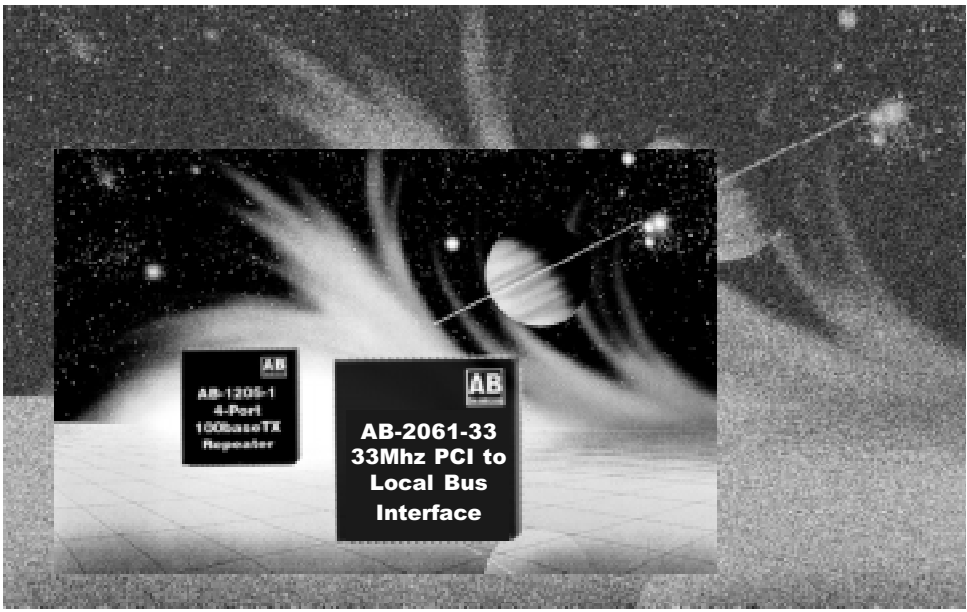
AB-2061G-33

33Mhz PCI to Local Bus Interface for 8 or 16 bit Local Bus Applications

32- to 16- bit PCI to local bus translator



Product Specification

AB Semicon AB-2061-33

32 bit to 8 or 16 bit PCI to
local bus translator
Product Specification

Disclaimer

AB Semicon Limited reserves the right to revise this publication and to make changes from time to time to the contents hereof without obligation to notify any person or organization of such revision or changes. AB Semicon Limited has endeavoured to ensure that the information in this publication is correct, but will not accept liability for any error or omission.

AB-2061-33

33Mhz PCI to Local Bus Interface

Features:

* High speed 33Mhz
* Ideal for interfacing 32-bit PCI bus to 8 or 16-bit local bus systems
* Mailbox (Door Bell) and Interrupt
* 100-pin PQFP package
* Low Power C-Mos Technology 0.6micron
* Byte aligned transfers
* Enhanced interprocessor handshaking

Introduction

The AB-2061 device described in this document is intended to support a subset of the PCI bus specification. However this does not preclude the use of this device on standard PCI buses, as the electrical specifications are the same and the minimum feature set is supported.

The device contains a number of shared registers for interprocessor communication and mechanisms for generating interrupts to the PCI and local buses.

The device supports the transfer of blocks in a way which reduces load on the local processor and also makes efficient use of the PCI bus by maintaining sensible size bursts and reducing the number of single data phase transactions to a minimum.

In order to decouple the relative speed differences between the two buses, the device incorporates a pair of FIFOs - First In First Out memory buffers, one for each direction. Each FIFO can hold four 32-bit words.

There is no direct access to the local memory space provided by this device, and the PCI side cannot set up DMA transfers which are solely under the control of the local processor.

Communication between buses is via the 32 byte shared register block or the mailbox registers.

AB-2061-33 has a 3.3v PCI bus interface which will operate at a maximum frequency of 33MHz. The local processor bus can be run at either 3.3 or 5v depending upon the power supply.

Chip Structure

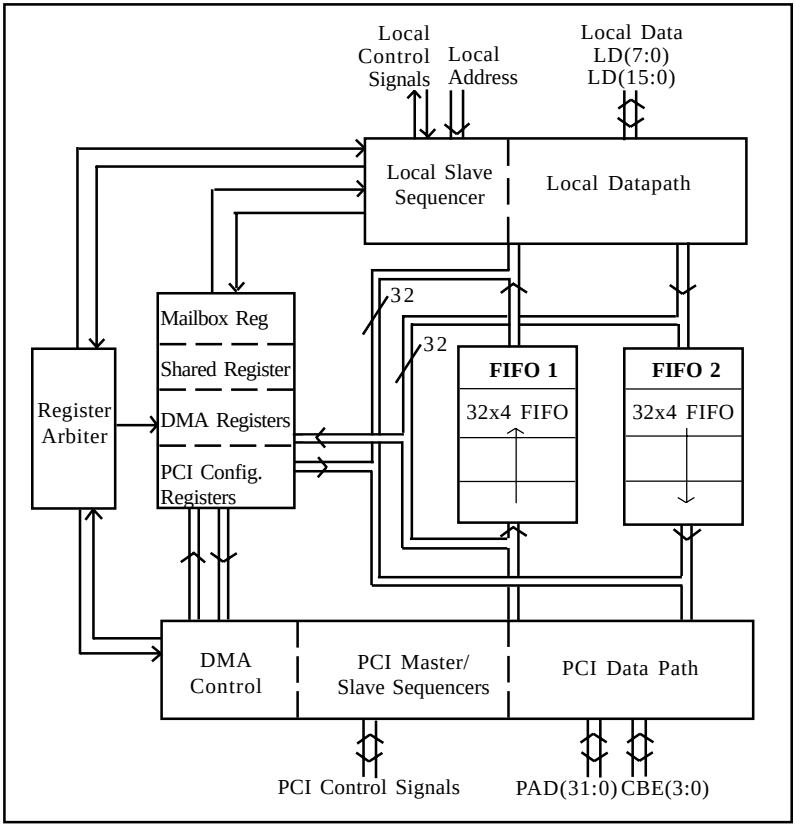


Figure 1
Functional Block Diagram

Applications

The chip can be used in many different applications where interfacing to a peripheral device such as a printer or a scanner is required. The device can be used to connect two processor systems and their respective buses with a high speed interface so that there is no degradation in processing time on either side of the systems by allowing fully asynchronous access from both sides of the system.

