NEC

VRC4372 I/O Controller with PCI Interface

Data Sheet

February 2000

Description

The VRC4372™ is a PCI-based I/O controller for high-performance, low-cost system implementation. Connected to the PCI bus, it is designed to support NEC's VR43xx CPU, interfacing between the VRC4373™ or VRC4375™ system controller and peripherals for tethered applications such as set-top boxes, printers, network computers and consumer electronics.

Features

- Compatible with NEC VR43xx CPUs and VRC4373 and VRC4375 system controllers
- PCI Bus Interface Controller
  - Compliant with the 3.3-volt PCI interface specification (version 2.1)
  - 5-volt PCI signaling in a controlled environment (with voltage spikes less than 7.6 volts and DC levels less than 6.6 volts)
  - PCI bus arbitration for one internal and six external PCI masters
- I/O Controller
  - ISA-like, 16-bit I/O bus and four independent direct memory access (DMA) channels
  - Ten chip selects and control signals for interfacing to external peripheral chips
  - N-to-3 interrupt controller to regulate interrupts from 14 input pins, two internal timers, the keyboard, the mouse, and four DMA channels can be encoded onto one of three outgoing interrupt lines
- Four programmable DMA channels
  - Unique $\overline{TC}$, EOP, DREQ, and $\overline{DACK}$ signals on each channel
  - Block or single transfers
  - Read or write requests
  - I/O device-demanded service requests via DREQx
  - Software-initiated requests
  - Channel suspend via the $\overline{MASK}$ register bit
  - I/O device transfer termination via EOP
  - Channel reload notification and termination notification via an interrupt
  - Efficient PCI bus data packing mode
  - Byte/short scattering/gathering capabilities (one per PCI word)
- Miscellaneous I/O
  - Timer and a beeper source
  - General-purpose control/status I/O pins
- Test Port
  - $\overline{TESTOE}$ signal that can be used to tristate all outputs
  - $\overline{SCAN}$ pin to place the chip in scan mode
- Data buffers for direct memory access channels and the I/O bus
- 33-MHz bus frequency

NEC VRC chipsets are designed for use with NEC VR Series microprocessors. NEC makes no claim as to the suitability of VRC chipsets for use with non-NEC microprocessors and does not warrant their performance, suitability or use in such applications.

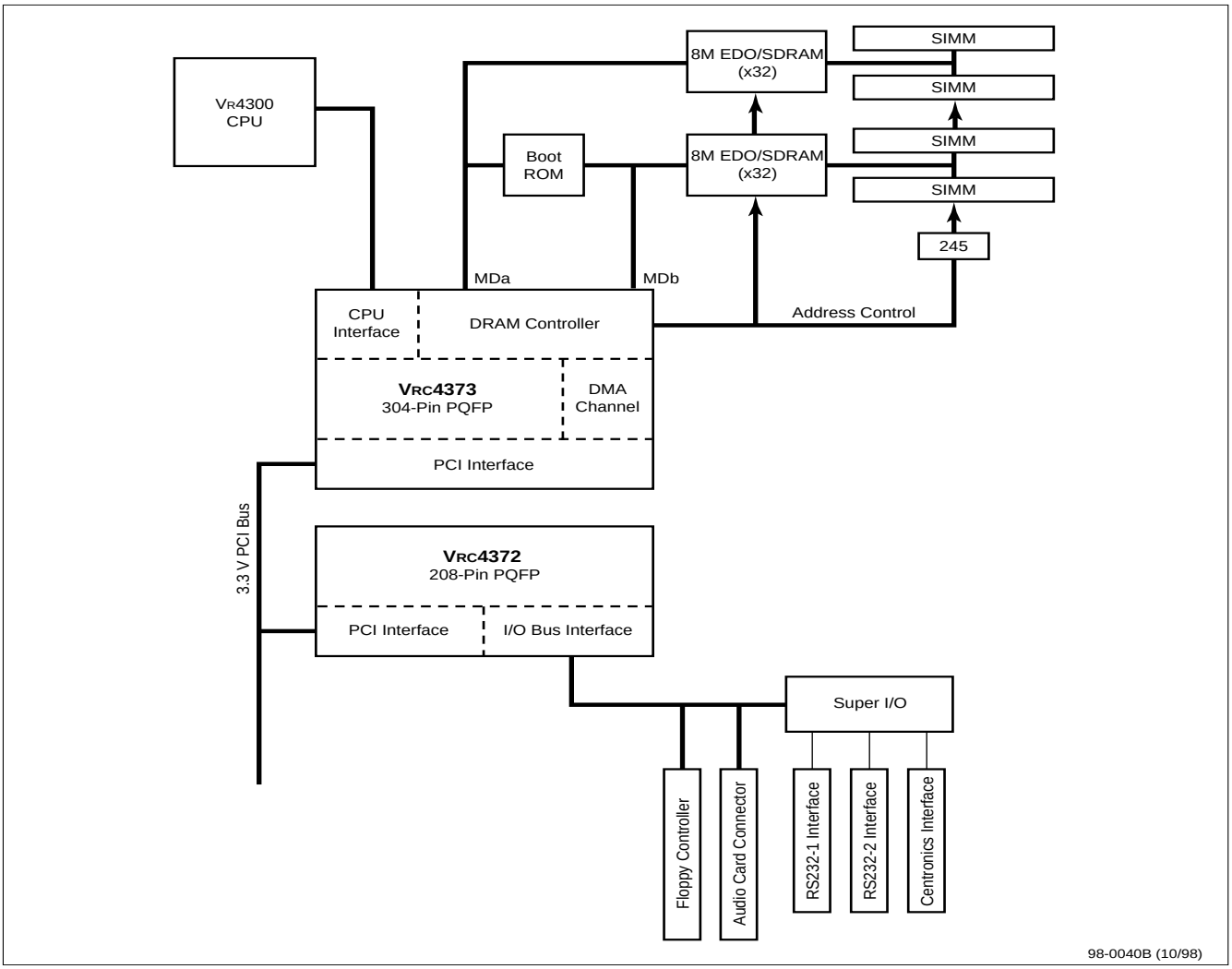
Ordering Information

Part Number	Package
VRC4372-1	208-pin PQFP

System Configuration

Figure 1 provides an example of a system that uses the VRC4372 controller.

Figure 1. System Connection



Reference Documents

- *PCI Local Bus Specification* Revision 2.1 and *PCI System Design Guide* Revision 1.0 (available from the Peripheral Component Interconnect Special Interest Group)
- NEC VR4300™ Microprocessor Data Sheet (document number U10116EJ4V0DS00)
- *IC Package Manual* (document number C10943XJ7V0IF00)

Contents

1.0 Pin Configuration

208-Pin Plastic Quad Flat Package (PQFP)

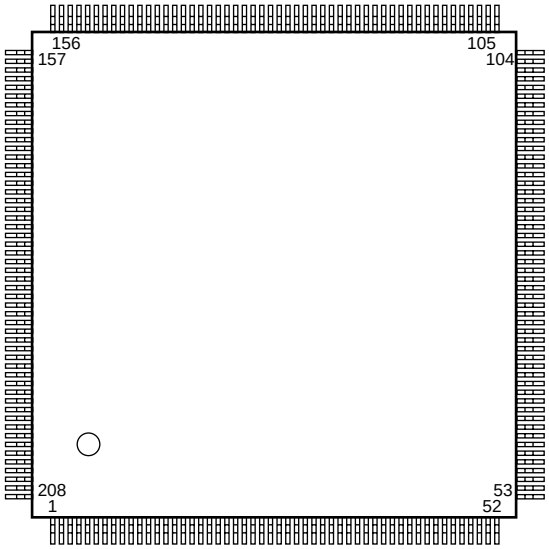


Table 1. Pin Assignments

Pin	Name	Pin	Name	Pin	Name	Pin	Name	Pin	Name
1	GND0	43	AD7	85	IOA1	127	IOD15	169	$\overline{INT4}$
2	GND1	44	AD6	86	IOA2	128	$\overline{SCAN}$	170	$\overline{INT5}$
3	$\overline{CBE3}$	45	AD5	87	IOA3	129	ARB	171	$\overline{INT6}$
4	IDSEL	46	GND10	88	IOA4	130	V_{DD7}	172	$\overline{INT7}$
5	AD23	47	V_{DD3}	89	IOA5	131	GND20	173	$\overline{INT8}$
6	AD22	48	AD4	90	$\overline{EOP\_TC1}$	132	GPIO0	174	$\overline{INT9}$
7	GND2	49	AD3	91	GND15	133	GPIO1	175	$\overline{INT10}$
8	AD21	50	AD2	92	IOD0	134	GPIO2	176	IPL2
9	AD20	51	GND11	93	IOA6	135	GPIO3	177	IPL1
10	AD19	52	GND12	94	IOA7	136	GPIO4	178	IPL0
11	AD18	53	V_{DD4}	95	$\overline{CS7}$	137	GPIO5	179	$\overline{GNT5}$
12	GND3	54	AD1	96	IOD1	138	GPIO6	180	$\overline{GNT4}$
13	V_{DD0}	55	AD0	97	IOD2	139	GPIO7	181	$\overline{GNT3}$
14	AD17	56	$\overline{IOREADY}$	98	GND16	140	GPIO8	182	GND25
15	AD16	57	$\overline{INT0}$	99	IOD3	141	GPIO9	183	V_{DD9}
16	$\overline{CBE2}$	58	$\overline{INT1}$	100	IOD4	142	GND21	184	$\overline{GNT2}$
17	$\overline{FRAME}$	59	$\overline{INT2}$	101	IOD5	143	GPIO10	185	$\overline{GNT1}$
18	GND4	60	$\overline{INT3}$	102	IOD6	144	IOBEEP	186	$\overline{REQ5}$
19	$\overline{IRDY}$	61	NC	103	IOD7	145	$\overline{INT12}$	187	$\overline{REQ4}$
20	$\overline{TRDY}$	62	$\overline{EOP\_TC3}$	104	V_{DD6}	146	$\overline{INT13}$	188	GND26
21	$\overline{DEVSEL}$	63	$\overline{EOP\_TC2}$	105	GND17	147	NC	189	$\overline{REQ3}$
22	GND5	64	$\overline{IOFRAME}$	106	GND18	148	$\overline{EOP\_TC0}$	190	$\overline{REQ2}$
23	$\overline{STOP}$	65	GND13	107	NC	149	DREQ1	191	$\overline{REQ1}$
24	$\overline{PERR}$	66	NC	108	NC	150	$\overline{DACK1}$	192	$\overline{RST}$

Note: NC = no connection

Table 1. Pin Assignments (continued)

Pin	Name	Pin	Name	Pin	Name	Pin	Name	Pin	Name
25	$\overline{SERR}$	67	DREQ3	109	NC	151	DREQ0	193	GND27
26	GND6	68	DREQ2	110	NC	152	$\overline{DACK0}$	194	CLK
27	V_{DD1}	69	$\overline{DACK3}$	111	$\overline{INT11}$	153	$\overline{WR0}$	195	$\overline{GNT0}$
28	PAR	70	$\overline{DACK2}$	112	NC	154	$\overline{RD0}$	196	$\overline{REQ0}$
29	$\overline{CBE1}$	71	$\overline{CS0}$	113	NC	155	GND22	197	V_{DD10}
30	AD15	72	$\overline{CS1}$	114	NC	156	GND23	198	GND28
31	AD14	73	$\overline{CS2}$	115	NC	157	V_{DD8}	199	AD31
32	GND7	74	$\overline{CS3}$	116	GND19	158	$\overline{CPRD0}$	200	AD30
33	AD13	75	$\overline{CS4}$	117	$\overline{TESTOE}$	159	$\overline{CPRD1}$	201	AD29
34	AD12	76	$\overline{CS5}$	118	$\overline{HBE}$	160	$\overline{CPWR0}$	202	AD28
35	AD11	77	$\overline{RD1}$	119	$\overline{LBE}$	161	$\overline{CPWR1}$	203	GND29
36	AD10	78	V_{DD5}	120	IOD8	162	$\overline{CPRD2}$	204	AD27
37	GND8	79	GND14	121	IOD9	163	$\overline{CPWR2}$	205	AD26
38	V_{DD2}	80	$\overline{WR1}$	122	IOD10	164	$\overline{CS6}$	206	AD25
39	AD9	81	$\overline{W\_R}$	123	IOD11	165	NC	207	AD24
40	AD8	82	$\overline{ALE}$	124	IOD12	166	$\overline{CS8}$	208	V_{DD11}
41	$\overline{CBE0}$	83	$\overline{BUFOE}$	125	IOD13	167	$\overline{CS9}$	—	—
42	GND9	84	IOA0	126	IOD14	168	$\overline{GND4}$	—	—

Note: NC = no connection

1.0.1 Pin Functions

Table 2. PCI Interface Pins

Pin(s)	I/O	Reset Value	Pull-up/ Pull-down	Description
AD[31:0]	I/O	Hi-Z	—	Multiplexed address/data bus
$\overline{CBE[3:0]}$	I/O	Hi-Z	—	PCI command/byte enable
PAR	I/O	Hi-Z	—	Parity of AD and $\overline{CBE}$
$\overline{FRAME}$	I/O	Hi-Z	—	PCI $\overline{FRAME}$
$\overline{TRDY}$	I/O	Hi-Z	—	PCI $\overline{TRDY}$
$\overline{IRDY}$	I/O	Hi-Z	—	PCI $\overline{IRDY}$
$\overline{STOP}$	I/O	Hi-Z	—	PCI $\overline{STOP}$
$\overline{DEVSEL}$	I/O	Hi-Z	—	PCI $\overline{DEVSEL}$
IDSEL	I	Hi-Z	—	PCI IDSEL
$\overline{REQ0}$	I/O	Hi-Z	—	PCI $\overline{REQ}$ input (ARB = V_{CC}) PCI $\overline{REQ}$ output (ARB = GND)
$\overline{REQ[5:1]}$	I		—	PCI $\overline{REQ}$ input
$\overline{GNT0}$	I/O	High Hi-Z	—	PCI $\overline{GNT}$ output (ARB = V_{CC}) PCI $\overline{GNT}$ input (ARB = GND)
$\overline{GNT[5:1]}$	O	High	—	PCI $\overline{GNT}$ output
CLK	I		—	PCI CLK input
$\overline{RST}$	I		—	Reset
$\overline{PERR}$	I/O	Hi-Z	—	Parity error
$\overline{SERR}$	I/O–oc	Hi-Z	—	System error

Table 3. I/O Bus Bidirectional/Output Pins

Pin(s)	I/O	Reset Value	Pull-up/ Pull-down	Max. DC Load (mA)	AC Load (pF)	Description
IOA[7:0]	O	Low	—	1	10–100	I/O address bus (eight LSB)
IOD[15:8] IOD[7:0]	I/O	Hi-Z	Pull-down, 50 KΩ	3	0–50 10–100	Multiplexed I/O address/data bus
$\overline{\text{HBE}}$	O	High	—	1	0–30	I/O bus high-byte enable
$\overline{\text{LBE}}$	O	High	—	1	0–30	I/O bus low-byte enable
$\overline{\text{ALE}}$	O	High	—	2	0–30	Latches IOA[23:8] from the IOD bus
$\overline{\text{IOFRAME}}$	O	High	—	1	0–30	Controls the I/O transfer
$\overline{\text{W\_R}}$	O	High	—	1	0–50	I/O write/read status
$\overline{\text{RD1}}$ $\overline{\text{RD0}}$	O	High	—	1 —	0–70 0–30	General I/O read strobes
$\overline{\text{WR1}}$ $\overline{\text{WR0}}$	O	High —	—	1 —	0–70 0–30	General I/O write strobes
$\overline{\text{BUFOE}}$	O	High	—	3	0–30	I/O data bus buffer enable
$\overline{\text{CPRD2}}$ $\overline{\text{CPRD[1:0]}}$	O	High	—	3 1	0–30 0–30	Decoded I/O read strobe, 2x F245 load Decoded I/O read strobes, CMOS load
$\overline{\text{CPWR2}}$ $\overline{\text{CPWR[1:0]}}$	O	High	—	2 1	0–30	Decoded I/O write strobe, 3x F373 load Decoded I/O write strobe, CMOS load
$\overline{\text{CS[9:8]}}$ $\overline{\text{CS[7:0]}}$	O	High	—	3 1	0–30	Decoded chip selects, 2x F245 load Decoded chip selects, CMOS load
IPL[2:0]	O	High	—	1	0–15	CPU interrupts
$\overline{\text{DACK[2:0]}}$	O	High	—	1	0–30	DMA acknowledge
$\overline{\text{DACK3}}$	O	Low	—	1	0–30	DMA acknowledge
$\overline{\text{EOP\_TC[3:0]}}$	I/O	Hi-Z	Pull-down, 50 KΩ	1	0–30	DMA terminal count and EOP

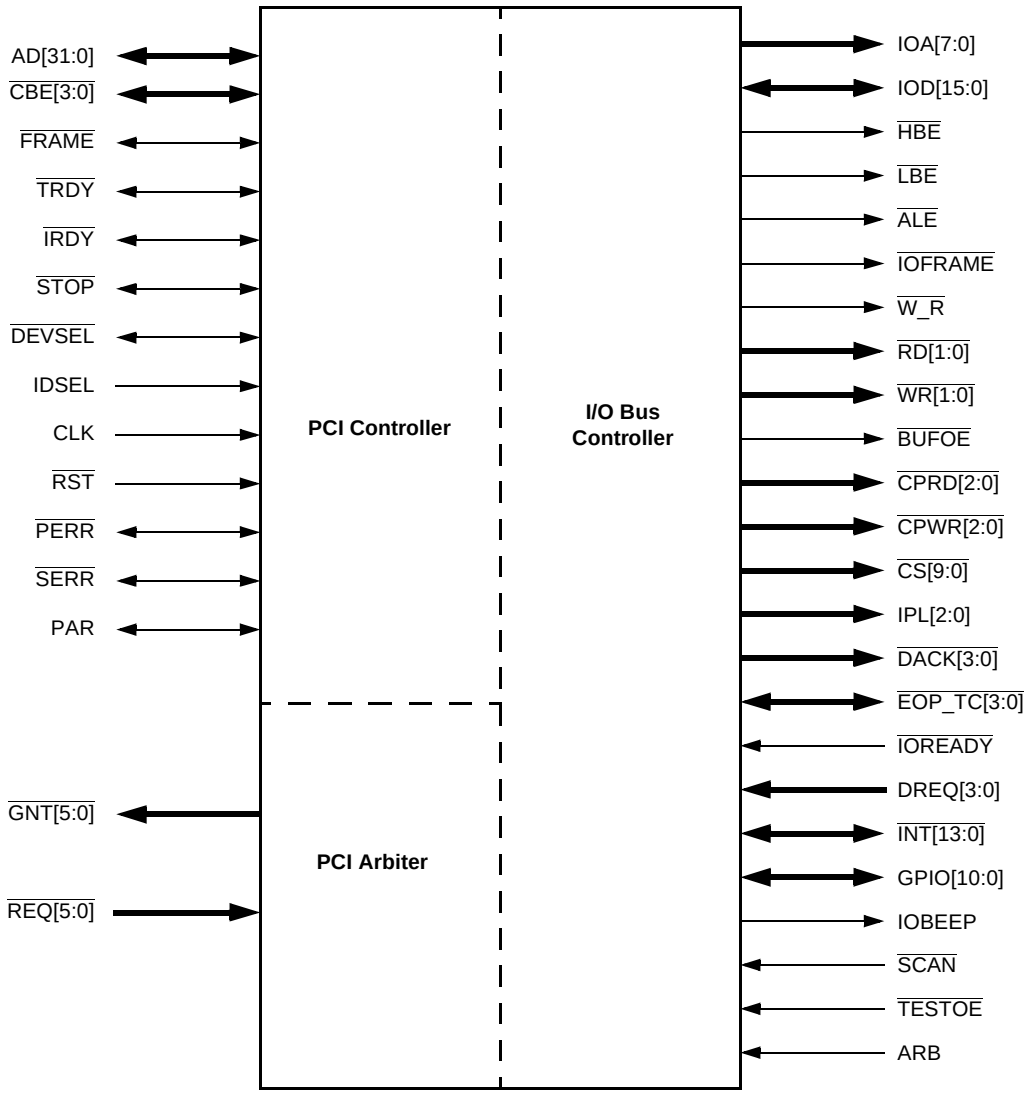
Table 4. I/O Bus Input Pins

Pin(s)	I/O	Reset Value	Pull-up/Pull-down	Description
$\overline{\text{IOREADY}}$	I	—	Pull-down, 50 KΩ	Ready indicates I/O bus cycle complete
DREQ[3:0]	I	—	Pull-down, 50 KΩ	DMA request
$\overline{\text{INT[13:0]}}$	I/O (Schmitt)	Hi–Z	Pull-down, 50 KΩ	Interrupt request when $\overline{\text{SCAN}}$ = VCC (input only); scan signals when $\overline{\text{SCAN}}$ = GND (may be input or output)

Table 5. Miscellaneous I/O Pins

Pin(s)	I/O	Reset Value	Pull-up/ Pull-down	DC Load (mA)	AC Load (pF)	Description
GPIO[10:0]	I/O	Hi–Z	Pull-down, 50 KB	2 (Note)	0–30	General-purpose status/control signals
IOBEEP	O	Low	—	1	30	Beeper output
$\overline{\text{SCAN}}$	I	—	—	—	—	Enables scan test mode when low
$\overline{\text{TESTOE}}$	I	—	—	1	0	Tristates all outputs when low
ARB	I	—	—	1	0	Internal arbitration when high; external arbitration when low

2.0 Block Diagram



3.0 PCI Bus Interface Controller

3.1 PCI Configuration Space Registers

The VRC4372 controller implements the 256-byte configuration space defined by the PCI bus specification (Table 6). Access to these registers is accomplished by read and write accesses in the PCI configuration space. In these modes, onboard logic asserts the IDSEL signal pin, thus selecting the VRC4372 configuration registers. Read accesses from undefined and reserved locations complete normally and return a logic 0. Undefined or reserved bits also read as logic 0, but the PCI specification requires that the software not rely on reserved bits having any particular value. Write accesses to undefined and reserved locations complete normally and the data is discarded.

