0] = 0x00
0x085	TIMER	Reserved	Bits [7:0] = 0x00
0x086	TIMERLSB	Read/Write Timer LSB	Bits [7:0]= 0x00
0x087	TIMERMSB	Read/Write Timer MSB	Bits [7:0]= 0x00
0x088	8XMODE	Read/Write	Bits [7:0] = 0x00
0x089	REGA	Reserved	Bits [7:0] = 0x00
0x08A	RESET	Write-only Self clear bits after executing Reset	Bits [7:0] = 0x00
0x08B	SLEEP	Read/Write Sleep mode	Bits [7:0]= 0x00
0x08C	DREV	Read-only Device revision	Bits [7:0] = Current Rev.
0x08D	DVID	Read-only Device identification	Bits [7:0] = 0x38
0x08E	REGB	Write-only	Bits [7:0] = 0x00
0x08F	MPIOINT	Read/Write MPIO interrupt mask	Bits [7:0] = 0x00
0x090	MPIOLVL	Read/Write MPIO level control	Bits [7:0] = 0x00
0x091	MPIO3T	Read/Write MPIO output control	Bits [7:0] = 0x00
0x092	MPIOINV	Read/Write MPIO input polarity select	Bits [7:0] = 0x00
0x093	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)	MPOINT	REGB	DVID	DREV
0x090-093	MPIO (read/write)	MPIOSEL	MPIOINV	MPIO3T	MPIOLVL

1.6.1 The Global Interrupt Register

The XR17V258 has a 32-bit wide register [INT0, INT1, INT2 and INT3] to provide interrupt information and supports two interrupt schemes. The first scheme is an 8-bit indicator representing all 8 channels with each bit representing each channel from 0 to 7. This permits the interrupt service routine to quickly determine which UART channels need servicing so that it can go to the appropriate UART channel interrupt service routines. INT0 bit [0] represents the interrupt status for UART channel 0 when its transmitter, receiver, line status, or modem port status requires service. Other bits in the INT0 register provide indication for the other channels with bit [7] representing UART channel 7 respectively.

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. This 3-bit code represents 7 interrupts corresponding to individual UART's transmitter, receiver, line status, modem port status. INT1, INT2 and INT3 registers provide the 24-bit interrupt status for all 8 channels. bits [10:8] representing channel 0 and bits [31:29] representing channel 7 respectively. All 8 channel interrupts status are available with a single DWORD read operation. This feature allows the host another method to quickly service the interrupts, thus reducing the service interval and host bandwidth requirement.

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

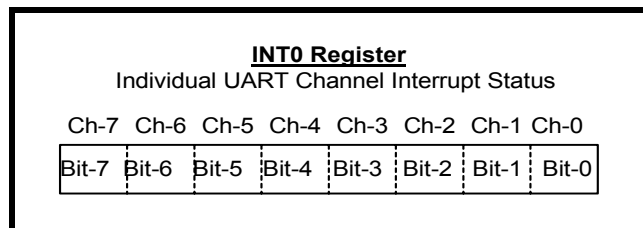


All bits start up zero. A special interrupt condition is generated by the V258 upon awakening from sleep after all eight channels were put to sleep mode earlier. This wake-up interrupt is cleared by a read to the INT0 register. **Figure 5** shows the 4-byte interrupt register and its make up.

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 [7] indicates channel 7. Logic 1 indicates the channel N [7:0] has called for service. The interrupt bit clears after reading the appropriate register of the interrupting channel register, see Interrupt Clearing section.

The INT0 register provides individual status for each channel



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. bits [10:8] represent channel 0 and go up to channel 7 with bits [31:29]. 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 space and not in other channel interrupt.