Figure 2 gives an overview of low-cost yet high-speed network interface applications for which this chip is suitable.

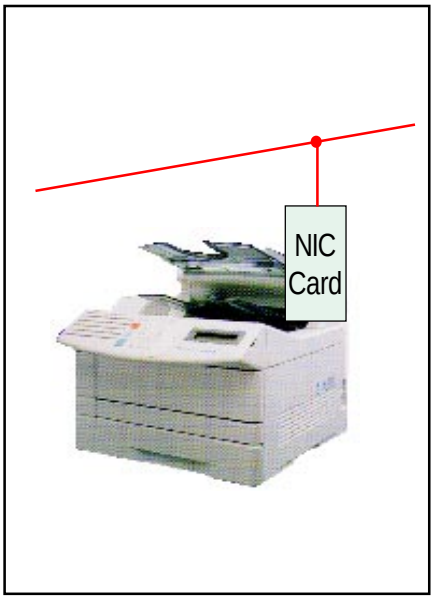


Figure 2

EXAMPLE for a Network Interface Card

Figure 3 shows a Network Interface with a bus connector to interface to a printer controller board.

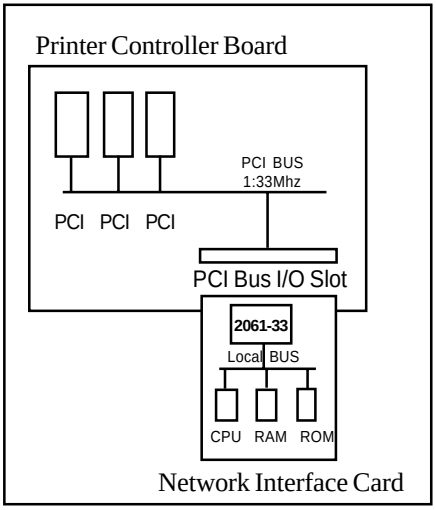
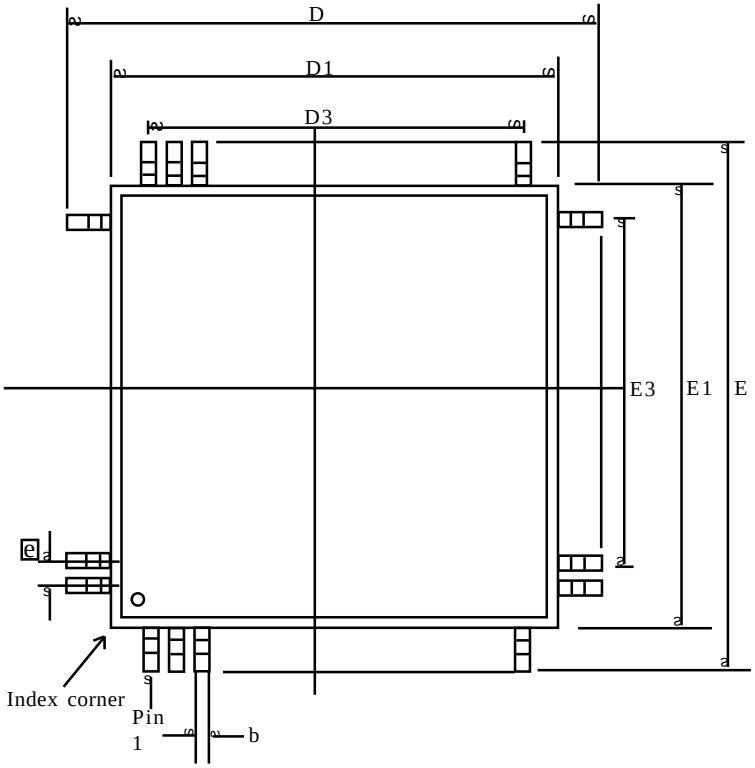


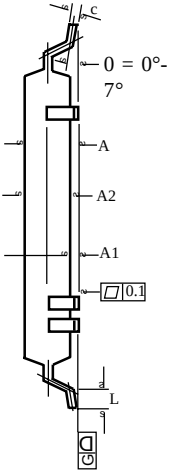
Figure 3

Pin-Out for AB-2061-33

The following diagram shows the pin-out for the AB-2061-33 PCI to local bus interface chip:



Packaging Information



Symbol	Control Dimensions in millimetres			Alternative Dimensions in inches		
	MIN	Nominal	MAX	MIN	Nominal	MAX
A	2.80		3.40	0.110		0.134
A1	0.25		0.85	0.010		0.033
A2	2.55		3.05	0.100		0.120
D	23.65		24.15	0.931		0.951
D1	19.80		20.20	0.780		0.795
D3	18.85 REF.			0.742 REF.		
E	17.65		18.15	0.695		0.715
E1	13.80		14.20	0.543		0.559
E3	12.35 REF.			0.486 REF.		
L	0.73		1.03	0.029		0.041
e	0.65 BSC.			0.026 BSC.		
b	0.22		0.38	0.009		0.015
c	0.11		0.23	0.004		0.009
	Pin features					
N	100					
ND	30					
NE	20					
NOTE	RECTANGULAR					

Conforms to JEDEC MO-112 CC-1 Iss. B.