Table 6. PCI Configuration Registers

Register Name	Symbol	Configuration Offset	R/W	Description
Vendor ID	VID	0x01–0x00	R	Manufacturer name
Device ID	DID	0x03–0x02	R	Device name
Command	PCICMD	0x05–0x04	R/W	Coarse control of PCI interface
Status	PCISTS	0x07–0x06	R/W	Status information for PCI-related events
Revision ID	RID	0x08	R	Device revision
Class code	CLASS	0x0b–0x09	R	Device type
Cache line size	CLSIZ	0x0c	R/W	System cache line size (words)
Master latency timer	MLTIM	0x0d	R/W	Value of latency timer for this master, in PCI clocks
Header type	HTYPE	0x0e	R	Layout for registers in the range 0x10–0xff
Register base address	REGADD	0x13–0x10	R/W	Start and size of register's region
I/O bus base address	IObusADD	0x17–0x14	R/W	Start and size of I/O memory space
Reserved	—	0x3b–0x18	—	—
Interrupt line	INTLIN	0x3c	R/W	Interrupt line routing information
Interrupt pin	INTPIN	0x3d	R	Interrupt pin name
Undefined	—	0x3f–0x3e	—	—
Disconnect timer	DSCTIM	0x40	R	Not implemented
Retry timer	RTYTIM	0x41	R/W	Maximum number of tries before terminating transaction. Reset value = 00
Trdy timer	TDYTIM	0x42	R/W	Not implemented
PCI arbiter priority control	PAPC	0x43	R/W	Priority scheme for granting the PCI bus

Table 7. Device and Vendor Identification

Bit(s)	Symbol	Name	Functional Description
15:0	VID	Vendor ID	Hardwired to 0x1033 for NEC PCI devices
31:16	DID	Device ID	Hardwired to 0x001A

Table 8. Command Register (PCICMD)

Bit(s)	Symbol	Name	Functional Description
0	IOEN	I/O space enable	Hardwired to 0; ignored
1	MEMEN	Memory space enable	Reset to 0; must set to 1 via software to enable the VRC4372 to respond to memory space accesses
2	BMAS	Bus master enable	Reset to 0; must be set to 1 to enable the VRC4372 to generate PCI accesses
3	SPC	Special cycles enable	Hardwired to 0; ignored
4	MWI	Memory write and invalidate enable	Hardwired to 0; ignored
5	VGA	Video Graphics Array	Hardwired to 0; ignored because the VRC4372 is not a VGA device
6	PER	$\overline{\text{PERR}}$ enable	Reset to 0
7	WAIT_CTL	Wait cycle control	Hardwired to 0; ignored
8	SERR_EN	$\overline{\text{SERR}}$ enable	Reset to 0
9	FBBE	Fast back-to-back enable	Hardwired to 0; ignored
15:10	Reserved	—	Read as 0

Table 9. Status Register (PCISTS)

Bit(s)	Symbol	Name	Functional Description
22:16	Reserved	—	Read as 0
23	FBBC	Fast back-to-back capable	Hardwired to 1; fast back-to-back enabled
24	DPR	Data parity reported	Data parity reported
26:25	DEVSEL	DEVSEL timing	Hardwired to 01 (medium response)
27	STA	Signaled target abort	Set whenever the VRC4372 signals a target abort
28	RTA	Received target abort	Set whenever the VRC4372 master receives a target abort
29	RMA	Received master abort	Set whenever the VRC4372 master generates a master abort
30	SSE	Signaled system error	Set whenever the VRC4372 detects a system error
31	DPE	Detected parity error	Set whenever the VRC4372 detects a parity error

Table 10. Revision Identification (RID) and Class Codes

Bit(s)	Symbol	Name	Functional Description
7:0	RID	Revision ID	Hardwired to 0x00
15:8	Prog	Programming interface	Hardwired to 0x00
23:16	SubCl	Subclass	Hardwired to 0x80
31:24	BaseCl	Base class	Hardwired to 0x06

Table 11. Built-In Self Test, Header Type, Master Latency Timer, and Cache Line Size

Bits(s)	Symbol	Functional Description
7:0	CLSIZ	Hardwired to 0x00. The VRC4372 rejects cache line fills/ spills (unnatural order bursts).
10:8	MLTIM	Hardwired to 000
15:11	MLTIM	Master Latency Timer. See the PCI specification (Sections 3.4.4.1 and 6.2.4).
23:16	HTYPE	Hardwired to 0x00
31:24	Reserved	—

Table 12. Register Base Address (REGADD)

Bits(s)	Symbol	Functional Description
15:0	Reserved	Hardwired to 0x0000; indicates that the VRC4372 registers should be located in a 32-bit memory space on a 64 KB boundary and are not prefetchable
31:16	REGADD	Maps the VRC4372 registers in memory on a 64 KB boundary

Table 13. I/O Memory Base Address (IOADD)

Bits(s)	Symbol	Functional Description
27:0	Reserved	Hardwired to 0x000000 to indicate that the VRC4372 I/O memory should be located in a 32-bit PCI memory space on a 256-MB boundary and is not prefetchable
31:28	IOADD	The higher four bits of IOADD are used to map the VRC4372 system memory on a 256-MB boundary.

Table 14. Interrupt Pin (INTPIN) and Interrupt Line (INTLIN)

Bit(s)	Symbol	Functional Description
7:0	INTLIN	Provides interrupt line routing information as required by the PCI specification (Section 6.2.4); can be used as a scratch pad in motherboard implementations
15:8	INTPIN	Hardwired to 0x01 to indicate that the VRC4372 uses $\overline{\text{INTA}}$. When the VRC4372 is used with add-in cards, this bit indicates that the VRC4372 is connected only to $\overline{\text{INTA}}$. When used in a motherboard implementation, the VRC4372 may drive as many as three interrupt lines.
31:16	Undefined	Read as 0

Refer to Section 5.3 for additional information.

Table 15. DSCTIM, RTYTIM, TDYTIM, PAPC

Bit(s)	Symbol	Functional Description
7:0	DSCTIM	Disconnect timer (not implemented); read as 0
15:8	RTYTIM	Retry timer; reset value is 00
23:16	TDYTIM	TRDY timer (not implemented); read as 0
25:24	PAPC	PCI arbiter priority control register; reset value is 00 Rotating fair (00); $\overline{REQ0}$ owns the bus during Idle mode Rotating alternate 0 (01); $\overline{REQ0}$ owns the bus during Idle mode Rotating alternate I (10); the VRC4372 owns the bus during Idle mode Not used (11)
26	TKYGNT	Take away grant; reset value is 00 When 0, $\overline{REQx}$ is granted and continues until the request is removed. When 1, $\overline{REQx}$ is granted, but the bus loses the GNTx to higher priority requests. Since only rotating priority schemes are used, all requests are at a higher priority. MLTIM determines the minimum bus tenure for a master.
31:27	Undefined	Read as 0

4.0 Control Registers

The VRC4372 I/O controller implements 64 KB of control register space (16 KB words). Beginning at the address specified by the REGADD register in the PCI header and occupying the lower 64 KB of that address space, the register space is divided into 16 regions of 4096 bytes each and selected by A[15:12]. Each register region is dedicated to one particular function.

Table 16. Register Allocation

Number	Address Offset	Description
0	0x0000–0x0fff	Control registers
—	0x1000–0xffff	Undefined

5.0 PCI Interface

The PCI interface performs the following functions:

- PCI master
- PCI target
- Register access

As a PCI master, the PCI interface used by the DMA controller performs memory read and write cycles on the PCI bus. Burst transfers are executed if the byte count requested by the DMA channel is greater than four (in multiples of four). Byte alignment is accomplished in the DMA data buffer section.

As a PCI target, the PCI interface is used for CPU accesses to the I/O bus and should enforce nonburst accesses by performing a target disconnect on the first word. The PCI target contains a posted write buffer that allows write cycles to complete with no PCI bus wait states. When write cycles are executed, data is captured by the posted write buffer and split into one to four I/O bus accesses, according to the byte enables and the device data size communicated by the I/O bus controller and as specified by the profile register for that device. The data is split according to little-endian ordering and correctly aligned. During splitting of the write data (emptying the I/O bus-posted write buffer), the target subsection causes the PCI interface to generate a target retry for all subsequent read or write accesses to the I/O bus.

Register access is also provided by the PCI interface. All VRC4372 registers are accessed as byte-writable 32-bit words directly through this subsection. Register accesses continue during PCI target-posted, write buffer emptying.

5.1 PCI Arbitration

The VRC4372 I/O controller features an internal PCI arbiter that can be configured by pulling the ARB pin high to arbitrate among six requestors (five external and one internal), as specified by the bits programmed in the PAPC register. A reset selects a rotating fair scheme. If the ARB pin is tied low, the VRC4372 does not perform arbitration for the PCI bus and uses a logic 0 request grant pair to request the PCI bus for internal requirements.

5.1.1 Rotating Fair

In this scheme, the priority of each requestor changes after every request, in round-robin fashion, to give every request a fair chance to get on the bus. The rotation sequence is as follows:

1. Internal request
2. Requestors 0, 1, 2, 3, 4, 5
3. Internal request

If any of the requestors is not on, then the next requestor in the sequence becomes the highest priority. After a requestor is granted the bus, it remains granted, depending upon the setting of the TKYGNT bit.

5.1.2 Rotating Alternate 0

In this scheme, $\overline{\text{REQ0}}$, if asserted, is granted the bus after every other transaction, as follows:

0, 1, 0, 2, 0, 3, 0, 4, 0, 5, 0, i (internal), and then repeat

After a requestor is granted the bus, it remains granted, depending upon the setting of the TKYGNT bit.

5.1.3 Rotating Alternate I

This scheme is identical to rotating alternate 0, except that the internal request is given the advantage, as follows:

i, 0, i, 1, i, 2, i, 3, i, 4, i, 5, and then repeat

After a requestor is granted the bus, it remains granted, depending on the setting of the TKYGNT bit.

5.2 PCI-to-I/O Bus Arbitration Control

When a PCI-to-I/O bus request cannot be granted immediately by the I/O bus arbiter, the VRC4372 I/O controller can wait up to four clock cycles before generating a PCI bus retry. Refer to Section 6.5.3 for additional information.

5.3 PCI Interface Timer (DSCTIM, RTYTIM)

5.3.1 Disconnect Timer (DSCTIM)

The DDCTIM times out a burst operation when the bridge is a target. Because the VRC4372 cannot take a burst operation as a target, this timer is not necessary.

5.3.2 Retry Timer (RTYTIM)

This timer programs the maximum number of tries that a master performs before terminating a transaction. On a reset, the value 0x00 is loaded into the timer to disable the feature. To enable the timer, a nonzero value is loaded. When the timer expires, $\overline{\text{SERR}}$ is asserted (if enabled) and a bit is set in the Status register of the affected DMA channel.

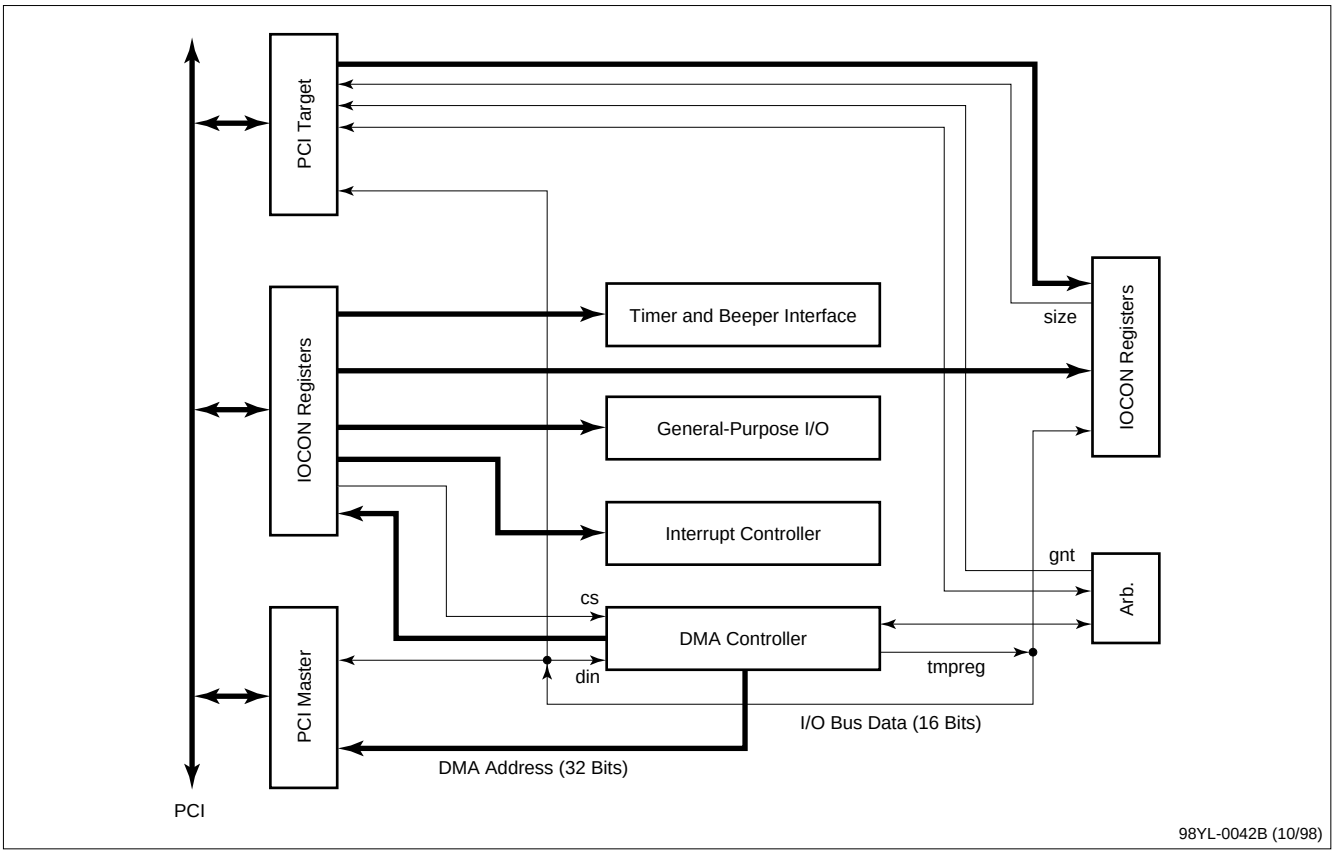
6.0 I/O Controller

6.1 Overview

The I/O controller (IOCON) block contains the following subblocks (Figure 2):

- Interface controller
- DMA controller
- Timer and beeper interface
- General-purpose I/O
- Interrupt controller

Figure 2. I/O Controller Block Diagram



6.1.1 I/O Register Memory Map

The I/O registers are grouped by function (Table 17):

Table 17. Control Register Memory Map

Offset Value	Register Name	Reference/Description
0x01ff–0x0100	General-Purpose I/O	Section 6.2.2 on page 15
0x02ff–0x0200	Interrupt Controller	Section 6.3.2 on page 17
0x03ff–0x0300	Timer and Beeper	Section 6.4.2 on page 22
0x04ff–0x0400	I/O Bus Interface	Section 6.5.2 on page 25
0x06ff–0x0500	DMA Controller	Section 6.6.2 on page 43
0xffff–0x0700	Undefined	Read as 0

6.2 General-Purpose I/O Pins

6.2.1 Overview

Eleven general-purpose I/O pins provide glueless access to system status and control information. All pins can be individually defined as either input or output via the GPDIR register. Pins configured as outputs receive output data from the GPOUT register. Pins configured as inputs or outputs can be read back via the GPIN register.

6.2.2 Signal and Pin Descriptions

The general-purpose I/O block consists of 11 bidirectional pins. When configured as outputs, these pins are clocked directly from the rising edge of the PCI CLK signal and do not float. When configured as inputs, these pins are considered static inputs and are not sampled.

Bit(s)	Symbol	Description
10:0	GPIO[10:0]	Default direction is input. No inversion occurs on these pins.

6.2.3 General-Purpose I/O Registers

Table 18. General-Purpose I/O Registers

Symbol	Offset Value	Reset Value	R/W	Description
GPDIR	0x0103–0x0100	Input direction	R/W	Sets the direction for each pin
GPOUT	0x0107–0x0104	0x0000 0000	R/W	Output value to each pin
GPIN	0x010b–0x0108	N/A	R	Input value from each pin
—	0x01ff–0x010c	N/A	R	Read as 0

6.2.4 General-Purpose Direction Register (GPDIR)

This register selects the GPIO pins to perform as either inputs or outputs.

Bit(s)	Symbol	Description
10:0	GPDIR[10:0]	Each bit controls the direction of the GPIO pin of the same number. At power-up, all bits are reset to 0 (input). When a bit is set to 1, the appropriate GPIO pin is output.
31:11	Unused	Read back as zero

6.2.5 General-Purpose Output Register (GPOUT)

This register contains the data to be placed on GPIO pins configured as outputs. Data written to input pins is ignored.

Bit(s)	Symbol	Description
10:0	GPOUT[10:0]	Each bit sends data onto the GPIO pin of the same number.
31:11	Unused	Read back as zero

6.2.6 General-Purpose Input Register (GPIN)

This register provides the capability to read the state of the GPIO pins at the pad. All pins, whether configured as input or output, can be read.

