OX9160
PCI Peripheral Bridge with EPP Parallel Port & 8/32 bit local bus

FEATURES

- 33MHz, 32-bit target PCI controller.
- Fully PCI 2.2 and PCI Power Management 1.0 compliant.
- 8- or 32-bit pass-through Local bus.
- IEEE1284 parallel port.
- Parallel port supports EPP mode for maximum data transfer rate to printers, removable drives etc.
- Most operations complete within one PCI frame (no retries).
- Supports shared interrupts
- 12 multi-purpose I/O pins which can be configured as interrupt input pins.
- EEPROM interface for optional reconfiguration.
- Local bus operation via I/O or memory mapping.
- Local bus supports Intel or Motorola mode signalling.
- Existing driver support for common I/O solutions.
- On-chip oscillator.
- 5.0V operation.
- Low power CMOS.
- 160 TQFP package.

DESCRIPTION

The OX9160 is a low-cost, general purpose PCI bridge solution designed to ease the migration to PCI of parallel port cards and instrumentation devices. It is configurable to provide either a Local bus interface or a bi-directional parallel port.

Using the local bus function, legacy devices can be easily accessed throught the target PCI interface, which is compliant with version 2.2 of the PCI Bus Specification and version 1.0 of PCI Power Management Specification. All reads and writes are completed with a minimum of PCI wait states, which ensures lower PCI bus occupancy than most similar PCI bridge solutions.

The local bus can be configured to operate with either 8- or 32-bit data, using either Intel x86 style or Motorola style signalling.

Alternatively the local bus can be disabled in favour of an integrated IEEE 1284 EPP parallel port. The parallel port is an IEEE 1284-compliant host interface, which supports SPP, PS2 (bidirectional) and EPP modes.

The local Bus function is extremely flexible, allowing the designer to customize the addressable space, divide it into chip-select regions, access devices via I/O or memory space mapping, and adjust the timings of all operations. The default register values have been selected to support many standard peripheral chips such as I/O controllers and other ISA-type devices, however all such parameters can be overwritten using an optional Microwire™ serial EEPROM.

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

OX9160 Data Sheet Revision 1.2 – June 2001
Part No. OX9160-TQC33-A

CONTENTS

1 BLOCK DIAGRAM

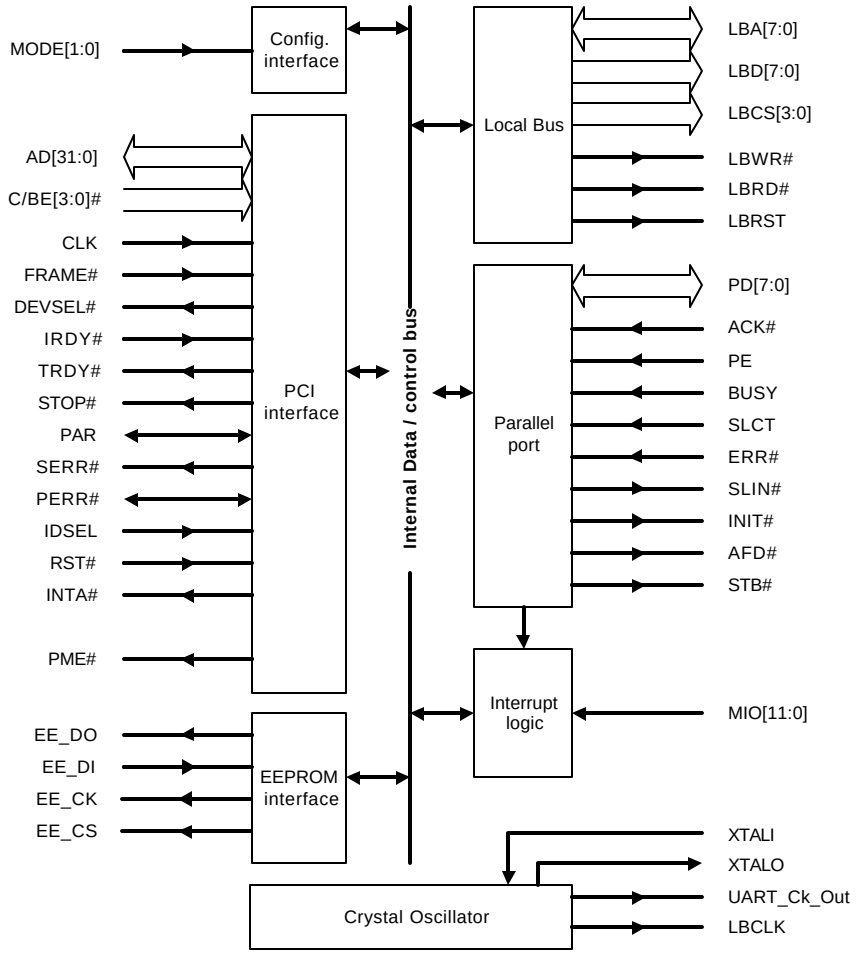
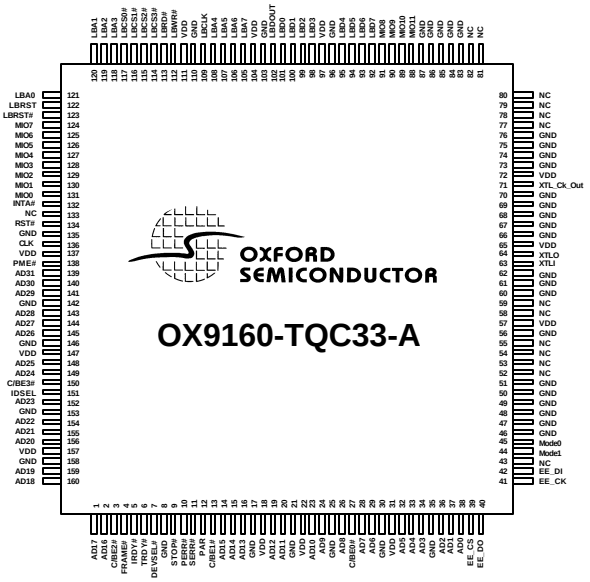


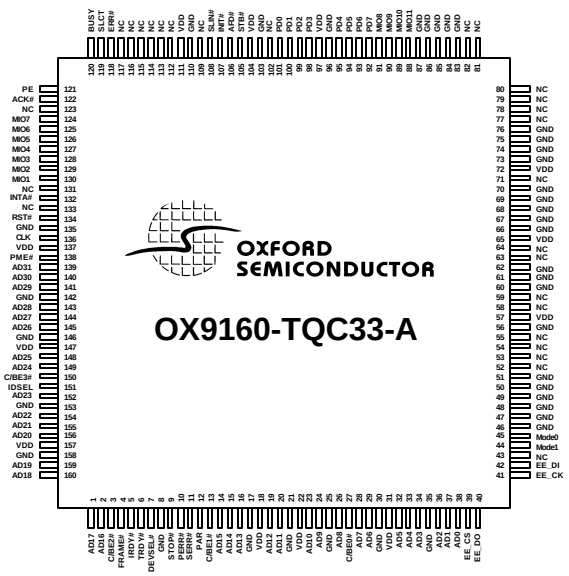
Figure 1 : OX9160 block diagram

2 PIN INFORMATION

Mode '00': 8-bit local bus



Mode '01': Parallel port



Mode '11': 32-bit local bus

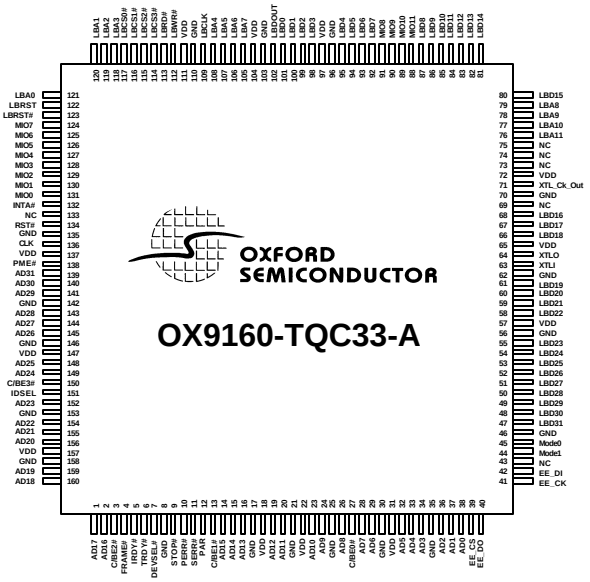


Figure 2: Pinout in all configurable modes (package = 160 TQFP)

3 PIN DESCRIPTIONS

Mode			Dir	Name	Description
00	01	11			
PCI Interface					
139, 140, 141, 143, 144, 145, 148, 149, 152, 154, 155, 156, 159, 160, 1, 2, 14, 15, 16, 19, 20, 23, 24, 26, 28, 29, 32, 33, 34, 36, 37, 38			P_I/O	AD[31:0]	Multiplexed PCI Address/Data bus
150, 3, 13, 27			P_I	C/BE[3:0]#	PCI Command/Byte enable
136			P_I	CLK	PCI system clock
4			P_I	FRAME#	Cycle Frame
7			P_O	DEVSEL#	Device Select
5			P_I	IRDY#	Initiator ready
6			P_O	TRDY#	Target ready
9			P_O	STOP#	Target Stop request
12			P_I/O	PAR	Parity
11			P_O	SERR#	System error
10			P_I/O	PERR#	Parity error
151			P_I	IDSEL	Initialization device select
134			P_I	RST#	PCI system reset
132			P_OD	INTA #	PCI interrupt
138			P_OD	PME#	Power management event
Local bus					
122	N/A	122	O	LBRST	Local bus active-high reset
123	N/A	123	O	LBRST#	Local bus active-low reset
102			O	LBDOUT	Local bus data out enable. This pin can be used by external transceivers; it is high when LBD[7:0] are in output mode and low when they are in input mode.
114-7	N/A	114-7	O	LBCS[3:0]#	Local bus active-low Chip-Select (Intel mode)
			O	LBDS[3:0]#	Local bus active-low Data-Strobe (Motorola mode)
112	N/A	112	O	LBWR#	Local bus active-low write-strobe (Intel mode)
			O	LBRDWR#	Local bus Read-not-Write control (Motorola mode)
113	N/A	113	O	LBRD#	Local bus active-low read-strobe (Intel mode)
			Z	Hi-Z	Permanent high impedance (Motorola mode)
105-8 118-21	N/A	N/A	O	LBA[7:0]	(8-bit mode) Local bus address signals
N/A	N/A	76-9, 105-8, 118-21	O	LBA[12:0]	(32-bit mode) Local bus address signals
92-5 98-101	N/A	N/A	I/O	LBD[7:0]	(8-bit mode) Local bus data signals
N/A	N/A	47-55, 58-61, 66-68, 80-87, 92-95, 98-101	I/O	LBD[31:0]	(32-bit mode) Local bus data signals