◆ Note: This package is rectangular

Functional Description of Signals

PCI Signals		
PIN Name	**Function**	**Number of Pins**
PAD(31:0)	Multiplexed Address/Data	32
GNT/REQ	Bus Arbitration	2
CBE(3:0)	Command/Byte Enables	4
PRST	Reset	1
INTA	Interrupt	1
IRDY/TRDY	Initiator/Target Ready	2
DEVSEL	Device Select	1
IDSEL	Initialisation Select	1
SERR/PERR	Error Reporting	2
FRAME	Cycle start/running	1
STOP	Stop Transaction	1
PAR	Parity	1
PCLK	PCI Bus Clock	1
Local Bus Signals		
LD(16:0)	Data Bus	16 (8)
RD / R/W	Read Strobe / Data direction	1
WR / E	Write Strobe / Data Strobe	1
CS	Chip Select	1
ALE	Address Latch	1
BHE	Byte Enable for 16 bit bus	1
LA(3:0)	Address Bus	4
LINT	Local Interrupt	1
MODE(2:0)	Local Bus Mode	3
WAIT	Local Wait Signal	1

Total Active Signals : 80 - 16 Bit bus 72 - 8 Bit bus

On Chip Resources

Configuration Registers
The PCI specification requires a minimum set of configuration registers, taking up some 64 bytes. These registers are accessible from the PCI bus via configuration cycles, though not all are writable. The local processor also requires access to the configuration registers in order to set them up with the correct values.
These registers are specified fully in the PCI Rev 2.1 Documentation.

Shared Register Block
The chip contains a block of registers 32 bytes in length. These are accessible via PCI from the address held in the configuration register, Base Address 0 via normal read/write cycles. Again the local processor also has access to these registers. One of these holds the offset register which contains pointers to various on-chip resources.

DMA Control
The DMA controller transfers data in both directions between the card local memory and the peripheral memory, although only the local processor will need to co-ordinate the transfers and be able to see the DMA control registers. Using DMA in this manner makes it easier to transfer data on the PCI bus in bursts, and hence decrease the amount of bus bandwidth taken up by the card. This could be important for certain types of device which may not work if the PCI bus is heavily loaded. A size of 16 bytes (4 long words) is ideal. Due to the potential difference in clock speeds and bandwidths between the local PCI buses, FIFOs are necessary to allow this. The controller allows one transfer of non-longword aligned blocks by the use of the DMA_MASK register.

Mailbox Registers
There are two mailbox registers for interprocessor communications which generate interrupts to the relevant processor when read from or written to. These interrupts are individually maskable.

Accessing On Chip Registers

Most registers in the device are not directly accessible; an indirect scheme is used whereby the internal address of the register required is written to a REGISTER_ADDRESS register and data transferred to and from the register via the REGISTER_DATAn port. Some registers are directly accessible for reasons of speed.

The chip appears to the local processor to be a set of 16 8-bit ports starting at some system-defined base address. These are:

Address Base Offset	Name	Function
0	REGISTER_ADDRESS	Indirect address register, holds the address of the register to be accessed through REGISTER_DATA0..3
1	RESOURCE_STATUS (R)	Read : Returns status of DMA and FIFOs
	RESOURCE_CONTROL (W)	Write : DMA/PCI control bits
2	FIFO_DATA0	Read : Data from PCI through FIFO1
3	FIFO_DATA1 (16 bit only)	Write : Data to PCI through FIFO2
4	REGISTER_DATA0	Byte 0 of indirectly accessed register
5	REGISTER_DATA1	Byte 1 of indirectly accessed register
6	REGISTER_DATA2	Byte 2 of indirectly accessed register
7	REGISTER_DATA3	Byte 3 of indirectly accessed register

RESOURCE_CONTROL Register

Bit	Name	Function
0	DMA1_ENABLE	When set to 1 DMA channel #1 will start. Writing a 0 will stop and reset the channel.
1	DMA1_PAUSE	Writing a 1 will cause DMA on channel #1 to pause until a 0 is written when it will recommence.
2	DMA2_ENABLE	When set to 1 DMA channel #2 will start Writing a 0 will stop and reset the channel.
3	DMA2_PAUSE	Writing a 1 will cause DMA on channel #2 to pause until a 0 is written when it will recommence.
4	FIFO2-FLUSH	Setting this bit to a 1 will allow the PCI bus controller to perform bursts of less than four long words. Used to flush last bytes from the FIFO.
5	PROTECT_REGISTERS	When set to 1 some registers in the shared register block are write protected.
6	LOCK_REGISTERS	When 1 all PCI bus accesses to the shared register block will be terminated with RETRY. Set to 0 after reset.
7	CONFIG_ENABLE	When 0 all PCI configuration cycles will be terminated with RETRY. Writing 1 will allow configuration cycles to proceed normally. Set to 0 after a reset.

RESOURCE_STATUS Register

Bit	Name	Function
0	DMA1_COMPLETE	A 1 indicates that the last word in a block has been placed into FIFO 1 by DMA channel #1.
1	DMA2_COMPLETE	A 1 indicates that DMA CHANNEL #2 has transferred the last word in a block to PCI memory space.
2	FIFO1_EMPTY	PCI > Local FIFO is empty.
3	FIFO2_FULL	Local > PCI FIFO is full.
4	DMA1_ERROR	Unrecoverable error during a transfer on channel 1.
5	DMA2_ERROR	Unrecoverable error during a transfer on channel 2.
6	SPARE	Undefined.
7	SPARE	Undefined.

PCI Configuration Registers

These are accessed by the local processor through an indirect method using REGISTER_ADDRESS and REGISTER_DATAn. The PCI uses configuration cycles to get at these.