Bit(s)	Symbol	Description
10:0	GPIN[10:0]	Each bit reads data back from the GPIO pin of the same number.
31:11	Unused	Read back as zero

6.3 Interrupt Controller

6.3.1 Overview

The interrupt controller block controls the 14 external interrupt pins and internal interrupt sources (two timer ticks, four DMA channel interrupts, one watchdog timer, and a host of PCI error sources) coded onto three output interrupt pins. This interrupt controller implements a very simple scheme that does not perform interrupt encoding.

Each of the 14 external interrupt lines ($\overline{\text{INT}}$[13:0]) can be programmed separately to falling edge, rising edge, active low, or active high; triggering occurs via the INTPOL and INTTRIG registers. Additionally, each of the 14 raw interrupt lines may be read using the INTPINS register.

Each of the three IPL[2:0] interrupt output pins has an associated Interrupt Source Mask register (INTMSK[2:0]) and Interrupt Status register (INTSTAT[2:0]).

6.3.2 Signal and Pin Descriptions

Symbol	Description
IPL[2:0]	These three, twice-clocked, active-low outputs are the concentrated interrupt lines to the CPU.
$\overline{\text{INT}}$[13:0]	These 14 inputs serve as external interrupts to the IPL interrupt pins. Each input can be programmed to perform as a rising-edge, falling-edge, active-low, or active-high interrupt.
$\overline{\text{DMA\_INT}}$[3:0]	These four signals originate internally, one from each of the four DMA channels. They are active low when asserted and are cleared at the DMA channel source.
TICK[1:0]	These two signals originate internally, one from each of the two timer channels. They are rising-edge when triggered and require clearing within the interrupt controller, using the INTSTAT write operation.
$\overline{\text{TOUT\_INT}}$	This signal originates internally from the I/O bus time-out block. This signal is active low when asserted and cleared using the IOTOUT register clearing mechanism described in Section 6.5.
$\overline{\text{PERR\_INT}}$	This signal originates internally from the PCI bus interface. It is triggered at the falling edge and requires clearing within the interrupt controller using the INSTAT write operation. This interrupt source signals that the VRC4372 has sensed that the PCI $\overline{\text{PERR}}$ signal is asserted during a valid PCI transaction originating from any device on the PCI bus (including VRC4372).
$\overline{\text{SERR\_INT}}$	This signal originates internally from the PCI bus interface. It is triggered at the falling edge and requires clearing within the interrupt controller using the INSTAT write operation. This interrupt source signals that the VRC4372 has sensed that the PCI $\overline{\text{SERR}}$ signal is asserted during a valid PCI transaction originating from any device on the PCI bus (including VRC4372).
$\overline{\text{SIG\_TA}}$	This signal originates internally from the PCI bus interface. It is triggered at the falling edge and requires clearing within the interrupt controller using the INSTAT write operation. This interrupt source signals that the VRC4372 has signalled a PCI “Target Abort” during a valid PCI target access to VRC4372.
$\overline{\text{REC\_TA}}$	This signal originates internally from the PCI bus interface. It is triggered at the falling edge and requires clearing within the interrupt controller using the INSTAT write operation. This interrupt source signals that the VRC4372 has received a PCI “Target Abort” during a valid PCI target access from a DMA channel.
$\overline{\text{SIG\_MA}}$	This signal originates internally from the PCI bus interface. It is triggered at the falling edge and requires clearing within the interrupt controller using the INSTAT write operation. This interrupt source signals that the VRC4372 has signalled a PCI “Master Abort” during a valid PCI master access from a DMA channel.
$\overline{\text{PCI\_ADD}}$	This signal originates internally from the PCI bus interface. It is triggered at the falling edge and requires clearing within the interrupt controller using the INSTAT write operation. This interrupt source signals that the VRC4372 has detected a parity error on the PCI address bus.
$\overline{\text{RET\_ERR}}$	This signal originates internally from the PCI bus interface. It is triggered at the falling edge and requires clearing within the interrupt controller using the INSTAT write operation. This interrupt source signals that the retry counter limit is reached on a PCI bus cycle originating from a DMA channel.

6.3.3 Interrupt Controller Registers

Table 19. Interrupt Controller Registers

Symbol	Offset Value	Reset Value	R/W	Description
INTPOL	0x0203–0x0200	0x0000 0000	R/W	$\overline{\text{INT}}$[13:0] input inversion control
INTTRIG	0x0207–0x0204	0x0000 0000	R/W	$\overline{\text{INT}}$[13:0] level/edge selection control
INTPINS	0x020b–0x0208	N/A	R	Raw input value from each $\overline{\text{INT}}$ pin
INTMSK0	0x020f–0x020c	0x0000 0000	R/W	IPL0 mask register
INTSTAT0	0x0213–0x0210	0x0000 0000	R/W	IPL0 status register
INTMSK1	0x0217–0x0214	0x0000 0000	R/W	IPL1 mask register
INTSTAT1	0x021b–0x0218	0x0000 0000	R/W	IPL1 status register
INTMSK2	0x021f–0x021c	0x0000 0000	R/W	IPL2 mask register
INTSTAT2	0x0223–0x0220	0x0000 0000	R/W	IPL2 status register
Unused	0x02ff–0x0224	N/A	R	Read as 0

Interrupt Polarity Control Register (INTPOL)

This register provides the ability to selectively invert each of the 14 interrupt input pins ($\overline{\text{INT}}$[13:0]) before they enter the interrupt controller. The register also provides the ability to deal with any active polarity or edge type that might be required by an external peripheral.

Bit(s)	Symbol	Description
13:0	INTPOL[13:0]	Each bit corresponds to the $\overline{\text{INT}}$ input pin of the same number. When a bit is clear, the corresponding $\overline{\text{INT}}$ interrupt input pin is brought into the controller without inversion to provide the capability to trigger on either a low level or a rising edge. When a bit is set, the corresponding input pin is brought into the controller inverted to provide the ability to trigger on either a high level or a falling edge.
31:14	Unused	Read back as zero

Interrupt Trigger Control Register (INTTRIG)

This register provides the ability to selectively trigger on either edge- or level-triggered interrupts for each of the incoming $\overline{\text{INT}}$ interrupt lines that have passed through the INTPOL block.

Bit(s)	Symbol	Description
Bits 13:0	INTTRIG[13:0]	Each bit corresponds to the $\overline{\text{INT}}$ input pin of the same number. When a bit is clear, the corresponding $\overline{\text{INT}}$ interrupt input pin is level-triggered (active low when the INTPOL bit is set to 0, and active high when INTPOL is set to 1). When a bit is set, the corresponding input pin is edge-triggered (rising edge when INTPOL is set to 0; falling edge when INTPOL is set to 1).
Bits 31:14	Unused	Read back as zero

Interrupt Pin Read Register (INTPINS)

This register provides the ability to read the raw interrupt pins. It does not latch the input pin states and it contains static inputs for unused interrupt input lines.

Bit(s)	Symbol	Description
13:0	INTPINS[13:0]	Each bit corresponds to the $\overline{\text{INT}}$ input pin of the same number. When a bit is clear, the corresponding $\overline{\text{INT}}$ interrupt input pin is low; when a bit is set, the corresponding $\overline{\text{INT}}$ pin is high. If dynamic input sources are used, then the buffer should be read repeatedly until two consecutive read cycles return the same value.
31:14	Unused	Read back as zero

Interrupt MASK Registers (INTMSK[2:0])

Each of these registers corresponds to the output interrupt pin IPL[2:0] of the same number. These three registers provide a means for selecting the interrupt input sources that are assigned to a particular IPL output pin. It is recommended that an interrupt source only be assigned to zero or to one IPL output and not all. Every possible interrupt source has a corresponding bit in each of these registers. When this bit is set, the interrupt source is enabled to the IPL output pin. When this bit is clear, the interrupt source is masked on that IPL output pin.

Bit(s)	Symbol	Description
13:0	$\overline{\text{INT}}$[13:0]	Each bit corresponds to an external qualified interrupt input line of the same number.
17:14	$\overline{\text{DMA\_INT}}$[3:0]	Interrupt mask bit for DMA channels 3 to 0
19:18	TICK[1:0]	Interrupt mask bit for timer TICK1 and TICK0
21:20	Unused	Undefined
22	TOUT_INT	Interrupt mask bit for an I/O bus timeout interrupt
23	$\overline{\text{PERR\_INT}}$	Interrupt mask bit for the PCI $\overline{\text{PERR\_INT}}$ signal
24	$\overline{\text{SERR\_INT}}$	Interrupt mask bit for the PCI $\overline{\text{SERR\_INT}}$ signal
25	$\overline{\text{SIG\_TA}}$	Interrupt mask bit for the PCI $\overline{\text{SIG\_TA}}$ signal
26	$\overline{\text{REC\_TA}}$	Interrupt mask bit for the PCI $\overline{\text{REC\_TA}}$ signal
27	$\overline{\text{SIG\_MA}}$	Interrupt mask bit for the PCI $\overline{\text{SIG\_MA}}$ signal
28	$\overline{\text{PCI\_ADD}}$	Interrupt mask bit for the PCI $\overline{\text{PCI\_ADD}}$ signal
29	$\overline{\text{RET\_ERR}}$	Interrupt mask bit for the PCI $\overline{\text{RET\_ERR}}$ signal
30	Unused	Read back as zero
31	Unused	Read back as zero

Interrupt Status Registers (INTSTAT[2:0])

Each of these registers corresponds to the output interrupt pin IPL[2:0] of the same number. These three registers provide a snapshot of the source of an IPL interrupt taken on one clock edge during the read cycle (synchronized). This register also provides a means for clearing edge-triggered interrupt sources. To clear an edge-triggered interrupt, a zero is written to this register in the bit position of the interrupt. And, since common edge-detection logic is used for an interrupt source to both IPL outputs, clearing a bit in one INTSTAT register clears the bit in all registers. Zeros written to level-triggered interrupt sources have no effect. Every possible interrupt source has a corresponding bit in each of these registers. When this bit is set, the interrupt source is driving the IPL pin. When this bit is clear, the interrupt source is not the cause of the IPL output assertion. Note that because level-triggered sources are not clocked, it is possible (but unlikely) that an interrupt could be generated and disappear before this register is read.

Bit(s)	Symbol	Description
13:0	$\overline{\text{INT}}$[13:0]	Each bit corresponds to an external qualified interrupt input line of the same number.
17:14	$\overline{\text{DMA\_INT}}$[3:0]	Interrupt status bit for DMA channels 3 to 0
19:18	TICK[1:0]	Interrupt status bit for timer TICK1 and TICK0
21:20	Unused	Undefined
22	$\overline{\text{TOUT\_INT}}$	Interrupt status bit for an I/O bus timeout interrupt
23	$\overline{\text{PERR\_INT}}$	Interrupt status bit for the PCI $\overline{\text{PERR\_INT}}$ signal
24	$\overline{\text{SERR\_INT}}$	Interrupt status bit for the PCI $\overline{\text{SERR\_INT}}$ signal
25	$\overline{\text{SIG\_TA}}$	Interrupt status bit for the PCI $\overline{\text{SIG\_TA}}$ signal
26	$\overline{\text{REC\_TA}}$	Interrupt status bit for the PCI $\overline{\text{REC\_TA}}$ signal
27	$\overline{\text{SIG\_MA}}$	Interrupt status bit for the PCI $\overline{\text{SIG\_MA}}$ signal
28	$\overline{\text{PCI\_ADD}}$	Interrupt status bit for the PCI $\overline{\text{PCI\_ADD}}$ signal
29	$\overline{\text{RET\_ERR}}$	Interrupt status bit for the PCI $\overline{\text{RET\_ERR}}$ signal
30	Unused	Read back as zero
31	Unused	Read back as zero

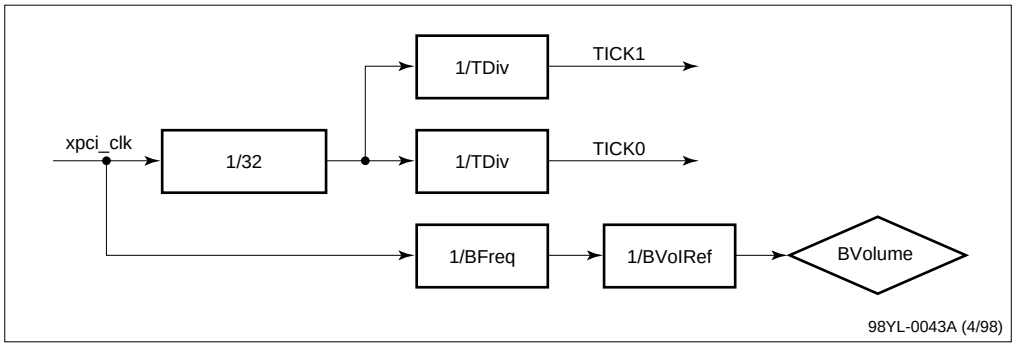
6.4

Timer and Beeper Interface

6.4.1 Overview

The timer section generates a system tick with a programmable frequency that can be used to issue a periodic interrupt. The beeper section generates the pulse-width-modulated waveform required for activating the speaker driver.

Figure 3. Timer and Beeper Operation



6.4.2 Signal and Pin Descriptions

The timer and beeper block consists of two internal signals and one output pin.

Symbol	Description
IOBEEP	This signal drives an operational amplifier (op-amp) that activates an external beeper. An internal version of this signal is also used by the DMA channels as a method to request DMA channel service.
TICK[1:0]	These two internal signals are used by the interrupt controller block to generate periodic interrupts to the CPU. The interrupt controller block is responsible for capturing, clearing, and masking these signals as appropriate.

6.4.3 Timer and Beeper Registers

Table 20. Timer/Beeper Registers

Symbol	Offset Value	Reset Value	R/W	Description
IOTIME0	0x0303–0x0300	0x0000 0000	R/W	Controls timer 0
IOTVAL0	0x0307–0x0304	0x0000 0000	R	Contains running value for timer 0
IOTIME1	0x030b–0x0308	0x0000 0000	R/W	Controls timer 1
IOTVAL1	0x030f–0x030c	0x0000 0000	R	Contains running value for timer 1
IOBEEP	0x0313–0x0310	0x0000 0000	R/W	Controls the pitch and volume of the beep
—	0x03ff–0x0314	—	R	Read as 0

6.4.4 Timer Control Register (IOTIMEx)

This register controls the timer “x.” Any write cycle to this register resets the counter to its initial value.

Bit(s)	Symbol	Description
15:0	TDiv	Tick generator divider. Sets the system tick between 16 Hz and 1 MHz. 0 disables the timer.
29:16	Unused	Read as 0
30	TIntEN	Timer interrupt enable (1)
31	Terminal	Timer terminal value reached (1). Set by the hardware when the timer reaches its terminal count. Reset by the hardware when the IOTIME register is read. This bit is updated regardless of the TIntEN value.

6.4.5 Timer Value Register (IOTVALx)

This register contains the running value for timer “x.”

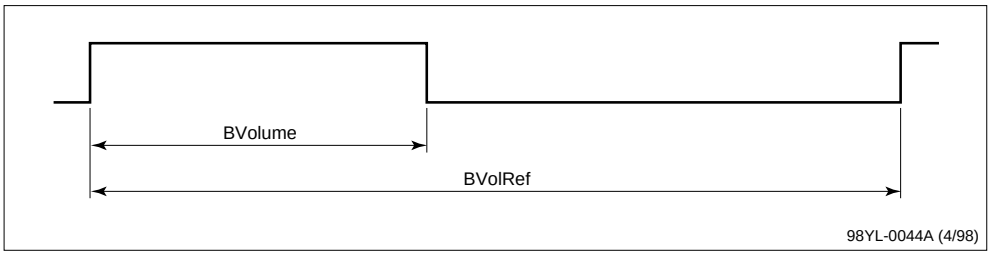
Bit(s)	Symbol	Description
15:0	Count	Tick generator count
31:16	Unused	Read as 0

6.4.6 Beeper Control Register (IOBEEP)

The beeper section generates a sound by dividing the system clock by BFreq, which sets the pitch. The period of the generated sound is subsequently divided into a high period and a low period, thus providing a variable volume through pulsewidth modulation (PWM).

Bit(s)	Symbol	Description
10:0	BVolRef	Sets the reference count width for the PWM output. When set to 0, disables the beeper. When set to a value of 2047 (for a 33-MHz system clock), provides pulse-width modulation with 11 bits of resolution and an output frequency in the range of 16 Hz to 16 kHz.
20:11	BVolume	Beeper volume control. Sets the pulse width (and therefore volume). BVolume has a valid range between 0 and BVolRef/2, for minimum to maximum volume, respectively.
21	Unused	Read as 0
31:22	BFreq	Beeper frequency divisor. Sets the pitch of the beeper to CLK/(BFreq x BVolRef).

Figure 4. Pulse Width Modulation



6.5

I/O Bus Interface Controller

6.5.1 Overview

The I/O bus within the VRC4372 controller is a mostly demultiplexed ISA-like bus that controls devices with up to 24 bits of address and up to 16 bits of data. The timing for I/O bus cycles is provided by the PCI CLK.

The I/O devices controlled by the VRC4372 controller are assigned a 256-MB memory region, located at the address specified by the IOADD register in the PCI header. This region is further divided into 16 subregions of 16 MB each. The first 10 subregions are assigned to the chip-select signals $\overline{CS}[9:0]$. The next three subregions are assigned to read and write chip-select pairs $\overline{CPRD}[2:0]$ and $\overline{CPWR}[2:0]$. The remaining three subregions are reserved for future use. Each subregion can host several external devices by decoding the lower address bits (starting with A23). To facilitate this additional decoding, the software should enforce fixed addresses on these lower bits. Along with these 16 addressable subregions, the I/O bus is also used for DMA transfers under the control of four DMA channels.

Table 21. Chip-Select Memory Map and Profile Register Assignment

A[27:0]	Chip-Select or DMA Channel	Profile Register
000 0000–0FF FFFF	CS0	IOPROF_00
100 0000–1FF FFFF	CS1	IOPROF_01
200 0000–2FF FFFF	CS2	IOPROF_02
300 0000–3FF FFFF	CS3	IOPROF_03
400 0000–4FF FFFF	CS4	IOPROF_04
500 0000–5FF FFFF	CS5	IOPROF_05
600 0000–6FF FFFF	CS6	IOPROF_06
700 0000–7FF FFFF	CS7	IOPROF_07
800 0000–8FF FFFF	CS8	IOPROF_08
900 0000–9FF FFFF	CS9	IOPROF_09
A00 0000–AFF FFFF	CP0	IOPROF_0A
B00 0000–BFF FFFF	CP1	IOPROF_0B
C00 0000–CFF FFFF	CP2	IOPROF_0C
D00 0000–FFF FFFF	—	Reserved
—	DMA0	IOPROF_10
—	DMA1	IOPROF_11
—	DMA2	IOPROF_12
—	DMA3	IOPROF_13
—	—	Reserved

Note: No addresses are assigned to the four DMA channels.

6.5.2 Signal and Pin Descriptions

The general interface of the I/O bus consists of output, input, and bidirectional pins. Unless otherwise specified, all outputs are clocked directly from the rising edge of the PCI CLK signal and do not float between valid accesses. Some outputs have programmable inversion capabilities controllable through the IOPOL register.

