OX12PCI840

Integrated Parallel Port and PCI interface

FEATURE

- IEEE1284 SPP/EPP/ECP parallel port
- Single function target PCI controller, fully PCI 2.2 and PCI Power Management 1.0 compliant
- 2 multi-purpose IO pins which can be configured as interrupt input pins
- Can be reconfigured using optional non-volatile configuration memory (EEPROM)
- 5.0V operation
- 100 pin PQFP package

DESCRIPTION

The OX12PCI840 is a single chip solution for PCI-based parallel expansion add-in cards. It is a single function PCI device.

For legacy applications the PCI resources are arranged so that the parallel port can be located at standard I/O addresses.

The efficient 32-bit, 33MHz target-only PCI interface is compliant with version 2.2 of the PCI Bus Specification and version 1.0 of PCI Power Management Specification. For full flexibility, all the default register values can be overwritten using an optional Microwire™ serial EEPROM.

The OX12PCI840 provides an IEEE1284 EPP/ECP parallel port which fully supports the existing Centronics interface.

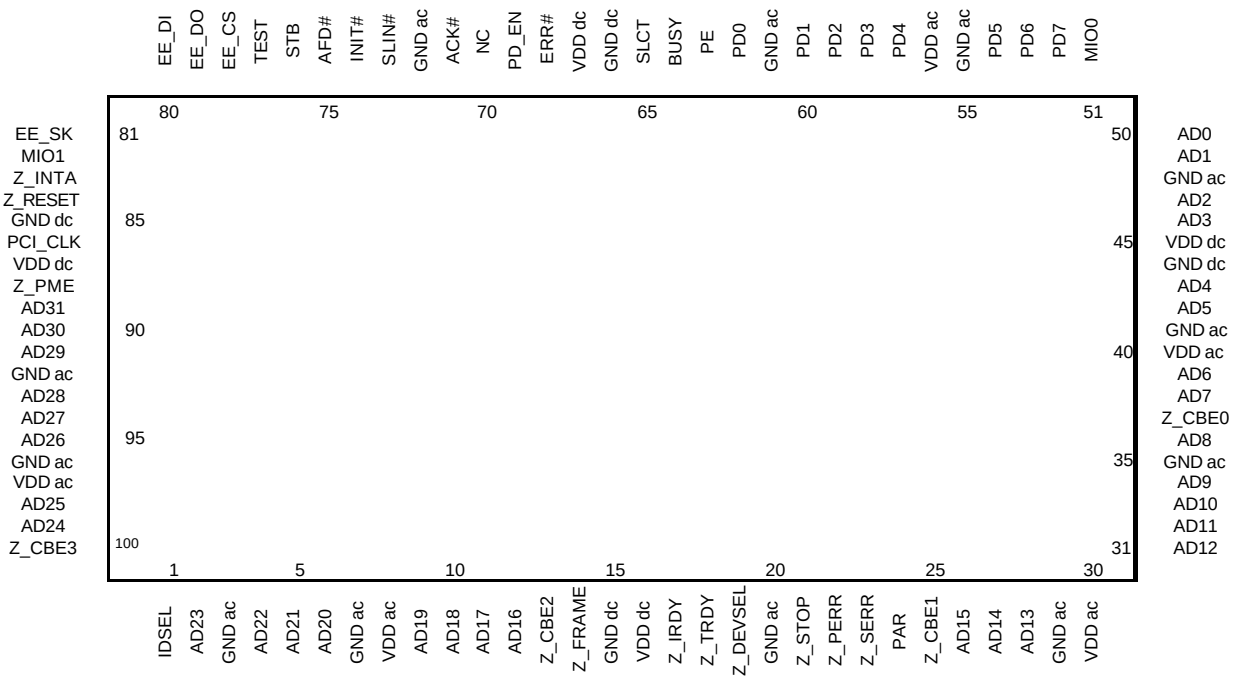
Oxford Semiconductor Ltd.
25 Milton Park, Abingdon, Oxon, OX14 4SH, UK
Tel: +44 (0)1235 824900 Fax: +44(0)1235 821141

OX12PCI840 1.2 – Dec 2001
Part No. OX12PCI840-PQC-A

CONTENTS

1 PIN INFORMATION

100 pin QFP



2 PIN DESCRIPTIONS

Pin Numbers	Dir[1]	Name	Description
PCI interface			
89,90,91,93,94,95,98,99,2,4,5,6,9, 10,11,12,26,27,28,31,32,33,34,36, 38,39,42,43,46,47,49,50	P_I/O	AD[31:0]	Multiplexed PCI Address/Data bus
100,13,25,37	P_I	C/BE[3:0]#	PCI Command/Byte enable
86	P_I	CLK	PCI system clock
14	P_I	FRAME#	Cycle Frame
19	P_O	DEVSEL#	Device Select
17	P_I	IRDY#	Initiator ready
18	P_O	TRDY#	Target ready
21	P_O	STOP#	Target Stop request
24	P_I/O	PAR	Parity
23	P_O	SERR#	System error
22	P_I/O	PERR#	Parity error
1	P_I	IDSEL	Initialisation device select
84	P_I	RST#	PCI system reset
83	P_OD	INTA#	PCI interrupt
88	P_OD	PME#	Power management event

Pin Numbers	Dir[1]	Name	Description
Parallel port			
71	I	ACK#	Acknowledge (SPP mode). ACK# is asserted (low) by the peripheral to indicate that a successful data transfer has taken place.
	I	INTR#	Identical function to ACK# (EPP mode).
63	I	PE	Paper Empty. Activated by printer when it runs out of paper.
64	I	BUSY	Busy (SPP mode). BUSY is asserted (high) by the peripheral when it is not ready to accept data
	I	WAIT#	Wait (EPP mode). Handshake signal for interlocked IEEE 1284 compliant EPP cycles.
73	OD	SLIN#	Select (SPP mode). Asserted by host to select the peripheral
	O	ADDRSTB#	Address strobe (EPP mode) provides address read and write strobe
65	I	SLCT	Peripheral selected. Asserted by peripheral when selected.
68	I	ERR#	Error. Held low by the peripheral during an error condition.
74	OD	INIT#	Initialise (SPP mode). Commands the peripheral to initialise.
	O	INIT#	Initialise (EPP mode). Identical function to SPP mode.
75	OD	AFD#	Auto Feed (SPP mode, open-drain)
	O	DATASTB#	Data strobe (EPP mode) provides data read and write strobe
76	OD	STB#	Strobe (SPP mode). Used by peripheral to latch data currently available on PD[7:0]
	O	WRITE#	Write (EPP mode). Indicates a write cycle when low and a read cycle when high
52,53,54,57, 58,59,60,62	I/O	PD[7:0]	Parallel data bus

Pin Numbers	Dir[1]	Name	Description
Multi-purpose & External interrupt pins			
82, 51	I/O	MIO[1:0]	Multi-purpose I/O pins. Can drive high or low, or assert a PCI interrupt
EEPROM pins			
81	O	EE_CK	EEPROM clock
78	O	EE_CS	EEPROM active-high Chip Select
80	IU	EE_DI	EEPROM data in. When the serial EEPROM is connected, this pin should be pulled up using 1-10k resistor. When the EEPROM is not used the internal pull-up is sufficient.
79	O	EE_DO	EEPROM data out.
Miscellaneous pins			
77	I	TEST	Test Pin : should be held low at all times
Power and ground[2]			
8,30,40,56,97	V	AC VDD	Supplies power to output buffers in switching (AC) state
16,45,67,87	V	DC VDD	Power supply. Supplies power to core logic, input buffers and output buffers in steady state
3,7,20,29,35,41,48,55,61,72,92,96	G	AC GND	Supplies GND to output buffers in switching (AC) state
15,44,66,85	G	DC GND	Ground (0 volts). Supplies GND to core logic, input buffers and output buffers in steady state

