OX16PCI958 DATA SHEET

Octal UART with PCI Interface

FEATURES

- Efficient 32-bit, 33 1/3 MHz multi-function, target-only PCI controller, compliant to PCI Local Bus Specification 3.0 & PCI Power Management Specification 1.1
- Eight UARTs fully software compatible with 16C550-type devices
- Compatible with existing 16C550/450 device drivers
- PCI 2.1, 2.2, 2.3 & 3.0 compliant
- Supports both 5.0-V & 3.3-V PCI signalling
- 32-byte deep FIFO per transmitter & receiver
- Baud rates up to 4.125 Mega-baud (using a 16.5 MHz input clock).
- Clock can be provided from crystal oscillator or external clock source
- Automated out-of-band flow control using CTS#/RTS#
- Configuration data is held in a small, low-cost serial Microwire™ compatible EEPROM
- Driver-facilitated DSR/DTR & Xon/Xoff handshaking
- 5-,6-,7- & 8-bit data framing
- 1, 1.5 or 2 stop bits
- UART enhancements:
  - Clock prescaler allows more baud rate options
  - Readable FIFO levels & tuneable trigger levels improve device driver performance
  - Programmable "synchronization factor" allows baud rates up to fclock/4
  - Extensions to standard register set are implemented in a safe, easy-to-use way
- Low-power design with separate power management control
- Operating temp. range : 0°C—70°C
- 160-pin QFP package
- Operation via IO or memory mapping
- Support for multiple wake-up events

DESCRIPTION

The OX16PCI958 contains eight UARTs (Universal Asynchronous Receiver-Transmitters) and a host interface suitable for direct connection to a PCI bus. Once installed and configured by the host OS, it provides an eight-byte programming interface to each UART. The UARTs are fully software-compatible with 16C550 devices. The device can be configured to fit the requirements of RS232 or RS422 applications.

The UARTs convert between RS232-format serial data on separate transmit and receive lines, and byte-wide I/O writes and reads on the host interface. Malformed incoming serial data is flagged along with the data in the receive FIFO. The state of the UART can be found at any time by reading status registers, and modem control (handshaking output) lines can be individually controlled.

Although polled-mode operation is possible, the UART will usually be operated on a host-interrupt basis. The interrupt system is designed to allow efficient handling of interrupt service requests from the UART, for example by using the prioritised interrupt identification register, readable FIFO levels, and tuneable FIFO trigger levels.

The internal transmitter and receiver logic runs at a programmable synchronisation factor of 4x, 8x, or 16x the serial baud rate. This internal clock is generated by dividing a reference clock by an integer divisor from 1 to (2^{16}–1). In this way the UART can accommodate a serial rate of up to 4 125 000 baud (using a 16.5 MHz input clock).

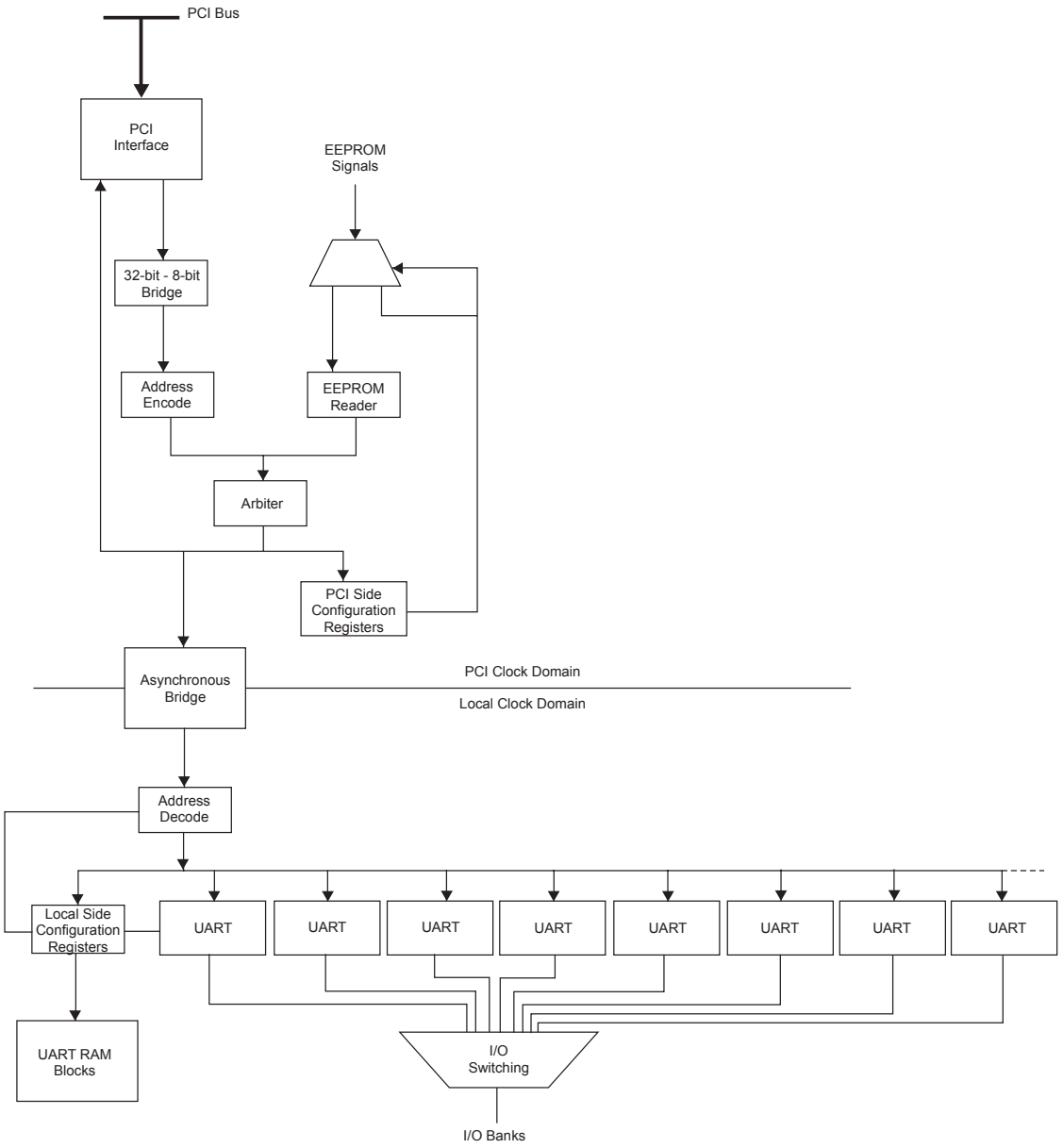
The OX16PCI958 provides a host interface that can be directly connected to a PCI bus. It responds to configuration accesses, and once configured it also responds to I/O and memory accesses for control of the UART. The data for configuration space is read from a small external serial EEPROM at start-up, together with information on how the OX16PCI958 should be set up.

Oxford Semiconductor Ltd.
25 Milton Park, Abingdon, Oxon, OX14 4SH, UK
Tel: +44 (0)1235 824900

OX16PCI958 DS-0022—Nov 2005
Part No. OX16PCI958—PQAG

CONTENTS

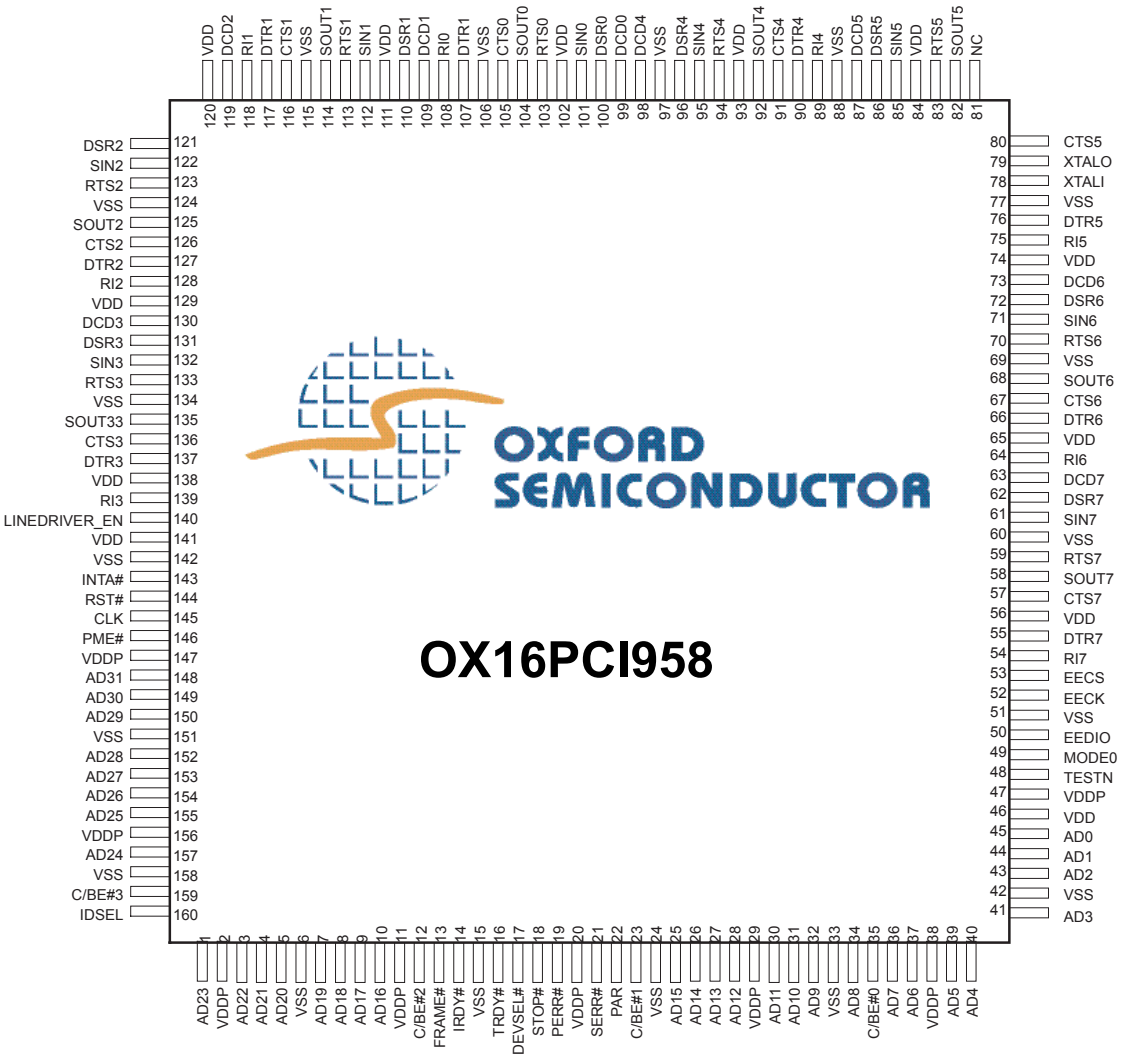
1. BLOCK DIAGRAM



Note: The connections between the UART RAM blocks and each of the UARTs are omitted for clarity.

2. PIN INFORMATION—160-PIN QFP

2.1. Pinout



2.2. Pin Descriptions

Table 1 lists the pin allocations, names and describes them.

Table 1 Pin Descriptions

Name	Pin	Dir	Description
			PCI bus signals
PAR	22	IO	
CLK	145	I	
RST#	144	I	
FRAME#	13	I	
IRDY#	14	I	
TRDY#	16	OT	
DEVSEL#	17	OT	
STOP#	18	OT	
PERR#	19	OT	
SERR#	21	OT	
INTA#	143	OT	
PME#	146	OT	
IDSEL	160	I	
AD[31:0]	148, 149,150, 152, 153, 154, 155, 157, 1, 3, 4, 5, 7, 8, 9, 10, 25, 26, 27, 28, 30, 31, 32, 34, 36, 37, 39, 40, 41, 43, 44, 45	IO	
C/BE#3	159	I	
C/BE#2	12	I	
C/BE#1	23	I	
C/BE#0	35	I	
			Chip configuration
EECS	53	O	
EECK	52	O	
EEDIO	50	IO	
			Local clock
XTALI	78	I	
XTALO	79	O	
			Local side
LD_EN	140	O	
			Power and ground
VSS (GND)	6, 15, 24, 33, 42, 49, 51, 60, 69, 77, 88, 97, 106, 115, 124, 134, 142, 151, 158		
VDD (5V)	56, 65, 74, 84, 93, 102, 111, 120, 129, 138		
VDDP	2, 11, 20, 29, 38, 46, 47, 141, 147, 156		

The VDDP pins provide power to the PCI I/O buffers, and must be connected to the +$V_{I/O}$ pins on the PCI connector.