Signal Name	Description
IOA[7:0]	These eight output pins drive the least-significant eight bits of the byte address during 8- and 16-bit-wide I/O bus accesses. These lines are driven valid during the first CLK cycle of a valid I/O bus access and remain valid until the completion of the same I/O bus access. Between valid I/O bus accesses, the last address on the bus is held.
IOD[15:0]	These 16 bidirectional pins implement a multiplexed address and data bus. During the first CLK cycle of a valid I/O bus access, the IOD bus (bit A[23:8] in the IOPROF_xx register of a region) can be configured to output the most significant 16 address bits (IOA[23:8]). During all subsequent CLK cycles of a valid I/O bus access, this bus is used for data transfers. Turnaround cycles are enforced between the address phase and the first data read phase in systems that use the $\overline{\text{BUFOE}}$ pin. To avoid contention in other systems with fast devices, either disable the address phase (bit A[23:8] in the IOPROF_xx register) or program the $\overline{\text{RD}}$ line.
$\overline{\text{HBE}}$	This output pin acts as a high-byte enable during accesses to 16-bit devices. It remains deasserted 1) during accesses to 8-bit devices and 2) during accesses to 16-bit devices that do not access the most-significant IOD bus byte. This signal is asserted during most-significant bit (MSB) access to 16-bit devices, has the same timing as the IOA[7:0] bus, and is held in its last state between valid I/O bus accesses.
$\overline{\text{LBE}}$	This output pin acts as a low-byte enable during accesses to 8- and 16-bit devices. It is asserted during any valid I/O bus access that requires data from the least-significant IOD bus byte (IOD[7:0]). This signal has the same timing as the IOA[7:0] bus and is held in its last state between valid I/O bus accesses.
$\overline{\text{ALE}}$	This output pin acts as a latch enable for capturing the upper address (IOA[23:8]) from the IOD bus. It has the same gross timing as IOFRAME. The timing of the asserting edge of $\overline{\text{ALE}}$ should be constrained to allow it to act as a latch enable for capturing the IOA[23:8] address bits from the IOD[15:0] bus with a standard 74F373. $\overline{\text{ALE}}$ is not synchronized with the PCI clock.
$\overline{\text{IOFRAME}}$	This output indicates that a valid I/O access is occurring on the I/O bus. It is clocked on the rising edge of CLK. $\overline{\text{IOFRAME}}$ is asserted on the first rising CLK edge of a valid I/O bus access and deasserted on the last rising edge before the end of a valid I/O bus access. This signal must remain deasserted between valid I/O bus accesses and must be used synchronously with the PCI clock.
$\overline{\text{W\_R}}$	This output indicates whether the current I/O bus cycle is a write or read cycle. This signal has the same timing as the IOA[7:0] bus and shall be held in its last state between valid I/O bus accesses. At power-up, the default polarity of this output is write = 1 and read = 0. This signal can be inverted via programming. Refer to “IOPOL (I/O Bus Output Polarity Control Register)” on page 30 for details.
$\overline{\text{RD}}[1:0]$	These outputs are identical copies of the same signal, collectively referred to as $\overline{\text{RD}}$. $\overline{\text{RD0}}$ and $\overline{\text{RD1}}$ act as a common read pulse signal during active I/O bus cycles. The timing of $\overline{\text{RD}}$ is programmable for each chip-select or DMA region via the I/O Profile register (IOPROF_xx) for the region. Both versions of this signal receive the same programming information and must remain deasserted between valid I/O bus accesses.

Signal Name	Description
WR[1:0]	These outputs are identical copies of the same signal, collectively referred to as WR. WR0 and WR1 act as a common write pulse signal during active I/O bus cycles. WR is programmable for each chip-select or DMA region via the I/O Profile register (IOPROF_xx) for the region. Both versions of the signal receive the same programming information and must remain deasserted between valid I/O bus accesses.
BUFOE	This output controls the output enable of an external IOD bus buffer and is clocked on the falling edge of PCI CLK. BUFOE can be enabled and disabled via the BUF_on bit in the IOPROF_xx register for each region. When enabled, BUFOE follows IOFRAME timing by half a clock cycle for all valid I/O bus accesses, except when bit A[23:8] (in IOPROF_xx) is also set and a read cycle is occurring. For this exception, BUFOE is asserted 1.5 cycles after IOFRAME and deasserted 0.5 clock cycles after IOFRAME. BUFOE is always deasserted between valid I/O bus accesses. This signal can be inverted via programming. Refer to “IOPOL (I/O Bus Output Polarity Control Register)” on page 30 for more information.
CPRD[2:0]	These three outputs function as strobes that are asserted during accesses to designated memory regions. Each CPRD output strobe is paired with a CPWR output strobe. Each CPRD/CPWR pair has a unique 16-MB memory space. Each CPRD may operate in one of two modes, depending upon the setting of the CS_low bit in the I/O Profile (IOPROF_xx) register for the region. When CS_low is clear, CPRD follows the RD timing during valid read accesses to its region. When CS_low is set, CPRD follows RD and WR timing during valid read and write accesses to its region. Between valid accesses to an assigned region, these outputs remain deasserted. At power-up, the default polarity of these outputs is defined as 1 = deasserted and 0 = asserted. This signal can be inverted via programming. Refer to “IOPOL (I/O Bus Output Polarity Control Register)” on page 30 for more information.
CPWR[2:0]	These three outputs function as strobes that are asserted during accesses to designated memory regions. Each CPWR output strobe is paired with a CPRD output strobe. Each CPWR/CPRD pair has a unique 16-MB memory space. Each CPWR may operate in one of two modes, depending upon the setting of the CS_low bit in the I/O Profile (IOPROF_xx) register for the region. When CS_low is clear, CPWR follows the WR timing during valid write accesses to its region. When CS_low is set, CPWR follows RD and WR timing during valid read and write accesses to its region. Between valid accesses to an assigned region, these outputs remain deasserted. At power-up, the default polarity of these outputs is defined as 1 = deasserted and 0 = asserted.
CS[9:0]	These 10 outputs function as clocked chip-selects asserted during accesses to uniquely assigned memory regions. Timing control for each output is programmable via the I/O Profile register (IOPROF_xx) for the region. At power-up, the default polarity of these outputs is defined as 1 = deasserted and 0 = asserted.
IOREADY	This input can be used to extend an I/O bus access to allow for externally generated timing control. Each region can be configured to either ignore or respect this input via the RDY_on bit in the I/O Profile register (IOPROF_xx) for the region. Refer to the description of the RDY_on bit on page 28 for further details.
TOUT_INT	This internal signal originates from the I/O bus timeout block and acts as an interrupt source to the interrupt controller block. This signal is active low, asserted, and cleared using the IOTOUT register clearing mechanism.

6.5.3 Registers

There are 17 identical profile registers, each of which controls the access profile of one chip-select or DMA channel. The timing profiles control I/O bus access traits such as data width, address width, and read/write pulse widths and positioning. During an I/O bus access, the I/O bus controller makes the information about the data width for the active chip-select or DMA channel available to the target subsection of the PCI interface.

Table 22. I/O Bus Profile Registers

Symbol	Offset Value	Reset Value	R/W	Description
IOPROF_00	0x0400	0x0000	R/W	Profile for CS0
IOPROF_01	0x0402	0x0000	R/W	Profile for CS1
IOPROF_02	0x0404	0x0000	R/W	Profile for CS2
IOPROF_03	0x0406	0x0000	R/W	Profile for CS3
IOPROF_04	0x0408	0x0000	R/W	Profile for CS4
IOPROF_05	0x040a	0x0000	R/W	Profile for CS5
IOPROF_06	0x040c	0x0000	R/W	Profile for CS6
IOPROF_07	0x040e	0x0000	R/W	Profile for CS7
IOPROF_08	0x0410	0x0000	R/W	Profile for CS8
IOPROF_09	0x0412	0x0000	R/W	Profile for CS9
IOPROF_0A	0x0414	0x0000	R/W	Profile for CPRDn0 and CPWRn0
IOPROF_0B	0x0416	0x0000	R/W	Profile for CPRDn1 and CPWRn1
IOPROF_0C	0x0418	0x0000	R/W	Profile for CPRDn2 and CPWRn2
—	0x041f–0x041a	—	R	Reserved
IOPROF_10	0x0420	0x0000	R/W	Profile for DMA channel 0
IOPROF_11	0x0422	0x0000	R/W	Profile for DMA channel 1
IOPROF_12	0x0424	0x0000	R/W	Profile for DMA channel 2
IOPROF_13	0x0426	0x0000	R/W	Profile for DMA channel 3
—	0x042f–0x0428	—	R	Read as 0
IOTOUT	0x0433–0x0430	0x0000 0000	R/W	Timeout control
IOPOL	0x0437–0x0434	0x0000 0000	R/W	Programmable inversion for output
—	0x04ff–0x0434	—	R	Read as 0

IOPROF_xx (I/O Bus Profile Registers)

Bit(s)	Symbol	Description
0	CS_low	This bit is used to position the asserting edge of $\overline{CS}$ or $\overline{DACK}/\overline{TC}$ within an I/O bus access. When clear, the $\overline{CS}/\overline{DACK}/\overline{TC}$ is asserted with IOA during the first clock cycle of an access. When set, the $\overline{CS}/\overline{DACK}/\overline{TC}$ bit is asserted during the second clock cycle of an access. This bit is used as a mode select during accesses involving $\overline{CPRD}[2:0]$ and $\overline{CPWR}[2:0]$. Refer to Section 6.5.2 for details.
2:1	CON_set	These bits are used to position the asserting edge of the control pulse ($\overline{RD}$ or $\overline{WR}$ or $\overline{CPRDX}$ or $\overline{CPWR}$) within an I/O bus access. The edge position is specified as follows: CON_set During CLK Cycle 0 2 1 3 2 4 3 5
5:3	CON_wid	These bits are used to control the width in CLK cycles when the control pulse ($\overline{RD}$ or $\overline{WR}$ or $\overline{CPRD}$ or $\overline{CPWR}$) remains asserted. This width is defined as 1 + (2 x CON_wid), described as follows: CON_wid Number of CLK Cycles 0 1 1 3 2 5 3 7 4 9 5 11 6 13 7 15
6	CON_hold	When set, this bit inserts two additional CLK cycles between the deasserting control pulse edge and the end of the I/O bus access. When clear, this bit inserts no additional CLK cycles.
7	CS_high	This bit is used to position the deasserting edge of $\overline{CS}$ or $\overline{DACK}/\overline{TC}$ within an I/O bus access. When clear, the $\overline{CS}/\overline{DACK}/\overline{TC}$ is deasserted with IOA during the clock following the last cycle of an access. When set, the $\overline{CS}/\overline{DACK}/\overline{TC}$ is deasserted during the last cycle of an I/O bus access. This bit is unused for $\overline{CPRD}[2:0]$ and $\overline{CPWR}[2:0]$.
8	RDY_on	$\overline{IOREADY}$ is sampled on the last rising edge of XCLK when $\overline{IOFRAME}$ is still asserted. If $\overline{IOREADY}$ is low at this time, the I/O cycle continues (as is the case when external ready is not used). If $\overline{IOREADY}$ is high, the I/O controller waits for a rising edge on XCLK, after which $\overline{IOREADY}$ is asserted to deassert $\overline{IOFRAME}$. When this condition occurs, data is sampled (for read accesses) and Data_hold states are started. When RDY_on is clear, the $\overline{IOREADY}$ input pin is ignored.
9	$\overline{D[16:8]}$	This bit defines the width of the IOD bus used by the external device. When clear, this bit configures the IOD bus width to one byte using IOD[7:0]. When set, this bit configures the IOD bus width to two bytes (where IOD[15:8] is the MSB and IOD[7:0] is the LSB).
10	A[23:8]	When this bit is set, the highest 16 bits of the I/O address bus (IOA[23:8]) are driven onto the I/O data bus (IOD[15:0]) during the first cycle of the access. The $\overline{IOFRAME}$ signal may be used to latch these address bits before the IOD bus is used for data transfers. When this bit is clear, the IOA[23:8] bits are not driven onto the IOD bus.

Bit(s)	Symbol	Description
11	BUF_on	When this bit is set, the BUFOE signal is asserted during valid I/O bus access to the region. When clear, BUFOE is not asserted during accesses to the region.
15:12	Unused	When read, returns 0

IOTOUT (I/O Bus Timeout Control Register)

This register controls the generation of a timeout error condition for regions that are ready-controlled (in other words, when RDY_on is set to 1 in the corresponding profile register). If RDY_on = 0, there is no timeout.

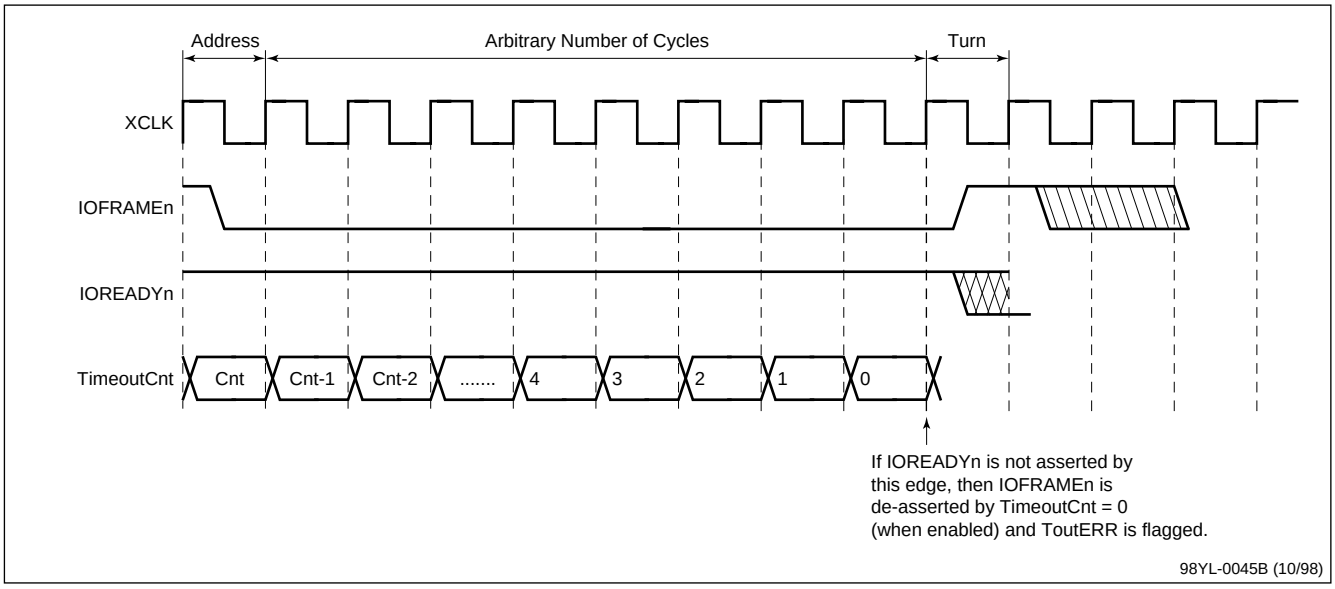
The TimeoutCnt field should be written with a value larger than the longest IOREADY-controlled access period. At every I/O bus access (DMA and regular) to an IOREADY-controlled region, the TimeoutCnt field is loaded into a counter and decremented on each rising CLK edge until IOREADY is returned asserted. When IOREADY is returned asserted, the counter is stopped and no timeout error is generated. If the counter value reaches zero before IOREADY has been returned asserted, then a timeout error condition occurs: the I/O bus controller deasserts IOFRAME as if IOREADY were low, completes the current access, and signals the error on the TOUT_INT internal signal (Figure 5).

A read cycle from this register resets the timeout condition (ToutERR) and clears the TOUT_INT signal without affecting the counter.

For additional information about error handling, refer to Section 8.0.

Bit(s)	Symbol	Description
9:0	TimeoutCnt	Number of CLK cycles to timeout
28:10	Unused	Read as 0
29	ArbTmrEn	PCI arbiter delayed retry bit. If this bit is 0, then the arbiter timer is disabled and the request on the PCI bus is retried immediately if the I/O bus is not granted immediately. If this bit is 1, then the PCI request waits four clocks for the I/O bus before generating a PCI bus retry.
30	ToutEN	Timeout enable. If this bit is set (1), timeout error reporting is enabled.
31	ToutERR	Timeout error. This bit is set by the hardware when a timeout is detected and reset by the hardware when the IOTOUT register is read. ToutERR is updated regardless of the value of ToutEN.

Figure 5. **I/O Bus Timeout Condition**



IOPOL (I/O Bus Output Polarity Control Register)

This register controls the polarity of the I/O bus pins that can be inverted glitch-free on both rising and falling edges. Each bit in this register controls the polarity of one or two output signals.Table 23 shows the bit assignments and default values.

Table 23. IOPOL Register Bit Assignments

Bit(s)	R/W	Signal Name	Reset Value
0	R/W	$\overline{W\_R}$	0
1	R/W	$\overline{CPRD0}$ $\overline{CPWR0}$	0
2	R/W	$\overline{CPRD1}$ $\overline{CPWR1}$	0
3	R/W	$\overline{CPRD2}$ $\overline{CPWR2}$	0
4	R/W	$\overline{CS0}$	0
5	R/W	$\overline{CS1}$	0
6	R/W	$\overline{CS2}$	0
7	R/W	$\overline{CS3}$	0
8	R/W	$\overline{CS4}$	0
9	R/W	$\overline{CS5}$	0
10	R/W	$\overline{CS6}$	0
11	R/W	$\overline{CS7}$	0
12	R/W	$\overline{CS8}$	0
13	R/W	$\overline{CS9}$	0
31:14	R	Unused (Note)	0
Note: Read back as 0			

6.5.4 I/O Bus Arbiter

The I/O bus controller features a simple arbiter that exists between the following two request sources, which are listed in order of priority:

- PCI target (CPU accesses to the I/O devices)
- DMA

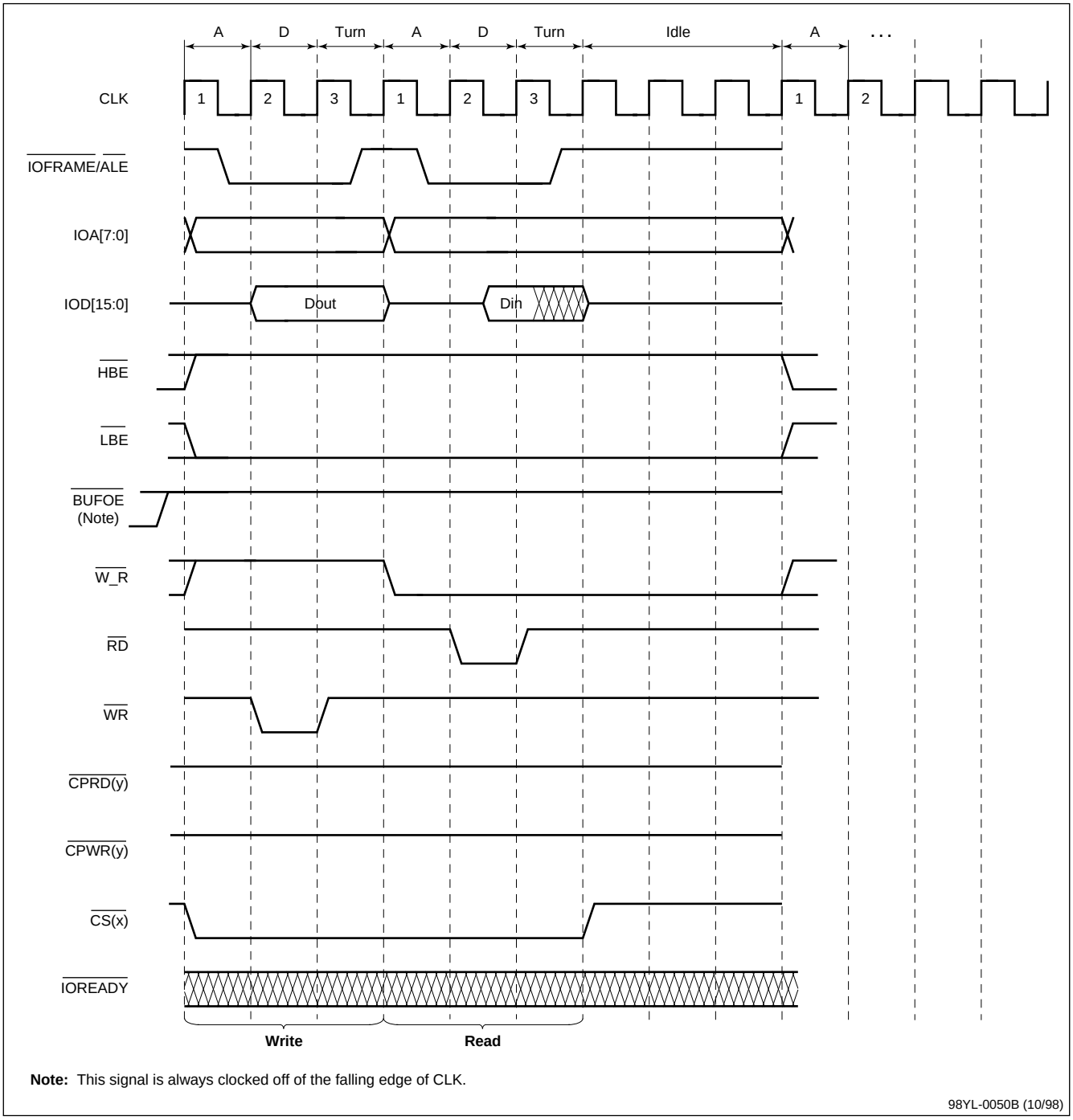
6.5.5 I/O Bus Protocol

The I/O bus is totally demultiplexed, except for an optional A[23:8] address frame that can be multiplexed onto the IOD[15:0] bus during the address phase of an I/O bus access. (See Figure 6 through Figure 15, at the end of this section, for examples of bus protocol.)