Parallel port					
N/A	122	N/A	I	ACK#	Acknowledge (SPP mode). ACK# is asserted (low) by the peripheral to indicate that a successful data transfer has taken place.
N/A	121	N/A	I	PE	Paper Empty. Activated by printer when it runs out of paper.
N/A	120	N/A	I	BUSY	Busy (SPP mode). BUSY is asserted (high) by the peripheral when it is not ready to accept data
N/A	108	N/A	OD	SLIN#	Select (SPP mode). Asserted by host to select the peripheral
N/A	119	N/A	I	SLCT	Peripheral selected. Asserted by peripheral when selected.
N/A	118	N/A	I	ERR#	Error. Held low by the peripheral during an error condition.
N/A	107	N/A	OD	INIT#	Initialize (SPP mode). Commands the peripheral to initialize.
N/A	106	N/A	OD	AFD#	Auto Feed (SPP mode, open-drain)
N/A	105	N/A	OD	STB#	Strobe (SPP mode). Used by peripheral to latch data currently available on PD[7:0]
N/A	Bus	N/A	I/O	PD[7:0]	Parallel data bus
EEPROM pins					
41			O	EE_CK	EEPROM clock
39			O	EE_CS	EEPROM active-high Chip Select
42			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 pullup is sufficient.
40			O	EE_DO	EEPROM data out.
Miscellaneous pins					
63			I	XTLI	Crystal oscillator input
64			O	XTLO	Crystal oscillator output. Maximum frequency 60MHz
71			O	XTL_Ck_Out	Buffered crystal clock output. This clock can drive TTL clock signals from a clock generator circuit connected at XTLI & XTLO. Can be enabled/disabled by software.
109			O	LBCLK	Buffered PCI clock. Can be enabled/disabled by software
44,45			I	Mode[1:0]	Mode selector: 00: 8-bit local bus 01: Parallel port 11: 32-bit local bus
Power & Ground					
18, 31, 57, 72, 97, 111, 147, 157			V	AC VDD	Supplies power to output buffers in switching (AC) state
22, 65, 104, 137			V	DC VDD	Power supply. Supplies power to core logic, input buffers and output buffers in steady state
8, 17, 25, 30, 35, 56, 70, 96, 110, 142, 146, 153, 158			G	AC GND	Supplies GND to output buffers in switching (AC) state
21, *46, 47, 48, 49, 50, 51, 60, 61,* 62, *66, 67, 68, 69, 73, 74, 75, 76, 83, 84, 85, 86, 87,* 103, 135,			G	DC GND	Ground (0 volts). Supplies GND to core logic, input buffers and output buffers in steady state

Table 1: Pin Descriptions

Multi-purpose & External interrupt pins					
131 N/A	N/A 131	131 N/A	I/O Z	MIO0 Hi-Z	Multi-purpose I/O 0. Can drive high or low, or assert a PCI interrupt Permanent high impedance
130			I/O	MIO1	Multi-purpose I/O 1. Can drive high or low, or assert a PCI interrupt.
129			I/O I	MIO2 PME_In	Multi-purpose I/O 2. When LCC[7] = 0, this pin can drive high or low, or assert a PCI interrupt. Input power management event. When LCC[7] is set this input pin can assert a function1 PME#
124-128 88-91	124-128 88-91	124-128 88-91	I/O	MIO[11:3]	Multi-purpose I/O pins. Can drive high or low, or assert a PCI interrupt

Note 1: Direction key:

I	Input	P_I	PCI input
IU	Input with internal pull-up	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 inside the device. 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, Local bus and parallel port pins.

Also, some GND pins (italicised) serve as GND in mode '00' and mode '01'; however they are multiplexed and function as address/data pins in mode '11'.

4 PCI TARGET CONTROLLER

4.1 Operation

The OX9160 responds to the following PCI transactions:-

- Configuration access: The OX9160 responds to type 0 configuration reads and writes if the IDSEL signal is asserted and the bus address is selecting a valid configuration register. The device will respond to the configuration transaction by asserting DEVSEL#. Data transfer then follows. Any other configuration transaction will be ignored by the OX9160.

- 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. For the parallel port and 8-bit Local bus functions, only byte accesses are supported; however the 32-bit bridge function also supports word and dword accesses. 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 OX9160. In other words, OX9160 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 OX9160 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 the actual controller.

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

The OX9160 will complete all transactions as disconnect-with-data, ie 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 OX9160 is reading from the serial EEPROM.

The OX9160 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 OX9160 as a target, so a bus master can perform faster sequences of write transactions to the Local bus when an inter-frame turn-around cycle is not required.

The device supports any combination of byte-enables to the PCI Configuration Registers, the Local Configuration registers (see Base Address 2 and 3) and the Local bus controller in 32-bit mode. If a byte-enable is not asserted, that byte is unaffected by a write operation and undefined data is returned upon a read.

The OX9160 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.

4.2 Configuration space

All required fields in the standard configuration space 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 31	16	15	0	Offset Address
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)				10h
Base Address Register 1 (BAR 1)				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
Reserved				20h
Reserved				24h
Reserved				28h
Subsystem ID		Subsystem Vendor ID		2Ch
Reserved				30h
Reserved			Cap_Ptr	34h
Reserved				38h
Reserved	Reserved	Interrupt Pin	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	
	8-bit local bus	32-bit local bus	Parallel port	EEPROM	PCI
Vendor ID	0x1415			W	R
Device ID	0x9511	0x9512	0x9513	W	R
Command	0x0000			-	R/W
Status	0x0290			W (bit 4)	R/W
Revision ID	0x00			-	R
Class code	0x068000	0x068000	0x070101	W	R
Header type	0x80			-	R
BAR 0	0x00000001			-	R/W
BAR 1	0x00000000			-	R/W
BAR 2	0x00000001			-	R/W
BAR 3	0x00000000			-	R/W
Subsystem VID	0x1415			W	R
Subsystem ID	0x0000			W	R
Cap ptr.	0x40			-	R
Interrupt line	0x00			-	R/W
Interrupt pin	0x01			-	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 local bus and 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 functions, according to which function is enabled and the size required. The addresses allocated can then be used to access the functions. The mapping of these BARs is shown in Table 4.

BAR	Local bus	Parallel port
0	Local bus (I/O mapped)	Parallel port base registers (I/O mapped)
1	Local bus (memory mapped)	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 8-bit local bus

BAR 0 and BAR 1 are used to access the Local bus. The system allocates a block of I/O space and a block of memory space according to the size requested.

I/O space

In order to minimise the usage of IO space, the block size for BAR0 (I/O access) is user definable in the range of 4 to 256 bytes. Having assigned the address range, the user can define two adjacent address bits to decode up to four chip selects internally. This facility allows glueless implementation of the local bus connecting to four external peripheral chips. The address range and the lower address bit for chip-select decoding (Lower-Address-CS-Decode) are defined in the Local bus Configuration register (see LT2[26:20] in section 1.1).

The 8-bit Local bus has eight address lines (LBA[7:0]) which correspond to the maximum IO address space. If the maximum allowable block size is allocated to the IO space (i.e. 256 bytes), then as access in IO space is byte aligned, LBA[7:0] equal PCI AD[7:0] respectively. When the user selects an address range which is less than 256 bytes, the unused upper address lines will be set to logic zero.

The region can be divided into four chip-select regions when the user selects the second uppermost non-zero address bit for chip-select decoding. For example if 32-bytes of IO space are reserved, the local bus address lines A[4:0] are active and the remaining address lines are set to zero. To generate four chip-selects the user should select A3 as the Lower-Address-CS-Decode. In this case A[4:3] will be used internally to decode chip-selects, asserting LBCS0# when the address offset is 00-07h, LBCS1# when offset is 08-0Fh, LBCS2# when offset is 10-17h, and LBCS3# when offset is 18- 1Fh.

The region can be divided into two chip-select regions by selecting the uppermost address bit to decode chip selects. In the above example, the user can select A4 as the Lower-Address-CS-Decode, thus using A[5:4] internally to decode chip selects. As in this example LBA5 is always zero, only chip-select lines LBCS0# and LBCS1# will be decoded into, asserting LBCS0# when address offset is 00-0Fh and LBCS1# when offset is 10-1Fh.

The region can be allocated to a single chip-select region by assigning an address bit beyond the selected range to Lower-Address-CS-Decode (but not above A8). In the above example, if the user selects A5 as the Lower-Address-CS-Decode, A[6:5] will be used to internally decode chip-selects. As in this example LBA[7:5] are always zero, only the chip select line LBCS0# may be selected. In this case address offset 00-1Fh asserts LBCS0# and the other chip-select lines remain inactive permanently.

Memory Space:

The memory base address registers have an allocated fixed size of 4K bytes in the address space. Since the Local bus has 8 address lines and the OX9160 only implements DWORD aligned accesses in memory space, the 256 bytes of addressable space per chip select is expanded to 1K. Unlike an I/O access, for a memory access the unused upper address lines are always active and the internal chip-select decoding logic ignores the user setting for Lower-Address-CS-Decode (LT2[26:23]) and uses PCI AD[11:10] to decode into 4 chip-select regions. When the Local bus is accessed in memory space, A[9:2] are asserted on LBA[7:0]. The chip-select regions are defined in Table 5.