Table 2 &

Table 3 list pin allocations for the local I/O banks.

Table 2 Local I/O Bank 0—3 Pin Allocations

IO#	Bank	Pin	Dir	Name
0	0	99	I	DCD
1		100	I	DSR
2		101	I	SIN
3		103	O	RTS
4		104	O	SOUT
5		105	I	CTS
6		107	O	DTR
7		108	I	RI
0	1	109	I	DCD
1		110	I	DSR
2		112	I	SIN
3		113	O	RTS
4		114	O	SOUT
5		116	I	CTS
6		117	O	DTR
7		118	I	RI
0	2	119	I	DCD
1		121	I	DSR
2		122	I	SIN
3		123	O	RTS
4		125	O	SOUT
5		126	I	CTS
6		127	O	DTR
7		128	I	RI
0	3	130	I	DCD
1		131	I	DSR
2		132	I	SIN
3		133	O	RTS
4		135	O	SOUT
5		136	I	CTS
6		137	O	DTR
7		139	I	RI

Table 3 Local I/O Bank 4—7 Pin Allocations

IO#	Bank	Pin	Dir	Name
0	4	98	I	DCD
1		96	I	DSR
2		95	I	SIN
3		94	O	RTS
4		92	O	SOUT
5		91	I	CTS
6		90	O	DTR
7		89	I	RI
0	5	87	I	DCD
1		86	I	DSR
2		85	I	SIN
3		83	O	RTS
4		82	O	SOUT
5		80	I	CTS
6		76	O	DTR
7		75	I	RI
0	6	73	I	DCD
1		72	I	DSR
2		71	I	SIN
3		70	O	RTS
4		68	O	SOUT
5		67	I	CTS
6		66	O	DTR
7		64	I	RI
0	7	63	I	DCD
1		62	I	DSR
2		61	I	SIN
3		59	O	RTS
4		58	O	SOUT
5		57	I	CTS
6		55	O	DTR
7		54	I	RI

3. PCI INTERFACE

The PCI interface conforms to revisions 2.1, 2.2, 2.3 and 3.0 of the PCI Specification, and version 1.1 of the PCI Power Management Specification.

Six base address registers are implemented in the OXPCI958:

- BAR0—128-byte memory-mapped region
- BAR1—128-byte I/O-mapped region
- BAR2—64-byte I/O-mapped region
- BAR3—16-byte I/O-mapped region
- BAR4—64-byte memory-mapped region
- BAR5—16-byte memory-mapped region

All memory regions are in 32-bit address space, and are marked as non-prefetchable.

3.1. Internal Address Map

The internal address map is referenced by the EEPROM to configure the UARTs. Table 4 shows how PCI accesses are mapped to internal addresses:

Table 4 PCI Address Mapping

BAR0, 1	PCI side	PCI bridge configuration	00h to 2Fh
		EEPROM control, power management	30h to 3Fh
	local side configuration	UART, SISR	40h to 7Fh
BAR2, 4	local functions	UARTs	80h to BFh
BAR3, 5	local functions	SISR	C0h to CFh

Notes:

- Addresses in the range 40h-7Fh are aliased with a period of 32, i.e., address bit 5 is not decoded in this range
- Addresses in the range C0h-FFh are aliased with a period of 16, i.e., address bits 5:4 are not decoded in this range. For example, if BAR4 is configured as C8000400h, a memory access at C8000413h, which is BAR4+13h, would access internal address 80h+13h = 93h
- The serial EEPROM reader can access any internal address

Table 5 lists the PCI register offsets.

Table 5 Register Offsets

BAR	Internal address	Use
BAR0, 1	00h – 19h	PCI setup registers, as described in section 3.4.
	30h	EEPROM-control register
	31h	Power-management control register
	34h	UART interrupt status
	35h	
	40h	UART-enable register
	41h	UART IO bank switching/rotation
	42h	SISR enable register
	4Ch	UART configuration
	50h	Global UART clock pre-divider
BAR2,4	80h-87h	UART 0 registers 0-7
	88h-8Fh	UART 1 registers 0-7
	90h-97h	UART 2 registers 0-7
	98h-9Fh	UART 3 registers 0-7
	A0h-A7h	UART 4 registers 0-7
	A8h-AFh	UART 5 registers 0-7
	B0h-B7h	UART 6 registers 0-7
	B8h-BFh	UART 7 registers 0-7
BAR3,5	C0h	SISR, if SISR enabled
	C0h-C1h	UART 0 registers 8-9, if SISR not enabled
	C2h-C3h	UART 1 registers 8-9
	C4h-C5h	UART 2 registers 8-9
	C6h-C7h	UART 3 registers 8-9
	C8h-C9h	UART 4 registers 8-9
	CAh-CBh	UART 5 registers 8-9
	CCh-CDh	UART 6 registers 8-9
	CEh-CFh	UART 7 registers 8-9
	other	RFU

Notes:

- Writes to undefined internal addresses are ignored, and reads from undefined internal addresses return zero
- For shared address ranges, the SISR takes priority over the UARTs

3.2. Configuration & Control Registers

Table 6 summarizes the configuration and control registers for quick reference.

Table 6 Configuration & Control Register Summary

A	use	D7	D6	D5	D4	D3	D2	D1	D0
30h	EEPROM-control register	EET2	EET1	RFU	EEDIO data in	EECS	EECK	EEDIO output enable	EEDIO data out
31h	Power-management control register	RFU		PM_DRIVER		PM_LCLK		PM_OSC	
34h	UART interrupt status (RO)	U7INT	U6INT	U5INT	U4INT	U3INT	U2INT	U1INT	U0INT
40h	UART-enable register	U7EN	U6EN	U5EN	U4EN	U3EN	U2EN	U1EN	U0EN
42h	SISR enable	SEN	RFU						
4Ch	UART configuration	RFU		1b	RFU				
50h	Global pre-divider	RFU1	RFU1	GCS1	RFU1	RFU	GCS0	RFU1	RFU

EEPROM-Control Register

The OX16PCI958 automatically takes control of the EECS, EECK and EEDIO pins after a deassertion of the host bus RESET signal, in order to read in configuration data. Afterwards, the signals may be controlled though accesses to this register.

Field (Bits)	Description
EET2 (7)	High—at least 70 PCI clock cycles have occurred since the register was last written. Cleared when the register is written. Set to the value of EET1 every 70th PCI clock cycle. This may be useful for ensuring that EEPROM timing constraints are met
EET1 (6)	Cleared when the register is written. Set every 70th PCI clock cycle
EEDIO data in (4)	Returns the current logic level on the EEDIO pin.
EECS (3)	Controls EECS output.
EECK (2)	Controls EECK output
EEDIO output enable (1)	1—the value last written to bit 0 is driven on the EEDIO pin 0—EEDIO pin is tri-stated
EEDIO data out (0)	Controls the logic level driven onto EEDIO when bit 1 is set.

This register is set to 00h on a PCI reset.

Power-Management Control Register

Each two-bit group represents a power-management level range, as shown in Table 7, defining whether an element is disabled, which is shown in Table 8.

Table 7 Power Management Group

Field (Bits)	Control Measure
PM_DRIVER	driver_en output deasserted
PM_LCLK	Local-side clock gated off
PM_OSC	Local-side oscillator disabled

Table 8 Element Disabling

Value	Description
0 0	Never disabled
0 1	Disabled in D1, D2 & D3
1 0	Disabled in D2 and D3
1 1	Disabled in D3 only

This register is set to 00h on a PCI reset.

UART Interrupt State

Each bit in this read-only register reports the interrupt status of the corresponding internal UART.

UART Enable

Each bit in this register enables the corresponding internal UART to be accessed on the internal bus, by either the PCI interface or the EEPROM reader.

SISR Enable

Bit 7 must be set to enable access to the shared interrupt status register (SISR).

This register is set to 80h on a PCI reset.

UART Configuration

Bit 5 must be set to binary 1 to ensure correct operation of the UART

Global UART Clock Pre-Divider

This register sets a pre-division value for all the internal UARTs.

Bit 5—One of the clock pre-division factors, see Table 9

Bit 2—One of the clock pre-division factors, see Table 9

After a reset, this register is set to F6h, giving a divide-by-8 clock setting for all UARTs. For the standard 14.7456 MHz external crystal, this gives a 1.8432 MHz effective clock to the UARTs.

For backwards compatibility, write only one of the four values in Table 9 to bits 5 and 2:

Table 9 Clock Pre-Division Values

Value	Divisor
F6h	8
F2h	4
D6h	2
D2h	1

The above register settings are recommended for backwards compatibility, but Table 10 shows how the actual control logic operates.

Table 10 Clock Division Logic

GCS1	GCS0	Division
1	1	8
1	0	4
0	1	2
0	0	1

3.3. PCI Configuration Space Registers

The PCI interface presents a type 0 configuration register set in configuration space, with the standard extension for power management. Table 11 summarizes the PCI configuration space registers.

Table 11 PCI Configuration Space Registers

Address Offset	Configuration register 31 24	23 16	15 8	7 0
00h	Device ID		Vendor ID	
04h	Status		Command	
08h	Class code			Revision ID
0Ch	BIST	Header type	Latency timer	Cache line size
10h	Base address register 0 (BAR0)			
14h	Base address register 1 (BAR1)			
18h	Base address register 2 (BAR2)			
1Ch	Base address register 3 (BAR3)			
20h	Base address register 4 (BAR4)			
24h	Base address register 5 (BAR5)			
28h	Cardbus CIS pointer			
2Ch	Subsystem device ID		Subsystem vendor ID	
30h	Expansion ROM base address register			
34h	RFU			Capabilities pointer
38h	RFU			
3Ch	Max_lat	Min_gnt	Interrupt pin	Interrupt line
40h	Power management capabilities (PMC)		Next Ptr (always 0)	Cap_ID (always 0)
44h	PM_Data	PMCSR_BSE	PMC Control/Status register (PMCSR)	

Device ID Register

This register is read-only via configuration accesses, and returns the current value of the DID register (see section 3.4 for details of this and other PCI set-up registers).

Vendor ID Register

This register is read-only via configuration accesses, and returns the current value of the VID register.

Status Register

This register records information on the PCI interface state, as described in the PCI specification.

Write 1 to bits 11, 14 and 15 to clear them, all others are read-only.

Bits	Description
15	1—parity error, even if parity error handling is disabled by bit 6 in the Command register
14	Set whenever the device asserts SERR#.
13	0
12	0
11	Set whenever the device terminates a transaction with Target-Abort.
10:9	Device select timing. Target access timing of the function via the DEVSEL# output. This device is a medium speed target device (01b)
8	0
7	0
5	0
4	1
3	Reflects the interrupt state in the device/function. INTA# is only asserted when the Interrupt Disable bit in Command is 0 & this Interrupt Status bit is a 1. Setting the Interrupt Disable bit to a 1 has no effect on the state of this bit.

Command Register

This register enables certain features of the PCI interface.

Bits	Description
10	1—disables INTA# assertion 0—enables INTA# assertion After RST# is 0
9	0
8	1—enables. the function to report address parity errors via SERR#
7	0
6	1—enables the function to report parity errors via PERR#
5	0
4	0
3	0
2	0
1	Controls the response to memory space accesses. 1—allows the device respond to the PCI bus memory accesses as a target
0	Controls the response to I/O accesses. 1—allows the device respond to the PCI bus I/O accesses as a target

Class-Code Register

This register is read-only via configuration accesses, and returns the current value of the PCC register.

Revision ID Register

This register is read-only via configuration accesses, and returns the current value of the REV register.

BIST Register

This byte always returns 00h, and writes to the byte are ignored, as there is no BIST function.

Header Type

This byte always returns 00h, indicating a type 0 header and a single-function device. Writes to the byte are ignored.

Latency Timer

Read-only register always returns 00h. (Not relevant for target-only PCI devices).