Table 1: Pin Descriptions

Note 1: Direction key:

I	Input	P_I	PCI input
ID	Input with internal pull-down	P_O	PCI output
O	Output	P_I/O	PCI bi-directional
I/O	Bi-directional	P_OD	PCI open drain
OD	Open drain		
NC	No connect	G	Ground
Z	High impedance	V	5.0V power

Note 2: Power & Ground

There are two GND and two VDD rails internally. One set of rails supply power and ground to output buffers while in switching state (called AC power) and another rail supply the core logic, input buffers and output buffers in steady-state (called DC rail). The rails are not connected internally. This precaution reduces the effects of simultaneous switching outputs and undesirable RF radiation from the chip. Further precaution is taken by segmenting the GND and VDD AC rails to isolate the PCI and Local Bus pins.

3 CONFIGURATION & OPERATION

The OX12PCI840 is a single function, target-only PCI device, compliant with the PCI Local Bus Specification, Revision 2.2 and PCI Power Management Specification, Revision 1.0.

The OX12PCI840 is configured by system start-up software during the bootstrap process that follows bus reset. The system scans the bus and reads the vendor and device identification codes from any devices it finds. It then loads device-driver software according to this information and configures the I/O, memory and interrupt resources. Device drivers can then access the functions at the assigned addresses in the usual fashion, with the improved data throughput provided by PCI.

There are a set of Local configuration registers that can be used to enable signals and interrupts, and configure timings. These can be set up by drivers or from the EEPROM.

All registers default after reset to suitable values for typical applications. However, all identification, control and timing registers can be redefined using an optional serial EEPROM. As an additional enhancement, the EEPROM can be used to program the parallel port, allowing pre-configuration, without requiring driver changes.

4 PCI TARGET CONTROLLER

4.1 Operation

The OX12PCI840 responds to the following PCI transactions:-

- Configuration access: The OX12PCI840 responds to type 0 configuration reads and writes if the IDSEL signal is asserted and the bus address is selecting the configuration registers for function 0. The device will respond to the configuration transaction by asserting DEVSEL#. Data transfer then follows. Any other configuration transaction will be ignored by the OX12PCI840.

- IO reads/writes: The address is compared with the addresses reserved in the I/O Base Address Registers (BARs). If the address falls within one of the assigned ranges, the device will respond to the IO transaction by asserting DEVSEL#. Data transfer follows this address phase. Only byte accesses are possible to the function BARs (excluding the local configuration registers for which WORD, DWORD access is supported). For IO accesses to these regions, the controller compares AD[1:0] with the byte-enable signals as defined in the PCI specification. The access is always completed; however if the correct BE signal is not present the transaction will have no effect.

- Memory reads/writes: These are treated in the same way as I/O transactions, except that the memory ranges are used. Memory access to single-byte regions is always expanded to DWORDs in the OX12PCI840. In other words, OX12PCI840 reserves a DWORD per byte in single-byte regions. The device allows the user to define the active byte lane using LCC[4:3] so that in Big-Endian systems the hardware can swap the byte lane automatically. For Memory mapped access in single-byte regions, the OX12PCI840 compares the asserted byte-enable with the selected byte-lane in LCC[4:3] and completes the operation if a match occurs, otherwise the access will complete normally on the PCI bus, but it will have no effect on either the parallel port or the local bus controller.

- All other cycles (64-bit, special cycles, reserved encoding etc.) are ignored.

The OX12PCI840 will complete all transactions as disconnect-with-data, i.e. the device will assert the STOP# signal alongside TRDY#, to ensure that the Bus Master does not continue with a burst access. The exception to this is Retry, which will be signalled in response to any access while the OX12PCI840 is reading from the serial EEPROM.

The OX12PCI840 performs medium-speed address decoding as defined by the PCI specification. It asserts the DEVSEL# bus signal two clocks after FRAME# is first sampled low on all bus transaction frames which address the chip. Fast back-to-back transactions are supported by the OX12PCI840 as a target, so a bus master can perform faster sequences of write transactions to the parallel port when an inter-frame turn-around cycle is not required.

The device supports any combination of byte-enables to the PCI Configuration Registers and the Local Configuration registers (see Base Address 2 and 3). If a byte-enable is not asserted, that byte is unaffected by a write operation and undefined data is returned upon a read.

The OX12PCI840 performs parity generation and checking on all PCI bus transactions as defined by the standard. If a parity error occurs during the PCI bus address phase, the device will report the error in the standard way by asserting the SERR# bus signal. However if that address/command combination is decoded as a valid access, it will still complete the transaction as though the parity check was correct.

The OX12PCI840 does not support any kind of caching or data buffering, other than that in the parallel port interface itself.

4.2 Configuration space

The OX12PCI840 is a single function device, with one configuration space. All required fields in the standard header are implemented, plus the Power Management Extended Capability register set. The format of the configuration space is shown in Table 2 overleaf.

In general, writes to any registers that are not implemented are ignored, and all reads from unimplemented registers return 0.

4.2.1 PCI Configuration Space Register map

Configuration Register Description				Offset Address
31	16	15	0	
Device ID		Vendor ID		00h
Status		Command		04h
Class Code			Revision ID	08h
BIST [1]	Header Type	Reserved	Reserved	0Ch
Base Address Register 0 (BAR0) - Function in I/O space				10h
Base Address Register 1 (BAR 1) - Function in I/O space				14h
Base Address Register 2 (BAR 2) – Local Configuration Registers in IO space				18h
Base Address Register 3 (BAR3) – Local Configuration Registers in Memory space				1Ch
Base Address Register 4 (BAR4) – Function in Memory Space				20h
Reserved				24h
Reserved				28h
Subsystem ID		Subsystem Vendor ID		2Ch
Reserved				30h
Reserved			Cap_Ptr	34h
Reserved				38h
Reserved	Reserved	InterruptPin	Interrupt Line	3Ch
Power Management Capabilities (PMC)		Next Ptr	Cap_ID	40h
Reserved	Reserved	PMC Control/Status Register (PMCSR)		44h

Table 2: PCI Configuration space

Register name	Reset value	Program read/write	
		EEPROM	PCI
Vendor ID	0x1415	W	R
Device ID	0x8403	W	R
Command	0x0000	-	R/W
Status	0x0290	W(bit 4)	R/W
Revision ID	0x00	-	R
Class code	0x070103	W	R
Header type	0x00	-	R
BAR 0	0x00000001	-	R/W
BAR 1	0x00000001	-	R/W
BAR 2	0x00000001	-	R/W
BAR 3	0x00000000	-	R/W
BAR 4	Reserved	-	R/W
Subsystem VID	0x1415	W	R
Subsystem ID	0x0001	W	R
Cap ptr.	0x40	-	R
Interrupt line	0x00	-	R/W
Interrupt pin	0x01	W	R
Cap ID	0x01	-	R
Next ptr.	0x00	-	R
PM capabilities	0x6C01	W	R
PMC control/ status register	0x0000	-	R/W

Table 3: PCI configuration space default values

4.3 Accessing logical functions

Access to the parallel port is achieved via standard I/O and memory mapping, at addresses defined by the Base Address Registers (BARs) in configuration space. The BARs are configured by the system to allocate blocks of I/O and memory space to the logical function, according to the size required by the function. The addresses allocated can then be used to access the function. The mapping of these BARs is shown in Table 4.