The timing of the data transfer and the end of the I/O bus frame is controlled through the deassertion of $\overline{\text{IOFRAME}}$ and the activity on the $\overline{\text{IOREADY}}$ pin. An I/O bus access transfers exactly one word of data (8 or 16 bits). The I/O bus does not support bursts.

The most basic access is three clock cycles long and occurs when CON_set, CON_wid, CON_hold, and RDY_on in the profile register are set to 0. This access consists of an Address + Data + Turn cycle for both reading and writing. The other fields within an I/O Profile register adjust pulse positioning and existence, but do not affect I/O bus access duration.

Figure 6. Basic Back-to-Back, Eight-Bit $\overline{CS}$ Access, Non-RDY, No Multiplexed Address, $\overline{BUFOE}$ Off

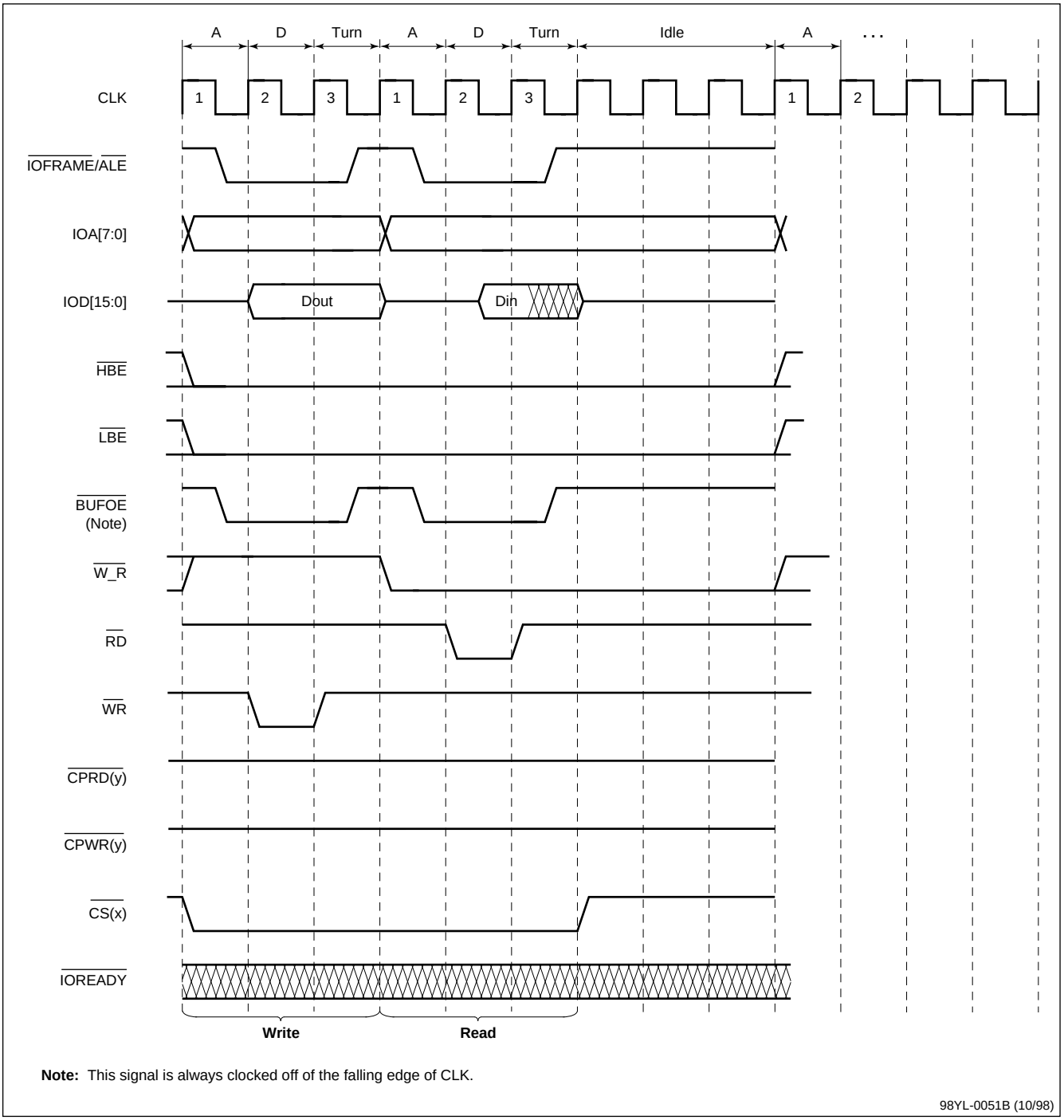


Note: This signal is always clocked off of the falling edge of CLK.

98YL-0050B (10/98)

11	10	9	8	7	6	5:3	2:1	0
BUF_on = 0	A[23:8] = 0	$\overline{A[16:8]}$ = 0	RDY_on = 0	CS_high = 0	CON_hold = 0	CON_wid = 0	CON_set = 0	CS_low = 0

Figure 7. Back-to-Back, 16-Bit $\overline{CS}$ Access, Two Bytes, Non-RDY, No Multiplexed Address, $\overline{BUFOE}$ On

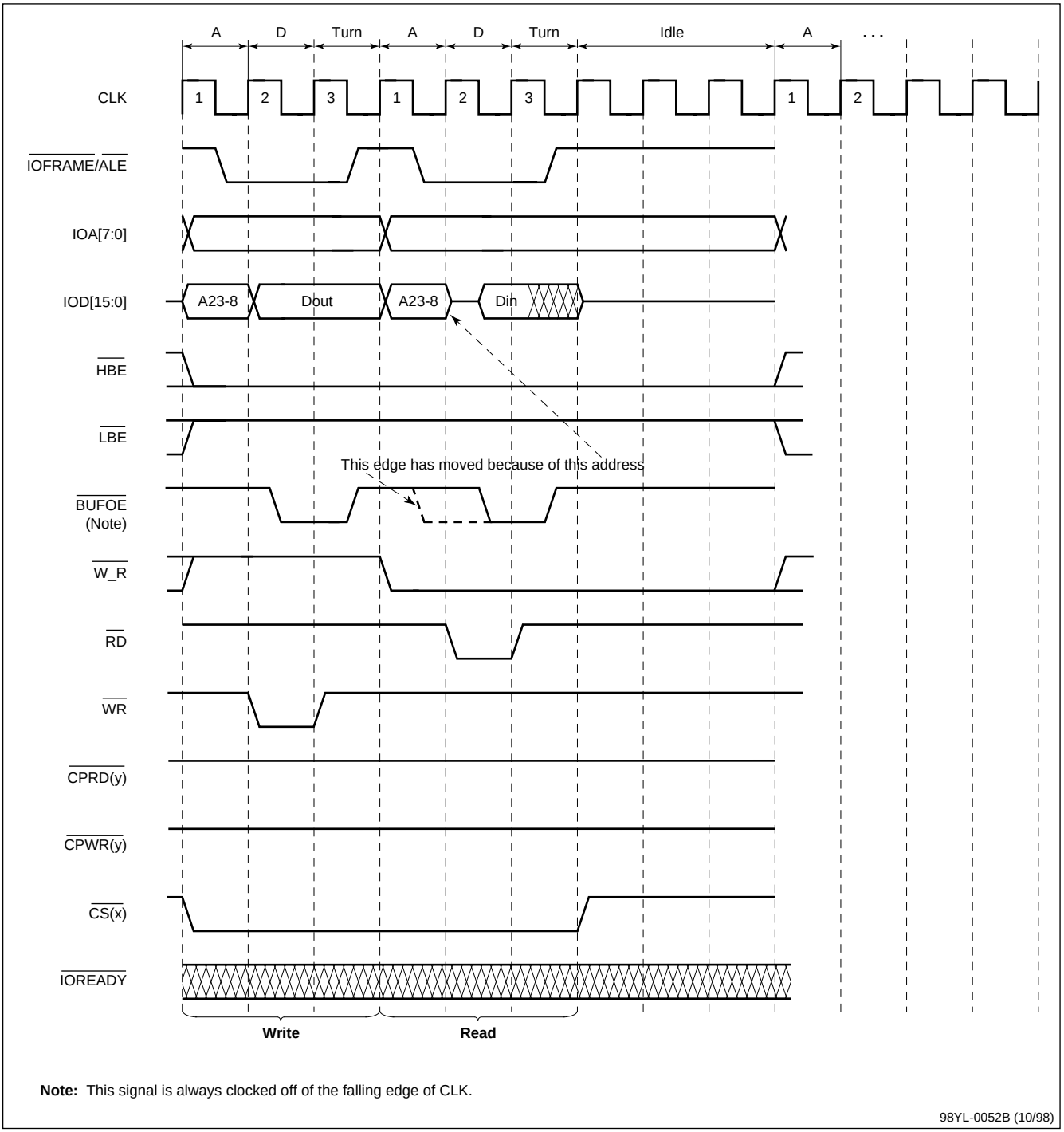


Note: This signal is always clocked off of the falling edge of CLK.

98YL-0051B (10/98)

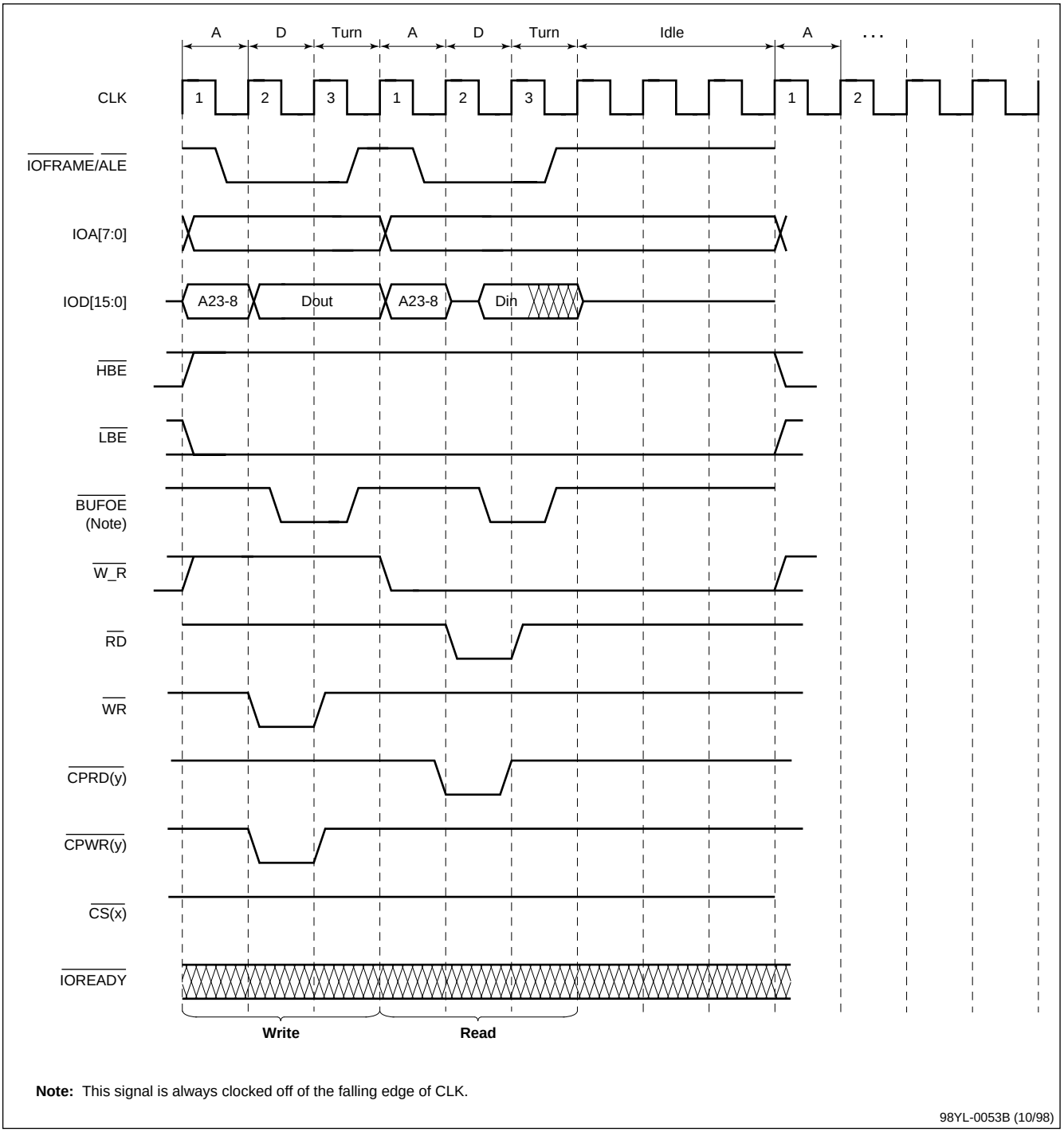
11	10	9	8	7	6	5:3	2:1	0
BUF_on = 1	A[23:8] = 0	$\overline{A[16:8]}$ = 1	RDY_on = 0	CS_high = 0	CON_hold = 0	CON_wid = 0	CON_set = 0	CS_low = 0

Figure 8. Back-to-Back, 16-Bit $\overline{CS}$ Access, High Byte, Non-RDY, Multiplexed Address, $\overline{BUFOE}$ On



11	10	9	8	7	6	5:3	2:1	0
BUF_on = 1	A[23:8] = 1	$\overline{A[16:8]}$ = 1	RDY_on = 0	CS_high = 0	CON_hold = 0	CON_wid = 0	CON_set = 0	CS_low = 0

Figure 9. Back-to-Back, 16-Bit $\overline{CP}$ Access, Low Byte, Non-RDY, Multiplexed Address, $\overline{BUFOE}$ On

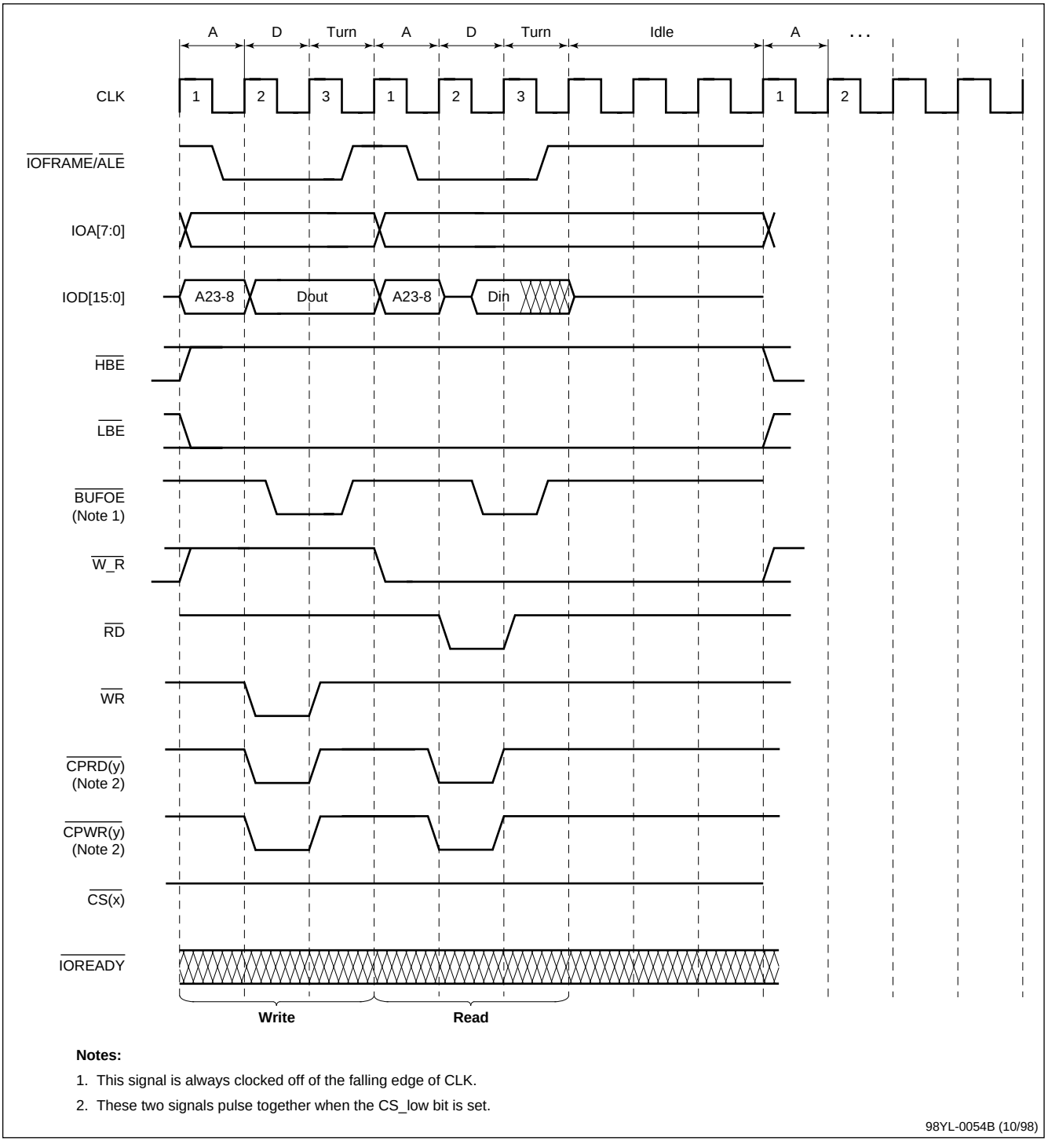


Note: This signal is always clocked off of the falling edge of CLK.