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

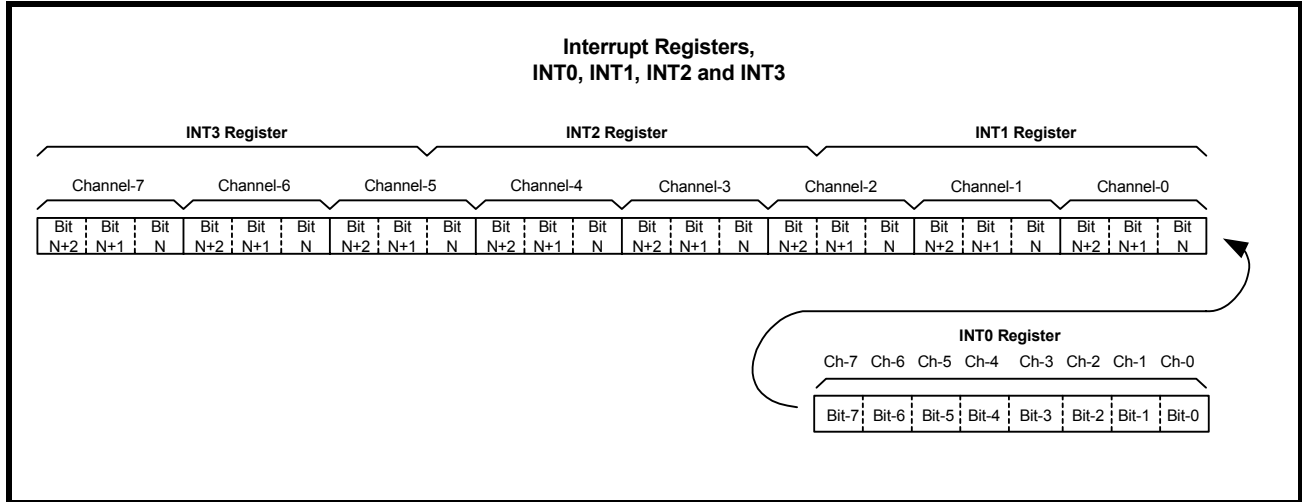


TABLE 8: UART CHANNEL [7:0] INTERRUPT SOURCE ENCODING

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 within channel 0, reserved in other channels.
7	1	1	1	TIMER Time-out. Available only within channel 0, reserved in other channels.

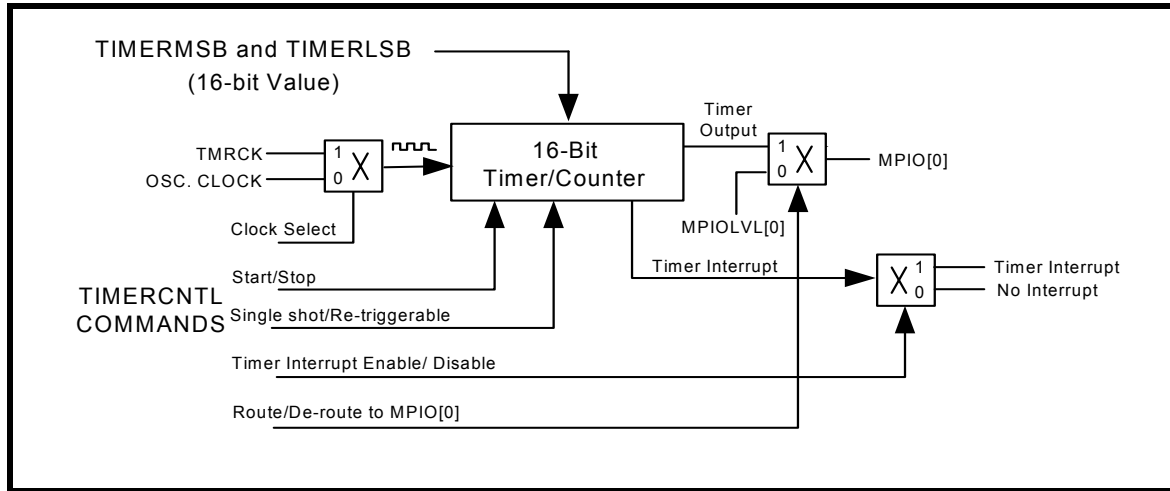
TABLE 9: UART CHANNEL [7:0] INTERRUPT CLEARING

Wake-up Indicator is cleared by reading the INTO register.
RXRDY and RXRDY Time-out is cleared by reading data in the RX FIFO until the RX FIFO is empty.
RX Line Status interrupt clears after reading the LSR register that is in the UART channel register set.
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 delta and special character detect interrupt clears after reading the ISR register that is in the UART channel register set.
TIMER Time-out interrupt clears after reading the TIMERCNTL register that is in the Device Configuration register set.
MPIO interrupt clears after reading the MPIOVLV register that is in the Device Configuration register set.

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

The XR17V258 has a general purpose 16-bit timer/counter. The crystal/clock at the XTAL1 input or an external clock at the TMRCK input pin can be selected as the clock source for the timer/counter. 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, TIMERSLB, 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

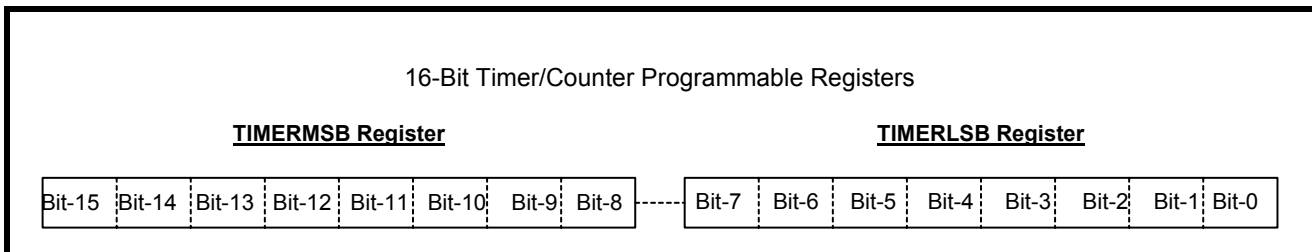


TIMERMSB [31:24] and TIMERSLB [23:16]