PCI Config. Addr.	Local Indirect Addr.	Name	Width	Function	Default Value
0	0(0,1)	Vendor ID	16	Interface Device Manufacturer ID code	1309h
2	0(3,2)	Device ID	16	Interface Device ID code	080Dh
4	1(0,1)	Command	16	PCI Command register	0000h
6	1(3,2)	Status	16	PCI Status register	0200h
8	2(0)	Revision ID	8	Interface Device Revision	00h
9	2(1)	Interface Register	8	Contains index to chip's offset register	18h
A	2(2)	Sub Class Code	8	PCI Sub Class code	80h
B	2(3)	Base Class Code	8	PCI Base Class code	FFh
C	3(0)	Cache Line Size	8	NOT IMPLEMENTED	00h
D	3(1)	Latency Timer	8	Maximum burst duration	00h
E	3(2)	Header Type	8	00h or 80h	00h
F	3(3)	BIST	8	NOT IMPLEMENTED	00h
10	4(3-0)	Base Address 0	32	Base addr. of card resources	00000**000**h
14	5(3-0)	Base Address 1	32	NOT IMPLEMENTED	**00000000h**
18	6(3-0)	Base Address 2	32	NOT IMPLEMENTED	**00000000h**
1C	7(3-0)	Base Address 3	32	NOT IMPLEMENTED	**00000000h**
20	8(3-0)	Base Address 4	32	NOT IMPLEMENTED	**00000000h**
24	9(3-0)	Base Address 5	32	NOT IMPLEMENTED	**00000000h**
28	A(3-0)	Cardbus CIS Pointer	32	NOT IMPLEMENTED	**00000000h**
2C	B(0,1)	Subsystem ID	16	Vendor assigned card ID no.	0000h
2E	B(3,2)	Subsystem Vendor ID	16	Card Manufacturer ID code	0000h
30	C(3-0)	ROM Base Address	32	NOT IMPLEMENTED	**00000000h**
34	D(3-0)	Reserved	32	NOT IMPLEMENTED	**00000000h**
38	E(3-0)	Reserved	32	NOT IMPLEMENTED	**00000000h**
3C	F(0)	Interrupt Line	8	Interrupt line routing	00h
3D	F(1)	Interrupt Pin	8	Interrupt Pin	00h
3E	F(2)	Min Grant	8	NOT IMPLEMENTED	**00h**
3F	F(3)	Max Latency	8	PCI Maximum Latency	00h

Only the upper 20 bits of Base Address 0 are writable, giving an address range of 4k. This cannot be changed from either the local or PCI bus. Local addresses are given as the indirect address to be written to REGISTER_ADDRESS followed by the byte number(s) i.e. the Vendor ID can be accessed by writing 0 to REGISTER_ADDRESS and then reading or writing REGISTER_DATA0 and REGISTER_DATA1.

Shared Register Block

These are accessed by the local processor through an indirect method using REGISTER_ADDRESS and REGISTER_DATAn. The PCI uses memory accesses and BaseAddress0 to locate these registers.

PCI Addr.	Local Indirect Addr.	Name	Width	Function (Interprocessor Communication Register)	Default Value
0	10(1,0)	LPG0	16	Interprocessor Communication Register	0000h
2	10(3,2)	LPG1	16	" " "	0000h
4	11(1,0)	LPG2	16	" " "	0000h
6	11(3,2)	LPG3	16	" " "	0000h
8	12(1,0)	LPG4	16	" " "	0000h
A	12(3,2)	Reserved	16	DO NOT USE	0000h
C	13(1,0)	BD0	16	" " "	0000h
E	13(3,2)	PG0	16	" " "	0000h
10	14(1,0)	PG1	16	" " "	0000h
12	14(3,2)	PG2	16	" " "	0000h
14	15(0)	PG3	8	" " "	0000h
15	15(1)	BD1	8	" " "	0000h
16	15(2)	PG4	8	" " "	0000h
17	15(3)	Reserved	8	DO NOT USE	0000h
18	16(3-0)	PG5	32	" " "	30343800h
1C	17(3-0)	PG6	32	" " "	00000000h

Note:

PG4, PG3, PG5, PG6 & BDI are write protected when the PROTECT_REGISTERS bit in RESOURCE_CONTROL is set.

LPGn - writable by the local processor

PGn - writable by the PCI

BDn - writable by either local or PCI

All shared registers are readable by either the local processor or the PCI bus.

DMA Control Registers

These can only be accessed by the local processor through an indirect method.

Local Indirect Addr.	Name	Width	Function
18(3-0)	DMA1_SOURCE_BASE	32	Base address of DMA transfers from PCI > Local
19(1,0)	DMA1_LENGTH	16	Length in long words of PCI > Local DMA
19(3,2)	DMA1_DEST_BASE	16	NOT IMPLEMENTED
1A(3-0)	DMA2_DEST_BASE	32	Base address of DMA transfers from Local > PCI
1B(1,0)	DMA2_LENGTH	16	Length in long words of Local > PCI DMA
1B(1,0)	DMA2_SOURCE_BASE	16	NOT IMPLEMENTED
24(0)	DMA_MASK	8	Byte masks for first and last word in DMA transfers
24(1)	RESOURCE_CONF	8	Bit 0 = local FIFO endian mode (see section on DMA controller units)

Interrupt Control Registers

PCI Address	Local Indirect Address	Name	Width	Function
30	1C(3-0)	INTERRUPT_SET	32	Written to by PCI to assert LINT Written by local to assert PCI #INTA.
34	1D(3-0)	INTERRUPT_CLEAR	32	Written to by the PCI to clear PCI INTA#. Written by local processor to clear LINT.
38	1E(3-0)	INTERRUPT_STATUS	32	Non zero to PCI when card is asserting INTA#. Non zero to Local when card is asserting Local INT.

The values written to INTERRUPT_SET and INTERRUPT_CLEAR are irrelevant.

Mailbox Registers

These registers are located in the memory space of PCI, addresses given are offsets from the base address assigned to the chip.

Local access is through the indirect addressing, using the REGISTER_ADDRESS and REGISTER_DATA0..3 ports.