Local bus Chip-Select (Data-Strobe)	PCI Offset from BAR 1 (Memory space)	
	Lower Address	Upper Limit
LBCS0# (LBDS0#)	000h	3FCh
LBCS1# (LBDS1#)	400h	7FCh
LBCS2# (LBDS2#)	800h	BFCh
LBCS3# (LBDS3#)	C00h	FFCh

Table 5: PCI address map for local bus (memory)

Note: The description given for I/O and memory accesses is for an Intel-type configuration for the Local bus. For Motorola-type configuration, the chip select pins are redefined to data strobe pins. In this mode the Local bus offers up to 8 address lines and four data-strobe pins.

4.3.2 PCI access to 32-bit local bus

Access to the Local bus in 32-bit mode is similar to 8-bit mode (see section 4.3.1) with the following exceptions:

- The local Bus offers a 32-bit bi-directional data bus and 12 bit address bus
- The PCI address signals 'AD[13:2]' are asserted on LBA[11:0]
- Block size in memory space is programmable by LT2[28:27] (see section 1.1)
- The Lower-Address-CS-Decode (LT2[26:23]) parameter is used to decode up to 4 chip selects

The block size allocation for chip-select regions is defined in Table 6.

Number of Chip selects	Memory block size (Kbytes)	LT2[28:27]	LT2[26:23]
1	16	'01'	'1010'
2	16	'01'	'1001'
4	16	'01'	'1000'
1	4	'00'	'1000'
2	4	'00'	'0111'
4	4	'00'	'0110'

Table 6: PCI access to 32-bit local bus (memory)

4.3.3 PCI access to parallel port

When the parallel port is enabled (Mode 01), access to the port works via BAR definitions as usual, except that there are two I/O BARs corresponding to two sets of registers defined to operate a bi-directional Parallel Port. Memory mapped access to the parallel port is not supported.

The user can change the I/O space block size of BAR0 by over-writing the default values in LT2[25:20] using the serial EEPROM (see section 1.1). For example the user can reduce the allocated space for BAR0 to 4-bytes by setting LT2[22:20] to '001'. The I/O block size allocated to BAR1 is fixed at 8-Bytes.

Legacy PC 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 are used to configure the controller. They are mapped to the I/O and memory addresses set up in BAR2 and BAR3, with the offsets defined for each register. Access is limited to byte only for I/O accesses; memory accesses can also be 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, external clock reference signals and the serial EEPROM. The individual bits are described below.

Bits	Description	Read/Write		Reset
		EEPROM	PCI	
1:0	Mode. These bits return the state of the Mode[1:0] pins.	-	R	XX
2	Enable crystal clock output. When this bit is set, the crystal oscillator output pin (XTL_Ck_Out) is active. When low, XTL_Ck_Out is permanently low.	W	RW	0
4:3	Endian Byte-Lane Select for memory access to 8-bit Local bus. 00 = Select Data[7:0] 10 = Select Data[23:16] 01 = Select Data[15:8] 11 = Select Data[31:24] Memory access to OX9160 is always DWORD aligned. When accessing 8-bit regions like the 8-bit Local bus and 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. These bits are ignored in 32-bit Local bus.	W	RW	00
6:5	Reserved. These bits are used for test purposes. The device driver must write zeros to these bits.	-	R	00
7	MIO2_PME Enable. A value of '1' enables the MIO2 pin to set the PME_Status in PMCSR register, and hence assert the PME# pin if enabled. A value of '0' disables MIO2 from setting the PME_Status bit (see section 4.6).	W	RW	0
23:8	Reserved. These bits are used for test purposes. The device driver must write zeros to these bits.	-	R	0000h
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	-	-	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[11:0] as follows.

Bits	Description	Read/Write		Reset
		EEPROM	PCI	
1:0	MIO0 Configuration Register (Mode[1:0]≠'01'). 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' Unused (Mode[1:0]='01'). When the Parallel Port is enabled, MIO[0] pin is unused and will remain in forcing output mode.	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
5:4	MIO2 Configuration Register (LCC[7]='0'). 00 -> MIO2 is a non-inverting input pin 01 -> MIO2 is an inverting input pin 10 -> MIO2 is output pin driving '0' 11 -> MIO2 is output pin driving '1' PME_Input (LCC[7]='1'). When LCC[7] is set, MIO2 pin is re-defined to PME_Input. It's polarity will be controlled by MIC[4]. It sets the sticky PME_Status bit.	W	RW	00
7:6	MIO3 Configuration Register. 00 -> MIO3 is a non-inverting input pin 01 -> MIO3 is an inverting input pin 10 -> MIO3 is an output pin driving '0' 11 -> MIO3 is an output pin driving '1'	W	RW	00
9:8	MIO4 Configuration Register. 00 -> MIO4 is a non-inverting input pin 01 -> MIO4 is an inverting input pin 10 -> MIO4 is an output pin driving '0' 11 -> MIO4 is an output pin driving '1'	W	RW	00
11:10	MIO5 Configuration Register. 00 -> MIO5 is a non-inverting input pin 01 -> MIO5 is an inverting input pin 10 -> MIO5 is an output pin driving '0' 11 -> MIO5 is an output pin driving '1'	W	RW	00
13:12	MIO6 Configuration Register. 00 -> MIO6 is a non-inverting input pin 01 -> MIO6 is an inverting input pin 10 -> MIO6 is an output pin driving '0' 11 -> MIO6 is an output pin driving '1'	W	RW	00
15:14	MIO7 Configuration Register. 00 -> MIO7 is a non-inverting input pin 01 -> MIO7 is an inverting input pin 10 -> MIO7 is an output pin driving '0' 11 -> MIO7 is an output pin driving '1'	W	RW	00
17:15	MIO8 Configuration Register. 00 -> MIO8 is a non-inverting input pin	W	RW	00

Bits	Description	Read/Write		Reset
		EEPROM	PCI	
	01 -> MIO8 is an inverting input pin 10 -> MIO8 is an output pin driving '0' 11 -> MIO8 is an output pin driving '1'			
19:18	MIO9 Configuration Register. 00 -> MIO9 is a non-inverting input pin 01 -> MIO9 is an inverting input pin 10 -> MIO9 is an output pin driving '0' 11 -> MIO9 is an output pin driving '1'	W	RW	00
21:20	MIO10 Configuration Register. 00 -> MIO10 is a non-inverting input pin 01 -> MIO10 is an inverting input pin 10 -> MIO10 is an output pin driving '0' 11 -> MIO10 is an output pin driving '1'	W	RW	00
23:22	MIO11 Configuration Register. 00 -> MIO11 is a non-inverting input pin 01 -> MIO11 is an inverting input pin 10 -> MIO11 is an output pin driving '0' 11 -> MIO11 is an output pin driving '1'	W	RW	00
31:24	Reserved	-	R	00h

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 Local bus. The timing parameters are programmed in 4-bit registers to define the assertion/de-assertion of the Local bus control signals. The value 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 following arrangement provides a flexible approach for users to define the desired bus timing of their peripheral devices. The timings refer to I/O or Memory mapped access to BAR0 and BAR1 respectively.

Bits	Description	Read/Write		Reset
		EEPROM	PCI	
3:0	Read Chip-select Assertion (Intel-type interface). Defines the number of clock cycles after the Reference Cycle when the LBCS[3:0]# pins are asserted (low) during a read operation from the Local bus.[1] These bits are unused in Motorola-type interface.	W	RW	0h
7:4	Read Chip-select De-assertion (Intel-type interface). Defines the number of clock cycles after the Reference Cycle when the LBCS[3:0]# pins are de-asserted (high) during a read from the Local bus.[1] These bits are unused in Motorola-type interface.	W	RW	3h (2h for parallel port)
11:8	Write Chip-select Assertion (Intel-type interface). Defines the number of clock cycles after the Reference Cycle when the LBCS[3:0]# pins are asserted (low) during a write operation to the Local bus.[1] These bits are unused in Motorola-type interface.	W	RW	0h

Bits	Description	Read/Write		Reset
		EEPROM	PCI	
15:12	Write Chip-select De-assertion (Intel-type interface). Defines the number of clock cycles after the reference cycle when the LBCS[3:0]# pins are de-asserted (high) during a write operation to the Local bus. [1] Read-not-Write De-assertion during write cycles (Motorola-type interface). Defines the number of clock cycles after the reference cycle when the LBRDWR# pin is de-asserted (high) during a write to the Local bus. [1]	W	RW	2h
19:16	Read Control Assertion (Intel-type interface). Defines the number of clock cycles after the Reference Cycle when the LBRD# pin is asserted (low) during a read from the Local bus. [1] Read Data-strobe Assertion (Motorola-type interface). Defines the number of clock cycles after the Reference Cycle when the LBDS[3:0]# pins are asserted (low) during a read from the Local bus. [1]	W	RW	0h (1h for parallel port)
23:20	Read Control De-assertion (Intel-type interface). Defines the number of clock cycles after the Reference Cycle when the LBRD# pin is de-asserted (high) during a read from the Local bus. [1] Read Data-strobe De-assertion (Motorola-type interface). Defines the number of clock cycles after the Reference Cycle when the LBDS[3:0]# pins are de-asserted (high) during a read from the Local bus. [1]	W	RW	3h (2h for parallel port)
27:24	Write Control Assertion (Intel-type interface). Defines the number of clock cycles after the Reference Cycle when the LBWR# pin is asserted (low) during a write to the Local bus. [1] Write Data-strobe Assertion (Motorola-type interface). Defines the number of clock cycles after the Reference Cycle when the LBDS[3:0]# pins are asserted (low) during a write to the Local bus. [1]	W	RW	0h (1h for parallel port)
31:28	Write Control De-assertion (Intel-type interface). Defines the number of clock cycles after the Reference Cycle when the LBWR# pin is de-asserted (high) during a write to the Local bus. [1] Write Data-strobe De-assertion (Motorola-type interface). Defines the number of clock cycles after the Reference Cycle when the LBDS[3:0]# pins are de-asserted (high) during a write cycle to the Local bus. [1]	W	RW	2h

Note 1: Only values in the range of 0h to Ah (0-10 decimal) are valid. Other values are reserved. These parameters apply to both 8-bit and 32-bit Local bus configurations. See notes in the following page.