Cache Line Size

This register is read-only and always returns 00h. (The device does not support cache line wrap mode)

Base Address Register 0

This register sets the PCI base address in memory space for access to local configuration registers. The register has bits 31-7 writable, and the remainder of the bits are always 0. This forms a request for 128 bytes of memory space with a 32-bit address, marked as non-prefetchable. Accesses made to the memory range defined by this BAR map to internal configuration registers at internal addresses 00h-07h.

Base Address Register 1

This register sets the PCI base address in I/O space for access to local configuration registers. The register has bits 31:7 writable; the remainder are always 01h. This forms a request for 128 bytes of I/O space. Accesses made to the I/O range defined by this BAR map to internal configuration registers at internal addresses 00h-07h.

Base Address Register 2

This base address register is for a mapping of 64 bytes in I/O space. Accesses made to the I/O range defined by this BAR map to internal UARTs at internal addresses 80h-BFh.

Base Address Register 3

This base address register is for a mapping of 16 bytes in I/O space. Accesses made to the I/O range defined by this BAR map to unused internal addresses C0h-CFh.

Base Address Register 4

This base address register is for a mapping of 64 bytes in 32-bit memory space (non-prefetchable memory). Accesses made to the memory range defined by this BAR map to internal UARTs at internal addresses 80h-BFh.

Base Address Register 5

This base address register is for a mapping of 16 bytes in 32-bit memory space (non-prefetchable memory). Accesses made to the memory range defined by this BAR map to internal addresses C0h-CFh.

Cardbus CIS Pointer

Hard-wired to zero, as this device is not for use in CardBus applications.

Subsystem Device ID Register

This register is read-only via configuration accesses, and returns the current value of the SDID register.

Subsystem Vendor ID register

This register is read-only via configuration accesses, and returns the current value of the SVID register.

Expansion ROM Base Address Register

Hard-wired to zero.

Capabilities Pointer

This register is read-only and always returns 40h, as this is where the power management registers are located.

Max_lat Register

Read-only register which always returns 00h. (Not relevant for target-only PCI devices)

Min_gnt Register

Read-only register which always returns 00h. (Not relevant for target-only PCI devices)

Interrupt Pin Register

This register is read-only via configuration accesses, but may be set to either 00h or 01h using local-register access via BAR0, BAR1 or EEPROM configuration. It is set to 01h following a PCI reset.

Interrupt Line Register

This register is read-write accessible via configuration accesses. It is set to 00h following a PCI reset.

Power-Management Registers

The Power Management Capabilities register is read-only via configuration accesses. It provides the host system with information on the power-management capabilities of the PCI device and returns the current value of the PMC register. It is set to a generic-configured value following a PCI reset.

This register is mostly just for the passing of power management information to the host system, but two of the fields also have an affect on the operation of the block:

Bits	Description
10	0—writing 10b to the PowerState bits in PMCSR (see below) leaves PowerState unchanged
9	0—writing 01b to the PowerState bits in the PMCSR (see below) leaves PowerState unchanged

The Power Management Control/Status register (PMCSR) has a mixture of read-only, read/write and read/clear-on-write bits.

Bits	Description
15	Set when the function would assert the PME# signal if bit 8 is enabled. Writing 1 clears it & causes the function to stop asserting PME#. Writing 0 has no effect. Cleared on PCI reset
14:13	Read-only scaling factor to be used when interpreting the value of the Power Management Data register. The value and meaning of this field will depend on which data value has been selected using bits 12:9
12:9	Read/write field used to select which data is to be reported through the Data register & bits 14:13. Resets to 0
8	1—enables PME# 0—disables PME# (default)
1:0	Determines the current power state of a function & sets it into a new power state using the values below. 00b—D0 01b—D1 10b—D2 11b—D3 If software attempts to write an unsupported, optional state to this field (i.e. D1 when D1_Support is not set, or D2 when D2_Support is not set), the write operation completes normally on the bus; however, the data is discarded and no state change occurs. Reset value 00b

The Power Management Bridge Support Extensions register at offset 46h is read-only, and always returns the value 00h.

The Power Management Data register at offset 47h is read-only, and returns a value which depends on which data value has been selected using PMCSR [12:9].

3.4. PCI Set-up Registers

Table 12 lists the PCI set up registers.

Table 12 Address Map

Address	Register	Register description	Default Value
00h	VID7:0	Vendor ID, lower byte	15h*
01h	VID15:8	Vendor ID, upper byte	14h*
02h	DID7:0	Device ID, lower byte	38h*
03h	DID15:8	Device ID, upper byte	95h*
04h	REV	PCI silicon revision	01h
05h	PCC7:0	PCI class code, programming interface byte	00h
06h	PCC15:8	PCI class code, subclass code byte	02h
07h	PCC23:16	PCI class code, base class code byte	07h
08h	SVID7:0	Subsystem Vendor ID, lower byte	15h*
09h	SVID15:8	Subsystem Vendor ID, upper byte	14h*
0Ah	SDID7:0	Subsystem Device ID, lower byte	00h*
0Bh	SDID15:8	Subsystem Device ID, upper byte	00h*
0Ch	PIP	Interrupt pin (bit 0 only used)	01h
0Dh		RFU	-
0Eh	PMC7:0	Power management capabilities, lower byte	02h
0Fh	PMC15:8	Power management capabilities, upper byte	00h
10h	PMD(0)	Power management data, for when data_select=0	00h
11h	PMD(1)	Power management data, for when data_select=1	00h
12h	PMD(2)	Power management data, for when data_select=2	00h
13h	PMD(3)	Power management data, for when data_select=3	00h
14h	PMD(4)	Power management data, for when data_select=4	00h
15h	PMD(5)	Power management data, for when data_select=5	00h
16h	PMD(6)	Power management data, for when data_select=6	00h
17h	PMD(7)	Power management data, for when data_select=7	00h
18h	PMDS(3:0)	Power management data_scale, for when data_select=0, 1, 2, 3	00h
19h	PMDS(7:4)	Power management data_scale, for when data_select=4, 5, 6, 7	00h

* Ensure these registers are written by the configuration EEPROM.

Most of these registers simply hold values which are presented in read-only registers in PCI configuration space (see section 3.3).

PMDS(3:0) and PMDS(7:4) contain eight 2-bits, packed as shown below, returned as a value for bits 14:13 in PMCSR when PMCSR [12:9] = n where n references PMDSn (see section 3.3).

4. UART FUNCTION

Each UART in the OX16PCI958 is identical. The depth of the FIFOs is 32 bytes.

Each UART converts between RS232-format serial data on separate transmit and receive lines, and byte-wide I/O writes and reads on the host interface. Malformed incoming serial data is flagged along with the data in the receive FIFO. The state of the UART can be found at any time by reading status registers, and modem control (handshaking output) lines can be individually controlled.

Although polled-mode operation is possible, the UARTs will usually be operated on a host-interrupt basis. The interrupt system is designed to allow efficient handling of interrupt service requests from the UART, for example by using the prioritised interrupt identification register, readable FIFO levels, and tuneable FIFO trigger levels.

The internal transmitter and receiver logic runs at a programmable synchronisation factor of 4x, 8x, or 16x the serial baud rate. This internal clock is generated by dividing a reference clock by an integer divisor from 1 to $(2^{16} - 1)$ and a fractional divisor from 8/8 to 255/8.

4.1. Programming

To prepare the UART for communication, it is necessary to first configure the serial channel using the control registers LCR, SFR, DLL and DLM. These set the number of data and stop bits, the parity setting and the baud rate. These registers can be changed at any time, but if data is being received or transmitted then corruption of the serial data is likely to occur.

It is also a good idea to enable FIFOs using FCR and UCR, to decrease the number of data-transfer services the UART will require. The trigger levels can also be set at this stage using RFTR, RITR and TITR, although the TL16C550-compatible method using FCR7:6 will still work. If appropriate, auto-flow control may be enabled by writing the MCR, and the same register sets the initial state of the output handshaking lines.

Once the serial channel is configured, interrupts can be enabled by writing IER and setting MCR3.

The interrupt handler can read the IIR to determine what type of event needs servicing: the interrupt types are prioritised so that if more than one event needs servicing, the most urgent one is indicated.

A “transmitter FIFO empty” interrupt is cleared as soon as the IIR is read, so if there is no data waiting to be transmitted then no further action is needed. To restart the flow of transmitted data, the usual practice is for the user-mode part of the device driver to add the data to the software transmit queue and then kick-start transmission by re-writing to the IER with its current value (with bit 0 set). This will re-enable the “transmitter FIFO empty” interrupt and the interrupt handler will handle the transfer of transmit data to the UART, pushing another block every time the FIFO becomes empty.

4.2. Accessible Registers

The internal registers of the UART are listed in Table 13, organized by function with both full name and mnemonic.

Table 13 Accessible UART Registers

Register Selection	Mnem.
Indexed register select	IRSR
Line control (bit 7)	LCR
Safety catch control	SCC

UART Data	Mnem.
Receiver buffer	RBR
Transmitter holding	THR

UART Control	Mnem.
LSB divisor latch	DLL
MSB divisor latch	DLM
Interrupt enable	IER
FIFO control	FCR
Line control	LCR
Modem control	MCR
Synchronisation factor	SFR
Clock prescaler	CPR
UART configuration	UCR
Port control	PCR
Receive FIFO flow-control trigger	RFTR
Receive FIFO interrupt trigger	RITR
Transmit FIFO interrupt trigger	TITR

UART Status (read only)	Mnem.
Interrupt identification	IIR
Line status	LSR
Modem status	MSR
Chip ID register	CIDR
Receive FIFO level	RFLR
Transmit FIFO level	TFLR

Build & Test	Mnem.
Scratch pad	SCR
Inverted scratch pad	ISCR

Individual bits within the registers are referred to by the register mnemonic with the bit number appended. For example, LCR7 refers to bit 7 of the line control register.

The register accessed when an I/O read or write is performed depends on the bits 2:0 of the internal address, the divisor latch access bit (DLAB, which is LCR7), and the Indexed Register Select register (IRSR). Registers with A2:0 from 0 to 7 are accessed through BAR2 or BAR4, and those with A2:0 from 8 to 9 are accessed through BAR3 or BAR5. See section 3.1 for details.

The transmitter holding register and receiver buffer register are used to transfer data for transmission and received data respectively. These are eight-bit registers, but the serial data may be 5, 6, 7 or 8 bits long: data is right-justified and padded with zeroes on the left. The UART always receives and transmits bit 0 first. The THR and RBR can be accessed at the same time as serial data transmission and reception are taking place, because the serializer and deserializer are separate from the data buffers/FIFOs.

Table 14 shows how to select the required UART register.

Table 14 Register Selection

IRSR	DLAB	A2:0	Mnemonic	Register
X	0	0	RBR	Receiver buffer register (read only)
X	0	0	THR	Transmitter holding register (write only)
X	1	0	DLL	LSB divisor latch
X	1	1	DLM	MSB divisor latch
X	0	1	IER	Interrupt enable register
X	X	2	IIR	Interrupt identification register (read only)
X	X	2	FCR	FIFO control register (write only)
X	X	3	LCR	Line control register
X	X	4	MCR	Modem control register
X	X	5	LSR	Line status register (read only)
X	X	6	MSR	Modem status register (read only)
X	X	6	IRSR	Indexed register select register (write only)
X	X	8	PCR	Port-control: alternate control for CPR
X	X	9	SCC	Safety-catch control
0	X	7	SCR	Scratch pad register
1	X	7	ISCR	Inverted scratch pad register
2	X	7	CIDR	Chip ID register (read only)
3	X	7	SFR	Synchronisation factor register
4	X	7	-	Reserved for compatibility
5	X	7	-	Reserved for compatibility
6	X	7	UCR	UART configuration register
7	X	7	-	Reserved for compatibility
8	X	7	RFLR	Receive FIFO level register
9	X	7	TFLR	Transmit FIFO level register
10	X	7	-	Reserved for compatibility
11	X	7	-	Reserved for compatibility
12	X	7	-	Reserved for compatibility
13	X	7	RFTR	Receive FIFO flow-control trigger register
14	X	7	RITR	Receive FIFO interrupt trigger register
15	X	7	TITR	Transmit FIFO interrupt trigger register
16	X	7	CPR	Clock prescaler register
17	X	7	WER	Wake event enable register
>17	X	7	-	Reserved for future use – do not read or write

X = irrelevant, 0 = low level, 1 = high level

The system programmer, using the host, can access any of the UART registers, as summarized in Table 15.