BAR	Mapping
0	Parallel port base registers (I/O mapped)
1	Parallel port extended registers (I/O mapped)
2	Local configuration registers (I/O mapped
3	Local configuration registers (memory mapped)
4	Unused
5	Unused

Table 4: Base Address Register definition

4.3.1 PCI access to parallel port

Access to the port works with two I/O BARs corresponding to the two sets of registers defined to operate an IEEE1284 ECP/EPP and bi-directional Parallel Port.

The user can change the I/O space block size of BAR0 or BAR by over-writing the default values using the serial EEPROM (see section 4.4).

Legacy parallel ports expect the upper register set to be mapped 0x400 above the base block, therefore if the BARs are fixed with this relationship, generic parallel port drivers can be used to operate the device in all modes.

Example: BAR0 = 0x00000379 (8 bytes at address 0x378)
BAR1 = 0x00000779 (8 bytes at address 0x778)

If this relationship is not used, custom drivers will be needed.

4.4 Accessing Local configuration registers

The local configuration registers are a set of device specific registers which can always be accessed. They are mapped to the I/O and memory addresses set up in BAR2 and BAR3, with the offsets defined for each register. I/O or memory accesses can be byte, word or dword accessed, however on little-endian systems such as Intel 80x86 the byte order will be reversed.

4.4.1 Local Configuration and Control register 'LCC' (Offset 0x00)

This register defines control of ancillary functions such as Power Management, endian selection and the serial EEPROM. The individual bits are described below.

Bits	Description	Read/Write		Reset
		EEPROM	PCI	
2:0	Reserved			000
4:3	Endian Byte-Lane Select for memory access to parallel port 00 = Select Data[7:0] 10 = Select Data[23:16] 01 = Select Data[15:8] 11 = Select Data[31:24] Memory access to OX12PCI840 is always DWORD aligned. When accessing the parallel port, this option selects the active byte lane. As both PCI and PC architectures are little endian, the default value will be used by systems, however, some non-PC architectures may need to select the byte lane.	W	RW	00
7:5	Power-down filter time. These bits define a value of an internal filter time for power-down interrupt request in power management circuitry in Function0. Once Function0 is ready to go into power down mode, OX12PCI840 will wait for the specified filter time and if Function0 is still in power-down request mode, it can assert a PCI interrupt (see section 4.6). 000 = power-down request disabled 010 = 129 seconds 001 = 4 seconds 011 = 518 seconds 1XX = Immediate	W	RW	000
10:8	Reserved: Power management test bits. The device driver must write zero to these bits	-	R	000
22:11	Reserved.	-	R	0000h
23	Parallel port Input (glitch) filters. Enabled when '1'	W	RW	0
24	EEPROM Clock. For PCI read or write to the EEPROM , toggle this bit to generate an EEPROM clock (EE_CK pin).	-	RW	0
25	EEPROM Chip Select. When 1 the EEPROM chip-select pin EE_CS is activated (high). When 0 EE_CS is de-active (low).	-	RW	0
26	EEPROM Data Out. For writes to the EEPROM, this output bit is the input-data of the EEPROM. This bit is output on EE_DO and clocked into the EEPROM by EE_CK.	-	RW	0
27	EEPROM Data In. For reads from the EEPROM, this input bit is the output-data of the EEPROM connected to EE_DI pin.	-	R	X
28	EEPROM Valid. A 1 indicates that a valid EEPROM program is present	-	R	X
29	Reload configuration from EEPROM. Writing a 1 to this bit re-loads the configuration from EEPROM. This bit is self-clearing after EEPROM read	-	RW	0
30	Reserved	-	R	0
31	Reserved	-	R	0

4.4.2 Multi-purpose I/O Configuration register 'MIC' (Offset 0x04)

This register configures the operation of the multi-purpose I/O pins 'MIO[1:0] as follows.

Bits	Description	Read/Write		Reset
		EEPROM	PCI	
1:0	MIO0 Configuration Register 00 -> MIO0 is a non-inverting input pin 01 -> MIO0 is an inverting input pin 10 -> MIO0 is an output pin driving '0' 11 -> MIO0 is an output pin driving '1'	W	RW	00
3:2	MIO1 Configuration Register 00 -> MIO1 is a non-inverting input pin 01 -> MIO1 is an inverting input pin 10 -> MIO1 is an output pin driving '0' 11 -> MIO1 is an output pin driving '1'	W	RW	00
4	MIO0_PME Enable. A value of '1' enables MIO0 pin to set the PME_Status in PMCSR register, and hence assert the PME# pin if enabled. A value of '0' disables MIO0 from setting the PME_Status bit.	W	RW	0
5	MIO1_PME Enable. A value of '1' enables MIO1 pin to set the PME_Status in PMCSR register, and hence assert the PME# pin if enabled. A value of '0' disables MIO1 from setting the PME_Status bit.	W	RW	0
6	MIO0 Power Down Request: A '1' enables MIO0 to control the power down request filter.	W	RW	0
7	MIO1 Power Down Request: A '1' enables MIO1 to control the power down request filter.	W	RW	0
31:8	Reserved	-	R	00

4.4.3 Local Bus Timing Parameter register 1 'LT1' (Offset 0x08):

The Local Bus Timing Parameter registers (LT1 and LT2) define the operation and timing parameters used by the internal local bus (that connects to the parallel port). **It is envisaged that these should not need to be changed by the user**. The timing parameters are programmed in 4-bit registers to define the assertion/de-assertion of the Local Bus control signals. The values programmed in these registers defines the number of PCI clock cycles after a Reference Cycle when the events occur, where the reference Cycle is defined as two clock cycles after the master asserts the IRDY# signal. The timings refer to I/O or Memory mapped accesses.

Bits	Description	Read/Write		Reset
		EEPROM	PCI	
3:0	Read Cycle start	W	RW	0h
7:4	Read Cycle end	W	RW	2h
11:8	Write Cycle start	W	RW	0h
15:12	Write Cycle end	W	RW	2h
19:16	Read Assertion	W	RW	1h
23:20	Read De-assertion	W	RW	2h
27:24	Write Assertion	W	RW	1h
31:28	Write De-assertion	W	RW	2h

Note 1: Only values in the range of 0h to Ah (0 -10 decimal) are valid. Other values are reserved. See notes in the following page.