4.4.4 Local bus Timing Parameter register 2 'LT2' (Offset 0x0C):

Bits	Description	Read/Write		Reset
		EEPROM	PCI	
3:0	Write Data Bus Assertion. This register defines the number of clock cycles after the Reference Cycle when the LBD pins actively drive the data bus during a write operation to the Local bus. [1]	W	RW	0h
7:4	Write Data Bus De-assertion. This register defines the number of clock cycles after the Reference Cycle when the LBD pins go high-impedance during a write operation to the Local bus. [1,2]	W	RW	Fh
11:8	Read Data Bus Assertion. This register defines the number of clock cycles after the Reference Cycle when the LBD pins actively drive the data bus at the end of a read operation from the Local bus. [1]	W	RW	4h (2h for parallel port)
15:12	Read Data Bus De-assertion. This register defines the number of clock cycles after the Reference Cycle when the LBD pins go high-impedance during at the beginning of a read cycle from the Local bus. [1]	W	RW	0h
19:16	Reserved.	-	R	0h
22:20	IO Space Block Size of BAR 0. 000 = Reserved 001 = 4 Bytes 010 = 8 Bytes 011 = 16 Bytes 100 = 32 Bytes 101 = 64 Bytes 110 = 128 Bytes 111 = 256 Bytes	W	R	'100' (='010' for parallel port)
26:23	Local bus Chip–select Parameter 'Lower-Address-CS-Decode'. [2] IO space in 8-bit Local bus: 0000 = A2, 0001 = A3, 0010 = A4, 0011 = A5, 0100 = A6, 0101 = A7, 0110 = A8, 0111 = A9, 1000 = Res, 1001 = Res, 1010 = Res, 1011 = Res, 1100 = Res, 1101 = Res, 1110 = Res, 1111 = Res Memory space and IO space in 32-bit Local bus: 0000 = A4, 0001 = A5, 0010 = A6, 0011 = A7, 0100 = A8, 0101 = A9, 0110 = A10, 0111 = A11, 1000 = A12, 1001 = A13, 1010 = A14, 1011 = Res, 1100 = Res, 1101 = Res, 1110 = Res, 1111 = Res	W	RW	'0001' (='0010' for parallel port)
28:27	Memory Space Block Size of BAR1 (32-bit Local bus only). 00 = 4 Kbytes 01 = 16 Kbytes 10 = Reserved 11 = Reserved When 8-bit Local bus or Parallel Port is selected (Mode[1:0]='00' or '01'), the Memory Block size is fixed at 4K and these bits are ignored.	W	R	00
29	Local bus Software Reset. When this bit is a 1 the Local bus reset pin is activated. When this bit is a 0 the Local bus reset pin is de-activated. [3]	-	RW	0
30	Local bus Clock Enable. When this bit is a 1 the Local bus clock (LBCK) pin is enabled. When this bit is a 0 LBCK pin is permanently low. The Local bus Clock is a buffered PCI clock.	W	RW	0
31	Bus Interface Type. When low (=0) the Local bus is configured to Intel-type operation, otherwise it is configured to Motorola-type operation. Note that when Mode[1:0] is '01', this bit is hard wired to 0.	W	RW	0

Note 1: Only values in the range of 0 to Ah (0-10 decimal) are valid. Other values are reserved as writing higher values causes the PCI interface to retry all accesses to the Local bus as it is unable to complete the transaction in 16 PCI clock cycles.

Note 2: The Lower-Address-CS-Decode parameter is described in sections 4.3.1 & 4.3.2. These bits are unused for Memory access to the 8-bit Local bus which uses a fixed decoding to allocate 1K regions to 4 chip selects. For further information on the Local bus, see section 5.

Note 3: The Local bus and the Parallel Port are both reset with PCI reset. In Addition, the user can issue the Software Reset Command.

LT2[15:0] enable the card designer to control the data bus during the idle periods. The default values will configure the Local bus data pins to remain forcing (LT2[7:4] = Fh). LT[15:8] is programmed to place the bus in high-impedance at the beginning of a read cycle and set it back to forcing at the end of the read cycle. For systems that require the data bus to stay in high-impedance, the card designer should write an appropriate value in the range of 0h to Ah to LT2[7:4]. This will place the data bus in high impedance at the end of the write cycle. Whenever the value programmed in LT2[7:4] does not equal Fh, the Local bus controller will ignore the setting of LT2[15:8] as the data bus will be high-impedance outside write cycles. In this case the card designer should place external pull-ups on the data bus pins LBD[7:0] (or LBD[32:0] in 32-bit mode).

While the configuration data is read from the external EEPROM, the LBD pins remain in the high-impedance state.
The timing registers define the Local bus timing parameters based on signal changes relative to a reference cycle which is defined as two PCI clock cycles after IRDY# is asserted for the first time in a frame. The following parameters are fixed relative to the reference cycle.

The Local bus address pins (LBA[7:0] in 8-bit Local bus, LBA[15:0] in 32-bit Local bus) are asserted during the reference cycle. In a write operation, the Local bus data is available during the reference cycle, however I/O buffers change direction as programmed in LT2[3:0]. In a Motorola type bus write operation, the Read-not-Write pin (LBRDWR#) is asserted (low) during the reference cycle. In a read cycle this pin remains high throughout the duration of the operation.

The default settings in LT1 & LT2 registers provide one PCI clock cycle for address and chip-select to control signal set-up time, one clock cycle for address and chip-select from control signal hold time, two clock cycles of pulse duration for read and write control signals and one clock cycle for data bus hold time. These parameters are acceptable for using external OX16C950, OX16C952 and OX16C954 devices connected to the Local bus, in Intel mode. Some redefinition will be required if the bus is to be operated in Motorola mode.

The user should take great care when programming the Local bus timing parameters. For example defining a value for chip-select assertion which is larger that the value defined for chip-select de-assertion or defining a chip-select assertion value which is greater than control signal assertion will result in obvious invalid local Bus cycles.

4.4.5 Global Interrupt Status and Control Register 'GIS' (Offset 0x1C)

Bits	Description	Read/Write		Reset
		EEPROM	PCI	
3:0	Reserved	-	R	0x0h
4	MIO0 (Mode[1:0]≠'01'). This bit reflects the state of the internal MIO[0]. The internal MIO[0] reflects the non-inverted or inverted state of MIO0 pin.[1] Parallel Port Interrupt (Mode[1:0]='01'). This bit reflects the state of the Parallel Port internal interrupt line.	- -	R R	X 0
15:5	These bits reflect the state of the internal MIO[11:2]. The internal MIO[11:2] reflect the non-inverted or inverted state of MIO[11:2] pins respectively.[1]	-	R	XXXh
19:16	Reserved.	-	R	Fh
20	MIO[0] Interrupt Mask (Mode[1:0]≠'01'). When set (=1) this bit enables MIO0 pin to assert a PCI interrupt. When cleared (=0) it prevents MIO0 pin from asserting a PCI interrupt.[1] Parallel Port Interrupt Mask (Mode[1:0]='01'). When set (=1) this bit enables the Parallel Port to assert a PCI interrupt. When cleared (=0) it prevents the Parallel Port from asserting a PCI interrupt.	W W	RW RW	1 1
31:21	MIO Interrupt Mask. When set (=1) these bits enable each MIO[11:1] pin to assert a PCI interrupt respectively. When cleared (=0) they prevent the respective pins from asserting a PCI interrupt.[1]	W	RW	7FFh

Note 1: The returned value is either the direct state of the corresponding MIO pin or its inverse as configured by the Multi-purpose I/O Configuration register 'MIC' (offset 0x14). As the internal MIO can assert a PCI interrupt, the inversion feature can define each external interrupt to be defined as active-low or active-high, as controlled by the MIC register.

4.5 PCI Interrupts

Interrupts in PCI systems are level-sensitive and can be shared. Interrupts can be triggered by the Parallel port, or from the MIO pins when using the Local bus function. The Parallel Port and MIO0 share the same interrupt status bit (GIS[4]).

All interrupts in the OX9160 are routed to the PCI interrupt pin, INTA#. 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 each 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 the sources of interrupt is available using the GIS register in the Local Configuration Register set, which can be accessed using I/O or Memory operations. This facility allows a device driver to quickly ascertain the source of interrupt and service it. The OX9160 also offers additional interrupt masking ability using GIS[31:20] (see section 4.4.5), allowing drivers to temporarily disable certain sources of interrupts

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 (Note: this effectively disables the MIO[0] interrupt).

4.6 Power Management

The OX9160 is compliant with PCI Power Management Specification Revision 1.0. The local bus can recognise 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 device can assert to the PME# pin to request that the system 'wake up.' 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 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 systemboard implementations, use of an isolator FET is recommended. If Power Management capabilities are not required, the PME# pin can be treated as no-connect.

The power-down request for the Local bus is application-dependent. The device driver can use any of the multi-purpose I/O lines, MIO[12:3] to issue a power-down request.

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

- The external XTL_Ck_Out pin is disabled regardless of the programmed value in LCC[2].
- The Local bus clock pin, LBCK, is disabled regardless of the programmed value in LT2[30].
- The PCI interrupt is disabled.
- Access to I/O or Memory BARs is disabled.

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

Devices on the local bus can issue a wake up request by using the MIO2 pin. When LCC[7] is set, a rising or falling edge of MIO2 will cause the OX9160 to issue a wake up request by setting PME_Status = (PMCSR[15]), if it is enabled by PMCSR[8]. When LCC[7] is set, the MIO2 pin will remain in input mode regardless of the value programmed in MIC[5], However MIC[4] still controls the input sense. PME_Status is a sticky bit which will be cleared by writing a '1' to it. While the PME_En (PMCSR[8]) bit is set, PME_Status will assert the PME# pin to inform the device driver that a power management wake up event has occurred. After a wake up event is signalled, the device driver is expected to return the function to the D0 power-state. Settings for wake up events are shown in Table 7.