Table 15 UART Register Summary

Register Mnemonic, Access†	Register Bit Number							
	Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
RBR (read only) A=0, DLAB=0	Data Bit 7 (MSB)	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0 (LSB)
THR (write only) A=0, DLAB=0	Data Bit 7	Data Bit 6	Data Bit 5	Data Bit 4	Data Bit 3	Data Bit 2	Data Bit 1	Data Bit 0
DLL A=0, DLAB=1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DLM A=1, DLAB=1	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
IER A=1, DLAB=0	0	0	Low power enable (ignored)	Sleep mode enable (ignored)	(EDSSI) Enable modem status interrupt	(ERLSI) Enable receiver line status interrupt	(ETHREI) Enable transmitter buffer empty interrupt	(ERBFI) Enable received data available interrupt
FCR (write only) A=2	Receiver Trigger (MSB)	Receiver Trigger (LSB)	0	0	DMA mode select (ignored)	Transmitter FIFO reset	Receiver FIFO reset	FIFO enable
IIR (read only) A=2	FIFOs Enabled‡	FIFOs Enabled‡	0	0	Interrupt ID Bit 3‡	Interrupt ID Bit 2	Interrupt ID Bit 1	0 if interrupt pending
LCR A=3	(DLAB) Divisor latch access bit	Set break	Stick parity	(EPS) Even parity select	(PEN) Parity enable	(STB) Number of stop bits	(WLSB1) Word length select bit 1	(WLSB0) Word length select bit 0
MCR A=4	0	0	AFE	Loop	OUT2 (interrupt enable)	OUT1	RTS	DTR
LSR (read only) A=5	Error in Receiver FIFO‡	Transmitter Empty (TEMT)	Transmit holding register (THRE)	Break Interrupt (BI)	Framing Error (FE)	Parity Error (PE)	Overrun Error (OE)	Data Ready (DR)
MSR (read only) A=6	(DCD) Data carrier detect	(RI) Ring indicator	(DSR) Data set ready	(CTS) Clear to send	(ΔDCD) Delta data carrier detect	(TERI) Trailing edge ring indicator	(ΔDSR) Delta data set ready	(ΔCTS) Delta clear to send
IRSR (write only) A=6	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PCR A=8	RFU	RFU	Clock select bit 1	Clock select bit 0	Hold CTS	RFU	RFU	RFU
SCC A=9	Indexed reg safety catch	RFU	RFU	RFU	RFU	RFU	RFU	RFU
SCR A=7, IRSR=0	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Register Mnemonic, Access†	Register Bit Number							
	Bit 7	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
ISCR A=7, IRSR=1	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CIDR (read only) A=7, IRSR=2	0	0	0	1	0	0	0	0
SFR A=7, IRSR=3	RFU	RFU	RFU	SF=16	SF=8	SF=4	RFU	RFU
UCR A=7, IRSR=6	RFU	RFU	RFU	RFU	RFU	RFU	Enable deep FIFOs	Force AFC on
RFLR (read only) A=7, IRSR=8	Error in Receiver FIFO	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TFLR (read only) A=7, IRSR=9	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RFTR A=7, RSR=13	RFU	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RITR A=7, RSR=14	RFU	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TITR A=7, RSR=15	RFU	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CPR A=7, RSR=16	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WER A=7, RSR=17	INT	RFU	RFU	SIN#	ΔDCD	TERI	ΔDSR	ΔCTS

† In this column, 'A' refers to the decimal value of the internal address bus. ‡ These bits are always 0 when FIFOs are disabled.

Master Reset

The UARTs are reset when PCI RESET# is asserted. Table 16 and Table 17 summarize the effect of reset on the UART circuits.

Table 16 Effect of RESET on UART Signals

UART Signal	Reset control	Signal Reset State
DTR#	Reset	High
RTS#	Reset	High
SOUT	Reset	High

Table 17 Effect of RESET on UART Registers

UART Register	Register reset state
LSR	Bits 7,4,3,2,1,0 cleared Bits 6 & 5 set
MCR	All bits cleared Note bits 7:6 permanently cleared
IER	All bits cleared Note bits 7:6 permanently cleared
FCR	All bits cleared
IIR	Bits 7,6,3,2,1 cleared Bit 0 is set
LCR	All bits are cleared
TFTR	All bits are cleared
MSR	Bits 3–0 cleared Bits 7–4 input signals
IRSR	All bits cleared
SCR	All bits cleared
LSB & MSB	All bits cleared
RBR	All bits cleared
THR	All bits cleared
RFTR	All bits cleared
RITR	All bits are cleared
UCR	All bits are cleared
WER	All bits are cleared

Table 18 Effect of RESET on UART Interrupts

Interrupt Type	Reset Control	Interrupt Reset State
modem status changes	Reset/Read MSR	Low
receiver data ready	Reset/Read RBR	Low
RCVR errors	Reset/Read LSR	Low
THRE	Reset/Read IIR/Write THR	Low

Table 19 Effect of RESET on UART FIFOs

FIFO	Reset Control	FIFO Reset State
Receiver FIFO	Reset FCR1–FCR0 ΔFCR0	FIFO empty
Transmitter FIFO	Reset FCR1–FCR0 ΔFCR0	FIFO empty

Serial Data Format

A 0 in RBR or THR corresponds to a logic low on SIN or SOUT, and a 1 in RBR or THR corresponds to a logic high on SIN or SOUT.

Bit 0 is always the least significant bit (LSB) and is the first bit to be serially transmitted or received.

A start bit or line break state corresponds to a logic low on SIN or SOUT, and a stop bit or inter-byte marking state corresponds to a logic high on SIN or SOUT.

4.3. Transmitter/Receiver Section

The status of the receiver is given by the Line Status Register (LSR).

The control of the receiver section and format of the data characters such as number of data bits, parity, etc is controlled by the Line Control Register (LCR). Note if parity is used (LCR3) then the polarity of parity LCR4 is required.

As serial asynchronous data is fed into the receiver serial data input terminal SIN, the UART continually looks for a high-to-low transition. Upon detection of the transition, an internal counter is reset and counts the SF× clock input to SF/2, which is the centre of the start bit. (SF is the Synchronisation Factor)

The receiver is prevented from assembling a false data character caused by noise on the SIN input, by verification of the start bits. Note: The start bit is valid only if SIN is still low.

The UART receiver section contains a Receiver Buffer Register (RBR) which is a FIFO and a Receiver Deserializer Register (RDR). Data fed into the receiver serial data input terminal SIN is deserialized by the RDR and is fed into RBR.

The control of the receiver section and format of the data characters such as number of data bits, parity, etc is controlled by LCR. Note if parity is used (LCR3) then the polarity of parity LCR4 is required.

The receiver timing is supplied by the baud clock generator.

In FIFO modes, FCR is used to enable and reset the receiver FIFO and also can be used to set data trigger levels for when interrupts are generated.

In non FIFO mode (16C450 style), when the received data available interrupt is enabled, an interrupt is generated when a character is placed in the receiver buffer register. When RBR is read, the interrupt is cleared.

Transmitter Holding Register & Multiplexer Register (THR & TMR)

The UART transmitter section contains a Transmitter Holding Register (THR), which is a FIFO, and a transmitter multiplexer register (TMR). THR receives data off the internal data bus and moves it into the TMR, while the transmitter is idle, which serializes the data and outputs it to the transmitter data serial output terminal SOUT.

The transmitter timing is supplied by the baud clock generator.

In FIFO modes, FCR is used to enable and reset the transmitter FIFOs and can be used to set data trigger levels for when interrupts are generated. For more details see Section 4.2.

In non FIFO mode (16C450), when the transmitter holding register empty interrupt is enabled, an interrupt is generated when THR is empty. When a character is loaded into the register, the interrupt is cleared.

Line Control Register (LCR)

LCR controls the format of the data character and is applicable to both transmitter and receiver. The LCR is read-writable. Its contents are described below.

Bits	Description
7	Divisor latch access bit (DLAB) 1—enables access to DLL & DLM 0—enables access to IER, THR &RBR
6	1—SOUT is forced to the spacing state (low)
5	1—If LCR3 is 1, parity bit transmission & reception is the state opposite to LCR4 value. If LCR4 is 1, even parity enabled. (or cleared parity enabled, if LCR5 is 1) 0—odd parity enabled (or set parity enabled, if LCR5 is 1) This forces parity to a known state
4	1—even parity enabled. (or cleared parity is enabled, if LCR5 is 1) 0—odd parity enabled (or set parity enabled, if LCR5 is 1).
3	1—a parity bit is generated between the last data word bit & stop bit in data transmitted & checked by the receiver 0—no parity is selected; see Table 20
2	Specifies either one or two stop bits in each transmitted character. 0—one stop bit is generated in the data 1—1½ or 2 stop bits are generated in the data: see Table 22. The receiver clocks only the first stop bit, regardless of the number of stop bits selected
1:0	These two bits specify the number of bits in each transmitted or received serial character; see Table 21

Table 20 Parity Selection

LCR5:3	Description
X X 0	No parity
0 0 1	Odd parity
0 1 1	Even parity
1 0 1	Set parity
1 1 1	Cleared parity

Table 21 Word Length Selection

LCR1:0	Description
0 0	Word length is 5 bits
0 1	Word length is 6 bits
1 0	Word length is 7 bits
1 1	Word length is 8 bits

Table 22 Stop Bit Length Selection

LCR2:0	Description
0 X X	1 stop bit generated
1 0 0	1½ stop bits generated
1 0 1	2 stop bits generated
1 1 0	2 stop bits generated
1 1 1	2 stop bits generated

Use the following steps to create a line break:

Note: no invalid characters are transmitted because of the break.

1. When THRE empty status occurs, load a zero byte
2. After the next THRE, set the break
3. When TEMT is set to high, wait for the transmitter to be idle
4. Clear the break when the transmission has to be re-established.

Line Status Register (LSR)

Read-only register that indicates the status of serial data reception.

Bits	Description
7	Set when a character with a parity, framing, or break error enters the FIFO. Cleared when LSR is read & no FIFO characters have an error flag set.
6	Set when no character is being transmitted, & no characters queued for transmission in THR or transmit FIFO
5	When FIFOs are disabled, set when THR is empty & the UART is ready for a new character to be written to it. When FIFOs are enabled, set when the FIFO is empty
4	Break interrupt indicator. A break interrupt or line break occurs when SIN is held low for longer than the normal transmission time for the start, data, parity & stop bits configured. The interrupt, whatever its length, is queued like a received character whose data bits are all cleared & a BI flag is attached to it in the receive FIFO. 1—the next character to be read from the RBR has its BI flag set The UART initially attempts to interpret the interrupt as a received character, so LSR3 is set because there was no valid stop bit & LSR2 may be set if parity bits are enabled
3	Framing error indicator. When a character without the expected stop bit is received, a framing error flag is attached to the character in the receive FIFO. 1—the next character to be read from the RBR has its FE flag set When a character is received with a framing error, the UART assumes that character synchronization is lost & attempts to resynchronize by assuming that what was sampled as the stop bit of a character is actually the start bit of the next character
2	Parity error indicator. When a character is received that does not have the expected value where the parity bit is expected, a parity error flag is attached to the character in the receive FIFO. 1—the next character to be read from the RBR has its PE flag set
1	Overrun error indicator. Set when a character is received & nowhere for it to be stored, i.e. the receive FIFO is full. The character & associated error flags are lost. Cleared when LSR is read
0	Data ready indicator. Set when a character can be read from the RBR or receive FIFO

Note: Writes to LSR are ignored, but it is recommended to avoid them because there may be unpredictable results on other UARTs.

Interrupt Enable Register (IER)

IER controls independent enable/disable for UART interrupt sources. A disabled source does not cause assertion of IREQ# and its code does not appear on the IIR.

Bit	Description
3	1—enables modem status interrupts
2	1—enables receiver line status interrupts
1	1—enables transmitter holding register empty interrupts
0	1—in FIFO modes, enables received data available & character time-out interrupts

Interrupt Identification Register (IIR)

IIR indicates the interrupt status of the UART and information about the FIFO status.

To ensure that the most time-critical interrupt sources are serviced first, IIR returns a code indicating the highest-priority interrupt sources that is currently active. The interrupt sources are prioritized as follows:

- (Highest priority) Receiver line status
- Receiver character time out
- Receiver data ready
- Transmitter holding register empty
- (Lowest priority) Modem status

The contents of the IIR are indicated in the table below.