PCI Addr. Local	Indirect Addr.	Name	Width	Default Value
40h	20h(3-0)	Printer Control Register (PCR)	32	00000000h
44h	21h(3-0)	Printer Status Register (PSR)	32	00000000h
48h	22h(3-0)	Device Control Register (DCR)	32	00000000h
4Ch	23h(3-0)	Device Status Register (DSR)	32	00000000h
70h	2Ch(3-0)	Printer Handshake Register (PHR)	32	00000000h
74h	2Dh(3-0)	Device Handshake Register (DHR)	32	00000000h

Printer Control Register 40h (20h)

Bit	POR Value	Local Access	PCI Access	Function
28	0	RD	RD/WR	1 = Enable PCI interrupt on local write to DHR
27	0	RD	RD/WR	1 = Enable PCI interrupt on local read from PHR
26	0	RD	RD/WR	1 = Enable PCI interrupt on Target Abort
25	0	RD	RD/WR	1 = Enable PCI interrupt on Master Abort
24	0	RD	RD/WR	1 = Software Reset - Generates local interrupt if enabled

Printer Status Register 44h (21h)

Bit	POR/RST Value	Local Access	PCI Access	Function
28	0	RD	RD/WR	When read, 1 = Local has written to DHR, Write 1 to clear
27	0	RD	RD/WR	When read, 1 = Local has read from PHR, Write 1 to clear
26	0	RD	RD	When read, 1 = Target Abort has occurred
25	0	RD	RD	When read, 1 = Master Abort has occurred

Device Control Register 48h (22h)

Bit	POR Value	Local Access	PCI Access	Function
28	0	RD/WR	RD	1 = Enable local interrupt on PCI write to PHR
27	0	RD/WR	RD	1 = Enable local interrupt on PCI read from DHR
26	0	RD/WR	RD	1 = Enable local interrupt on Target Abort
25	0	RD/WR	RD	1 = Enable local interrupt on Master Abort
24	0	RD/WR	RD	1 = Enable local interrupt on Software Reset

Device Status Register 4Ch (23h)

Bit	POR/RST Value	Local Access	PCI Access	Function
28	0	RD/WR	RD	When read, 1 = PCI has written to PHR, Write 1 to clear
27	0	RD/WR	RD	When read, 1 = PCI has read from DHR, Write 1 to clear
26	0	RD	RD	When read, 1 = Target Abort has occurred
25	0	RD	RD	When read, 1 = Master Abort has occurred
24	0	RD	RD	When read, 1 = PCI has set the software

Printer Handshake Register 70h (2Ch)

This is the mailbox used to pass data from PCI to local. When a PCI write to the PHR occurs an interrupt to the local processor is generated. When the local processor reads the register an interrupt to PCI is generated (the interrupt is actually generated when byte 3 is read, so the local processor should follow the usual convention of reading that byte last). Each interrupt is individually maskable.

Device Handshake Register 74h (2Dh)

This mailbox is used to pass data from local to PCI. When a local write to the DHR occurs an interrupt to PCI is generated (again, always write byte 3 last). When PCI reads the DHR an interrupt to the local processor is generated. Each interrupt is individually maskable.

I/O Pin Assignment

Pin ID	*Signal*	*Designator*
1	*LA<1>*	*Local Address Bus 1*
2	*LA<0>*	*Local Address Bus 2*
3		*Vss*
4		*Vdd (5v)*
5	*RD*	*Read Strobe*
6	*WR*	*Write Strobe*
7	*ALE*	*Address Latch*
8	*BHE*	*Upper Byte Enable (16 bit bus)*
9	*CS*	*Chip Select*
10	*MODE<2>*	*Local Bus Mode 2*
11	*MODE<1>*	*Local Bus Mode 1*
12	*MODE<0>*	*Local Bus Mode 0*
13	*LD<15>*	*Local Data 15*
14	*LD<14>*	*Local Data14*
15	*LD<13>*	*Local Data13*
16	*LD<12>*	*Local Data12*
17	*LD<11>*	*Local Data11*
18	*LD<10>*	*Local Data10*
19	*LD<9>*	*Local Data 9*
20	*LD<8>*	*Local Data 8*
21	*LD<7>*	*Local Data 7*
22	*LD<6>*	*Local Data 6*
23	*LD<5>*	*Local Data 5*
24	*LD<4>*	*Local Data 4*
25	*LD<3>*	*Local Data 3*

I/O Pin Assignment

Pin ID	*Signal*	*Designator*
26	*LD<2>*	*Local Data 2*
27	*LD<1>*	*Local Data 1*
28		*Vdd (5v)*
29	*LD<0>*	*Local Data 0*
30	*LWAIT*	*Wait (Local)*
31	*LINT*	*Interrupt (Local)*
32		*Vss*
33	*PRST*	*Reset*
34	*PCLK*	*PCI Bus Clock*
35	*GNT*	*Bus Arbitration - Grant*
36	*REQ*	*Bus Arbitration - Request*
37	*PAD<31>*	*PCI Address and Data 31*
38	*PAD<30>*	*PCI Address and Data 30*
39	*PAD<29>*	*PCI Address and Data 29*
40		*VssAC*
41	*PAD<28>*	*PCI Address and Data 28*
42		*VssDC*
43		*VddDC (3v)*
44	*PAD<27>*	*PCI Address and Data 27*
45	*PAD<26>*	*PCI Address and Data 26*
46	*PAD<25>*	*PCI Address and Data 25*
47	*PAD<24>*	*PCI Address and Data 24*
48	*CBE<3>*	*Command/Byte Enable 3*
49	*IDSEL*	*Initialisation Select*
50	*PAD<23>*	*PCI Address and Data 23*