LCC[7]	MIC[4]	MIO2 Rising	MIO2 Falling	PME_Status
0	X	X	X	Remains unchanged
1	0	yes	X	Gets set
1	0	no	X	Remains unchanged
1	1	X	Yes	Gets set
1	1	X	No	Remains unchanged

Table 7: Local bus Wake-up configuration

5 LOCAL BUS

5.1 Overview

The OX9160 incorporates a bridge from PCI to the Local bus. It allows card developers to easily migrate legacy soluations to PCI.

When Mode[1:0] is '00', the Local bus is comprised of a bi-directional 8-bit data bus, an 8-bit address bus, up to four chip selects, and a number of control signals that allow for easy interfacing to standard peripherals. It also provides twelve active-high or active-low interrupt inputs.

When Mode[1:0] is '11', the Local bus is comprised of a bi-directional 32-bit data bus, a 12-bit address bus, up to four chip selects, and the same control signals and interrupts as in 8-bit mode.

The local bus is configured by LT1 and LT2 (see sections 4.4.3 & 4.4.4) in the Local Configuration Register space. By programming these registers the card developer can alter the characteristics of the local bus to suit the characteristics of the peripheral devices being used.

5.2 Operation

The local bus can be accessed via I/O and memory space, at addresses defined by BAR 0 and BAR 1 of configuration space. The mapping to the devices will vary with the application, but the bus is fully configurable to facilitate simple development.

The operation of the local bus is synchronised to the PCI bus clock. A buffered PCI clock signal is output on pin LBCLK if it has been enabled by setting LT2[30].

The eight bit bi-directional pins LBD[7:0] (LBD[31:0] in 32-bit mode) drive the output data onto the bus during local bus write cycles. For reads, the device latches the data read from these pins at the end of the cycle.

The local bus address is placed on pins LBA[7:0] (LBA[11:0] in 32-bit mode) at the start of each local bus cycle and will remain latched until the start of the subsequent cycle. If the maximum allowable block size (256 bytes) is allocated to the local bus in I/O space, then as access in I/O space is byte aligned, AD[7:0] are asserted on LBA[7:0]. If a smaller address range is selected, the corresponding upper address lines will be set to logic zero.

The control bus is comprised of up to four chip-select signals LBCS[3:0]#, a read strobe LBRD# and a write strobe LBWR#, in Intel-type interfaces. For Motorola-type interfaces, LBWR# is re-defined to perform read/write control signal (LBRDWR#) and the chip-select signals (LBCS[3:0]#) are re-defined to data-strobe (LBDS[3:0]#).

A reference cycle is defined, as two PCI clock cycles after the master asserts the IRDY# signal for the first time within a frame. In general, all the local bus control signals change state in the first cycle after the reference cycle, with offsets to provide suitable setup and hold times for common peripheral devices. However, all the timings can be increased / decreased independently in multiples of PCI clock cycles. This feature enables the card designer to override the length of read or write operations, the address and chip-select set-up and hold timing, and the data bus hold timing so that add-in cards can be configured to suit different speed peripheral devices connected to the Local bus. The designer can also program the data bus to remain in the high impedance state or actively drive the bus during idle periods.

The local bus will always return to an idle state, where no chip-select (data-strobe in Motorola mode) signal is active, between adjacent accesses. During read cycles the local bus interface latches data from the bus on the rising edge of the clock where LBRD# (LBDS[3:0]# in Motorola mode) goes high. Card designers should ensure that their peripherals provide the OX9160 with the specified data set-up and hold times with respect to this clock edge.

The local bus cannot accept burst transfers from the PCI bus. If a burst transfer is attempted the PCI interface will signal 'disconnect with data' on the first data phase. The local bus does accept 'fast back-to-back' transactions from PCI.

A PCI target must complete the transaction within 16 PCI clock cycles from assertion of the FRAME# signal, otherwise it should signal a retry. During a read operation from the Local bus, OX9160 waits for master-ready signal (IRDY#) and computes the number of remaining cycles to the de-assertion of the read control signal. If the total number of PCI clock cycles for that frame is greater than 16 clock cycles, OX9160 will post a retry. The master would normally return immediately and complete the operation in the following frame.

5.3 Configuration & Programming

The configuration registers for the local bus controller are described in sections 4.4.3 & 4.4.4. The values of these registers after reset allow the host system to identify the function and configure its base address registers. Alternatively many of the default values can be re-programmed during device initialisation through use of the optional serial EEPROM (see section 7).

The I/O space block can be varied in size from 4 bytes to 256 bytes (32 bytes is the default) by setting LT2[22:20] accordingly. Varying the block size means that I/O space can be allocated efficiently by the system, whatever the application.

The I/O block can then be divided into one, two or four chip-select regions, depending on the setting in LT2[26:23]. To divide the area into four chip-select region, the user should select the second uppermost non-zero address bit as the Lower-Address-CS-decode. To divide into two regions, the user should select the uppermost address bit. If an address bit beyond the selected range is selected, the entire I/O space is allocated to CS0#. For example, if 32 bytes of I/O space are reserved, the active address lines are A[4:0]. To divide this into four regions, the Lower Address CS parameter should be set to A3, by programming the value '0001' into LT2[26:23]. To select two regions, choose A4, and to maintain one region, select any value greater than A4.

In 8-bit mode, the memory space block is always 4K bytes, and always divided into four chip-select regions of 1K byte each.

In 32-bit mode, again the I/O space can be varied in size from 4 bytes to 256 bytes. It is also possible to increase the memory space block size from 4K bytes to 16K bytes. Also in 32-bit mode, the Lower-Address-CS-Decode parameter afftects division of the I/O space AND memory space into chip-select regions.

A soft reset facility is provided so software can independently reset the peripherals on the local bus. The local bus reset signals, LBRST and LBRST#, are always active during a PCI bus reset and also when the configuration register bit LT2[29] is set to 1.

5.4 Clock references

The clock enable bit LT2[30], when set enables a copy of the PCI bus clock output on the local bus pin LBCLK. If a reference clock is desired this signal should be used as splitting the PCICLK signal will violate the PCI specification.

A buffered crystal clock can also be asserted on the XTL_Ck_Out pin; this means that a single oscillator can be used to drive devices such as UARTs on the local bus. To make use of the XTL_Ck_Out function, a crystal oscillator circuit should be connected to the XTLI and XTLO pins, as shown in Figure 3.

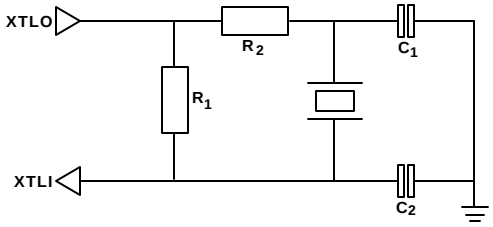


Figure 3: Crystal Oscillator Circuit

Frequency Range (MHz)	C_1 (pF)	C_2 (pF)	R_1 (W)	R_2 (W)
1-8	68	22	220k	470R
8-60	33-68	33 – 68	220k-2M2	470R

Table 8: Component values

Note: For better stability use a smaller value of R_1. Increase R_1 to reduce power consumption.
The total capacitive load (C1 in series with C2) should be that specified by the crystal manufacturer (nominally 16pF).

6 BIDIRECTIONAL PARALLEL PORT

6.1 Operation and Mode selection

The OX16PCI954 offers a compact, low power, IEEE-1284 (EPP-only) compliant host-interface parallel port, designed to interface to many peripherals such as printers, scanners and external drives. It supports compatibility modes, SPP, NIBBLE and PS2, as well as EPP mode. The register set is compatible with the Microsoft® register definition. To enable the parallel port function, the Mode[1:0] pins should be set to '01'. The system can access the parallel port via two 8-byte blocks of I/O space; BAR0 contains the address of the basic parallel port registers, BAR1 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.

6.1.1 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 to access the parallel port. The default configuration allocates 8 bytes to BAR0 in I/O space.

6.1.2 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.

6.1.3 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 rgisters. 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.

6.1.4 ECP mode (not supported)

The Extended Capabilities Port 'ECP' mode is not supported.

6.2 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]).

6.3 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							
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							
-	400h	-	Reserved							
-	401h	-	Reserved							
ECR	402h	R/W	Mode[2:0]			Reserved – Must write '00001'				
-	403h	-	Reserved							

Table 9: 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.

6.3.1 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.

6.3.2 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 6.1.3)

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.

6.3.3 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 inactive.

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".

6.3.4 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.

6.3.5 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.

6.3.6 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
These bits are reserved and must always be set to "00001".

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

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

7 SERIAL EEPROM

7.1 Specification

The OX9160 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 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 OX9160 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. 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.

The OX9160 requires a total of bytes of EEPROM data to program all the EEPROM writable registers. The 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.

7.2 EEPROM Data Organisation

The serial EEPROM data is divided in four 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 OX9160 registers from the serial EEPROM. The general EEPROM data structure is shown in Table 10.

DATA Zone	Size (Words)	Description
0	One	Header
1	One or more	Local Configuration Registers
2	Two	Identification Registers
3	Two or more	PCI Configuration Registers

Table 10: EEPROM data format

7.2.1 Zone0: Header

The header identifies the EEPROM program as valid.

Bits	Description
15:4	These bits should return 0x950 to identify a valid program. Once the OX9160 reads 0x950 from these bits, it sets LCC[28] to indicate that a valid EEPROM program is present.
3	Reserved. Write '0' to this bit.
2	1 = Zone1 (Local Configuration) exists 0 = Zone1 does not exist
1	1 = Zone2 (Identification) exists 0 = Zone2 does not exist
0	1 = Zone3 (PCI Configuration) exists 0 = Zone3 does not exist

The programming data for each zone follows the proceeding zone if it exists. For example a Header value of 0x9507 indicates that all zones exist and they follow one another in sequence, while 0x9505 indicates that only Zones 1 and 3 exist where the header data is followed by Zone1 WORDs, and since Zone2 is missing Zone1 WORDs are followed by Zone3 WORDs.