Bits	Description
7:6	Set when FCR0 is set, i.e., the UART is in a FIFO mode
3:1	Identifies the highest-priority interrupt currently active, as indicated in Table 23.
0	Cleared when any interrupt is active; set when no interrupt sources are active

Table 23 Interrupt ID Codes in IIR3:0

Value 3:0	Interrupt	Priority Level	Source	Clear Mechanism
0110	Receiver line status	1	OE, PE, FE, or BI are set in the LSR	Read LSR
1100	Character time-out	2	During the last four character times, at least one character has been waiting in the receive FIFO, and the FIFO has been inactive.	Read RBR
0100	Received data available	3	The receive FIFO has reached its trigger level.	Read RBR until FIFO drops below the trigger level
0010	THRE	4	THRE is set in the LSR: the UART is ready to be given more data to transmit.	Read IIR or write to THR
0000	Modem status	5	At least one of the MSR3:0 bits are set, because CTS#, DSR#, RI#, or DCD# have changed	Read MSR
0001	None	None	No interrupt source active	-

FIFO Control Register (FCR)

Write only register at the same location as IIR. It enables and clears the FIFOs, and sets the trigger level of the receiver FIFO.

Bits	Description
7:6	Trigger level for receiver FIFO interrupt
2	1—all bytes in transmitter FIFO are cleared & counter reset to 0. Does not clear the multiplexer register. Write 1 to clear
1	1—clears all bytes in the receiver FIFO & resets counter. Does not clear the multiplexer register. Write 1 to clear
0	1—enables transmit & receive FIFOs; also enables write access to other FCR bits, otherwise they are not programmed. An alteration to this bit clears the FIFOs

Table 24 Receiver FIFO Trigger Level

FCR7	FCR6	Trigger Level (Bytes)		Desc.
		UCR1=0	UCR1=1	
0	0	01	01	Not empty
0	1	04	8	At least quarter-full
1	0	08	16	At least half-full
1	1	14	28	At least seven-eighths full

4.4. FIFO Interrupt Mode Operation

When the receiver FIFO and receiver interrupts are enabled (FCR0=1, IER0=1, IER2=1), a receiver interrupt occurs as follows:

1. When the FIFO has reached its programmed trigger level, the received data available interrupt is issued to the host. This is cleared when the FIFO drops below its programmed trigger level.

2. In addition to when the FIFO trigger level is reached, and as the interrupt, is cleared when the FIFO drops below the trigger level, the IIR receive data available indication also occurs.

3. The receiver line status interrupt (IIR = 06) holds a much higher priority than the received data available (IIR = 04) interrupt.

4. When a character is transferred from the deserializer to the receiver FIFO, the data ready bit (LSR0) is set. When the FIFO is empty, it is cleared.

When the receiver FIFO and receiver interrupts are enabled:

1. FIFO time-out interrupt occurs when the following conditions exist:
   a. A minimum of one character is still present in the FIFO.
   b. More than four continuous character times have passed (if two stop bits are programmed, the second one is included in this time delay) before a new serial character is received.
   c. More than four continuous character times have passed since the reading of the FIFO was carried out by the host. This causes a maximum character received to interrupt an issued delay of 160 ms at 300 baud with a 12-bit character.
2. The FIFO interrupt is cleared when a time-out interrupt occurs. When the host reads one character from the receiver FIFO, this causes the timer to reset. The non-occurrence of a time-out interrupt, causes the time-out timer to reset after a new character is received, or after the host reads the receiver FIFO.

When the transmitter FIFO and THRE interrupt are enabled (FCR0 = 1, IER1 = 1), transmit interrupts occur as follows:

3. When the transmit FIFO is empty, it causes the transmitter holding register interrupt [IIR3:0 = 2] to occur. When either the THR is written to, or the IIR is read, this event causes the interrupt to be cleared [IIR3:0 = 1].
4. A minimum of two bytes in the transmit FIFO are required, at the same instance since the last time that THRE = 1, or this causes the transmit FIFO empty indicator (LSR5 (THRE) = 1) to be delayed one character time minus the last stop bit time. The first transmitter interrupt is instantaneous after changing FCR0 when it is enabled.

The behavior of the THRE interrupt is summarized in Table 25 terms of what events cause it to become set and cleared.

Table 25 THRE Interrupt Behavior

Event	Effect on THRE interrupt
UART reset or IER1 cleared	Interrupt cannot occur until IER1 set
IER written with bit 1 set	Interrupt is set immediately if transmit FIFO empty
Transmit FIFO becomes empty	Interrupt set
IIR read	Interrupt cleared
THR written	Interrupt cleared
Number of bytes in transmit FIFO drops from TITR+1 to TITR	Interrupt set

4.5. FIFO Polled Mode Operation

If any, or all of the Interrupt enable masks are cleared (IER0, IER1, IER2, IER3, or all four = 0) data and error conditions are still available from the UART by using a polling method.

The user application or driver can check transmitter, and/or receiver FIFO status by querying the Line Status Register (LSR).

In Polled mode, the FIFOs continue to store data in the expected fashion with the exception that trigger level or time-out flags are not generated.

Receive FIFO Level Register (RFLR)

This read-only register allows a much faster emptying of the receiver FIFO, by eliminating the need to perform an LSR read before each read of the RBR. If RFLR7 is clear, then the device driver can immediately perform N reads of the RBR, where N is the value given by RFLR6:0.

Bits	Description
7	Set if any of the entries in the receiver FIFO has any of its error flags (parity, framing, or break error) set. Cleared when the host reads LSR & there are no subsequent errors in the FIFO.
6:0	Returns a value between 0 and 31, relating to the number of data entries stored in the receiver FIFO.

Transmit FIFO Level Register (TFLR)

This read-only register returns a value between 0 and 32, relating to the number of available spaces in the transmit FIFO.

Receive FIFO Flow-Control Trigger Register (RFTR)

This register allows an arbitrary trigger level to be set for auto-RTS flow control (see section 4.7). If the value in this register is non-zero, it is used instead of the trigger level set by FCR7:6. Valid values are 0 to 31.

Receive FIFO Interrupt Trigger Register (RITR)

This register allows an arbitrary trigger level to be set for the received data available. If the value in this register is non-zero, it is used instead of the trigger level set by FCR7:6. Valid values are 0 to 31.

Transmit FIFO Interrupt Trigger Register (TITR)

This register allows an arbitrary trigger level to be set for the THRE interrupt. Valid values are 0 to 31. If the value in this register is non-zero, then when the number of bytes in the transmit FIFO drops from TITR+1 to TITR the THRE interrupt state will be set. The THRE interrupt is still also generated when the transmit FIFO becomes empty, allowing for the case where less than TITR characters have been written to the transmit FIFO.

Modem Control Register (MCR)

The MCR controls the interface with the modem or data set as described in Figure 20. MCR can be written and read. The RTS# and DTR# outputs are directly controlled by their control bits in this register (unless the UART is in loopback mode). A high input asserts a low signal (active) at the output terminals. The MCR bits are shown below:

Bit	Description
5	1—enables auto flow-control as described in section 4.7. Auto flow-control may also be enabled by UCR0
4	Provides a local loopback feature for diagnostic testing of the channel. See section 4.6.
3	1—enables external serial channel interrupt
1	1—RTS# output is forced low 0—RTS# output is forced high The RTS# output of the serial channel can be input into an inverting line driver to obtain the proper polarity input at the modem or data set
0	1—DTR# output is forced low 0—DTR# output is forced high The DTR# output of the serial channel can be input into an inverting line driver in order to obtain the proper polarity input at the modem or data set

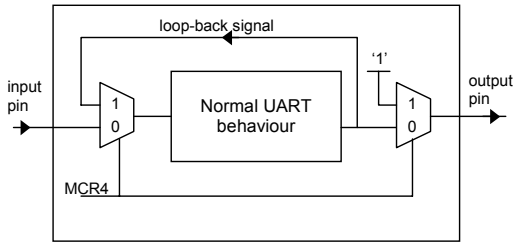
Note: MCR7:6 are permanently cleared.

4.6. Loopback Mode

The UART enters loopback mode when MCR4 is set, which is useful for testing. Serial data and modem control outputs SOUT, DTR#, RTS#, OUT1#, and OUT2# are forced into an inactive (high) state so that attached devices are not affected. The signals which normally feed these pins are fed into the serial data and modem control inputs (SIN, CTS#, DSR#, DCD#, and RI#), which are disconnected from their input pins.

In this mode, data transmitted is immediately received, allowing the host to test the UART transmit and receive data paths. Interrupt control continues to operate based on the state of the looped-back signals rather than the actual SOUT, DTR#, RTS#, OUT1#, and OUT2# input pins.

Internal signal normally controlled by this pin	is controlled by the signal which normally goes to this output pin: output is forced high
SIN	SOUT
CTS#	DTR#
DTR#	DSR#
DCD#	OUT1#
RI#	OUT2#



Modem Status Register (MSR)

MSR allows the state of the modem status lines CTS#, DSR#, RI#, and DCD# to be read by the host. Bits 7-4 of this read-only register reflect the assertion state of the corresponding input pins and bits 3:0 indicate whether changes have occurred on these inputs since MSR was last read.

When the UART is operating under interrupts, the host can be notified of changes in the modem status lines by enabling modem status interrupts (set IER3), in which case a priority-5 interrupt is generated when any of MSR3:0 become set. The interrupt is generated whether the setting of MSR3:0 is caused by changes on the modem status lines or by changes of MCR3:0 in loopback mode.

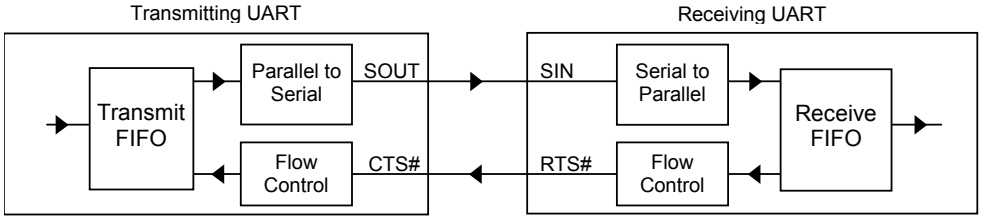
Bit	Description
7	Set when the DCD# input is low. In loopback mode (MCR4 is set), MSR7 reflects the value last written to MCR3
6	Set when the RI# input is low. In loopback mode, MSR6 reflects the value last written to MCR2
5	Set when the DSR# input is low. In loopback mode, MSR5 reflects the value last written to MCR0
4	Set when the CTS# input is low. In loopback mode, MSR4 reflects the value last written to MCR1
3	1—DCD# input has changed state since the last time the MSR was read
2	1—RI# input has changed from a low to a high state since the last time the MSR was read. High-to-low transitions on RI# do not affect TERI
1	1—DSR# input has changed state since the last time the MSR was read
0	1—CTS# input has changed state since the last time the MSR was read

4.7. Auto Flow Control

Auto flow control consists of two functional parts: auto-CTS and auto-RTS. These features allow the flow of serial data to be throttled, preventing data loss due to receive buffer overruns, without relying on fast interrupt service. RTS# and CTS# are often used for flow control, but if the host has to perform that function then delays in servicing interrupts can mean that the flow control is not quick enough, and data can be lost. When a pair of UARTs are connected that both have auto flow control enabled, then loss-free data transfer is possible whatever the interrupt latencies of the host systems.

The RTS# output of the receiving UART must be connected to the CTS# input of the transmitting UART, as shown below. Usually a full-duplex link will be used and so the diagram below will be only half the system, with both UARTs having a transmitting and a receiving side and being connected symmetrically.

Figure 1 Auto Flow Control (Auto-RTS & Auto-CTS) Example



4.7.1. Auto-RTS

Auto-RTS attempts to control the flow of serial data in to the UART. When enabled, it automatically signals to a connected device that it should stop transmitting, because there is a risk of the receive FIFO overrunning. This is done by making the state of the RTS# output dependent on the level of the receive FIFO: RTS# is asserted when the receive FIFO is below a certain trigger level, and deasserted when the receive FIFO is at or above that trigger level. The OX16PCI958 UARTs support the setting of the trigger level to four predefined levels using FCR7:6 and also allow a precise value to be set using RFTR.