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

$$\text{Timer output frequency} = \text{Timer input clock} / 16\text{-bit Timer value}$$

The least-significant bit of the timer is being bit [0] of the TIMERSLB 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 selected as clock source
- Timer output not routed to MPIO[0]
- Timer stopped

TABLE 10: TIMER CONTROL REGISTERS

TIMERCNTL [7:4]	Reserved
TIMERCNTL [3:0]	<p>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.</p> <p>0001: Enable Timer Interrupt</p> <p>0010: Disable Timer Interrupt</p> <p>0011: Select One-shot mode</p> <p>0100: Select Re-triggerable mode</p> <p>0101: Select Internal Crystal Oscillator output as clock input for the Timer</p> <p>0110: Select External Clock input through the TMRCK pin for the Timer</p> <p>0111: Route Timer output to MPIO[0] pin</p> <p>1000: De-route Timer output from MPIO[0]</p> <p>1001: Start Timer</p> <p>1010: Stop Timer</p> <p>1011: Reset Timer</p>

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 $N (= P \text{ clocks})$ 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:

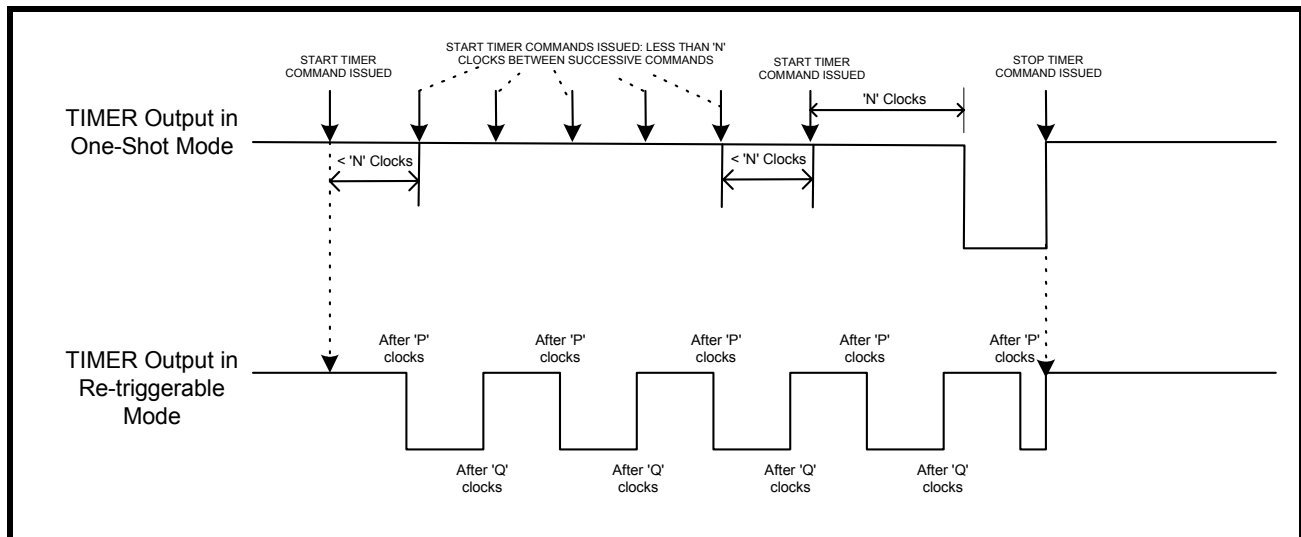
- Any write to TIMER MSB, LSB registers
- Issue of any command other than 'Stop Timer' and 'Reset Timer' ('Start Timer' is not allowed)

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 (MPIOVLV, MPIOSEL etc) lose control over MPIO[0] and get the control back only when the Timer output is de-routed from MPIO[0].