7.2.2 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 1.1).

Note: Not all of the registers in the Local Configuration Register set are writable by EEPROM. If bit2 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 11.

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 11: Zone 1 data format

Table 12 shows which Local Configuration registers are writable from the EEPROM. Note that an attempt by the EEPROM to write to any other offset locations can result in unpredictable behaviour.

Offset	Bits	Description	Reference
0x00	1:0	Must be '00'.	
0x00	2	Enable crystal clock output.	LCC[2]
0x00	4:3	Endian byte-lane select.	LCC[4:3]
0x00	6:5	Power-down filter.	LCC[6:5]
0x00	7	MIO2_PME enable.	LCC[7]
0x04	7:0	Multi-purpose IO configuration.	MIC[7:0]
0x05	7:0	Multi-purpose IO configuration.	MIC[15:8]
0x06	7:0	Multi-purpose IO configuration.	MIC[23:16]
0x08	7:0	Local bus timing parameters	LT1[7:0]
0x09	7:0	Local bus timing parameters	LT1[15:8]
0x0A	7:0	Local bus timing parameters	LT1[23:16]
0x0B	7:0	Local bus timing parameters	LT1[31:24]
0x0C	7:0	Local bus timing parameters	LT2[7:0]
0x0D	7:0	Local bus timing parameters	LT2[15:8]
0x0E	3:0	Must be '0000'.	
0x0E	6:4	IO Space Block size.	LT2[22:20]
0x0E	7	Lower-Address-CS-Decode.	LT2[23]
0x0F	2:0	Lower-Address-CS-Decode.	LT2[26:24]
0x0F	4:3	Memory space block size in 32-bit Local bus.	LT2[28:27]
0x0F	5	Must be '0'.	
0x0F	6	Local bus clock enable.	LT2[30]
0x0F	7	Bus interface type.	LT2[31]
0x1E	4	MIO0/Parallel Port interrupt mask	GIS[20]
0x1E	7:5	Multi-purpose IO interrupt mask.	GIS[23:21]
0x1F	7:0	Multi-purpose IO interrupt mask.	GIS[31:24]

Table 12: EEPROM-writable Local Configuration Registers

7.2.3 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 13.

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 Zone1 (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 13: Zone 2 data format

7.2.4 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, one or more configuration WORDs, and a function end WORD. The function header WORD is defined as 0x8001, and the end WORD is defined as 0x0000.

The data between the function header WORD and the end WORD consists of one or more configuration WORDs, which contain the address offset and a byte of data for the PCI Configuration Space of the function. The format of configuration WORDs for the PCI Configuration Registers are described below.

Bits	Description
15	'0' = This is the last configuration WORD. '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 Extended Capabilities register is 0x06. Offset values are tabulated in section 4.2.
7:0	8-bit value of the register to be programmed

Table 14: Zone 3 data format (data)

Table 15 shows which PCI Configuration registers are writable from the EEPROM.

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.
0x42	7:0	Power Management Capabilities bits 7 to 0.
0x43	7:0	Power Management Capabilities bits 15 to 8.

Table 15: EEPROM-writable PCI configuration registers

Table 16 shows an example zone 3 which sets the device ID with two configuration words.

Description	Data
Function header word	0x8001
Set Device ID[7:0] – one more config word	0x82CD
Set Device ID[15:8] – no more config words	0x03AB
end WORD	0x0000

Table 16: Example zone 3 data

8 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 17: Absolute maximum ratings

Symbol	Parameter	Min	Max	Units
V_{DD}	DC supply voltage	4.5	5.5	V
T_C	Commercial temperature	0	70	°C
T_I	Industrial temperature	-40	85	°C

Table 18: Recommended operating conditions

9 DC ELECTRICAL CHARACTERISTICS

9.1 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	35	48.1	mA
	Operating supply current in Power-down mode	2	34.4	

Table 19: 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

9.2 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 20: 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

10 AC ELECTRICAL CHARACTERISTICS

10.1 PCI Bus

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

10.2 Local bus

By default, the Local bus control signals change state in the cycle immediately following the reference cycle, with offsets to provide setup and hold times for common peripherals in Intel mode. The tables below show these default values; however each of these can be increased or decreased by an number of PCI clock cycles by adjusting the parameters in registers LT1 and LT2.

Symbol	Parameter	Min	Max	Units
t_{ref}	IRDY# falling to reference LBCLK	Nominally 2 PCI clock cycles		
t_{za}	Reference LBCLK to Address Valid	TBD	TBD	ns
t_{ard}	Address Valid to LBRD# falling	TBD	TBD	ns
t_{zrcs1}	Reference LBCLK to LBCS# falling	TBD	TBD	ns
t_{zrcs2}	Reference LBCLK to LBCS# rising	TBD	TBD	ns
t_{csrd}	LBCS# falling to LBRD# falling	TBD	TBD	ns
t_{rdcs}	LBRD# rising to LBCS# rising	TBD	TBD	ns
t_{zrd1}	Reference LBCLK to LBRD# falling	TBD	TBD	ns
t_{zrd2}	Reference LBCLK to LBRD# rising	TBD	TBD	ns
t_{drd}	Data bus floating to LBRD# falling	TBD	TBD	ns
t_{zd1}	Reference LBCLK to data bus floating at the start of the read transaction	TBD	TBD	ns
t_{zd2}	Reference LBCLK to data bus driven by OX9160 at the end of the read transaction	TBD	TBD	ns
t_{sd}	Data bus valid to LBRD# rising	TBD	TBD	ns
t_{hd}	Data bus valid after LBRD# rising	TBD	TBD	ns

Table 21: Read operation from Intel-type Local bus

Symbol	Parameter	Min	Max	Units
t_{ref}	IRDY# falling to reference LBCLK	Nominally 2 PCI clock cycles		
t_{za}	Reference LBCLK to Address Valid	TBD	TBD	ns
t_{awr}	Address Valid to LBWR# falling	TBD	TBD	ns
t_{zwcs1}	Reference LBCLK to LBCS# falling	TBD	TBD	ns
t_{zwcs2}	Reference LBCLK to LBCS# rising	TBD	TBD	ns
t_{cswr}	LBCS# falling to LBWR# falling	TBD	TBD	ns
t_{wrcs}	LBWR# rising to LBCS# rising	TBD	TBD	ns
t_{zwr1}	Reference LBCLK to LBWR# falling	TBD	TBD	ns
t_{zwr2}	Reference LBCLK to LBWR# rising	TBD	TBD	ns
t_{zdv}	Reference LBCLK to data bus valid	TBD	TBD	ns
t_{zdf}	Reference LBCLK to data bus high-impedance	TBD	TBD	ns
t_{wrdi}	LBWR# rising to data bus invalid	TBD	TBD	ns

Table 22: Write operation to Intel-type Local bus

Symbol	Parameter	Min	Max	Units
t_{ref}	IRDY# falling to reference LBCLK	Nominally 2 PCI clock cycles		
t_{za}	Reference LBCLK to Address Valid	TBD	TBD	ns
t_{ads}	Address Valid to LBDS# falling	TBD	TBD	ns
t_{zrds1}	Reference LBCLK to LBDS# falling	TBD	TBD	ns
t_{zrds2}	Reference LBCLK to LBDS# rising	TBD	TBD	ns
vt_{drd}	Data bus floating to LBDS# falling	TBD	TBD	ns
t_{zd1}	Reference LBCLK to data bus floating at the start of the read transaction	TBD	TBD	ns
t_{zd2}	Reference LBCLK to data bus driven by OX9160 at the end of the read transaction	TBD	TBD	ns
t_{sd}	Data bus valid to LBDS# rising	TBD	TBD	ns
t_{hd}	Data bus valid after LBDS# rising	TBD	TBD	ns

Table 23: Read operation from Motorola-type Local bus

Symbol	Parameter	Min	Max	Units
t_{ref}	IRDY# falling to reference LBCLK	Nominally 2 PCI clock cycles		
t_{za}	Reference LBCLK to Address Valid	TBD	TBD	ns
t_{ads}	Address Valid to LBDS# falling	TBD	TBD	ns
t_{zw1}	Reference LBCLK to LBRDWR# falling	TBD	TBD	ns
t_{zw2}	Reference LBCLK to LBRDWR# rising	TBD	TBD	ns
t_{wds}	LBRDWR# falling to LBDS# falling	TBD	TBD	ns
t_{dsw}	LBDS# rising to LBRDWR# rising	TBD	TBD	ns
t_{zwds1}	Reference LBCLK to LBDS# falling	TBD	TBD	ns
t_{zwds2}	Reference LBCLK to LBDS# rising	TBD	TBD	ns
t_{zdv}	Reference LBCLK to data bus valid	TBD	TBD	ns
t_{zdf}	Reference LBCLK to data bus high-impedance	TBD	TBD	ns
t_{dsdi}	LBDS# rising to data bus invalid	TBD	TBD	ns