I/O Pin Assignment

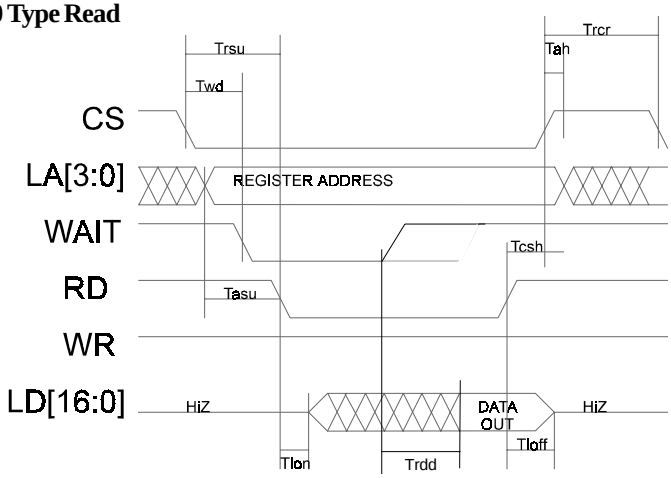
Pin ID	*Signal*	*Designator*
51	*PAD<22>*	*PCI Address and Data 22*
52		*VssAC*
53		*VssDC*
54	*PAD<21>*	*PCI Address and Data 21*
55	*PAD<20>*	*PCI Address and Data 20*
56	*PAD<19>*	*PCI Address and Data 19*
57	*PAD<18>*	*PCI Address and Data 18*
58	*PAD<17>*	*PCI Address and Data 17*
59		*VddDC (3v)*
60	*PAD<16>*	*PCI Address and Data 16*
61		*VddAC (3v)*
62	*CBE<2>*	*Command/Byte Enable 2*
63	*FRAME*	*Cycle Start*
64	*IRDY*	*Initiator Ready*
65	*TRDY*	*Target Ready*
66	*DEVSEL*	*Device Select*
67	*STOP*	*Stop Transaction*
68	*INTA*	*PCI Interrupt*
69		*VddAC (3v)*
70		*VssAC*
71	*PERR*	*Parity Error*
72		*VssDC*
73		*VddDC (3v)*
74	*SERR*	*System Error*
75	*PAR*	*Parity*

I/O Pin Assignment

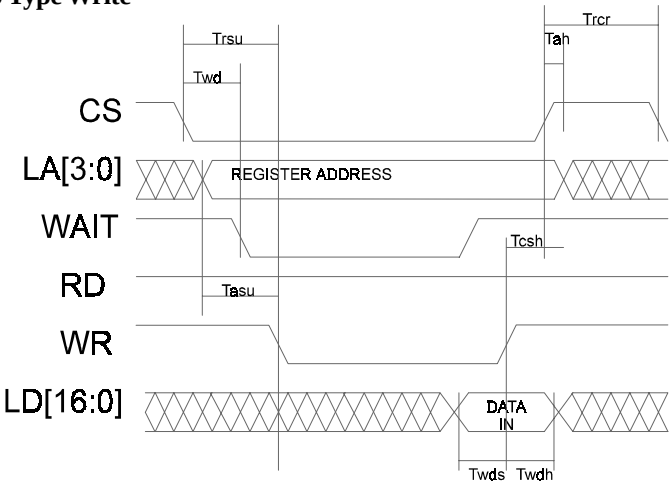
Pin ID	*Signal*	*Designator*
76	*CBE<1>*	*Command/Byte Enable 1*
77	*PAD<15>*	*PCI Address and Data 15*
78		*VddDC (3v)*
79	*PAD<14>*	*PCI Address and Data 14*
80	*PAD<13>*	*PCI Address and Data 13*
81	*PAD<12>*	*PCI Address and Data 12*
82	*PAD<11>*	*PCI Address and Data 11*
83	*PAD<10>*	*PCI Address and Data 10*
84	*PAD<9>*	*PCI Address and Data 9*
85	*PAD<8>*	*PCI Address and Data 8*
86	*CBE<0>*	*Command/Byte Enable 0*
87	*PAD<7>*	*PCI Address and Data 7*
88		*VssDC*
89	*PAD<6>*	*PCI Address and Data 6*
90		*VssAC*
91		*VddAC (3v)*
92	*PAD<5>*	*PCI Address and Data 5*
93	*PAD<4>*	*PCI Address and Data 4*
94	*PAD<3>*	*PCI Address and Data 3*
95	*PAD<2>*	*PCI Address and Data 2*
96	*PAD<1>*	*PCI Address and Data 1*
97	*PAD<0>*	*PCI Address and Data 0*
98	*DMARREQ*	*DMA Read Request*
99	*DMAWREQ*	*DMA Write Request*
100	*LA<2>*	*Local Address Bus 2*

To minimise noise on output pins it is advisable to ensure a degree of isolation between the AC (noisy) power supply and the DC (quiet) supply pins. i.e. they should be connected to power planes using separate vias'.