4.4.4 Local Bus Timing Parameter/Bar sizing register 2 'LT2' (Offset 0x0C):

Bits	Description	Read/Write		Reset
		EEPROM	PCI	
3:0	Reserved: 0h must be written to this location	W	RW	0h
7:4	Reserved: Fh must be written to this location	W	RW	Fh
11:8	Reserved: 2h must be written to this location	W	RW	2h
15:12	Reserved: 0h must be written to this location	W	RW	0h
19:16	Reserved.	-	R	0h
22:20	IO Space Block Size of BAR0 000 = Reserved 100 = 32 Bytes 001 = 4 Bytes 101 = 64 Bytes 010 = 8 Bytes 110 = 128 Bytes 011 = 16 Bytes 111 = 256 Bytes	W	R	'010'
23	Reserved	-	R	0h
26:24	IO Space Block Size of BAR1 000 = Reserved 100 = 32 Bytes 001 = 4 Bytes 101 = 64 Bytes 010 = 8 Bytes 110 = 128 Bytes 011 = 16 Bytes 111 = 256 Bytes	W	R	'001'
28::27	Reserved	-	R	000
29	Reserved:0 must be written to this location	-	RW	0
31:30	Reserved:00 must be written to this location	W	RW	00

4.4.5 Global Interrupt Status and Control Register 'GIS' (Offset 0x10)

Bits	Description	Read/Write		Reset
		EEPROM	PCI	
1:0	Reserved	-	R	0x0h
2	MIO0 This bit reflects the state of the internal MIO[0]. The internal MIO[0] reflects the non-inverted or inverted state of MIO0 pin.	-	R	X
3	MIO1 This bit reflects the state of the internal MIO[0]. The internal MIO[0] reflects the non-inverted or inverted state of MIO0 pin.	-	R	X
17-4	Reserved	-	R	0
18	MIO0 INTA enable When set (1) allows MIO0 to assert a PCI interrupt on the INTA line. State of MIO0 that causes an interrupt is dependant upon the polarity set by MIC(1:0)	W	RW	0
19	MIO1 INTA enable When set (1) allows MIO1 to assert a PCI interrupt on the INTA line. State of MIO1 that causes an interrupt is dependant upon the polarity set by MIC(3:2)	W	RW	0
20	Power-down Interrupt This is a sticky bit. When set, it indicates a power-down request issued and would normally have asserted a PCI interrupt if bit 21 was set (see section 7.9). Reading this bit clears it.	-	R	X
21	Power-down interrupt enable. When '1' a power down request is allowed to generate an interrupt.	W	RW	0
22	Parallel port interrupt status	-	R	0
23	Parallel port interrupt enable	W	RW	1
31:24	Reserved	-	R	000h

4.5 PCI Interrupts

Interrupts in PCI systems are level-sensitive and can be shared. There are three sources of interrupt in the OX12PCI840, two from Multi-Purpose IO pins (MIO1 to MIO0) and one from the parallel port.

All interrupts are routed to the PCI interrupt pin INTA#. The default routing asserts Function0 interrupts on INTA#. This default routing may be modified (to disable interrupts) by writing to the Interrupt Pin field in the configuration registers using the serial EEPROM facility. The Interrupt Pin field is normally considered a hard-wired read-only value in PCI. It indicates to system software which PCI interrupt pin (if any) is used by a function. The interrupt pin may only be modified using the serial EEPROM facility, and card developers must not set any value which violates the PCI specification. Note that OX12PCI840 only has one PCI interrupt pin - INTA#. If in doubt, the default routings should be used. Table 5 relates the Interrupt Pin field to the device pin used.

Interrupt Pin	Device Pin used
0	None
1	INTA#
2 to 255	Reserved

Table 5: 'Interrupt pin' definition

During the system initialisation process and PCI device configuration, system-specific software reads the interrupt pin field to determine which (if any) interrupt pin is used by the function. It programmes the system interrupt router to logically connect this PCI interrupt pin to a system-specific interrupt vector (IRQ). It then writes this routing information to the Interrupt Line field in the function's PCI configuration space. Device driver software must then hook the interrupt using the information in the Interrupt Line field.

Interrupt status for all sources of interrupt is available using the GIS register in the Local Configuration Register set, which can be accessed using I/O or Memory accesses.

All interrupts can be enabled / disabled individually using the GIS register set in the Local configuration registers. When an MIO pin is enabled, an external device can assert a PCI interrupt by driving that pin. The sense of the MIO external interrupt pins (active-high or active-low) is defined in the MIC register. The parallel port can also assert an interrupt.

4.6 Power Management

The OX12PCI840 is compliant with PCI Power Management Specification Revision 1.0. The function implements its own set of Power Management registers and supports the power states D0, D2 and D3. Power management is accomplished by power-down and power-up requests, asserted via interrupts and the PME# pin respectively. The PME# pin is de-asserted when the sticky PME_Status bit is cleared.

Power-down request is not defined by Power Management 1.0. It is a device-specific feature and requires a bespoke device driver implementation. The device driver can either implement the power-down itself or use a special interrupt and power-down features offered by the device to determine when the device is ready for power-down.

The PME# pin can, in certain cases, activate the PME# signal when power is removed from the device, which will cause the PC to wake up from Low-power state D3(cold). To ensure full cross-compatibility with system board implementations, use of an isolator FET is recommended. If Power Management capabilities are not required, the PME# pin can be treated as no-connect.

4.6.1 Power Management using MIO

The power-down request for the Parallel port is application-dependent. Provided that the necessary enables have been set in the local registers, the multi-purpose I/O pins MIO(1:0) can be used to generate a powerdown request. The MIO state that governs powerdown is the inverse of the MIO state that asserts the INTA line (if that option were to be enabled). This means that when the external device is not interrupting it will begin the powerdown cycle. For greater flexibility in the generation of the power down request,, a powerdown filter is also available to ensure that the relevant MIO pins remain stable for a selectable period before a powerdown request is issued.

Function0 implements the PCI Power Management power-states D0, D2 and D3. Whenever the device driver changes the power-state to state D2 or D3, Function0 takes the following actions:-

- The PCI interrupt for Function0 is disabled.
- Access to I/O or Memory BARs of Function0 is disabled.

However, access to the configuration space is still enabled. The device driver can optionally assert/de-assert any of its selected (design dependent) MIO pins to switch off VCC, disable other external clocks, or activate shut-down modes to any external devices.

Function0 can issue a wake up request by using the MIO pins. When MIC[7] or MIC[6] is set, rising or falling edge of the relevant MIO pin will cause Function0 to issue a wake up request by setting PME_Status = (PMCSR[15]), if it is enabled by PMCSR[8] of Function0. PME_Status is a sticky bit which will be cleared by writing a '1' to it. After a wake up event is signalled, the device driver is expected to return the function to the D0 power-state.

5 BI-DIRECTIONAL PARALLEL PORT

5.1 Operation and Mode selection

The OX12PCI840 offers a compact, low power, IEEE-1284 compliant host-interface parallel port, designed to interface to many peripherals such as printers, scanners and external drives. It supports compatibility modes, SPP, NIBBLE, PS2, EPP and ECP modes. The register set is compatible with the Microsoft® register definition. The system can access the parallel port via two blocks of I/O space; BAR0 (8 bytes) contains the address of the basic parallel port registers, BAR1 (4 bytes) contains the address of the upper registers. These are referred to as the 'lower block' and 'upper block' in this section. If the upper block is located at an address 0x400 above the lower block, generic PC device drivers can be used to configure the port, as the addressable registers of legacy parallel ports always have this relationship. If not, a custom driver will be needed.

5.1.1 SPP mode

SPP (output-only) is the standard implementation of a simple parallel port. In this mode, the PD lines always drive the value in the PDR register. All transfers are done under software control. Input must be performed in nibble mode.

Generic device driver-software may use the address in I/O space encoded in BAR0 of function 1 to access the parallel port. The default configuration allocates 8 bytes to BAR0 in I/O space.

5.1.2 PS2 mode

This mode is also referred to as bi-directional or compatible parallel port. In this mode, directional control of the PD lines is possible by setting & clearing DCR[5]. Otherwise operation is similar to SPP mode.