4.7.2. Auto-CTS

Auto-CTS controls the flow of serial data from the UART. When enabled, the transmitter will not start transmitting a new data byte unless the CTS# input is asserted. If data transmission is stopped in this way, it restarts as soon as CTS# is asserted. The UART never sends partial data bytes.

4.7.3. Enabling Auto-RTS & Auto-CTS

It is possible to have neither auto-RTS nor auto-CTS enabled (with manual control of RTS#), or both enabled, or auto-CTS enabled with RTS# held deasserted. Auto-RTS and auto-CTS are activated by setting bits MCR5 and MCR1, but auto flow control is also affected by MCR1, MCR4, and UCR0.

Clock Prescaler Register (CPR)

This register allows more fine-grained control over the baud rate than is possible using the divisor alone. Before being divided by the divisor value stored in DLL and DLM, the clock is prescaled, being divided by (CPR/8). Valid values of the CPR are 8 to 255 inclusive. The reset value of the CPR is 8, i.e. the prescaler has no effect. This register is automatically set to 8, 16 32 or 64 when the PCR is written.

Synchronization Factor Register (SFR)

The UART uses an internal clock that is faster (by a fixed integer factor) than the baud rate selected. The factor by which it is faster is called the synchronisation factor (SF). Received data is clocked into the UART every SF clock cycles, and the position of these enabled cycles is selected so as to clock in data from as close to the middle of each serial bit as possible. A higher value of SF gives a sampling time which is closer to the centre of the bits, and hence gives greater tolerance of line noise, but also gives higher power consumption and a lower maximum baud rate.

This register allows the programmer to select the synchronisation factor used. Allowed values for writing to this register are 04h, 08h, and 10h.

Programmable Baud Rate Generator

The internal clock used by the UART transmit and receive units is generated by taking the clock input fed into XTALI and dividing it first by the global clock divider, then by a prescaling factor, and then by a 16-bit integer value (the “divisor”). The baud rate calculation is:

baud rate =
XTALI input freq ÷ (global predivider x (CPR/8) x divisor x SF)

The divisor is stored in two 8-bit divisor latches. Setting the divisor latches is a necessary step in the initialization of the UART before data can be transmitted or received.

Loading a divisor of zero (all bits cleared in DLL and DLM) stops the clocking of the receiver and transmitter, i.e. it gives a baud rate of zero. No serial data is transmitted or received, no data enters the receive FIFO or leaves the transmit FIFO. The level on SOUT can still be changed by writing LCR6, and all other UART functions continue to operate.

Port Control Register (PCR)

Bit 3 allows the CTS input of the UART to be held in a ‘true’ state, regardless of the state of the CTS# input pin: see Table 26.

Table 26 Hold CTS Values

Value	Description
0	CTS reflects state of CTS# pin
1	CTS held true

4.8. Chip Type Identification

To determine whether a UART is a OX16PCI958 UART, and to clear the extended registers safety catch:

- Ensure that bit 4 of IER is cleared
- Write 80h to LCR
- Write 00h to DLM
- Set X=23h
- Repeat the following 42 times:
  - Write the value X to DLM
  - Set X=X x 2
  - Set bit 0 of X to the exclusive-or of bit 7 and bit 6 of X
- If IER4 is set, the chip is an OX16PCI958 or a future device with the same extended-register system. Write 2 to IRSR and read CIDR to identify the device type. If IER4 is clear, the chip is not a known device.

Do not make any other read or write access to the UART while writing the sequence of values to the DLM. The sequence should be: 00h, 23h, 47h, 8Fh, 1Eh, 3Ch, 79h, F2h, E4h, C8h, 91h, 22h, 45h, 8Bh, 16h, 2Ch, 59h, B3h, 67h, CEh, 9Dh, 3Ah, 75h, EAh, D4h, A9h, 53h, A7h, 4Fh, 9Fh, 3Eh, 7Dh, FAh, F4h, E8h, D0h, A1h, 43h, 87h, 0Eh, 1Ch, 38h, 71h.

Extended Registers Safety Catch

Some host systems may violate the TL16Cx50 specifications and write to the MSR, which sets IRSR in the OX16PCI958 UART and could lead to failure due to an apparently failed scratch register or corruption of the extended registers. To prevent this problem, the IRSR is protected by a safety catch. After any reset of the UART a safety-catch is engaged which causes writes to the MSR/IRSR address to be ignored.

To clear the safety catch and enable access to the extended register set, an OX16PCI958-aware device driver may perform the chip type identification sequence to enable writes to the IRSR until the next UART reset. Alternatively, the safety catch can be written and read directly at bit 7 of the SCC register.

Chip ID Register (CIDR)

Read-only register which provides an identification code so software can distinguish between versions of the OX16PCI958, or future components with a similar interface. Codes are given in Table 27

Table 27 Chip Identification Code Values

Value	Description
00-0Fh	Reseved
10h	UART conforming to this specification
11-1Fh	Reserved for devices with register sets compatible with the OX16PCI958. Device drivers may assume the device conforms with this specification, though additional features may also be present.
others	Reserved

UART Configuration Register (UCR)

Read-write register with reset value taken from the configuration EEPROM.

Bit	Description
2	1—UART generates transmit-buffer-empty interrupts until < 16 spaces are left in the transmit FIFO, or no data is written
1	1—FIFOs are always 32 deep when FCR0 is set
0	1—when MCR4 is cleared, auto flow control is always on, even when the driver does not enable it (should only be used if the attached device is using RTS-CTS handshaking for flow control)

Table 28 Operation when UCR0 is Cleared

MCR5	MCR1	Flow Control Configuration
1	1	Auto-RTS# & auto-CTS# enabled (auto flow control enabled)
1	0	Auto-CTS# only enabled
0	X	Auto-RTS# & auto-CTS# disabled

Table 29 Operation when UCR0 is Set

MCR4	MCR5	MCR1	Flow Control Configuration
0	X	1	Auto-RTS# and auto-CTS# enabled (auto flow control enabled)
0	X	0	Auto-CTS# only enabled, RTS# deasserted
1	1	1	Auto-RTS# and auto-CTS# enabled (auto flow control enabled)
1	1	0	Auto-CTS# only enabled
1	0	X	Auto-RTS# and auto-CTS# disabled

Scratchpad Register (SCR)

The 8-bit read/write scratch register has no affect on either channel in the UART. It is intended to be used by the programmer to hold data temporarily.

Inverted Scratchpad Register (ISCR)

Writes to this register set the scratchpad register. Reads return the current scratch register value with all bits inverted.

Wake Event Enable Register (WER)

This register controls the serial port events that can trigger a wake event, e.g. on the PCI bus. One wake event source requires the UART's clock to be running, but five are entirely asynchronous and require no clock.

Bit	Description
7	1—a wake event is generated whenever an enabled interrupt event occurs. Requires the UART clock to be running
4	1—a wake event tis generated whenever SIN goes low
3	1—a wake event is generated whenever DCD# has a changed state since the last time the UART was clocked
2	1—a wake event is generated whenever RI# goes low
1	1—a wake event is generated whenever DSR# has a changed state since last time the UART was clocked
0	1—a wake event is generated whenever CTS# has a changed state since the last time the UART was clocked

The wake event from the UART is a rising edge on the wake output: it is the job of external circuitry to latch this signal if needed.

4.9. SISR Function

A "Shared Interrupt Status Register" (SISR) function can be enabled. The purpose of this function is to help device drivers arbitrate between the multiple UARTs that may be requesting an interrupt service. The function consists of a single 8-bit read-only register. The binary value of this value will either be FFh, indicating that there is no UART requesting an interrupt service, or a value from 0 to 7 inclusive indicating the number of the UART that the device driver should service next. This value is determined using a round-robin algorithm.

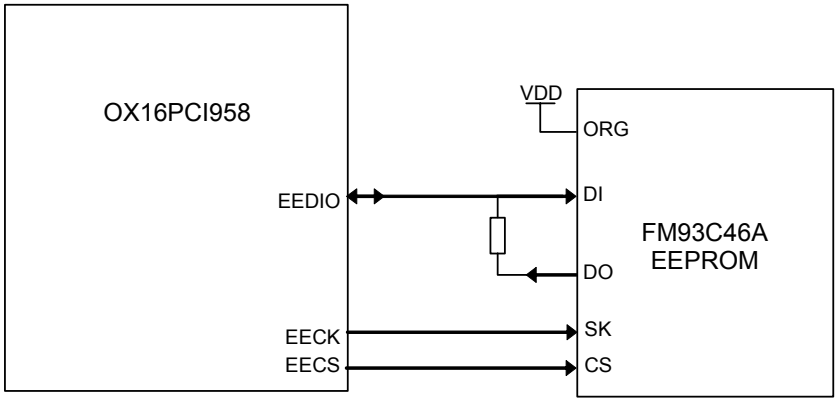
There is no requirement to use the SISR function, UARTs may be serviced in any order—the SISR is only included as an aid to fair servicing of the ports.

5. EEPROM

5.1. The EEPROM Reader

After the host bus RESET signal has been asserted and then released, the OX16PCI958 reads the attached Microwire-type serial EEPROM to obtain configuration information. The EEPROM must be in 16-bit mode, and connected for 3-wire operation, as shown in Figure 2.

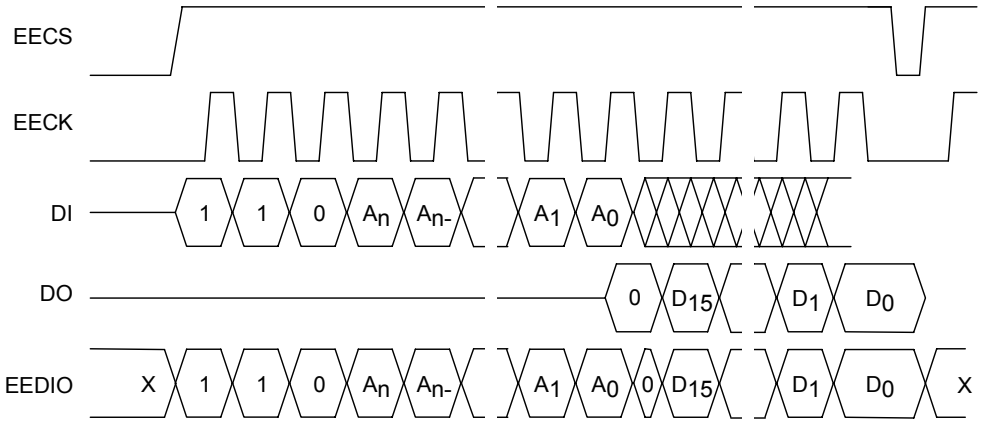
Figure 2 Example Circuit for EEPROM Connection



The EEPROM must be set up to provide 16-bit data, e.g. by tying the ORG pin high on an AT96C46.

Any serial EEPROM with a 16-bit data and Microwire-compatible read instruction, where the number of address bits is either 6 or 8, should be suitable for use with the OX16PCI958. The EEPROM does not have to have a sequential read feature. This means that most 93C46, 93C56 and 93C66 parts should be suitable.

Figure 3 EEPROM Read Waveform



The interface timings are designed to be suitable for EEPROMs with maximum clock frequencies down to 250 kHz. The EEPROM reader unit sets up EECS and EEDIO 70 PCI clock cycles before generating a rising edge on EECK, and it samples EEDIO 70 PCI clock cycles after the EECK rising edge. The EECK clock cycle lasts for 140 PCI clock cycles, giving an EEPROM clock rate of 238 kHz when the PCI clock is running at $33^{1}/_{3}$ MHz.

Figure 4 EEPROM Signal Timings During Initialization

Type	Signal	Timing	Min	Nom	Max	Unit
Generated	EECK	Cycle period	4140	140 x T_{PCI}	-	ns
Generated	EECK	Low time	2040	70 x T_{PCI}		ns
Generated	EECK	High time	2040	70 x T_{PCI}		ns
Generated	EECS	Set-up before SK↑	2040	70 x T_{PCI}	-	ns
Generated	EECS	Hold after SK↓	2040	70 x T_{PCI}	-	ns
Generated	EECS	CS low time	2040	70 x T_{PCI}		ns
Generated	SBDIO	Set-up before SK↑	2040	70 x T_{PCI}	-	ns
Generated	SBDIO	Hold after SK↑	2040	70 x T_{PCI}	-	ns
Allowed	SBDIO	Delay to valid after SK↑	-	70 x T_{PCI}	2040	ns
Allowed	SBDIO	Hold after SK↑	-	70 x T_{PCI}	2040	ns
Allowed	SBDIO	Delay to high-Z after SK↑	-	70 x T_{PCI}	2040	ns

These values are calculated using the worst-case PCI clock period of 30 ns. 70 x T_{PCI} is 2.10 µs for a $33^1/_3$ MHz clock.