Z80 Type Read

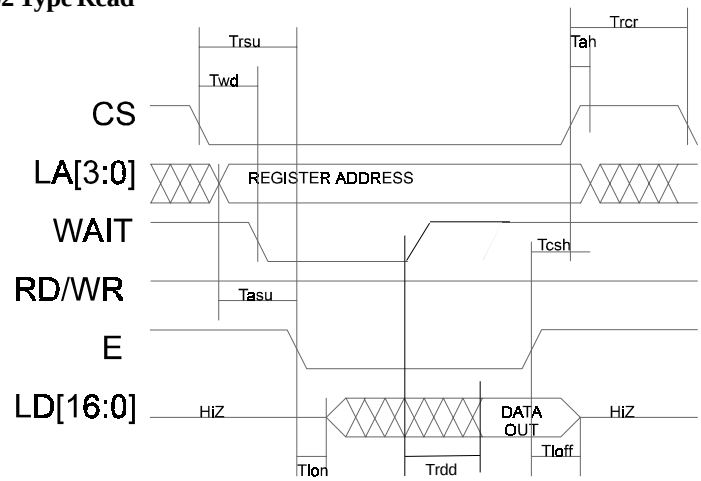


Z80 Type Write

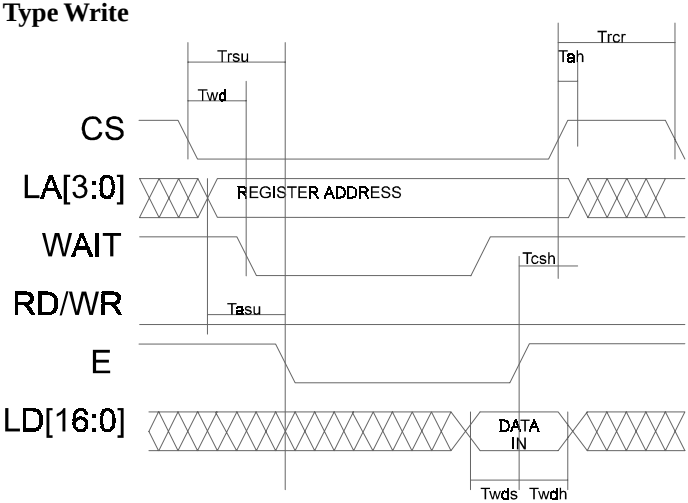


Trsu	Chip select setup time to read/write strobe asserted
Tasu	Address setup time to read/write strobe asserted
Tcsh	Chip select hold time from read/write strobe deasserted
Tah	Address hold from chip select deasserted
Trcr	Recovery time to next assertion of chip select
Twd	Time to assert WAIT from chip select
Tlon	Data out turnon time from read strobe asserted
Tloff	Data out turnoff time from read strobe deasserted
Twds	Write data setup time
Twdh	Write data hold time
Trdd	Read data valid from WAIT deasserted

6502 Type Read

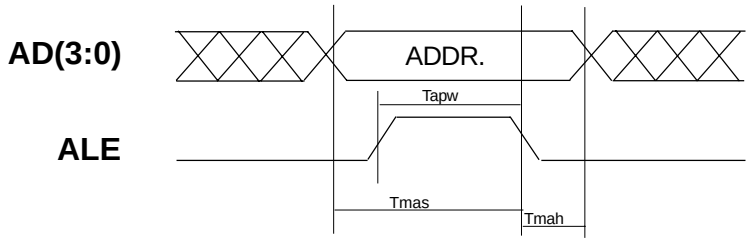


6502 Type Write



Trsu	Chip select setup time to read/write strobe asserted
Tasu	Address setup time to read/write strobe asserted
Tcsh	Chip select hold time from read/write strobe deasserted
Tah	Address hold from chip select deasserted
Trcr	Recovery time to next assertion of chip select
Twd	Time to assert WAIT from chip select
Tlon	Data out turnon time from read strobe asserted
Tloff	Data out turnoff time from read strobe deasserted
Twds	Write data setup time
Twdh	Write data hold time
Trdd	Read data valid from WAIT deasserted

Multiplexed Bus Mode



Tapw	ALE Pulse Width
Tmas	Multiplexed address setup
Tmah	Multiplexed address hold

Local Bus Controller Unit

The Local Bus Controller Unit (LBCU) sequences all the operations necessary to transfer data to and from the microprocessor, both in slave mode (register accesses) and master mode (DMA transfers).

The LBCU is made up of three units, the Local Slave Sequencer Unit (LSSU), the Local Data Path Unit (LDPU) and the Local Master Sequencer Unit (LMSU).

(In AB-2061 the LMSU is not present or active).

The LSSU handles all transactions when the microprocessor is driving the local bus; these transactions are exclusively register accesses.

When the DMA unit requires to transfer data from the local memory to the on-chip FIFOs or vice versa, it signals the LMSU which obtains control of the local bus using the BUSREQ and BUSGNT lines, and then sequences the spitting or assembly of 32 bit words for transfer over 8 or 16 bits. (Only in local bus mastering capable variants).

The LDPU is the collection of multiplexers required to split or assemble the bytes/ words and long words.

Local Bus Configuration
Local bus configuration is determined at reset by the levels on the MODE[2:0] pins. The value on these pins MUST NOT change outside the period where PRST is asserted.

Mode	**= 1**	**= 0**
0	6502 Type strobes - E, R/W	Z80 Type strobes - RD, WR
1	Non multiplexed bus	Multiplexed bus
2	8 bit data	16 bit data

DMA Controller Units

The DMA Controller units (DCUs) contain the control logic, pointers and counters to schedule, sequence and provide source and destination addresses for data passing through the FIFOs. These units ensure that the maximum burst length possible is always used. In the case of AB2061 this is 4 longwords (32 bytes).

DMA transfers can be byte aligned within the PCI memory space, operation of this feature is described in the DMA_MASK section.

AB2061 is not capable of directly placing data into a memory on the local bus. The job of retrieving data from and writing data to the FIFOs is up to the local processor or some other DMA capable device. FIFO access has been specially optimised to allow transfer to local memory by relatively simple DMA controllers such as those present on many microcontroller devices.

Initialising DMA Transfers

DMA Channel 1 always moves data from the PCI memory space to the FIFOs and hence into the local memory. Channel 2 moves data from local memory into PCI memory space.

Channel1:	DMA1_SOURCE_BASE	Holds a 32 bit address. This is the first location in the PCI memory space of the block to be copied to local memory.
	DMA1_LENGTH	This 16 bit register holds the length in longwords of the block to be copied to local memory.
Channel2:	DMA2_DEST_BASE	Holds a 32 bit address. This is the first location in the PCI memory space of the block where data from local memory will be placed
	DMA2_LENGTH	A 16 bit value which specifies the length of the transfer in longwords.
	DMA_MASK	This 8 bit value determines which bytes are written in the first and last words of the block, thus allowing blocks to be byte aligned.

DMA Status

The RESOURCE_STATUS register contains bits that allow the local processor to monitor the progress or otherwise of ongoing DMA transfers.

DMA1_COMPLETE — The last word of data has been placed in FIFO1 by DMA channel 1. This DOES NOT however indicate that the data is in local memory yet. (See FIFO1_EMPTY)

DMA2_COMPLETE — DMA Channel2 has moved the last word from FIFO2 into the PCI memory space.

FIFO1_EMPTY — FIFO1 (PCI to LOCAL) is empty of data. If there is more data expected, the local processor must wait until this flag is reset. If DMA1_COMPLETE and FIFO1_EMPTY are both set the program can assume the whole block has now been fetched.