5.1.3 EPP mode

To use the Enhanced Parallel Port 'EPP' the mode bits (ECR[7:5]) must be set to '100'. The EPP address and data port registers are compatible with the IEEE 1284 definition. A write or read to one of the EPP port registers is passed through the parallel port to access the external peripheral. In EPP mode, the STB#, INIT#, AFD# AND SLIN# pins change from open-drain outputs to active push-pull (totem pole) drivers (as required by IEEE 1284) and the pins ACK#, AFD#, BUSY, SLIN# and STB# are redefined as INTR#, DATASTB#, WAIT#, ADDRSTB# and WRITE# respectively.

An EPP port access begins with the host reading or writing to one of the EPP port registers. The device automatically buffers the data between the I/O registers and the parallel port depending on whether it is a read or a write cycle. When the peripheral is ready to complete the transfer it takes the WAIT# status line high. This allows the host to complete the EPP cycle.

If a faulty or disconnected peripheral failed to respond to an EPP cycle the host would never see a rising edge on WAIT#, and subsequently lock up. A built-in time-out facility is provided in order to prevent this from happening. It uses an internal timer which aborts the EPP cycle and sets a flag in the PSR register to indicate the condition. When the parallel port is not in EPP mode the timer is switched off to reduce current consumption. The host time-out period is 10µs as specified with the IEEE-1284 specification.

The register set is compatible with the Microsoft® register definition. Assuming that the upper block is located 400h above the lower block, the registers are found at offset 000-007h and 400-402h.

5.1.4 ECP mode

The Extended Capabilities Port 'ECP' mode is entered when ECR[7:5] is set to '011'. ECP mode is compatible with Microsoft® register definition of ECP, and IEEE-1284 bus protocol and timing. This implementation of the ECP port supports the optional decompression of received compressed data, but does not compress transmit data.

Assuming that the upper block is located 400h above the lower block, the registers are found at offset 000-007h and 400-402h.

5.2 Parallel port interrupt

The parallel port interrupt is asserted on INTA#. It is enabled by setting DCR[4]. When DCR[4] is set, an interrupt is asserted on the rising edge of the ACK# (INTR#) pin and held until the status register is read, which resets the INT# status bit (DSR[2]).

5.3 Register Description

The parallel port registers are described below. (NB it is assumed that the upper block is placed 400h above the lower block).

Register Name	Address Offset	R/W	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPP (Compatibility Mode) Registers										
PDR	000h	R/W	Parallel Port Data Register							
ecpAFifo	000h	R/W	ECP FIFO : Address / RLE							
DSR (EPP mode)	001h	R	nBUSY	ACK#	PE	SLCT	ERR#	INT#	1	Timeout
(Other modes)	001h	R	nBUSY	ACK#	PE	SLCT	ERR#	INT#	1	1
DCR	002h	R/W	0	0	DIR	INT_EN	nSLIN#	INIT#	nAFD#	nSTB#
EPPA [1]	003h	R/W	EPP Address Register							
EPPD1 [1]	004h	R/W	EPP Data 1 Register							
EPPD2 [1]	005h	R/W	EPP Data 2 Register							
EPPD3 [1]	006h	R/W	EPP Data 3 Register							
EPPD4 [1]	007h	R/W	EPP Data 4 Register							
EcpDFifo	400h	R/W	ECP Data FIFO							
TFifo	400h	R/W	Test FIFO							
CnfgA	400h	R	Configuration A Register – always 90h							
CnfgB	401h	R	0	int	'000000'					
ECR	402h	R/W	Mode[2:0]			Must write '00001'				
-	403h	-	Reserved							

Table 6: Parallel port register set

Note 1: These registers are only available in EPP mode.
Note 2 : Prefix 'n' denotes that a signal is inverted at the connector. Suffix '#' denotes active-low signalling

The reset state of PDR, EPPA and EPPD1-4 is not determinable (i.e. 0xXX). The reset value of DSR is 'XXXXX111'. DCR and ECR are reset to '0000XXXX' and '00000001' respectively.

5.3.1 Parallel port data register 'PDR'

PDR is located at offset 000h in the lower block. It is the standard parallel port data register. Writing to this register in mode 000 will drive data onto the parallel port data lines. In all other modes the drivers may be tri-stated by setting the direction bit in the DCR. Reads from this register return the value on the data lines.

5.3.2 ECP FIFO Address / RLE

A data byte written to this address will be interpreted as an address if bit(7) is set, otherwise an RLE count for the next data byte. Count = bit(6:0) + 1.

5.3.3 Device status register 'DSR'

DSR is located at offset 001h in the lower block. It is a read only register showing the current state of control signals from the peripheral. Additionally in EPP mode, bit 0 is set to '1' when an operation times out (see section 5.1.3)

DSR[0]:
EPP mode: Timeout
logic 0 ⇒ Timeout has not occurred.
logic 1 ⇒ Timeout has occurred (Reading this bit clears it).

Other modes: Unused
This bit is permanently set to 1.

DSR[1]: Unused
This bit is permanently set to 1.

DSR[2]: INT#
logic 0 ⇒ A parallel port interrupt is pending.
logic 1 ⇒ No parallel port interrupt is pending.

This bit is activated (set low) on a rising edge of the ACK# pin. It is de-activated (set high) after reading the DSR.

DSR[3]: ERR#
logic 0 ⇒ The ERR# input is low.
logic 1 ⇒ The ERR# input is high.

DSR[4]: SLCT
logic 0 ⇒ The SLCT input is low.
logic 1 ⇒ The SLCT input is high.

DSR[5]: PE
logic 0 ⇒The PE input is low.
logic 1 ⇒The PE input is high.

DSR[6]: ACK#
logic 0 ⇒ The ACK# input is low.
logic 1 ⇒ The ACK# input is high.

DSR[7]: nBUSY
logic 0 ⇒ The BUSY input is high.
logic 1 ⇒ The BUSY input is low.

5.3.4 Device control register 'DCR'

DCR is located at offset 002h in the lower block. It is a read-write register which controls the state of the peripheral inputs and enables the peripheral interrupt. When reading this register, bits 0 to 3 reflect the actual state of STB#, AFD#, INIT# and SLIN# pins respectively. When in EPP mode, the WRITE#, DATASTB# AND ADDRSTB# pins are driven by the EPP controller, although writes to this register will override the state of the respective lines.

DCR[0]: nSTB#
logic 0 ⇒ Set STB# output to high (inactive).
logic 1 ⇒ Set STB# output to low (active).

During an EPP address or data cycle the WRITE# pin is driven by the EPP controller, otherwise it is inactive.

DCR[1]: nAFD#
logic 0 ⇒ Set AFD# output to high (inactive).
logic 1 ⇒ Set AFD# output to low (active).

During an EPP address or data cycle the DATASTB# pin is driven by the EPP controller, otherwise it is inactive.

DCR[2]: INIT#
logic 0 ⇒ Set INIT# output to low (active).
logic 1 ⇒ Set INIT# output to high (inactive).

DCR[3]: nSLIN#
logic 0 ⇒ Set SLIN# output to high (inactive).
logic 1 ⇒ Set SLIN# output to low (active).

During an EPP address or data cycle the ADDRSTB# pin is driven by the EPP controller, otherwise it is i nactive.

DCR[4]: ACK Interrupt Enable
logic 0 ⇒ ACK interrupt is disabled.
logic 1 ⇒ ACK interrupt is enabled.