5.2. EEPROM Data Format

The contents of the EEPROM are used to generate a series of internal register writes in the OX16PCI958 – the data specifies a sequence of addresses to be written to and the byte to be written. Table 30 shows how the data in the EEPROM is organized.

Table 30 EEPROM Data Organization

Address	Bits 15:8 of data	Bits 7:0 of data
00h	sync. byte: must be 10h	end address in EEPROM
01h	internal address	data to write
02h	internal address	data to write
…	…	…
end address	internal address	data to write

After reset, the block reads the first word in the EEPROM. This word must contain the binary pattern 00010000 in its top 8 bits. The block uses this fact to work out the number of address bits required by the EEPROM in the following reads, thus allowing a wide range of EEPROM devices to be used.

Now that the number of EEPROM address bits is known, the block knows where the data begins within the serial stream, and it reads the first EEPROM word again. It latches the bottom 8 bits of this word internally, and this value is used as the end-address of the data in the EEPROM.

In the next phase, the block reads word 1 through to *end-address* in sequence, stopping after it has read the end-address word. It considers the top 8 bits of each word as an internal address and the bottom 8 bits as data. The block performs a write on its internal bus interface, using the address and data from the EEPROM word.

5.3. Example EEPROM Data

This section explains how to determine what EEPROM data must be written to the device and gives a worked example.

To decide on the EEPROM data, first list the sequence of values that need to be written to registers and then work out the internal address for each register. Using the information in Table 30, EEPROM information is as follows:

- The first word (address 00h) in the EEPROM must have an upper byte of 10h, and a lower byte containing the value *number of register writes*.
- Following words must have an upper byte of *internal address* and a lower byte of *data to write*. For example, to configure a product with eight serial ports, the following configuration is required:
  - Set the PCI Vendor IDs to 1415h, and the Device IDs to 9538h
  - Enable UARTs 0 to 7
  - Set the clock prescaler for UART 0 to 1Ch to divide clock by 3.5 (this is just to demonstrate a complex EEPROM sequence)
  - Set the PCI ID codes by writing registers in the internal address range 00h-19h (see Table 12 on page 15).
  - Set the UART enable and bank switching registers at internal addresses 40h, 41h and 4Ch, as described in section 3.2.

Setting the clock prescaler requires a sequence of writes, because the CPR is an indexed register. This type of setting is usually done by the device driver controlling the card, but the operation is included here as an example of how to do such a configuration, perhaps for use with an old device driver. In this worst-case example, it takes seven EEPROM words to change one UART register, but to add more register changes would not need so much EEPROM space.

Before accessing an indexed register, the indexed register safety catch must be switched off, using the following sequence:

1. Switch off the safety catch for UART 0
2. Set IRSR for UART 0 to 16 (the index of the CPR)
3. Write the CPR value
4. Set IRSR for UART 0 back to 0, so that the scratch register is seen at offset 7 by default, for backwards compatibility
5. Switch the safety catch for UART 0 back on

Table 31 shows how the example information above translates to EEPROM data.

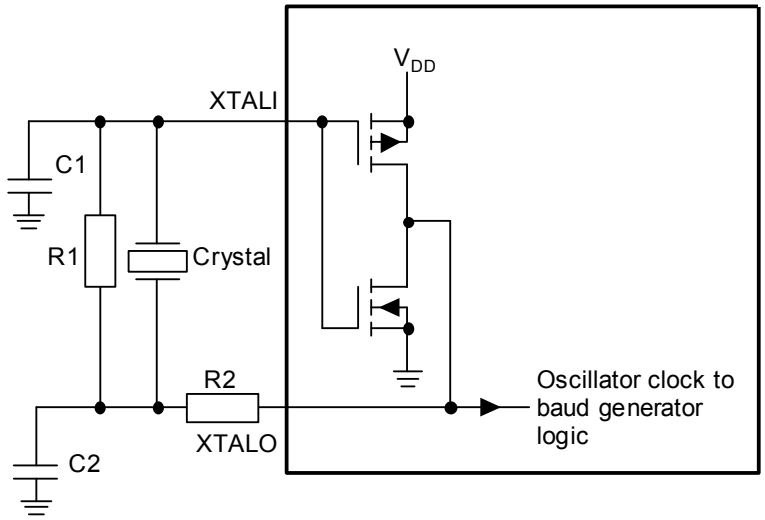
Table 31 Example EEPROM Data

Address in EEPROM	Bits 15:8 of EEPROM data: internal address	Bits 7:0 of EEPROM data: write data	Notes
00h	10h	12h	sync. byte and end address
01h	00h	15h	VID lower byte
02h	01h	14h	VID upper byte
03h	08h	15h	SVID lower byte
04h	09h	14h	SVID upper byte
05h	02h	38h	DID lower byte
06h	03h	95h	DID upper byte
07h	0Ah	08h	SDID lower byte
08h	0Bh	95h	SDID upper byte
09h	40h	FFh	Enable UARTs 0-8
0Dh	C1h	00h	safety catch off for UART 0
0Eh	86h	10h	set IRSR=16
0Fh	87h	1Ch	set CPR=1Ch
10h	86h	00h	set IRSR=0
11h	C1h	80h	safety catch on

6. CLOCK/OSCILLATOR PINS

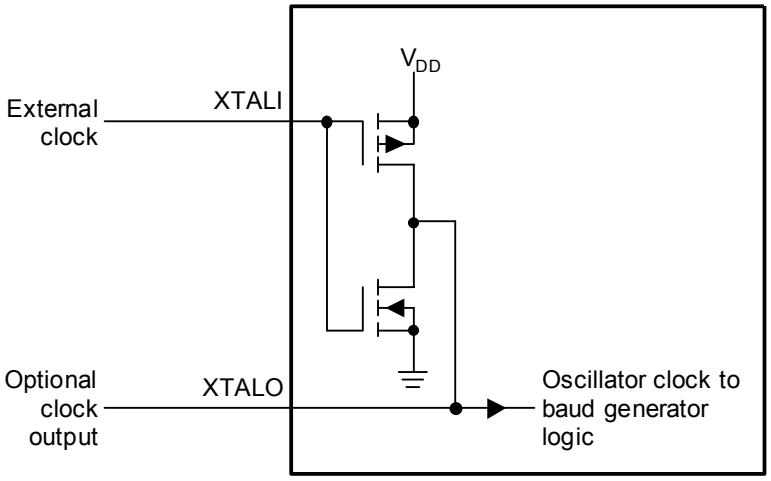
The OX16PCI958 provides a clock input and a logically inverted output suitable for driving a crystal oscillator as shown below:

Figure 5 OX16PCI958 Clock Provisions



Alternatively, the XTALI pin may be driven from an external clock signal, and the XTALO pin used as an optional clock output.

Figure 6 OX16PCI958 Using External Clock Input



7. OPERATING CONDITIONS

Table 32 Absolute Maximum Ratings

	Min	Max	Unit
DC Supply voltage	-0.3	7.0	V
Input voltage		VDD + 0.3	V
Input current		10	µA
Storage temperature	-55	150	ºC

7.1. Recommended Operating Conditions

	Min	Max	Unit
Supply voltage, VDD	4.5	5.5	V
Supply voltage, VDDP	3.0	5.5	V
Operating free-air temperature, TA	0	70	ºC
Junction temperature, TJ		150	ºC
Clock frequency (PCI clock, 'CLK')	0	331/3	MHz
Oscillator/clock frequency (UART clock, 'XTALI')	14.7	16	MHz

7.2. DC Characteristics

Table 33 CMOS Type Input, Supply at +5V ± 10%

Parameter	Min	Max	Unit
Input low voltage, VIL		0.3 x VDD	V
Input high voltage, VIH	0.7 x VDD		V
Input leakage (no pull-up)	-10	10	µA

Table 34 TTL Type Input, Supply at +5V ± 10%

Parameter	Min	Max	Unit
Input low voltage, VIL		0.8	V
Input high voltage, VIH	2.0		V
Input leakage (no pull-up)	-10	10	µA

Table 35 CMOS or TTL Type Output, Supply at +5V ± 10%

Parameter	Min	Max	Unit
Output low voltage, VOL (sinking rated current)	2.4		V
Output high voltage, VOH (sourcing rated current)		0.4	V
3-state output leakage current, IOZ	-10	10	µA

8. I/O ELECTRICAL & TIMING SPECIFICATIONS

The host interface timings comply with all requirements of PCI specifications.

In

Table 36, the columns have the following meanings:

I, O or B	I—input O—output B—bi-directional I/O
Drv Str	Output drive strength in mA
CMOS / TTL	Type of voltage thresholds used; see details in the following section
CTO	Clock-to-output timings, in ns. If only one number is given, this is the maximum
Ext Load	External load (in pF) used to verify output timings
ITC	Input-to-clock timings, in ns, i.e. setup time
PU/PD	Indicates whether internal pull-up or pull-down resistors are fitted. All pull-ups have a 100 kΩ nominal resistance, although they are not strictly ohmic in nature
Special	Particular I/O specifications, as follows: PCI Universal— I/O conforms to the PCI 5V signaling specification when the VDDP pins are at 5V, and conforms to the PCI 3.3V signalling specification when the VDDP pins are at 3.3V. IEEE 1284 level 2— I/O conforms to the "level 2" signalling specification defined in IEEE 1284-2000, when fitted with an external 22 Ω series resistor (all signals) and 1.2 kΩ pull-up resistors (input and bi-directional pins) Power/power voltage— power or I/O pin is connected to the 3.3/5V I/O group used for the PCI interface, or the 5V group used for the other I/Os

Table 36 Pin Electrical & Timing Characteristics

Pin No.	Pin Name	I, O or B	Drv Str (in mA)	CMOS / TTL	CTO / ns	Ext Load / pF	ITC / ns	PU/PD	Special	Signalling/ power voltage (in V)
1	AD23	B		CMOS	2-11		7	-	PCI Universal	3.3/5
2	VDDP									3.3/5
3	AD22	B		CMOS	2-11		7	-	PCI Universal	3.3/5
4	AD21	B		CMOS	2-11		7	-	PCI Universal	3.3/5
5	AD20	B		CMOS	2-11		7	-	PCI Universal	3.3/5
6	VSS									
7	AD19	B		CMOS	2-11		7	-	PCI Universal	3.3/5
8	AD18	B		CMOS	2-11		7	-	PCI Universal	3.3/5
9	AD17	B		CMOS	2-11		7	-	PCI Universal	3.3/5
10	AD16	B		CMOS	2-11		7	-	PCI Universal	3.3/5
11	VDDP									3.3/5
12	C/BE#2	I	-	CMOS	-	-	7	-	PCI Universal	3.3/5
13	FRAME#	I	-	CMOS	-	-	7	-	PCI Universal	3.3/5
14	IRDY#	I	-	CMOS	-	-	7	-	PCI Universal	3.3/5
15	VSS									
16	TRDY#	O		CMOS	2-11		-	-	PCI Universal	3.3/5
17	DEVSEL#	O		CMOS	2-11		-	-	PCI Universal	3.3/5
18	STOP#	O		CMOS	2-11		-	-	PCI Universal	3.3/5
19	PERR#	O		CMOS	2-11		-	-	PCI Universal	3.3/5
20	VDDP									3.3/5
21	SERR#	O		CMOS	2-11		-	-	PCI Universal	3.3/5
22	PAR	B		CMOS	2-11		7	-	PCI Universal	3.3/5
23	C/BE#1	I	-	CMOS	-	-	7	-	PCI Universal	3.3/5
24	VSS									
25	AD15	B		CMOS	2-11		7	-	PCI Universal	3.3/5
26	AD14	B		CMOS	2-11		7	-	PCI Universal	3.3/5
27	AD13	B		CMOS	2-11		7	-	PCI Universal	3.3/5
28	AD12	B		CMOS	2-11		7	-	PCI Universal	3.3/5
29	VDDP									3.3/5
30	AD11	B		CMOS	2-11		7	-	PCI Universal	3.3/5
31	AD10	B		CMOS	2-11		7	-	PCI Universal	3.3/5
32	AD9	B		CMOS	2-11		7	-	PCI Universal	3.3/5
33	VSS									
34	AD8	B		CMOS	2-11		7	-	PCI Universal	3.3/5
35	C/BE#0	I	-	CMOS	-	-	7	-	PCI Universal	3.3/5
36	AD7	B		CMOS	2-11		7	-	PCI Universal	3.3/5
37	AD6	B		CMOS	2-11		7	-	PCI Universal	3.3/5
38	VDDP									3.3/5
39	AD5	B		CMOS	2-11		7	-	PCI Universal	3.3/5
40	AD4	B		CMOS	2-11		7	-	PCI Universal	3.3/5
41	AD3	B		CMOS	2-11		7	-	PCI Universal	3.3/5
42	VSS									
43	AD2	B		CMOS	2-11		7	-	PCI Universal	3.3/5