98YL-0053B (10/98)

11	10	9	8	7	6	5:3	2:1	0
BUF_on = 1	A[23:8] = 1	$\overline{A[16:8]}$ =1	RDY_on = 0	CS_high = X	CON_hold = 0	CON_wid = 0	CON_set = 0	CS_low = 0

Figure 10. Back-to-Back, 8-Bit $\overline{CP}$ Access, Non-RDY, Multiplexed Address, $\overline{BUFOE}$ On



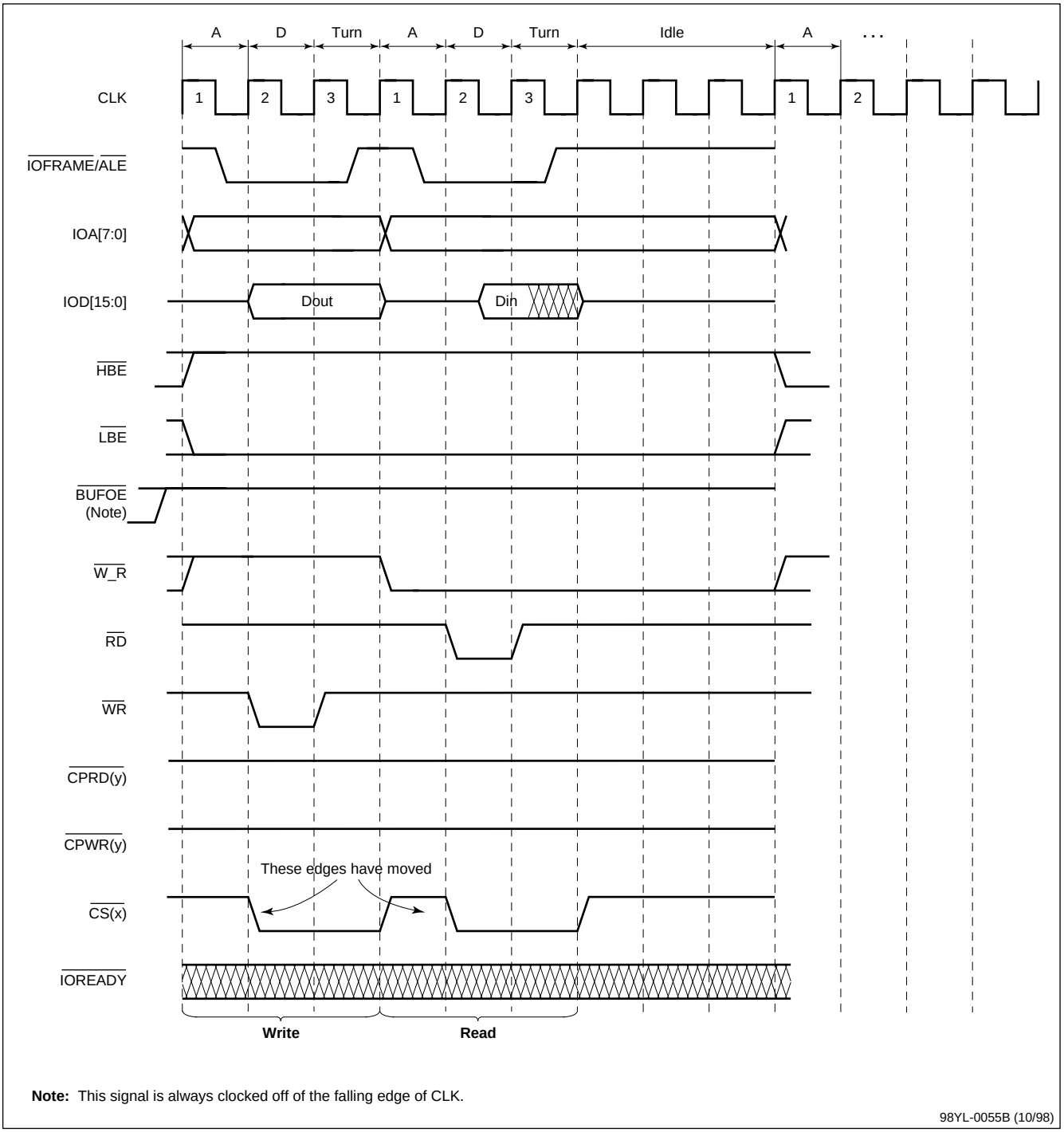
Notes:

1. This signal is always clocked off of the falling edge of CLK.
2. These two signals pulse together when the CS_low bit is set.

98YL-0054B (10/98)

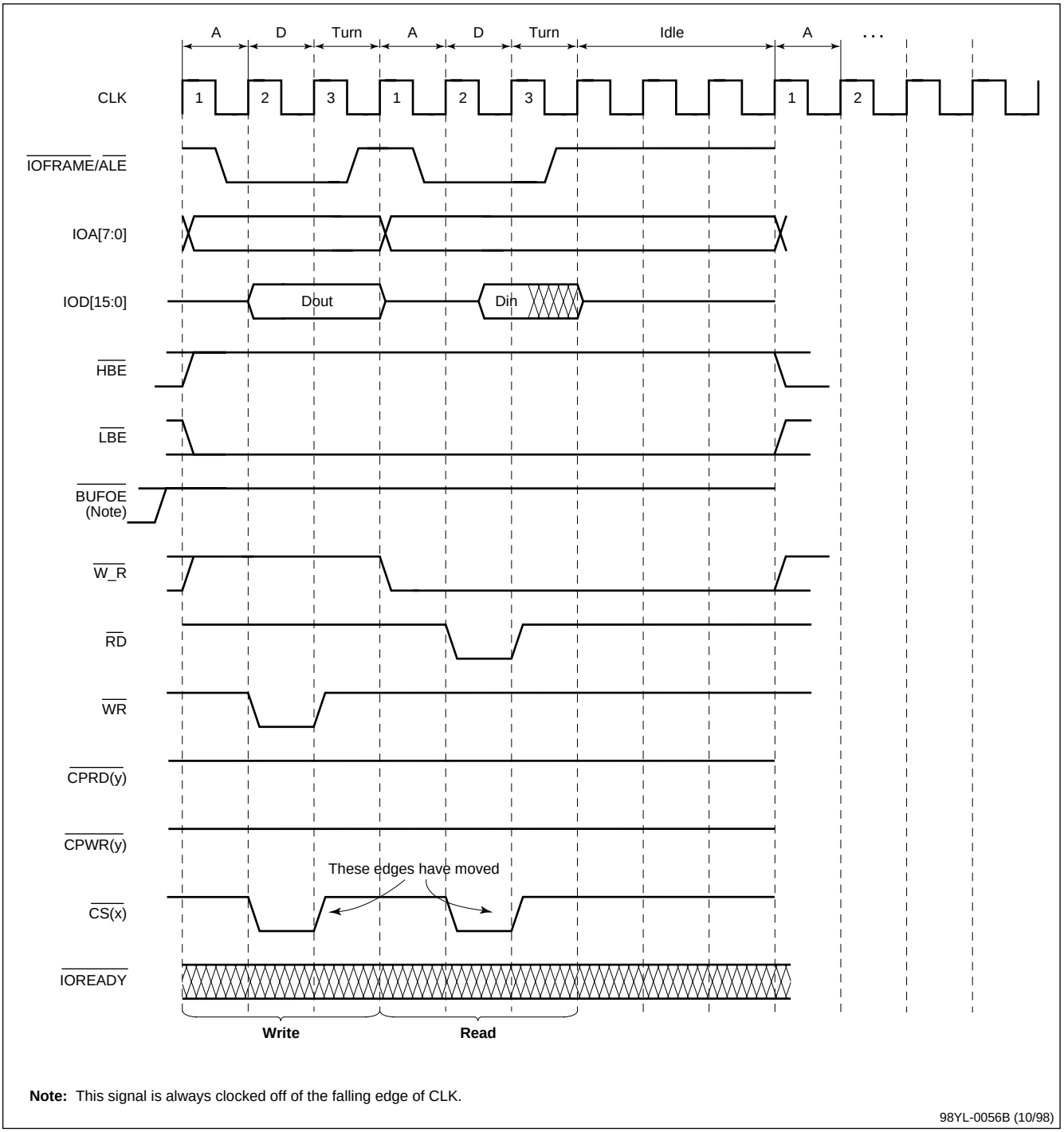
11	10	9	8	7	6	5:3	2:1	0
BUF_on = 1	A[23:8] = 1	$\overline{A[16:18]}$ = 1	RDY_on = 0	CS_high = X	CON_hold = 0	CON_wid = 0	CON_set = 0	CS_low = 0

Figure 11. Back-to-Back, 8-Bit $\overline{CS}$ Access with Delayed Assertion (Non-RDY)



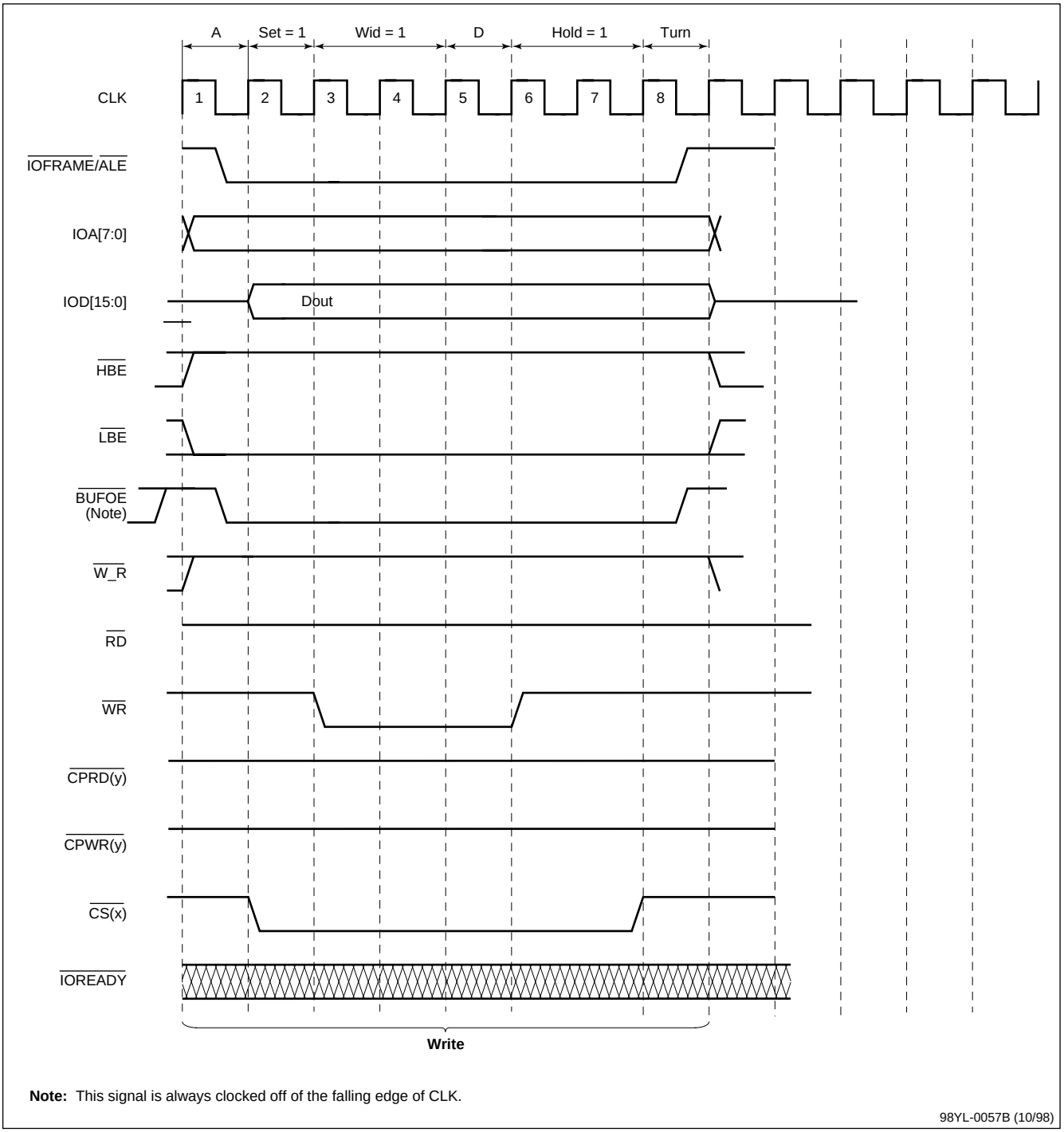
11	10	9	8	7	6	5:3	2:1	0
BUF_on = 1	A[23:8] = 0	$\overline{A[16:8]}$ = 0	RDY_on = 0	CS_high = 0	CON_hold = 0	CON_wid = 0	CON_set = 0	CS_low = 1

Figure 12. Back-to-Back, 8-Bit $\overline{CS}$ Access, with Delayed Assert and Early Deassertion (Non-RDY)



11	10	9	8	7	6	5:3	2:1	0
BUF_on = 0	A[23:8] = 0	$\overline{A[16:8]}$ = 0	RDY_on = 0	CS_high = 1	CON_hold = 0	CON_wid = 0	CON_set = 0	CS_low = 1

Figure 13. Byte Write $\overline{CS}$ Access with Non-Zero Setup, Width, and Hold Values (Non-RDY)

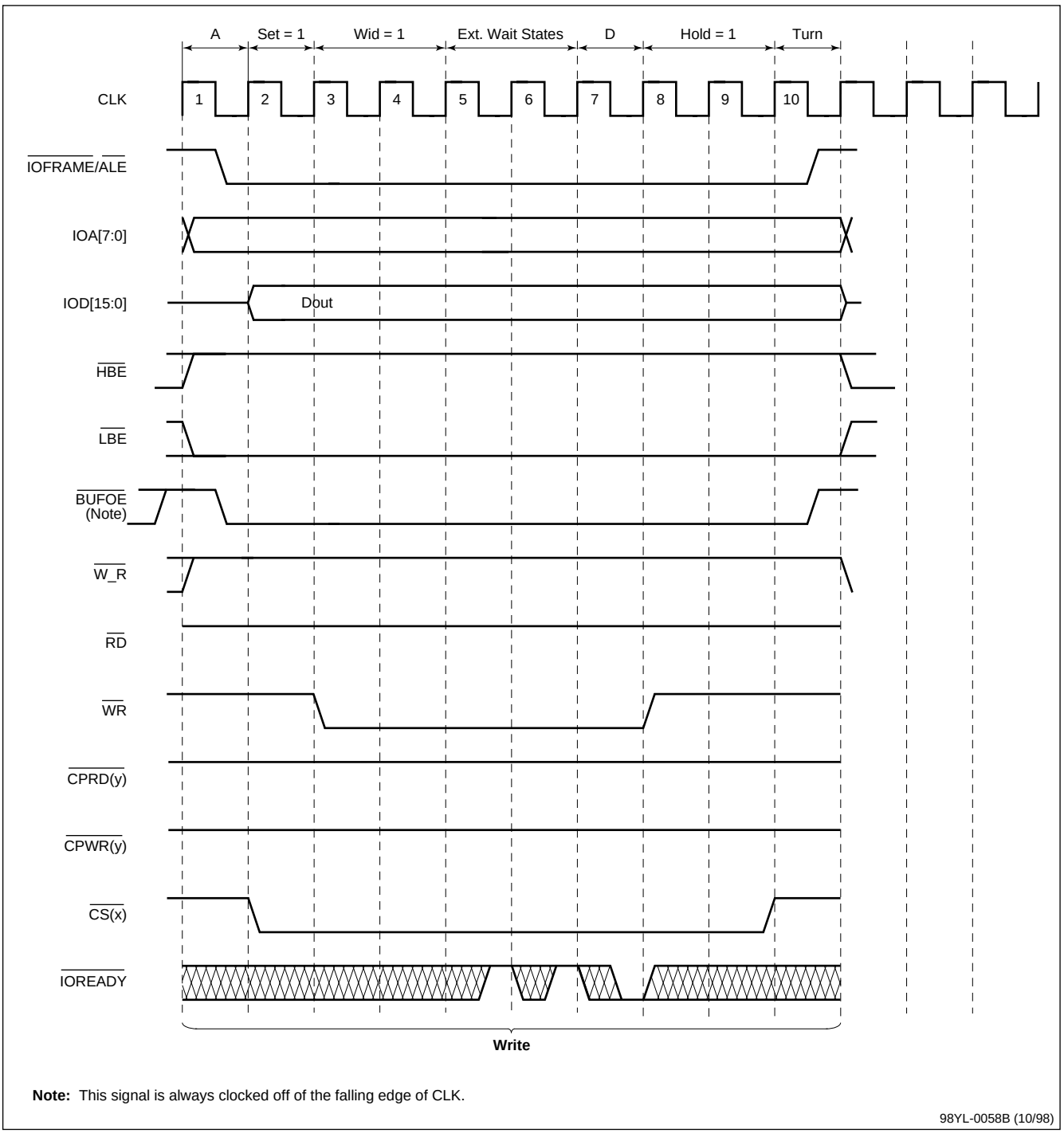


Note: This signal is always clocked off of the falling edge of CLK.

98YL-0057B (10/98)

11	10	9	8	7	6	5:3	2:1	0
BUF_on = 1	A[23:8] = 0	$\overline{A[16:8]}$ = 0	RDY_on = 0	CS_high = 1	CON_hold = 1	CON_wid = 1	CON_set = 1	CS_low = 1

Figure 14. Byte Write $\overline{CS}$ Access with Non-Zero Setup, Width, and Hold Values, RDY On

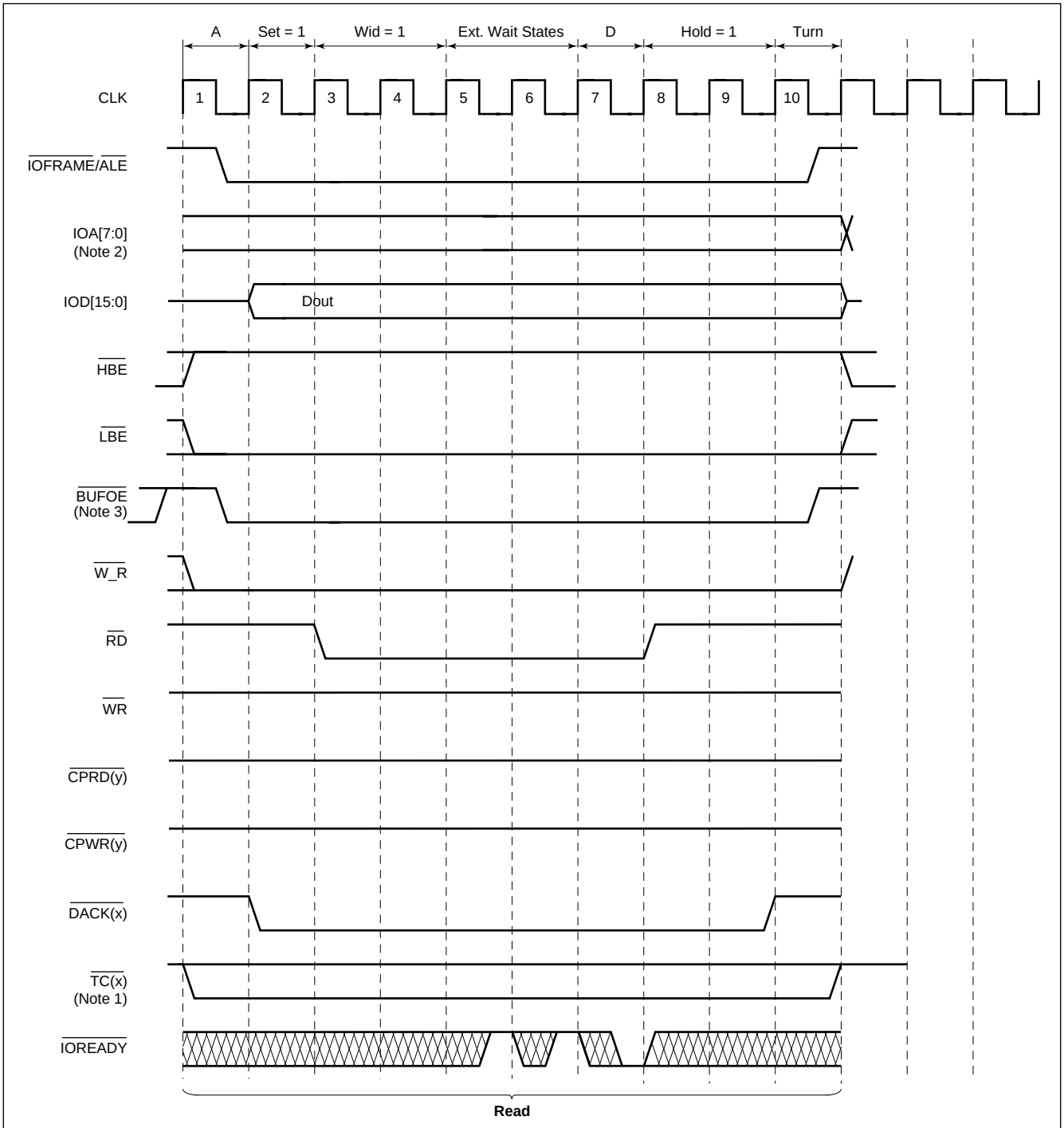


Note: This signal is always clocked off of the falling edge of CLK.