DCR[5]: DIR
logic 0 ⇒ PD port is output.
logic 1 ⇒ PD port is input.

This bit is overridden during an EPP address or data cycle, when the direction of the port is controlled by the bus access (read/write)

DCR[7:6]: Reserved
These bits are reserved and always set to "00".

5.3.5 EPP address register 'EPPA'

EPPA is located at offset 003h in lower block, and is only used in EPP mode. A byte written to this register will be transferred to the peripheral as an EPP address by the hardware. A read from this register will transfer an address from the peripheral under hardware control.

5.3.6 EPP data registers 'EPPD1-4'

The EPPD registers are located at offset 004h-007h of the lower block, and are only used in EPP mode. Data written or read from these registers is transferred to/from the peripheral under hardware control.

5.3.7 ECP Data FIFO

Hardware transfers data from this 16 bytes deep FIFO to the peripheral when DCR(5) = '0'. When DCR(5) = '1' hardware transfers data from the peripheral to this FIFO.

5.3.8 Test FIFO

Used by the software in conjunction with the full and empty flags to determine the depth of the FIFO and interrupt levels.

5.3.9 Configuration A register

ECR[7:5] must be set to '111' to access this register. Interrupts generated will always be level, and the ECP port only supports an **implD** of '001'.

5.3.10 Configuration B register

ECR[7:5] must be set to '111' to access this register. Read only, all bits will be set to 0, except for bit[6] which will reflect the state of the interrupt.

5.3.11 Extended control register 'ECR'

The Extended control register is located at offset 002h in upper block. It is used to configure the operation of the parallel port.

ECR[4:0]: Reserved - write
These bits are reserved and must always be set to "00001".

ECR[0]: Empty - read
When DCR[5} = '0'
logic 0 ⇒ FIFO contains at least one byte
logic 1 ⇒ FIFO completely empty
When DCR[5} = '1'
logic 0 ⇒ FIFO contains at least one byte
logic 1 ⇒ FIFO contains less than one byte

ECR[1]: Full - read
When DCR[5} = '0'
logic 0 ⇒ FIFO has at least one free byte
FIFO completely full
When DCR[5} = '1'
logic 0 ⇒ FIFO has at least one free byte
logic 1 ⇒ FIFO full

ECR[2]: serviceIntr - read
When DCR[5} = '0'
logic 1 ⇒ writeIntrThreshold (8) free bytes or more in FIFO
When DCR[5} = '1'
logic 1 ⇒ readIntrThreshold (8) bytes or more in FIFO

ECR[7:5]: Mode – read / write
These bits define the operational mode of the parallel port.

logic '000'	SPP
logic '001'	PS2
logic '010'	Reserved
logic '011'	ECR
logic '100'	EPP
logic '101'	Reserved
logic '110'	Test
logic '111'	Config

6 SERIAL EEPROM

6.1 Specification

The OX12PCI840 can be configured using an optional serial electrically-erasable programmable read only memory (EEPROM). If the EEPROM is not present, the device will remain in its default configuration after reset. Although this may be adequate for some applications, many will benefit from the degree of programmability afforded by this feature. The EEPROM also allows configuration accesses to the parallel port, which can be useful for default set ups.

The EEPROM interface is based on the 93C46/56 serial EEPROM devices which have a proprietary serial interface known as Microwire™. The interface has four pins which supply the memory device with a clock, a chip-select, and serial data input and output lines. In order to read from such a device, a controller has to output serially a read command and address, then input serially the data. The 93C46/56 and compatible devices have a 16-bit data word format but differ in memory size (and number of address bits).

The OX12PCI840 incorporates a controller module which reads data from the serial EEPROM and writes data into the configuration register space. It performs this operation in a sequence which starts immediately after a PCI bus reset and ends either when the controller finds no EEPROM is present or when it reaches the end of its data. *NOTE: that any attempted PCI access while data is being downloaded from the serial EEPROM will result in a retry.* The operation of this controller is described below. Following device configuration, driver software can access the serial EEPROM through four bits in the device-specific Local Configuration Register LCC[27:24]. Software can use this register to manipulate the device pins in order to read and modify the EEPROM contents.

Note that 93C46 and 93C56 EEPROM devices offer 128 and 256 bytes of programmable data respectively.

A Windows® based utility to program the EEPROM is available. For further details please contact Oxford Semiconductor (see back cover).

Microwire™ is a trade mark of National Semiconductor. For a description of Microwire™, please refer to National Semiconductor data manuals.

6.2 EEPROM Data Organisation

The serial EEPROM data is divided in five zones. The size of each zone is an exact multiple of 16-bit WORDs. Zone0 is allocated to the header. A valid EEPROM program must contain a header. The EEPROM can be programmed from the PCI bus. Once the programming is complete, the device driver should either reset the PCI bus or set LCC[29] to reload the OX12PCI840 registers from the serial EEPROM. The general EEPROM data structure is shown in Table 7.

DATA Zone	Size (Words)	Description
0	One	Header
1	One or more	Local Configuration Registers
2	One to four	Identification Registers
3	Two or more	PCI Configuration Registers
4	Multiples of 2	Function Access

Table 7: EEPROM data format

6.2.1 Zone0: Header

The header identifies the EEPROM program as valid.

Bits	Description
15:4	These bits should return 0x840 to identify a valid program. Once the OX12C840 reads 0x840 from these bits, it sets LCC[28] to indicate that a valid EEPROM program is present.
3	1 = Zone1 (Local Configuration) exists 0 = Zone1 does not exist
2	1 = Zone2 (Identification) exists 0 = Zone2 does not exist
1	1 = Zone3 (PCI Configuration) exists 0 = Zone3 does not exist
0	1 = Zone4 (Function Access) exists 0 = Zone4 does not exist

The programming data for each zone follows the proceeding zone if it exists. For example a Header value of 0x840F indicates that all zones exist and they follow one another in sequence, while 0x8405 indicates that only Zones 2 and 4 exist where the header data is followed by Zone2 WORDs, and since Zone3 is missing Zone2 WORDs are followed by Zone4 WORDs.

6.2.2 Zone1: Local Configuration Registers

The Zone1 region of EEPROM contains the program value of the vendor-specific Local Configuration Registers using one or more configuration WORDs. Registers are selected using a 7-bit byte-offset field. This offset value is the offset from Base Address Registers in I/O or memory space (see section 4.4).

Note: Not all of the registers in the Local Configuration Register set are writable by EEPROM. If bit3 of the header is set, Zone1 configuration WORDs follow the header declaration. The format of configuration WORDs for the Local Configuration Registers in Zone1 are described in Table 8.

Bits	Description
15	'0' = There are no more Configuration WORDs to follow in Zone1. Move to the next available zone or end EEPROM program if no more zones are enabled in the Header. '1' = There is another Configuration WORD to follow for the Local Configuration Registers.
14:8	These seven bits define the byte-offset of the Local configuration register to be programmed. For example the byte-offset for LT2[23:16] is 0x0E.
7:0	8-bit value of the register to be programmed

Table 8: Zone 1 data format

6.2.3 Zone2: Identification Registers

The Zone2 region of EEPROM contains the program value for Vendor ID and Subsystem Vendor ID. The format of Device Identification configuration WORDs are described in Table 9.