Table 24: Write operation to Motorola-type Local bus

11 TIMING WAVEFORMS

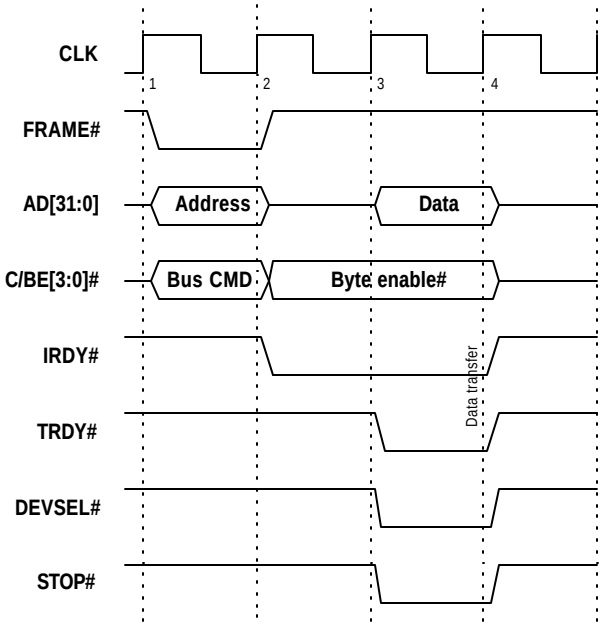


Figure 4: PCI Read transaction from Local Configuration registers

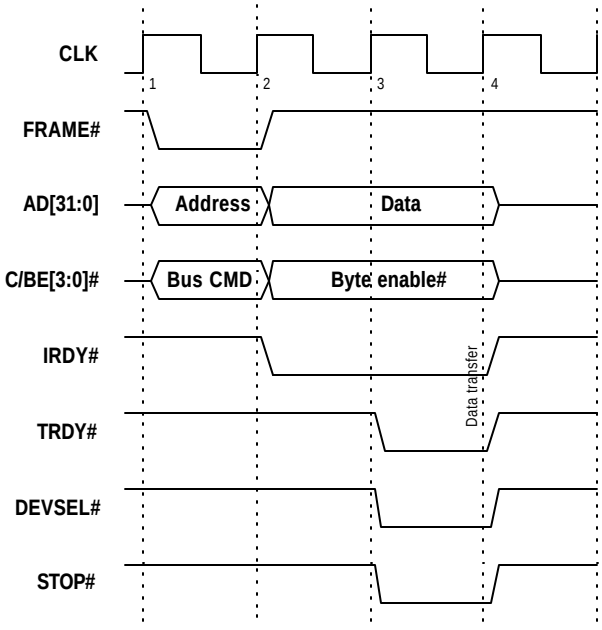


Figure 5: PCI Write transaction to Local Configuration Registers

TIMING WAVEFORMS

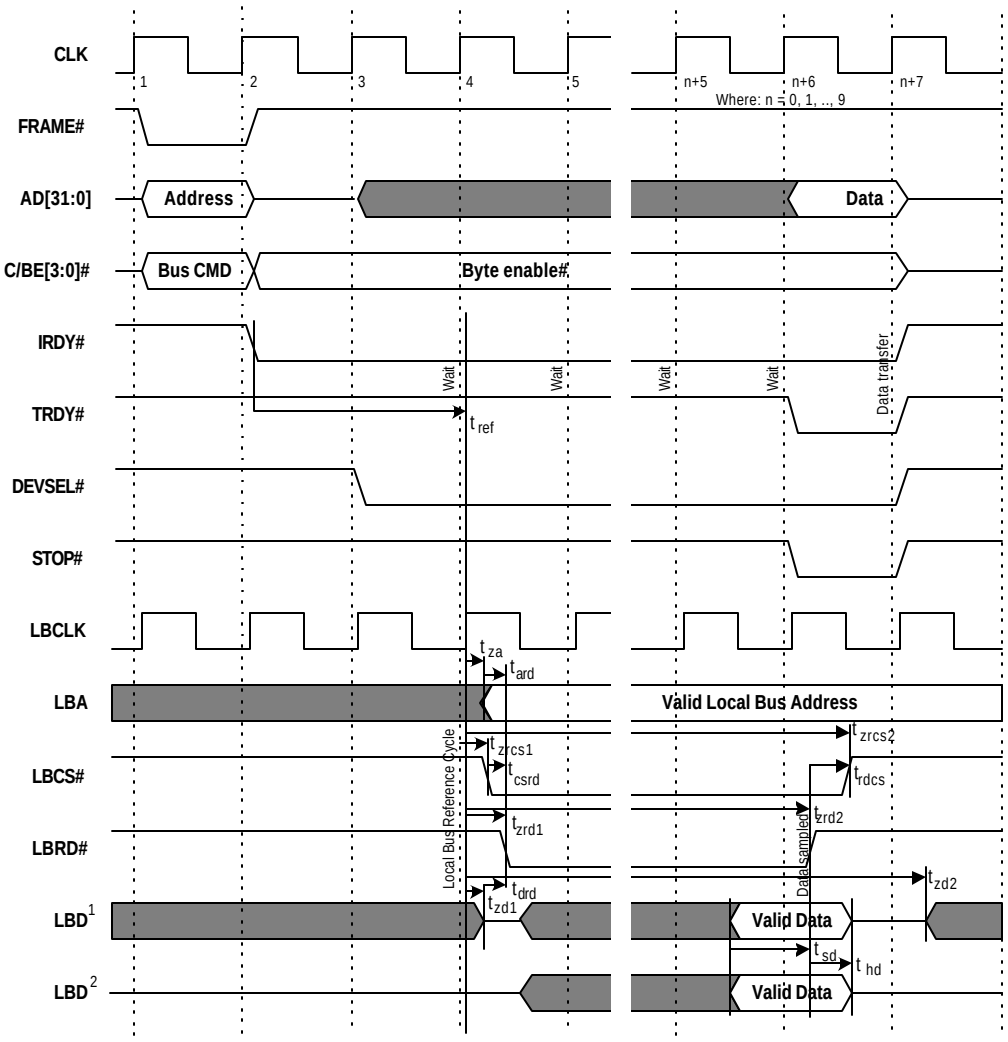


Figure 6: PCI Read Transaction from Intel-type Local bus

TIMING WAVEFORMS

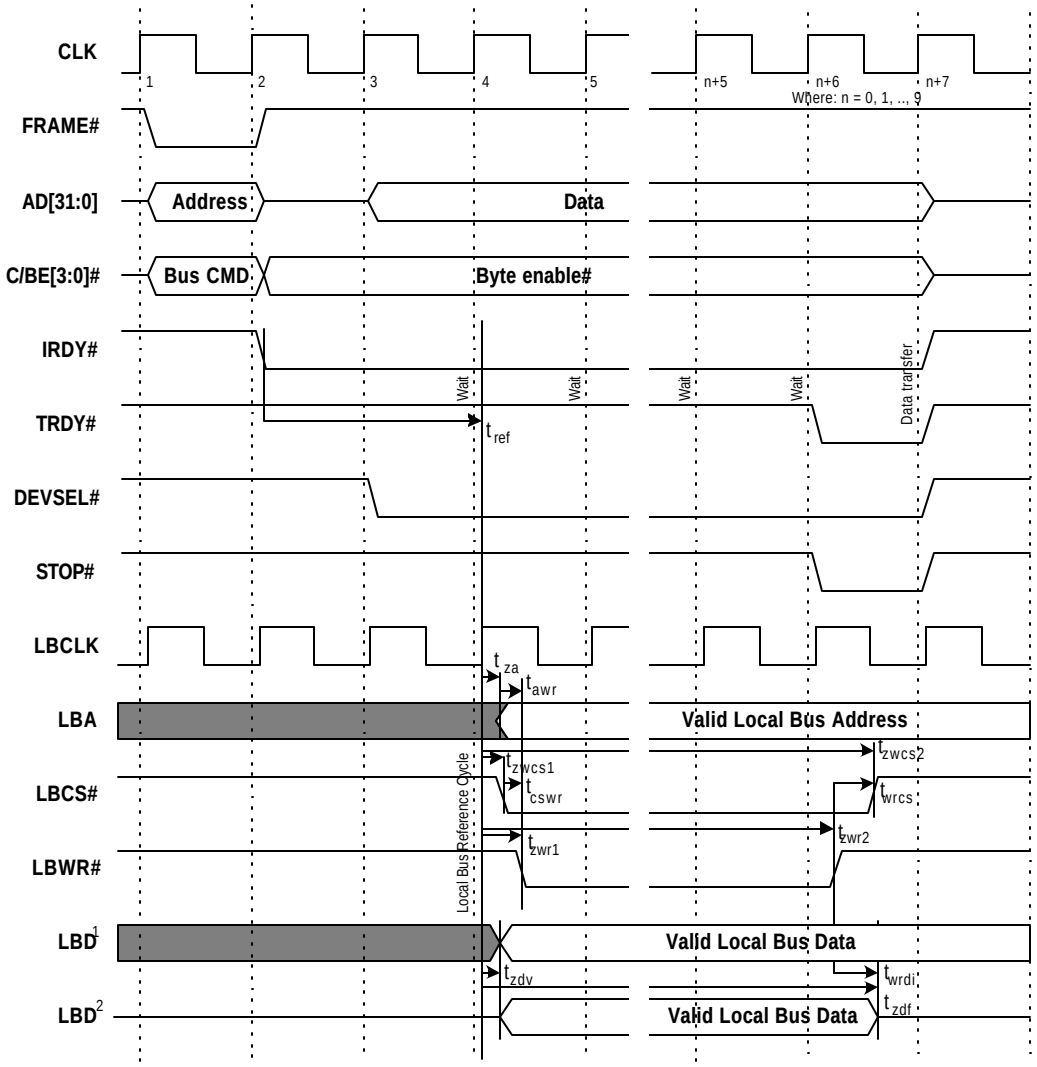


Figure 7: PCI Write Transaction to Intel-type Local bus