98YL-0058B (10/98)

11	10	9	8	7	6	5:3	2:1	0
BUF_on = 1	A[23:8] = 0	$\overline{A[16:8]}$ = 0	RDY_on = 1	CS_high = 1	CON_hold = 1	CON_wid = 1	CON_set = 1	CS_low = 1

Figure 15. Byte Read DMA Access with Non-Zero Setup, Width, and Hold Values, RDY On **(Note 1)**



Notes:

1. This access type is identical to a regular chip-select access, with the possible addition of $\overline{TC}$.
2. The address is held from the last IO bus cycle. No address is put out during a DMA access.
3. This signal is always clocked off of the falling edge of CLK.

98YL-0059B (10/98)

11	10	9	8	7	6	5:3	2:1	0
BUF_on = 1	$\overline{A[23:8]}$ = 0	$\overline{A[16:8]}$ = 0	RDY_on = 1	CS_high = 1	CON_hold = 1	CON_wid = 1	CON_set = 1	CS_low = 1

6.6

DMA Controller

6.6.1 Overview

The DMA controller within the VRC4372 provides four DMA channels for I/O device- or software-initiated transfers between the I/O controller and the PCI bus. Communication takes place via unique $\overline{TC}$ or EOP, DREQ, and $\overline{DACK}$ signals on each channel. The DMA controller communicates with the PCI interface block and the I/O bus controller block to perform the necessary transfers. The DMA channels can be programmed to provide the following features:

- Block or single transfers per request
- Read or write requests
- I/O device-demanded service requests via DREQ[x], or program-initiated software requests
- Channel suspend requests via the $\overline{MASK}$ register bit
- I/O device transfer termination via the EOP
- Channel reload notification and termination notification via an interrupt
- Efficient PCI bus data packing mode capability
- Byte or short scattering and gathering capabilities (one per PCI word)

Registers and status bits are associated with the control and operation of each channel.

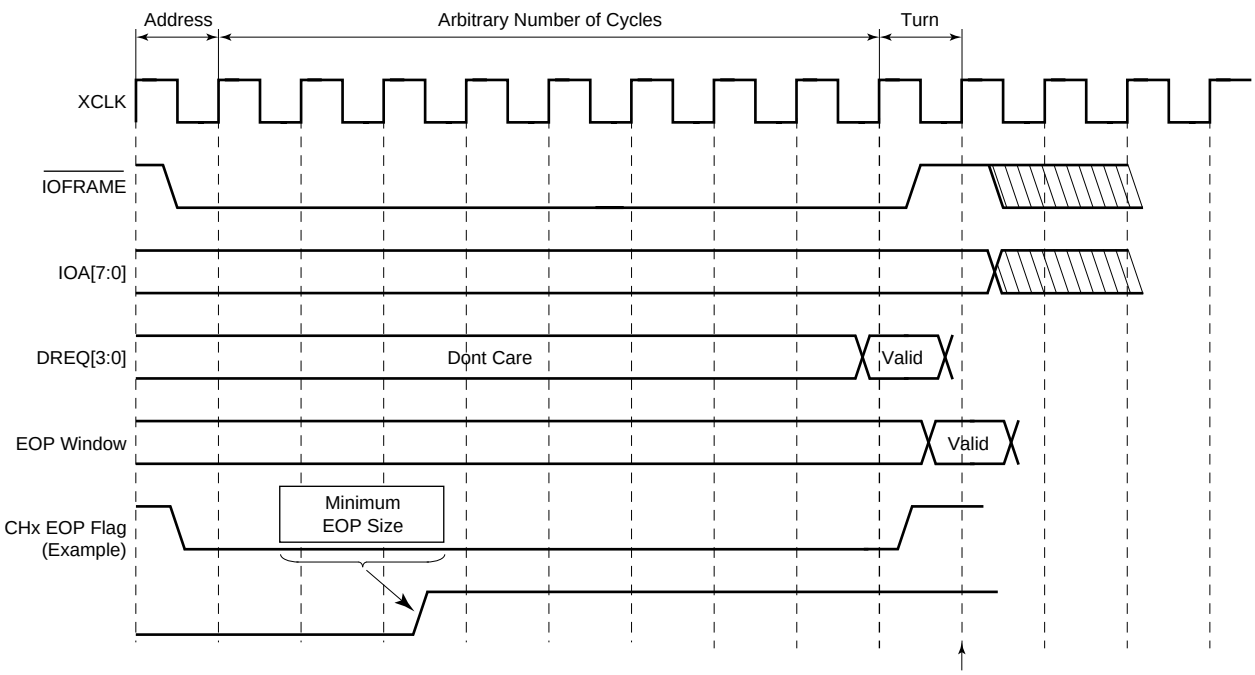
- A current address and current count register pair are used to address the current DMA buffer in PCI memory space.
- A reload start and reload count register pair provides a reload mechanism, through which channel chaining can be accomplished.
- The fifth and sixth registers provide the channel with mode and status control.
- A global control register specifies global DMA attributes such as arbitration scheme selection.

6.6.2 Signal and Pin Descriptions

The I/O bus DMA controller consists of four output, four input, and four bidirectional pins. All outputs are clocked directly from the rising edge of the PCI CLK signal and do not float between valid accesses. All output and bidirectional pins have programmable inversion capabilities controllable through the DMA_OPIN register. The following is a list of I/O bus DMA controller pin descriptions.

Symbol	Description
$\overline{\text{DACK}}$[3:0]	These four outputs function as clocked DMA acknowledgments for four DMA channels. $\overline{\text{DACK}}$[x] is assigned to DMA channel "x". Timing control is individually programmable via the I/O Profile register (IO_PROF_1x) for the channel and has timing similar to the chip select pin. At power-up, the default polarity of these outputs is defined as 1 = deasserted and 0 = asserted.
$\overline{\text{EOP\_TC}}$[3:0]	These four bidirectional pins perform two functions for the four DMA channels: (1) terminal count output and (2) end of packet input. $\overline{\text{EOP\_TC}}$[x] is assigned to DMA channel "x." At power-up, the pins default to the EOP input function. Each channel can be individually programmed to use its $\overline{\text{EOP\_TC}}$ pin as a $\overline{\text{TC}}$ output or as an EOP input by setting or clearing the appropriate bit in the DMA_POL register. When programmed as an EOP input, the pin can be used by the DMA channel to terminate the current DMA buffer prematurely. When programmed as a $\overline{\text{TC}}$ output, the EOP input functionality is not available to the channel and should be disabled via the EOP_EN bit in the DMAMODE register of the channel. The EOP (active high) input functionality can be enabled via the EOP_EN bit in the DMAMODE register for each channel. This EOP input pin, when enabled for a channel, allows the channel to prematurely terminate either a DMA channel operation or a DMA channel buffer. A DMA channel (EMODE set) operation can be terminated regardless of the state of the RLD bit for the channel. The EOP pin indicates that the current DMA transfer is the last transfer for the channel. The channel must be reconfigured to restart data transfers. A DMA channel buffer (EMODE clear) can be terminated, but still reload the channel with the next buffer if the RLD bit for the channel is set. If asynchronous operation for this pin is required, then the pin must be asserted for a duration of at least two clock periods to allow internal synchronization. More specifically, if this signal is sampled high during any clock of the valid period, the packet is terminated after completion of the current transfer. Figure 16 shows an example of valid EOP assertion. When $\overline{\text{EOP\_TC}}$[3:0] has been programmed as a $\overline{\text{TC}}$ output, it is asserted instead of (and at the normal time of) IOA[7:0] during valid I/O bus DMA channel accesses. It signals the last DMA data transfer of every buffer in a channel's chain. At all other times, this signal remains deasserted. At power-up, the default polarity when used as a $\overline{\text{TC}}$ output is defined as: 1 = deasserted and 0 = asserted.
DREQ[3:0]	Each channel has one DREQ input pin that is used to request a DMA transfer. Each DREQ may be selected as active high or active low by the D_POL bit in the channel's DMAMODE register. These inputs are asynchronous; however, for guaranteed deterministic recognition during contiguous data transfers, they should meet setup and hold time requirements around the rising edge of CLK where the I/O bus Turn state is clocked. Refer to the DMA controller descriptions for more details. Figure 16 depicts the DMA signal relationships.
$\overline{\text{DMA\_INT}}$[3:0]	These four internal signals correspond to each of the DMA channels and provide an interrupt source to the interrupt controller block within the VRC4372 controller. $\overline{\text{DMA\_INT}}$[3:0] are active-low asserted and are cleared at the DMA channel source. This signal can be inverted via programming. Refer to "DMA Output Polarity Control Register: DMAPOL" on page 53 for details.

Figure 16. Example DMA Signal Relationships



Notes:

1. The EOP window refers to the period of time that the VRC4372 is sampling the EOP input for channel termination (during a DMA transfer involving the same channel).

2. The minimum EOP size refers to the minimum high time on EOP during the EOP window for guaranteed termination recognition.

98YL-0046B (10/98)

6.6.3 Data Organization and Packing Modes

Each DMA channel can be configured to move data between the I/O bus and the PCI bus.

- The $\overline{D[16:8]}$ bit in the I/O bus DACK profile register sets the I/O bus device data width to either 8 or 16 bits.
- The PACK bit in a channel's DMAMODE control register allows the efficient transfer of data across the PCI bus by packing consecutive PCI addressed bytes/shorts into (a) single or multiple PCI word(s) before a transfer takes place using the assigned DMA channel buffer.
- The STPSZ bit in a channel's DMAMODE control register sets the increment value for the PCI byte address to allow scatter and gather operations to and from the PCI address space.
- When used in conjunction with the PACK bit, STPSZ specifies the prepacked address step size (not the packed word address step size).

Table 24 describes the legal packing, step size, and I/O bus width combinations.

Table 24. Data Packing and Sizes

$\overline{D[16:8]}$ (Note 1)	Pack	Step Size	Description
0	0	0	One-byte-wide I/O device; one-byte PCI transfers; no current address increments (Note 2)
0	0	1	One-byte-wide I/O device; one-byte PCI transfers; one-byte current address increments (Note 2)
0	0	2	One-byte-wide I/O device; one-byte PCI transfers; two-byte current address increments (Note 2)
0	0	4	One-byte-wide I/O device; one-byte PCI transfers; four-byte current address increments (Note 2)
0	1	0	Unsupported
0	1	1	One-byte-wide I/O device; one-byte or greater PCI transfers; one-byte current address increments (Note 3)
0	1	2	Unsupported
0	1	4	Unsupported
1	0	0	Two-byte-wide I/O device; two-byte PCI transfers; no current address increments (Note 2)
1	0	1	Unsupported
1	0	2	Two-byte-wide I/O device; two-byte PCI transfers; two-byte current address increments (Note 2)
1	0	4	Two-byte-wide I/O device; two-byte PCI transfers; four-byte current address increments (Note 2)
1	1	0	Unsupported
1	1	1	Unsupported
1	1	2	Two-byte-wide I/O device; two-byte or greater PCI transfers; two-byte current address increments (Note 4)
1	1	4	Unsupported

Notes:
1. This field is in the I/O bus DACK Profile register (IOPROF_1x) for each DMA channel.
2. PCI addresses increment in the same way as the current address increments.
3. A packing register takes multiple I/O bus accesses and packs them into one or more PCI word access(es). This mode is only available if the start address is word aligned. PCI bus byte addresses are incremented by four.
4. A packing register takes two or more double-byte I/O bus accesses and packs them into one or more PCI word access(es). This mode is only available if the start address is word aligned. PCI bus byte addresses are incremented by four.

6.6.4 Block and Single Transfer Modes

Each channel can be configured via the MODE bit to perform either one transfer per request (Single Transfer mode) or a block transfer per request (Block Transfer mode). In either case, requests can be sourced from the DREQ pin for the channel, from the beeper block, or from the SW_REQ bit for the channel. In Block Transfer mode, data transfers proceed until one of the termination conditions occurs (as described in Section 6.6.8). When a channel is configured for Block Transfer mode, it must rearbitrate for the next transfer at the end of every I/O bus access. In essence, a channel in Block Transfer mode may be unable to transfer its data consecutively.

$\overline{\text{MASK}}$ Bit

Data transfers can be suspended by clearing the $\overline{\text{MASK}}$ bit. Once the channel is unmasked, operation resumes immediately, without a new request.

6.6.5 DMA Channel Request Sources and Masking (or Suspending)

Three request sources are available for each DMA channel.

1. *Software-initiated requests, which are initiated by setting the SW_REQ bit in the channel DMAMODE register.* Once the request is completed, the bit is reset. This request mode is not masked by the $\overline{\text{MASK}}$ bit in the DMAMODE register.
2. *I/O device-initiated requests via the DREQ pin assigned to the channel.* The $\overline{\text{MASK}}$ bit in the DMAMODE register must be set to allow requests to pass through to the channel controller. The SRC bit must be clear to allow DREQ requests (rather than IOBEEP requests) to be the source of the DMA request. Masking or suspending a channel can be accomplished by clearing the $\overline{\text{MASK}}$ bit to 0.

   Note: Clearing the $\overline{\text{MASK}}$ bit also suspends a block transfer; however, once the $\overline{\text{MASK}}$ bit is reset, the block transfer resumes. A new transfer request is unnecessary once the channel is unmasked.
3. *Beeper-initiated requests via an internal signal driven from the beeper output, which generates one request for every rising edge of the IOBEEP signal.* This mode of request facilitates the use of an external DAC for medium-quality audio. The $\overline{\text{MASK}}$ bit in the DMAMODE register must be set to allow requests to pass through to the channel controller. The SRC bit must be set to allow IOBEEP requests (rather than DREQ pin requests) to be the source of the DMA request. Masking or suspending a channel can be accomplished by clearing the $\overline{\text{MASK}}$ bit to zero.

   Note: Clearing the $\overline{\text{MASK}}$ bit also suspends a block transfer; however, once the $\overline{\text{MASK}}$ bit is reset, the block transfer resumes. A new transfer request is unnecessary once the channel is unmasked.

6.6.6 DMA Channel Initialization

The following procedure explains how to initialize a DMA channel.

1. Ensure that the channel is disabled (not active) by verifying that the IDLE bit is set for that channel.
2. Ensure that interrupts are disabled for the channel by verifying that the I_EN bit is set to 0 in the DMAMODE register for the channel.
3. Write the reload register pair with the start address and byte count of the DMA buffer.
4. Configure the channel by setting bits in the channel's DMAMODE register as required (refer to "Mode Control Registers: DMAMODEx" on page 52).
5. If chaining is desired for the channel, mask the channel requests by ensuring that the channel's $\overline{\text{MASK}}$ bit is clear until after the chain is linked (step 8). Program-initiated requests should also be avoided until the channel chain is linked.
6. Enable the channel by setting the CH_EN bit in the channel's DMAMODE register. This step transfers the reload register pair into the current register pair.
7. If chaining is desired for the channel, write the reload register pair with the starting address and byte count of the next DMA buffer for the channel. This step can be skipped if chaining is not required.
8. If chaining is desired for the channel, set the RLD bit in the DMAMODE register to enable the automatic reload mechanism once the current register pair is exhausted. This step can be skipped if chaining is not required.
9. Unmask the channel by setting the $\overline{\text{MASK}}$ bit to 1 in the DMAMODE register for the channel. Program-initiated requests can also commence at this time. Enable channel interrupts, if required, by setting the channel's I_EN bit in the DMAMODE register.

6.6.7 DMA Channel Reloading (or Chaining)

To avoid hazards when chaining, observe the following:

1. Follow the initialization guidelines described in Section 6.6.6.
2. Monitor the reload operation either by enabling the interrupt for the channel during the initialization phase, which triggers an interrupt after the reload occurs, or by polling for a zero on the RLDx bit in the channel's DMASTAT register.
3. Mask the channel by clearing the $\overline{\text{MASK}}$ bit to 0 in the channel's DMAMODE register.
4. Verify that the channel has not completed by reading the IDLE bit from the channel's DMASTAT register. If the bit is set, then the channel is done and a new chain must be started. If the bit is clear, then the channel has not completed and another link can be added.

   Note: If the IDLE bit is clear, then the channel Finite State Machine (FSM) must check the RLD bit in the DMAMODE register. Correct operation is guaranteed when the RLD bit is checked and found clear and the IDLE bit is set on the next clock.
5. Write the reload starting and counting register pair with the next DMA buffer.
6. Set the RLD bit in the DMAMODE register.
7. Unmask the channel by setting the $\overline{\text{MASK}}$ bit in the channel's DMAMODE register.
8. Repeat steps 2 through 7 for additional links.

6.6.8 DMA Channel Termination

DMA channel termination occurs when one of the following conditions becomes true:

- The channel runs out of data (in other words, all channel chains have finished). This condition occurs when the RLD bit in the DMAMODE register is not set before the current buffer is emptied.
- The EOP pin is sampled when it is asserted during a DMA transfer initiated by the channel, while EOP-controlled termination is enabled for the channel and the EMODE bit is set. If the EMODE bit is clear and EOP-controlled termination is enabled, then EOP assertion terminates the current buffer and performs a reload if one is pending.
- The channel is abnormally terminated by software deassertion of the CH_EN bit in the channel's DMAMODE register, or by error detection during a DMA-channel-initiated transfer on either the I/O bus or the PCI bus. DMA channel packing buffers are cleared.

  Note: This termination mode does not guarantee data delivery.

DMA channel termination is signaled by the assertion of the IDLE bit in the DMASTAT register corresponding to the channel.

6.6.9 DMA Interrupt Handling

Each DMA channel can be enabled to issue an interrupt using a dedicated interrupt signal (DMA_INT[3:0]), which can be enabled or masked by the I_EN bit in the DMAMODE register for each channel. The INTSTAT register, located in the interrupt controller, includes four interrupt sources (DMA_INT[3:0]), one for each DMA channel.

If enabled, an interrupt is generated for a channel any time the current channel buffer empties or is terminated by any termination condition. To facilitate the reload operation, the interrupt, triggered by the emptying of a channel buffer, occurs after the reload occurs; this allows the software to change the value of the reload registers.

To clear the interrupt bit for a channel, the software toggles the I_EN bit from 1 to 0 to 1 in the DMAMODE register of the channel being serviced.

6.6.10 DMA Buffer Length Description

The starting address points to the first data in the DMA buffer. The byte count contains the number of bytes to be transferred. Odd byte count transfers to a 16-bit device are not supported. All 16-bit transfers must start and end on an even-byte address boundary.

6.6.11 Packing Buffer Operation

Each DMA channel has a packing buffer that can be used to pack byte or double-byte I/O bus transactions into word or longer (burst) PCI transfers. To enable packing, the PACK bit must be set in the channel's DMAMODE register and the current start address must be word-aligned with no restriction on the byte count length.

Buffers for channels 2 and 3 are completely filled after they have been completely emptied, or after the current byte count reaches zero at each link in a chained DMA transaction and the current DMA buffer has completed.