Bits	Description
15	'0' = There are no more Zone2 (Identification) bytes to program. Move to the next available zone or end EEPROM program if no more zones are enabled in the Header. '1' = There is another Zone2 (Identification) byte to follow.
14:8	0x00 = Vendor ID bits [7:0]. 0x01 = Vendor ID bits [15:8]. 0x02 = Subsystem Vendor ID [7:0]. 0x03 = Subsystem Vendor ID [15:8]. 0x03 to 0x7F = Reserved.
7:0	8-bit value of the register to be programmed

Table 9: Zone 2 data format

6.2.4 Zone3: PCI Configuration Registers

The Zone3 region of EEPROM contains any changes required to the PCI Configuration registers (with the exception of Vendor ID and Subsystem Vendor ID which are programmed in Zone2). This zone consists of a function header WORD, and one or more configuration WORDs for that function. The function header is described in Table 10.

Bits	Description
15	'0' = End of Zone 3. '1' = Define this function header.
14:3	Reserved. Write zeros.
2:0	Function number for the following configuration WORD(s). '000' = Function0 Other values = Reserved.

Table 10: Zone 3 data format (Function Header)

The subsequent WORDs for each function contain the address offset and a byte of programming data for the PCI Configuration Space belonging to the function number selected by the proceeding Function-Header. The format of configuration WORDs for the PCI Configuration Registers are described below.

Bits	Description
15	'0' = This is the last configuration WORD in for the selected function in the Function-Header. '1' = There is another WORD to follow for this function.
14:8	These seven bits define the byte-offset of the PCI configuration register to be programmed. For example the byte-offset of the Interrupt Pin register is 0x3D. Offset values are tabulated in section 4.2.
7:0	8-bit value of the register to be programmed

Table 11: Zone 3 data format (data)

Table 12 shows which PCI Configuration registers are writable from the EEPROM for each function.

Offset	Bits	Description
0x02	7:0	Device ID bits 7 to 0.
0x03	7:0	Device ID bits 15 to 8.
0x06	3:0	Must be '0000'.
0x06	4	Extended Capabilities.
0x06	7:5	Must be '000'.
0x09	7:0	Class Code bits 7 to 0.
0x0A	7:0	Class Code bits 15 to 8.
0x0B	7:0	Class Code bits 23 to 16.
0x2E	7:0	Subsystem ID bits 7 to 0.
0x2F	7:0	Subsystem ID bits 15 to 8.
0x3D	7:0	Interrupt pin.
0x42	7:0	Power Management Capabilities bits 7 to 0.
0x43	7:0	Power Management Capabilities bits 15 to 8.

Table 12: EEPROM-writable PCI configuration registers

6.2.5 Zone4: Function Access

Zone 4 allows the parallel port to be configured, prior to PCI access. This can be useful for patching designs to work with generic drivers, enabling interrupts, etc. Each 8-bit (function) access is equivalent to accessing the function through I/O bars 0 and 1, with the exception that a function read access does not return any data (discarded). Each entry in zone 4 comprises 2 16 bit words. The format is as shown in Table 14.

1st WORD of FUNCTION ACCESS PAIR

Word	Bits	Description
	15	'1' - another WORD to follow
	14:12	BAR number to access 000 for BAR 0 001 for BAR 1 *others reserved*
	11	'0' : Read access (data discarded) '1' : Write access
	10:8	Reserved – write 0's
	7:0	I/O address to access This is the location (I/O offset from the relevant Base Address) that needs to be written/read.

2nd WORD of FUNCTION ACCESS PAIR

Word	Bits	Description
	15	'1' – another function access WORD pair to follow. '0' – no more function access pairs. End EEPROM program.
	14:8	Reserved – write 0's
	7:0	Data to be written to location. *Field unused for function access READS.*

7 OPERATING CONDITIONS

Symbol	Parameter	Min	Max	Units
V_{DD}	DC supply voltage	-0.3	7.0	V
V_{IN}	DC input voltage	-0.3	V_{DD} + 0.3	V
I_{IN}	DC input current		+/- 10	mA
T_{STG}	Storage temperature	-40	125	°C

Table 13: Absolute maximum ratings

Symbol	Parameter	Min	Max	Units
V_{DD}	DC supply voltage	4.5	5.5	V
T_C	Temperature	0	70	°C

Table 14: Recommended operating conditions

8 DC ELECTRICAL CHARACTERISTICS

8.1 Non-PCI I/O Buffers

Symbol	Parameter	Condition	Min	Max	Units
V_{DD}	Supply voltage	Commercial	4.75	5.25	V
V_{IH}	Input high voltage	TTL Interface [1] TTL Schmitt trig	2.0 2.0		V
V_{IL}	Input low voltage	TTL Interface [1] TTL Schmitt trig		0.8 0.8	V
C_{IL}	Cap of input buffers			5.0	pF
C_{OL}	Cap of output buffers			10.0	pF
I_{IH}	Input high leakage current	$V_{in} = V_{DD}$	-10	10	µA
I_{IL}	Input low leakage current	$V_{in} = V_{SS}$	-10	10	µA
V_{OH}	Output high voltage	I_{OH} = 1 µA	V_{DD} – 0.05		V
V_{OH}	Output high voltage	I_{OH} = 4 mA [2]	2.4		V
V_{OL}	Output low voltage	I_{OL} = 1 µA		0.05	V
V_{OL}	Output low voltage	I_{OL} = 4 mA [2]		0.4	V
I_{OZ}	3-state output leakage current		-10	10	µA

Symbol	Parameter	Typical	Max	Units
I_{CC}	Operating supply current in normal mode	TBD	TBD	mA
	Operating supply current in Power-down mode	TBD		

Table 15: Characteristics of non -PCI I/O buffers

Note 1: All input buffers are TTL with the exception of PCI buffers

Note 2: I_{OH} and I_{OL} are 12 mA for PD/LBDB[7:0] and other Parallel Port Outputs. They are 4 mA for all other non-PCI outputs

8.2 PCI I/O Buffers

Symbol	Parameter	Condition	Min	Max	Unit
DC Specifications					
V_{CC}	Supply voltage		4.75	5.25	V
V_{IL}	Input low voltage		-0.5	0.8	V
V_{IH}	Input high voltage		2.0	V_{CC} + 0.5	V
I_{IL}	Input low leakage current	V_{IN} = 0.5V		-70	µA
I_{IH}	Input high leakage current	V_{IN} = 2.7V		70	µA
V_{OL}	Output low voltage	I_{OUT} = -2 mA		0.55	V
V_{OH}	Output low voltage	I_{OUT} = 3 mA, 6mA	2.4		V
C_{IN}	Input pin capacitance			10	pF
C_{CLK}	CLK pin capacitance		5	12	pF
C_{IDSEL}	IDSEL pin capacitance			8	pF
L_{PIN}	Pin inductance			10	nH
AC Specifications					
	Switching current	0 < V_{OUT} ☐ 1.4	-44		
$I_{OH(AC)}$	high	1.4 < V_{OUT} ☐ 2.4	-44 (V_{OUT} - 1.4)/0.024		mA
		3.1 < V_{OUT} ☐ V_{CC}		Eq. A	
	(Test point)	V_{OUT} = 3.1		-142	
	Switching current	V_{OUT} ☐ 2.2	95		
$I_{OL(AC)}$	low	2.2 > V_{OUT} > 0.55	V_{OUT} / 0.023		mA
		0.71 > V_{OUT} > 0		Eq. B	
	(Test point)	V_{OUT} = 0.71		206	
I_{CL}	Low clamp current	-5 < V_{IN} < -1	-25 + (V_{IN} +1)/ 0.015		mA
I_{HL}	High clamp current	V_{CC}+4 < V_{IN} < V_{CC}+1	25+ (V_{IN} -V_{CC} -1)/ 0.015		mA
$Slew_R$	Output rise slew rate	0.4V to 2.4V	1	5	V/nS
$Slew_F$	Output fall slew rate	2.4V to 0.4V	1	5	V/nS

Table 16: Characteristics of PCI I/O buffers

Eq. A : I_{OH} = 11.9 * (V_{OUT} - 5.25) * (V_{OUT} + 2.45) for 3.1 < V_{OUT} ☐ V_{CC}
Eq. B : I_{OL} = 78.5 * V_{OUT} * (4.4 - V_{OUT}) for 0.71 > V_{OUT} > 0

9 AC ELECTRICAL CHARACTERISTICS

9.1 PCI Bus

The timings for PCI pins comply with PCI Specification for the 5.0 Volt signalling environment.