FIFO2_FULL — FIFO2 (LOCAL to PCI) if full and no more data should be written into it.

DMA1_ERROR
DMA2_ERROR — These flags indicate that an attempt to read or write the PCI memory space failed for an non-recoverable reason. AB2061 regards PCI Master Abort (no device responds) and PCI Target Abort events as non-recoverable.

Starting, Stopping and Resetting DMA

In order to get either DMA channel to re-read its BASE and LENGTH registers, the appropriate ENABLE bit MUST be written as a '0'. To start a channel write the BASE and LENGTH values, write a '0' to the ENABLE bit, followed immediately by a '1'. Writing a 1 to a channel that is already running will not disturb its operation in any way. To stop a channel in operation, write '0' to the appropriate ENABLE bit. Note that this will STOP and RESET the channel. In order to pause without resetting the counters and pointers write a '1' to the appropriate PAUSE bit. The pause function will not stop a burst in progress once AB2061 has asserted REQuest to the PCI bus.

DMA Channel 2 will normally only try to initiate a burst on the PCI bus when FIFO2 is full (4 longwords). In cases where this is undesirable such as a block of

3 or less longwords are left, the FIFO2_FLUSH bit must be set. This bit allows DMA2 to initiate a burst with any quantity of data in FIFO2. Do not forget to reset this bit after completion of the block transfer as it can dramatically reduce the transfer speed and bus efficiency of AB2061.

NEVER attempt to reset the CONFIG_ENABLE bit in the RESOURCE_CONTROL register when any DMA is running. Doing so causes AB2061 to enter a factory test mode and will cause unpredictable PCI memory space corruption.

Reading and Writing Data

DMA data is read from and written to the FIFO_DATA register. This is either a 8 or 16 bit port depending on the data bus width. In 8 bit mode four sucessive reads will retrieve one full 32 bit word from FIFO1 and 4 writes will place one 32 bit word into FIFO2. The order that the bytes are written to and retrieved from the fifos is determined by the ENDIAN bit in the RESOURCE_CONFIG register. In 16 bit mode FIFO_DATA must be read and written as a 16 bit wide port, two reads/writes are required per 32 bit word.

As an example, assume the top word of FIFO1 contains the value 00C0FFEEh.

In 8 bit mode with ENDIAN=0 four reads of FIFO_DATA return EE, FF, C0, 00
ENDIAN=1 four reads of FIFO_DATA return 00, C0, FF, EE

In 16 bit mode with ENDIAN=0 two reads of FIFO_DATA return FFEE, 00C0
ENDIAN=1 two reads of FIFO_DATA return 00C0, FFEE

Writing data to FIFO2 follows the same ordering convention.

The internal 'byte/word pointers' for this function are reset when a '0' is written to the associated channel's DMA_ENABLE bit.

Local bus DMAC signals

AB2061 has a DMA request pin for each channel, this allows a simple twochannel DMAC on the local bus to read and write the fifos without CPU intervention.

DMARREQ - Active low and asserted when FIFO1 contains data

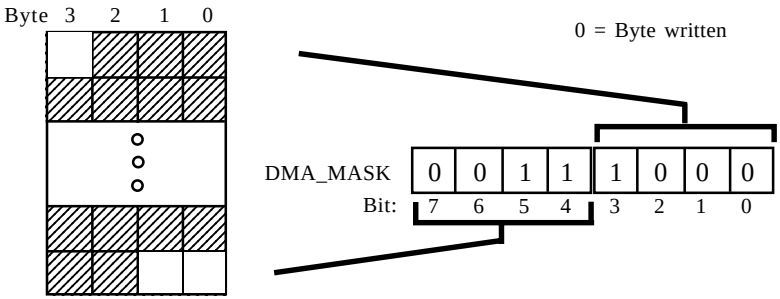
DMAWREQ - Active low and asserted when FIFO2 has space for data

Non Aligned DMA Transfers

AB2061 supports the transfer of non word aligned blocks from the local processor to PCI. Non aligned transfers from PCI to local are not supported however.

This feature is implemented through the DMA_MASK register which is 8 bits wide. The lower nibble determines the pattern output on the PCI byte enables when writing the first word of a block (and hence which bytes are written). The upper nibble has the same effect on the last word of the block.

Example: To write the block shown below, the DMA_MASK register should be written with the value 38h.



Shading indicates bytes to be transferred

Notes

1) It is possible to write non contiguous data with this feature, however this is strongly discouraged and may result in some targets signalling a system error on SERR.

2) When the local processor is writing data to the FIFO, it must always write to FIFO_PCI_DATA4 last, regardless of whether that byte will be written in the word. Failure to do this will result in the loss of the entire word of data.

3) When performing transfers of one word (or less), the upper and lower nibbles of the DMA_MASK register should be written to the same value.

Electrical Specification

VDD_3	*+3.3V ± 10%*
VDD_5	*+5.0V ± 10% +3.3V ± 10%*
Icu	*20mA (typ) 40mA (max)*
GND and GNDc	*0V*
Input - Low	*< 0.7V*
Input - High	*> 1.8V*
Output - Low	*< 0.6V*
Output - High	*> 2.6V*
Input - Load	*5pF*

Operating Temperature:
Range 0°C - 70°C
Humidity 90% (Non condensing)

Storage Temperature:
Range -10°C - +80°C
Humidity 95% (Non condensing)

Product has to be used within 6-7 hours after unpacking.

Distributed in Japan
Rikei Corporation
1-26-2 Nishi-Shinjuku
Shinjuku-Ku
Tokyo 163-05
Japan
Tel: +81 3 3345 2189
Fax: +81 3 3344 3949

AB Semicon Limited
AB Semicon House
62 Victoria Road
Burgess Hill
West Sussex
RH15 9LH
Tel: +44 (0) 1444 870408
Fax: +44 (0) 1444 870452