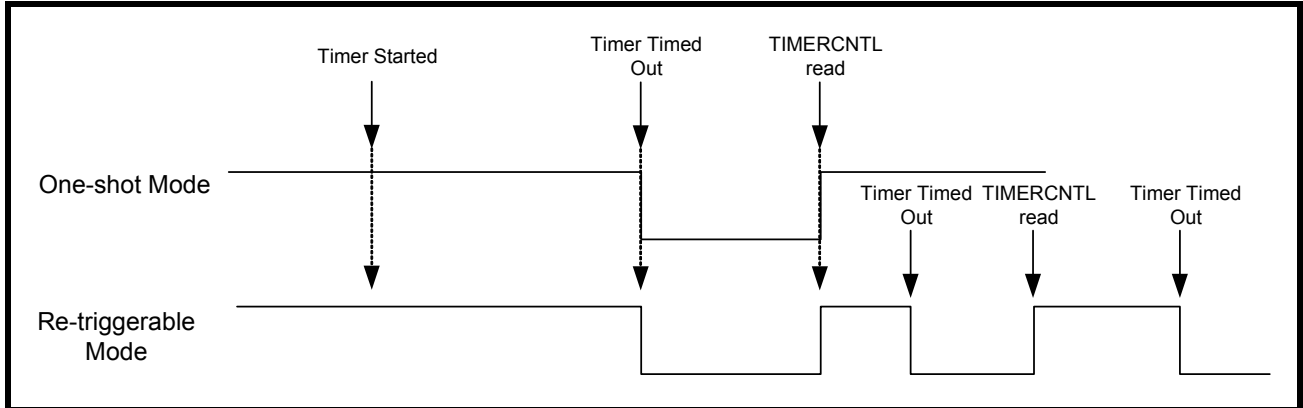
FIGURE 7. TIMER OUTPUT IN ONE-SHOT AND RE-TRIGGERABLE MODES



Timer Interrupt

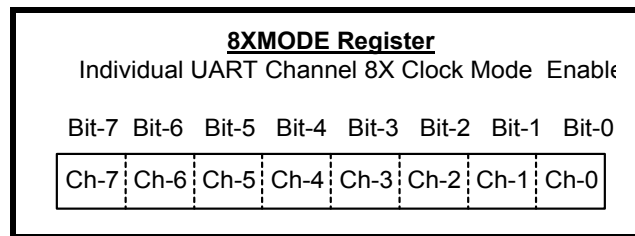
In the one-shot mode, the Timer will issue an interrupt upon timing out which is 'N' clocks after the Timer is started. In the re-triggerable mode, the Timer will keep issuing an interrupt every 'N' clocks which is on every rising edge of the Timer output. The Timer interrupt can be cleared by reading 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)

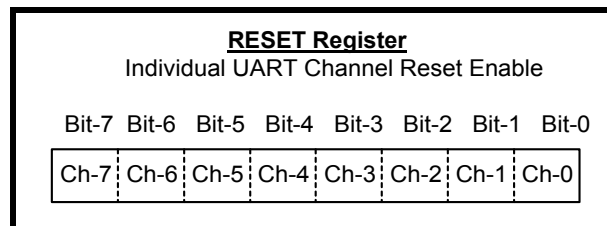
Each bit selects 8X or 16X sampling rate for that UART channel, bit [0] is channel 0. Logic 0 (default) selects normal 16X sampling with logic one selects 8X sampling rate. Transmit and receive data rates will double by selecting 8X.



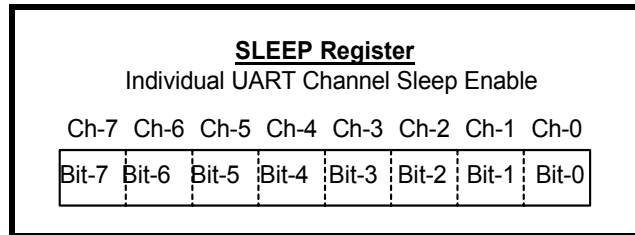
1.6.4 REGA [15:8] (default 0x00) Reserved

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

The 8-bit Reset register [RESET] provides the software with the ability to reset the UART(s) when there is a need. Each bit is self-clearing 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 22](#) for details. .



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



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 (Logic 0 (default) is to disable and logic 1 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] = 0x0)

When all 8 UART channels are put to sleep, the on-chip oscillator shuts off to further conserve power. In this case, the V258 is awakened by any of the following events occurring at any of the 8 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 V258 is ready after 32 crystal clocks to ensure full functionality. Therefore, if the V258 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 V258 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 0x48 from this register indicates the device is a XR17V258. 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 XR17V258 is 0x48.

DREV [7:0]

Revision number of the XR17V258. 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 all 8 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)	Logic 0 (default) write to each UART configuration registers individually.
	Logic 1 enables simultaneous write to all 8 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 V258 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, MPIOVLV, MPIO3T, MPIOINV and MPIOSEL. If all 8 pins are set for inputs, all 8 interrupts would be ORed together. The ORed 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 20.

1.6.10 MPIO REGISTERS

There are 5 registers that select, control and monitor the 8 multipurpose inputs and outputs. **Figure 9** shows the internal circuitry.