10 TIMING WAVEFORMS

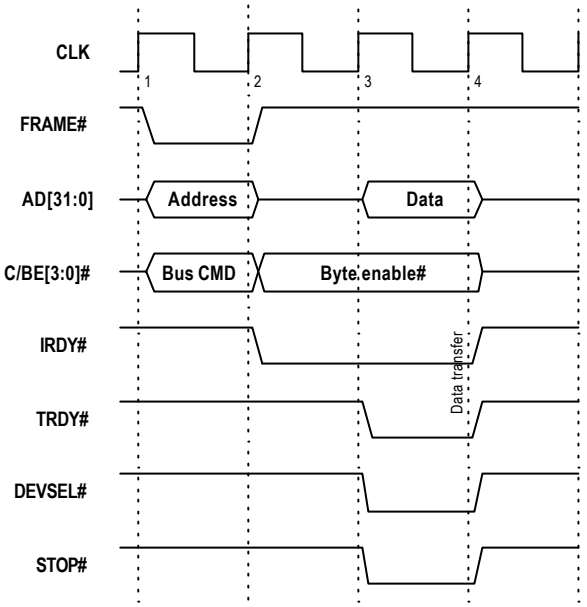


Figure 1: PCI Read transaction from Local Configuration registers

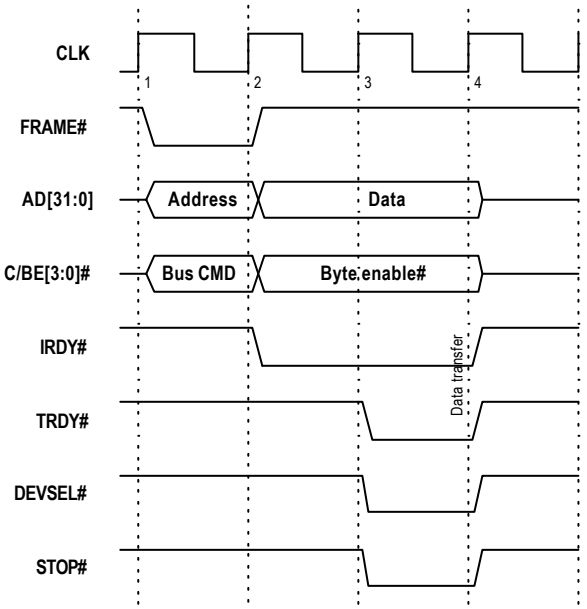
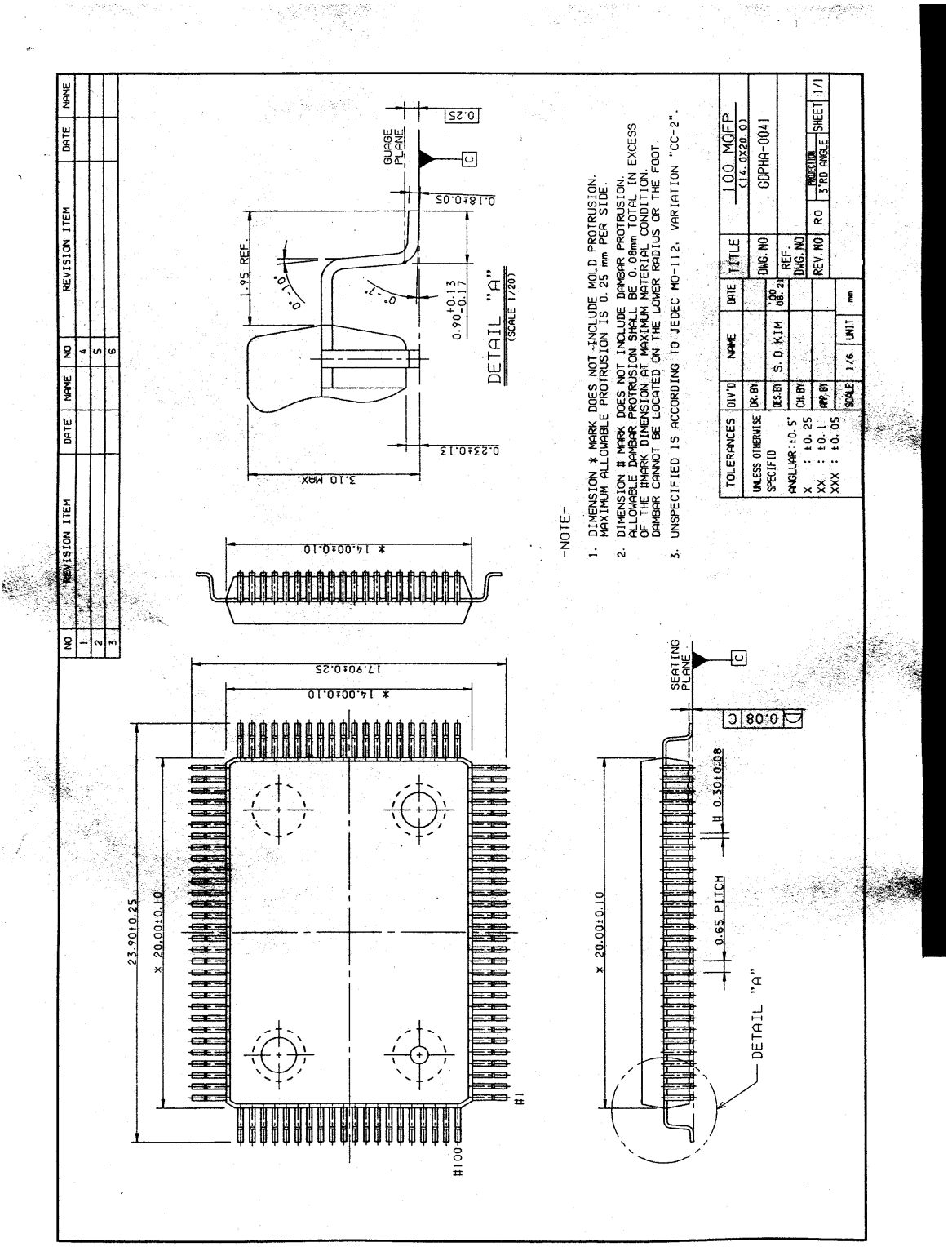


Figure 2: PCI Write transaction to Local Configuration Registers

11 PACKAGE DETAILS



12 NOTES

This page has been intentionally left blank

13 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

DISCLAIMER