Pin No.	Pin Name	I, O or B	Drv Str (in mA)	CMOS / TTL	CTO / ns	Ext Load / pF	ITC / ns	PU/PD	Special	Signalling/ power voltage (in V)
44	AD1	B		CMOS	2-11		7	-	PCI Universal	3.3/5
45	AD0	B		CMOS	2-11		7	-	PCI Universal	3.3/5
46	VDD									3.3/5
47	VDDP									3.3/5
48	TESTN	I	-	CMOS	-	-		-	global tri-state control	5
49	VSS	I	-	CMOS	-	-			Tie to GND	5
50	EEDIO	B		CMOS		12		PU		5
51	VSS									
52	EECK	O		CMOS		7		-		5
53	EECS	O		CMOS		7		-		5
54	RI7	B	14	TTL	36		20	PU	IEEE 1284 level 2	5
55	DTR7	O	2	TTL	36	17	-	-		5
56	VDD									5
57	CTS7	I	-	TTL	-		20	PU		5
58	SOUT7	O	14	TTL	36	17	-	-	IEEE 1284 level 2	5
59	RTS7	O	14	TTL	36	17	-	-	IEEE 1284 level 2	5
60	VSS									
61	SIN7	I	-	TTL	-		20	PU		5
62	DSR7	I	-	TTL	-		20	PU		5
63	DCD7	B	14	TTL	36		20	PU	IEEE 1284 level 2	5
64	RI6	B	14	TTL	36		20	PU	IEEE 1284 level 2	5
65	VDD									5
66	DTR6	O	2	TTL	36	17	-	-		5
67	CTS6	I	-	TTL	-		20	PU	IEEE 1284 level 2	5
68	SOUT6	O	14	TTL	36	17	-	-	IEEE 1284 level 2	5
69	VSS									
70	RTS6	O	14	TTL	36	17	-	-	IEEE 1284 level 2	5
71	SIN6	I	-	TTL	-		20	PU		5
72	DSR6	I	-	TTL	-		20	PU		5
73	DCD6	B	14	TTL	36		20	PU	IEEE 1284 level 2	5
74	VDD									5
75	RI5	B	14	TTL	36		20	PU	IEEE 1284 level 2	5
76	DTR5	O	2	TTL	36	17	-	-		5
77	VSS									
78	XTALI	I		*					Crystal oscillator output	5
79	XTALO	O		*					Crystal oscillator input	5
80	CTS5	I	-	TTL	-		20	PU		5
81	NC									
82	SOUT5	O	2	TTL	36	17	-	-		5
83	RTS5	O	2	TTL	36	17	-	-		5
84	VDD									5
85	SIN5	I	-	TTL	-		20	PU		5
86	DSR5	I	-	TTL	-		20	PU	IEEE 1284 level 2	5
87	DCD5	B	14	TTL	36		20	PU	IEEE 1284 level 2	5
88	VSS									

Pin No.	Pin Name	I, O or B	Drv Str (in mA)	CMOS / TTL	CTO / ns	Ext Load / pF	ITC / ns	PU/PD	Special	Signalling/ power voltage (in V)
89	RI4	B	14	TTL	36		20	PU	IEEE 1284 level 2	5
90	DTR4	O	2	TTL	36	17	-	-		5
91	CTS4	I	-	TTL	-		20	PU	IEEE 1284 level 2	5
92	SOUT4	O	2	TTL	36	17	-	-		5
93	VDD									5
94	RTS4	O	2	TTL	36	17	-	-		5
95	SIN4	I	-	TTL	-		20	PU	IEEE 1284 level 2	5
96	DSR4	I	-	TTL	-		20	PU	IEEE 1284 level 2	5
97	VSS									
98	DCD4	B	14	TTL	36		20	PU	IEEE 1284 level 2	5
99	DCD0	I	-	TTL	-		20	PU		5
100	DSR0	I	-	TTL	-		20	PU		5
101	SIN0	I	-	TTL	-		20	PU		5
102	VDD									5
103	RTS0	O	2	TTL	36	17	-	-		5
104	SOUT0	O	2	TTL	36	17	-	-		5
105	CTS0	I	-	TTL	-		20	PU		5
106	VSS									
107	DTR0	O	2	TTL	36	17	-	-		5
108	RI0	I	-	TTL	-		20	PU		5
109	DCD1	I	-	TTL	-		20	PU		5
110	DSR1	I	-	TTL	-		20	PU		5
111	VDD									5
112	SIN1	I	-	TTL	-		20	PU		5
113	RTS1	O	2	TTL	36	17	-	-		5
114	SOUT1	O	2	TTL	36	17	-	-		5
115	VSS									
116	CTS1	I	-	TTL	-		20	PU		5
117	DTR1	O	2	TTL	36	17	-	-		5
118	RI1	I	-	TTL	-		20	PU		5
119	DCD2	I	-	TTL	-		20	PU		5
120	VDD									5
121	DSR2	I	-	TTL	-		20	PU		5
122	SIN2	I	-	TTL	-		20	PU		5
123	RTS2	O	2	TTL	36	17	-	-		5
124	VSS									
125	SOUT2	O	2	TTL	36	17	-	-		5
126	CTS2	I	-	TTL	-		20	PU		5
127	DTR2	O	2	TTL	36	17	-	-		5
128	RI2	I	-	TTL	-		20	PU		5
129	VDD									5
130	DCD3	I	-	TTL	-		20	PU		5
131	DSR3	I	-	TTL	-		20	PU		5
132	SIN3	I	-	TTL	-		20	PU		5
133	RTS3	O	2	TTL	36	17	-	-		5

Pin No.	Pin Name	I, O or B	Drv Str (in mA)	CMOS / TTL	CTO / ns	Ext Load / pF	ITC / ns	PU/PD	Special	Signalling/ power voltage (in V)
134	VSS									
135	SOUT3	O	2	TTL	36	17	-	-		5
136	CTS3	I	-	TTL	-		20	PU		5
137	DTR3	O	2	TTL	36	17	-	-		5
138	VDD									5
139	RI3	I	-	TTL	-		20	PU		5
140	LINEDRIVER_EN	O	6	CMOS	36	17	-	-		5
141	VDD									3.3/5
142	VSS									
143	INTA#	O		CMOS			-	-	PCI Universal	3.3/5
144	RST#	I	-	CMOS	-	-	-	-	PCI Universal	3.3/5
145	CLK	I	-	CMOS	-	-	7	-	PCI Universal	3.3/5
146	PME#	O		CMOS			-	-	PCI Universal	3.3/5
147	VDDP									3.3/5
148	AD31	B		CMOS	2-11		7	-	PCI Universal	3.3/5
149	AD30	B		CMOS	2-11		7	-	PCI Universal	3.3/5
150	AD29	B		CMOS	2-11		7	-	PCI Universal	3.3/5
151	VSS									
152	AD28	B		CMOS	2-11		7	-	PCI Universal	3.3/5
153	AD27	B		CMOS	2-11		7	-	PCI Universal	3.3/5
154	AD26	B		CMOS	2-11		7	-	PCI Universal	3.3/5
155	AD25	B		CMOS	2-11		7	-	PCI Universal	3.3/5
156	VDDP									3.3/5
157	AD24	B		CMOS	2-11		7	-	PCI Universal	3.3/5
158	VSS									
159	C/BE#3	I	-	CMOS	-	-	7	-	PCI Universal	3.3/5
160	IDSEL	I	-	CMOS	-	-	7	-	PCI Universal	3.3/5

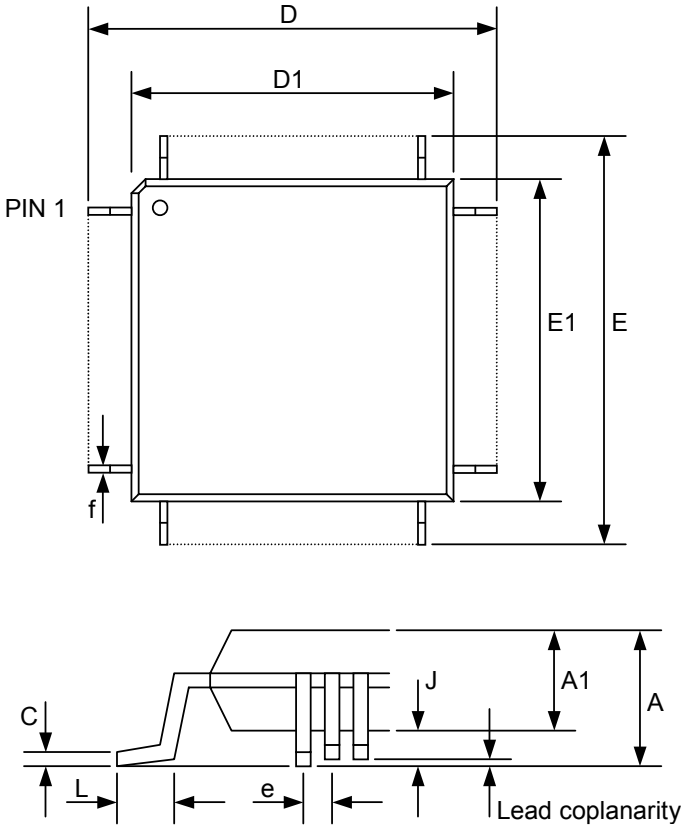
Note:

Although the PME# pin is an open-collector output, it is not suitable for direct connection to the PCI bus. The PME# pin must be isolated from the host system when the OX16PCI958 power source is absent, so that it does not cause unwanted wake events. See section 7 of the PCI Bus Power Management Interface Specification, revision 1.1, for guidance on implementing the isolation.

9. PACKAGE INFORMATION

The package is a standard 160-pin QFP package (JEDEC ref MS-022) with 0.65 mm lead pitch and a 4.1 mm max height from PCB, as shown in Figure 7.

Figure 7 OX16PCI958 Package



	Min	Max	Basic	Unit
A		4.10		mm
A1	3.20	3.60		mm
C			0.23	mm
D			31.20	mm
D1			28.00	mm
E			31.20	mm
E1			28.00	mm
J	0.25	0.50		mm
L	0.73	1.03		mm
e			0.65	mm
f	0.22	0.40		mm
Lead coplanarity		0.10		mm

Plastic body dimensions do not include flash or protrusion, max allowable 0.25 mm per side.

10. GLOSSARY

BSC Basic spacing between centres

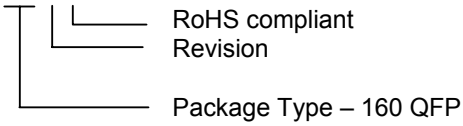
QFP Quad Flat Pack

RFU Reserved for future use: register bits described as RFU should be cleared during writes, and ignored during reads. They will be clear when read, but for future compatibility this should not be assumed.

RO Read-only

11. ORDERING INFORMATION

OX16PCI958-PQAG



12. CONTACT DETAILS

Oxford Semiconductor Ltd.

25 Milton Park
Abingdon
Oxfordshire
OX14 4SH
United Kingdom

Telephone:	+44 (0)1235 824900
Fax:	+44 (0)1235 821141
Sales e-mail:	sales@oxsemi.com
Tech support e-mail:	support@oxsemi.com
Web site:	http://www.oxsemi.com