TIMING WAVEFORMS

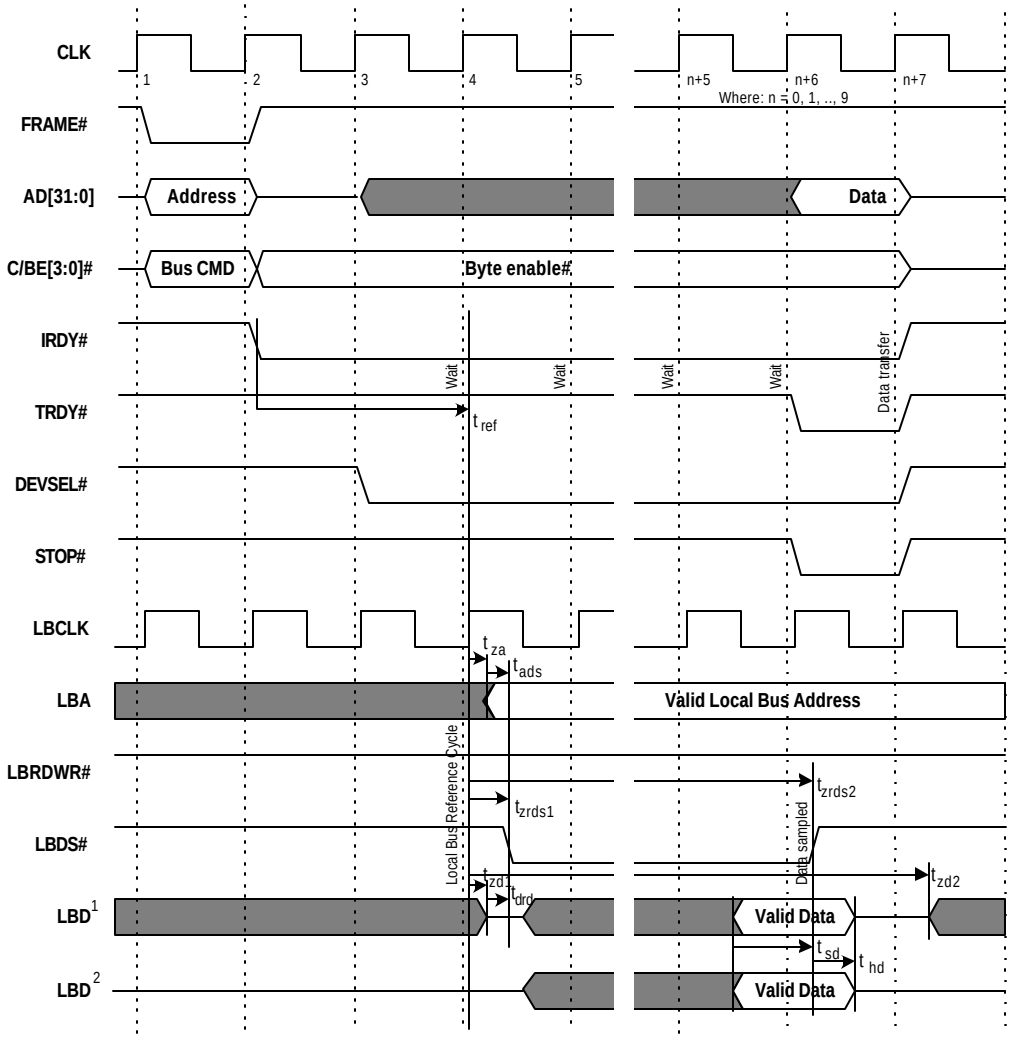


Figure 8: PCI Read Transaction from Motorola-type Local bus

TIMING WAVEFORMS

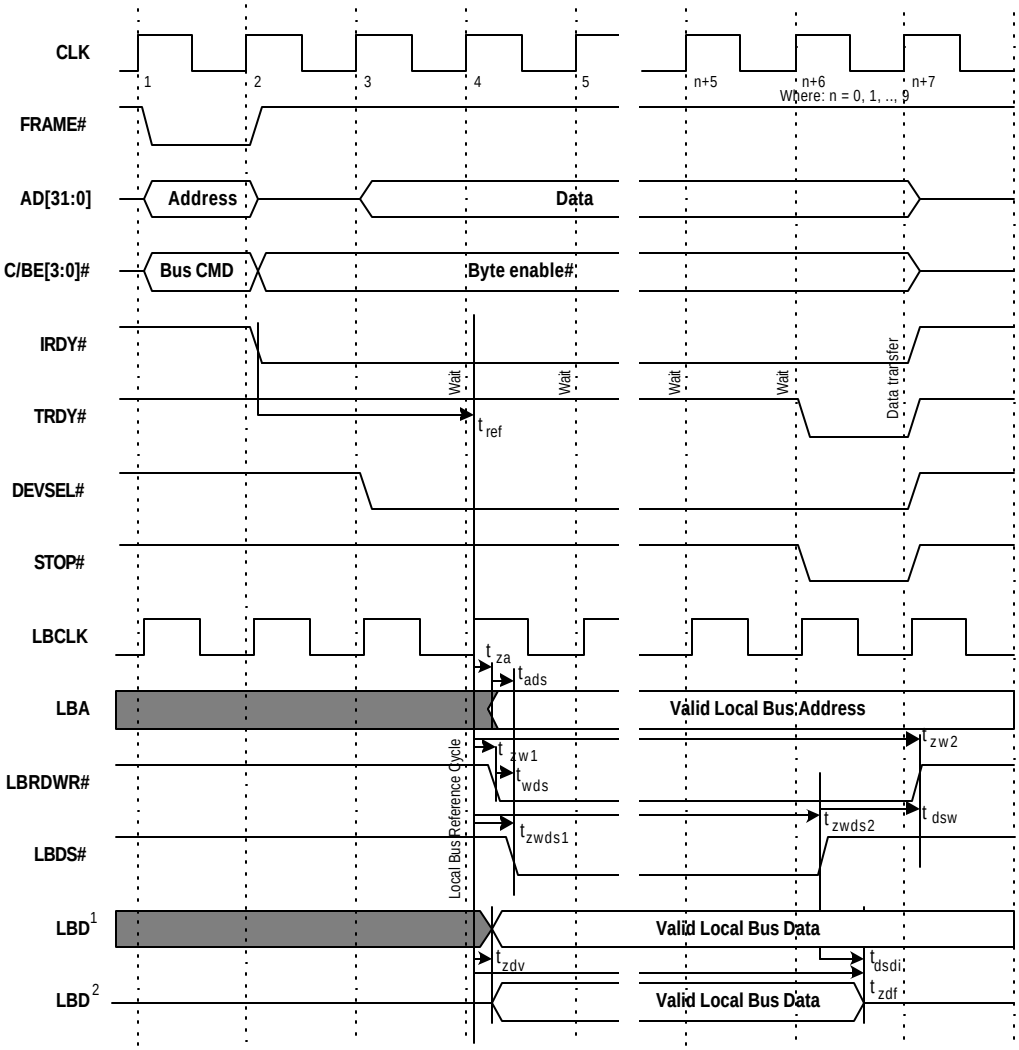


Figure 9: PCI Write Transaction to Motorola-type Local bus

12 PACKAGE INFORMATION

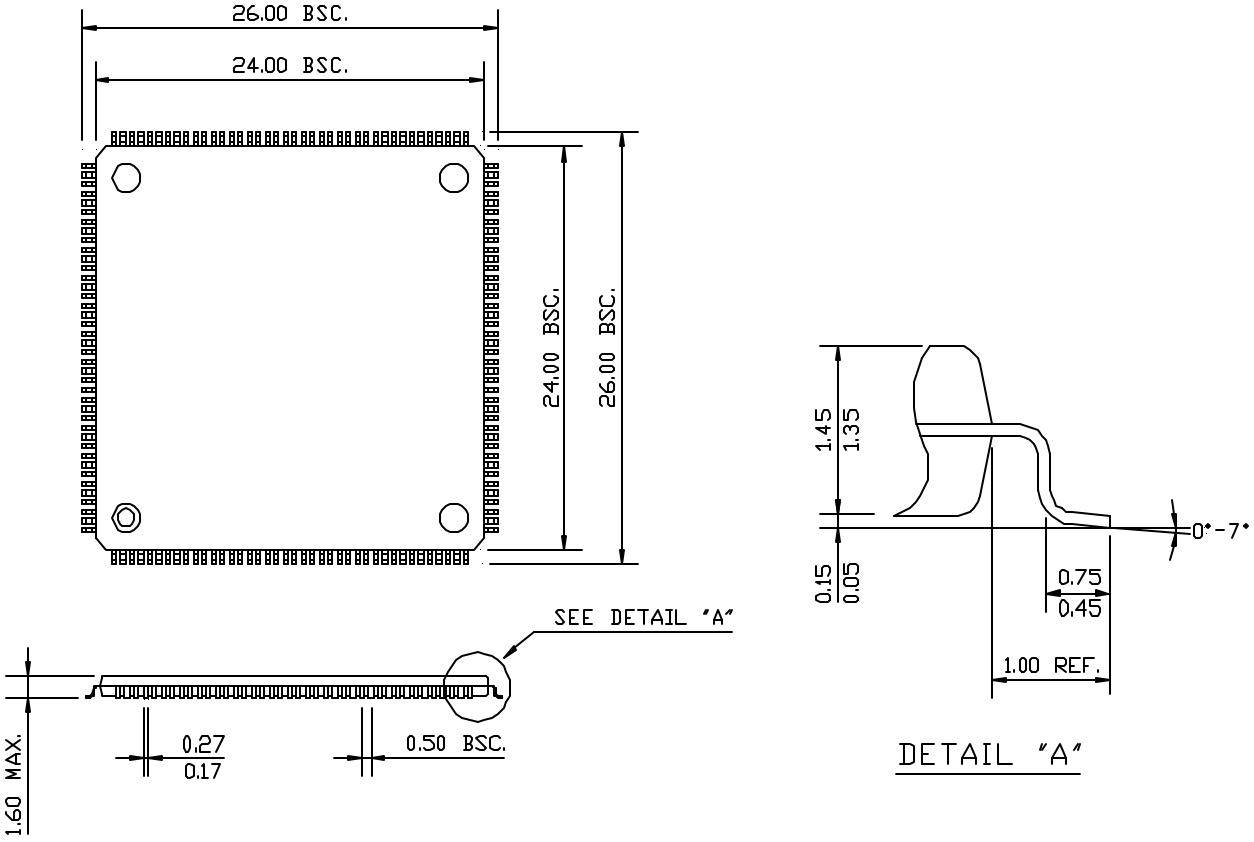


Figure 10: 160 pin Thin Quad Flat Pack (TQFP) package

13 ORDERING INFORMATION

OX9160-TQC33-A

Revision

Package Type – 160 TQFP

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
Web site: http://www.oxsemi.com

DISCLAIMER