Buffers for channels 0 and 1 operate almost identically to those for channels 2 and 3, except they emptied after they reach half capacity. While a packing buffer for channel 2 or 3 is receiving service from the PCI bus, the DMA channel it serves is ignored. For buffers 0 and 1, the DMA channel being serviced continues to receive service (even during PCI bus transactions) until the buffer is completely full or empty. For additional information on these buffers, refer to Section .

6.6.12 DMA Register Summary and Memory Map

Table 25 defines the DMA register set and memory map in the VRC4372 DMA controller.

Table 25. DMA Registers

Symbol	Offset Value	Reset Value	R/W	Description
DMACADD0	0x0503–0x0500	0x0000 0000	R	Current address for channel 0
DMACCNT0	0x0507–0x0504	0x0000 0000	R	Current byte count for channel 0
DMARSTRT0	0x050b–0x0508	0x0000 0000	R/W	Reload start address for channel 0
DMARCNT0	0x050f–0x050c	0x0000 0000	R/W	Reload byte count for channel 0
DMACADD1	0x0513–0x0510	0x0000 0000	R	Current address for channel 1
DMACCNT1	0x0517–0x0514	0x0000 0000	R	Current byte count for channel 1
DMARSTRT1	0x051b–0x0518	0x0000 0000	R/W	Reload start address for channel 1
DMARCNT1	0x051f–0x051c	0x0000 0000	R/W	Reload byte count for channel 1
DMACADD2	0x0523–0x0520	0x0000 0000	R	Current address for channel 2
DMACCNT2	0x0527–0x0524	0x0000 0000	R	Current byte count for channel 2
DMARSTRT2	0x052b–0x0528	0x0000 0000	R/W	Reload start address for channel 2
DMARCNT2	0x052f–0x052c	0x0000 0000	R/W	Reload byte count for channel 2
DMACADD3	0x0533–0x0530	0x0000 0000	R	Current address for channel 3
DMACCNT3	0x0537–0x0534	0x0000 0000	R	Current byte count for channel 3
DMARSTRT3	0x053b–0x0538	0x0000 0000	R/W	Reload start address for channel 3
DMARCNT3	0x053f–0x053c	0x0000 0000	R/W	Reload byte count for channel 3
Reserved	0x05ff –0x0540	—	R	Read as 0
DMAMODE0	0x0603– 0x0600	0x0000 0000	R/W	Mode control register for channel 0
DMAMODE1	0x0607–0x0604	0x0000 0000	R/W	Mode control register for channel 1
DMAMODE2	0x060b–0x0608	0x0000 0000	R/W	Mode control register for channel 2
DMAMODE3	0x060f–0x060c	0x0000 0000	R/W	Mode control register for channel 3
Reserved	0x063f–0x0610	—	R	Read as 0
DMASTAT0	0x0643–0x0640	—	R	Status register channel 0
DMASTAT1	0x0647–0x0644	—	R	Status register channel 1
DMASTAT2	0x064b–0x0648	—	R	Status register channel 2
DMASTAT3	0x064f–0x064c	—	R	Status register channel 3
Reserved	0x067f–0x0650	—	R	Read as 0
DMAGLOB	0x0683–0x0680	0x0000 0000	R/W	Global DMA control register
DMAPOL	0x0687–0x0684	0x0000 0008	R/W	DMA output pin polarity control
Undefined	0x06ff–0x0688	—	R	Read as 0

Current Address Registers: DMACADDx

Bit(s)	Symbol	Description
31:0	Add[31:0]	The full byte address of the current PCI address

Current Byte Count Registers: DMACCNTx

Bit(s)	Symbol	Description
23:0	CNT[23:0]	The byte count of the remaining bytes to be transferred from or to the I/O bus. While packing mode is enabled, this count does not reflect the data bytes remaining from the PCI bus.
31:24	Unused	Read back as 0

Reload Address Registers: DMARADDx

Bit(s)	Symbol	Description
31:0	Add[31:0]	The full byte address of the reload PCI address

Reload Byte Count Registers: DMARCNTx

Bit(s)	Symbol	Description
23:0	CNT[23:0]	The byte count of the reload buffer size
31:24	Unused	Read back as 0

Mode Control Registers: DMAMODEx

Bit(s)	Symbol	Description
0	RLD	When this bit is set, the channel automatically reloads the current registers from the reload registers and continues the channel operation with the next buffer. After the channel completes the register transfer, this bit is cleared.
1	$\overline{\text{R\_W}}$	When this bit is set, the channel performs read cycles from the I/O bus. When cleared, the channel performs write cycles to the I/O bus.
2	CH_EN	When this bit is set, channel operation is enabled; clearing this bit terminates channel operation. When a channel finishes, either normally or abnormally, this bit is automatically cleared.
3	Reserved	This reserved bit can be both read and written; resets to 0.
4	EOP_EN	When this bit is set, EOP pin termination is enabled; when this bit is cleared, EOP pin termination is masked. See the EMODE bit description (bit 12) for EOP termination modes.
5	D_POL	When this bit is set, DREQ pin polarity is active low; when this bit is cleared, DREQ pin polarity is active high.
6	I_EN	When this bit is set, an interrupt is generated after the final I/O bus transaction of the current buffer channel operation and the reload operation occurs before the interrupt is issued.
7	PACK	When this bit is set, a packing mechanism is enabled that takes multiple I/O bus transactions and packs them into one PCI word access. When this bit is cleared, each I/O bus transaction generates a PCI bus transaction of the same width.
8	SW_REQ	When this bit is set, one S/W DREQ request is generated. This bit is not masked by the $\overline{\text{MASK}}$ bit, and is reset at the completion of the request.
9	MODE	When this bit is set, Block Transfer mode is selected; when this bit is cleared, Single Transfer mode is selected. When Block Transfer mode is selected, the channel operates until a termination condition occurs or until masked by the $\overline{\text{MASK}}$ bit.
11:10	STPSZ	These bits define the current address increment step size as follows: 0: No increment 1: Increment by one-byte address 2: Increment by two-byte address 3: Increment by four-byte address
12	EMODE	When this bit is set and EOP is enabled (the EOP_EN bit is set), EOP assertion causes the channel to terminate whether or not a reload is pending. When this bit is cleared and EOP is enabled, EOP assertion causes only the current buffer of a DMA chain to terminate and a channel reload occurs if enabled.
13	SRC	When this bit is set, the DMA request source is the beeper block within the controller and one DMA request is generated at the rising edge of IOBEEP. When this bit is cleared, the DMA request source is the DREQ input pin for the channel.
30:14	Reserved	Read back as zero
31	$\overline{\text{MASK}}$	When this bit is set, DREQ pin requests are enabled; when this bit is cleared, DREQ pin requests are masked. This bit can be used to suspend channel operation during reload operations to allow linking of a DMA chain for both Single and Block mode transfers.

DMA Status Registers: DMASTATx

Each DMA channel features a status register. Bits 2 through 7 are cleared from the previous DMA chain when CH_EN is set during DMA channel initialization for the next chain.

Bit(s)	Symbol	Description
0	IDLE	When set, this bit indicates that channel x has completed or is idle. The IDLE bit follows the state of the CH_EN bit for channel x.
1	RLD	When set, this bit indicates that channel x has a reload pending.
2	PERR	When set, this bit indicates a parity error (PERR pin asserted) on a PCI bus cycle originated by channel x. The parity error does not cause the channel to terminate abnormally.
3	SERR	When set, this bit indicates a system error (SERR pin asserted) on a PCI bus cycle originated by channel x. The system error causes the channel to terminate abnormally.
4	REC_TA	When set, this bit indicates a target abort on a PCI bus cycle originated by channel x. The target abort error causes the channel to terminate abnormally.
5	SIG_MA	When set, this bit indicates a master abort on a PCI bus cycle originated by channel x. This error is generated when no device responds to a PCI address generated by the DMA channel. The master abort error causes the channel to terminate abnormally.
6	RET_ERR	When set, this bit indicates the retry counter limit is reached on a PCI bus cycle originated by channel x. The retry counter limit error causes the channel to terminate abnormally.
7	ToutERR	When set, this bit indicates a timeout error on the I/O bus during a transaction originated by channel x. The timeout error causes the channel to terminate abnormally.
31:8	Reserved	Read back as zero

DMA Global Control Register: DMAGLOB

Bit(s)	Symbol	Description
0	ARB	When this bit is set, the round-robin arbitration scheme is used. When this bit is cleared, the fixed-priority arbitration scheme is used. With fixed-priority arbitration, channel 0 has the highest priority, followed by channel 1, channel 2, and channel 3.
31:1	Reserved	These bits are reserved for future channels and read back as 0.

DMA Output Polarity Control Register: DMAPOL

This register controls the polarity of the DMA pins that can be inverted. Each bit in this register is used to invert one output. Both edges of the inverted signal must be glitch-free. Table 26 details the bit assignments and default values.

Table 26. DMAPOL Register Bit Assignments

Bit(s)	R/W	Signal Name	Reset Value	Description
0	R/W	$\overline{\text{DACK0}}$	0	Active low
1	R/W	$\overline{\text{DACK1}}$	0	Active low
2	R/W	$\overline{\text{DACK2}}$	0	Active low
3	R/W	$\overline{\text{DACK3}}$	1	Reset to an active-high signal. During reset, this pin is low. Supports the dual functionality of the PC87334's $\overline{\text{PDACK}}$ pin.
4	R/W	$\overline{\text{TC0}}$	0	Multiplexed onto $\overline{\text{EOP\_TC0}}$
5	R/W	$\overline{\text{TC1}}$	0	Multiplexed onto $\overline{\text{EOP\_TC1}}$
6	R/W	$\overline{\text{TC2}}$	0	Multiplexed onto $\overline{\text{EOP\_TC2}}$
7	R/W	$\overline{\text{TC3}}$	0	Multiplexed onto $\overline{\text{EOP\_TC3}}$
8	R/W	$\overline{\text{EOP\_TC0}}$	0	Input (default direction)
9	R/W	$\overline{\text{EOP\_TC1}}$	0	Input (default direction)
10	R/W	$\overline{\text{EOP\_TC2}}$	0	Input (default direction)
11	R/W	$\overline{\text{EOP\_TC3}}$	0	Input (default direction)
31:12	R	Unused	0	Read back as 0

7.0 Buffers

7.1 Data Buffering

The VRC4372 chip contains data buffers that isolate the PCI bus from the slower I/O bus. Each DMA channel has its own buffers that prevent unnecessary flushing and improve performance. In addition, a four-byte posted write buffer is provided for the PCI slave write transfers to the I/O bus.

7.2 DMA Buffers

DMA buffers are bidirectional, so that both read and write data can be stored. These buffers are also used for dword assembly and disassembly operations, which are required because I/O bus devices can be either one or two bytes wide.

The size of each buffer was chosen on the basis of expected worst-case latencies on the PCI bus and the transfer rate of each channel. Assuming PCI latencies of 24–30 μs, the following buffer sizes are implemented:

DMA Ch0	32 bytes for 1 MB/s transfer rate device
DMA Ch1	16 bytes for 0.5 MB/s transfer rate device
DMA Ch2	4 bytes for 0.1 MB/s transfer rate device
DMA Ch3	4 bytes for 0.1 MB/s transfer rate device

7.3 PCI-Posted Write Buffer

Write transfers by a PCI master device to the VRC4372 chip are handled by the PCI target interface and by the data stored in the four-byte PCI-posted write buffer. The data from this buffer may be destined to an internal I/O register block or to the I/O bus. The write transfers to the I/O bus contain an associated address, the data is disassembled, and write cycles (byte or word) to the I/O bus are performed to complete the transfer.

7.4 Buffer Management

Whenever data is temporarily stored in the buffers of a bridge chip, such as the VRC4372 I/O controller, data coherency problems may arise. To avoid data coherency problems, an algorithm is implemented in the buffer management logic. In order for the DMA buffers to write data to the PCI bus, the buffers must be full, a terminal count must be reached, or a synchronization event must be executed. The synchronization event can be defined as a read operation of the I/O bus or an internal register by a PCI master device.

For channels 0 and 1, the flushing or filling of the DMA buffers is initiated when they are half filled or half empty. For channels 2 and 3, the buffers are serviced once they have completely filled and emptied. During servicing of the channels 2 and 3 DMA buffers, the DMA device is stalled until the data buffers are filled and emptied.

On a read command, a buffer is filled, and the data disassembled and sent to the I/O bus. This read data is invalidated on a subsequent write to any DMA channels, or when any terminal condition occurs for the channel.

If the PACK bit in the DMA profile register is cleared, data buffering for that channel is disabled. On a write cycle to PCI, data may still be written into the buffer, but the PCI write cycle begins immediately and the I/O bus is stalled until after the write cycle is complete. Similarly, on a read cycle from PCI, only one or two bytes of data are read and transferred to the I/O bus; data buffers may be used for staging purposes.

A posted write buffer is scheduled to be written to the I/O bus immediately, and subsequent cycles from PCI are retried by the PCI target until this buffer is empty. And, since the PCI target has higher priority than the DMA, this buffer is emptied before the bus is granted to a DMA channel. DMA data buffers remain valid during this time.

Figure 17. Data Buffers

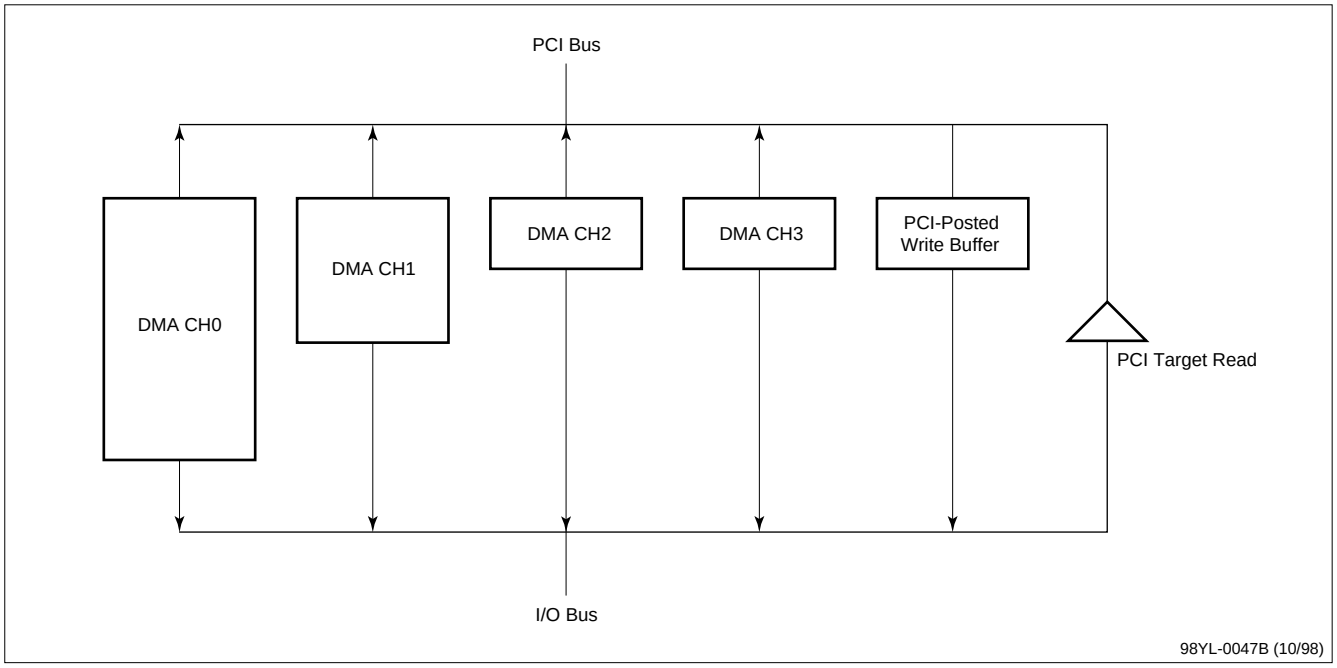
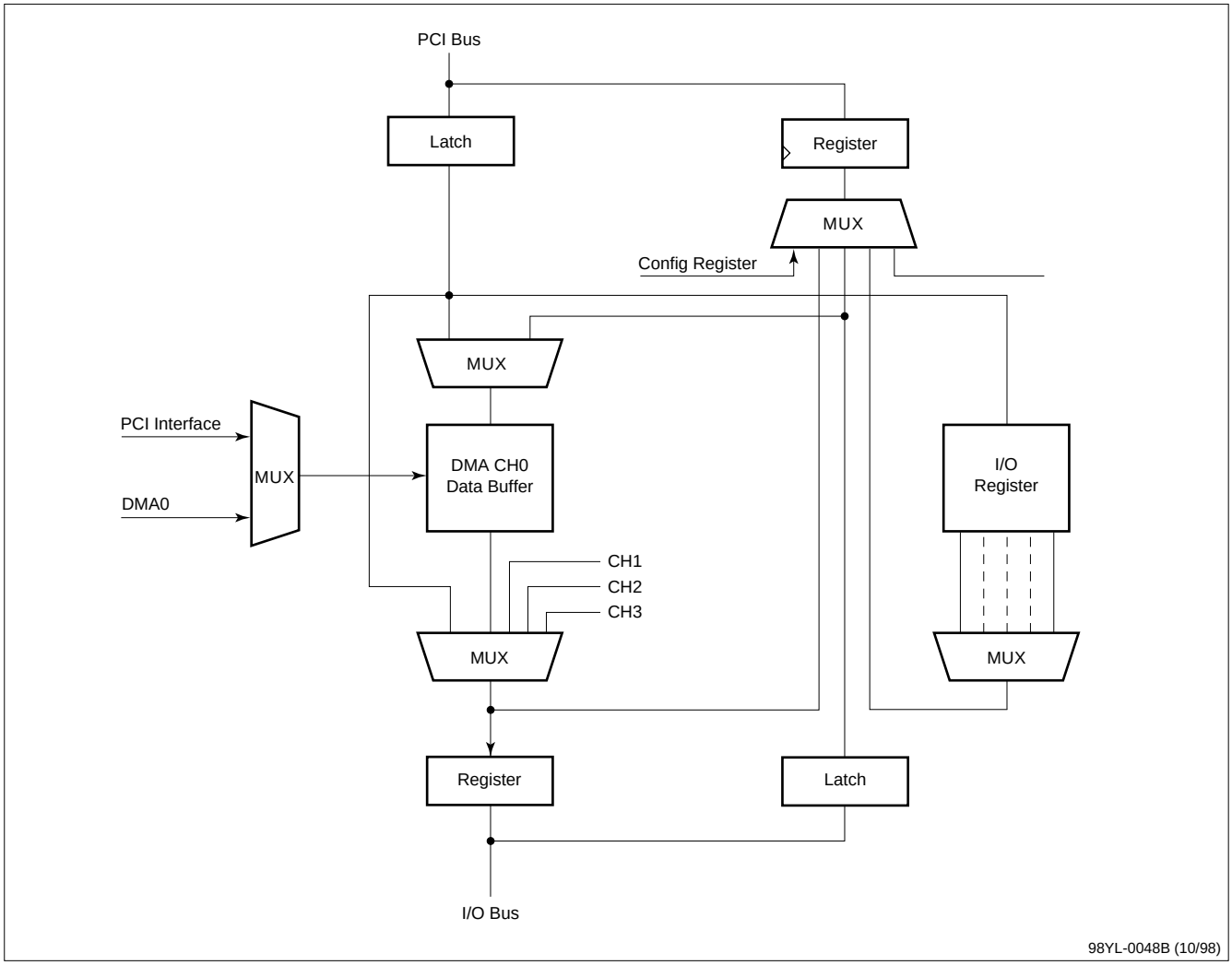


Figure 18. Bus Interface



8.0 Error Handling

Table 27 summarizes the causes and effects of various error modes within the VRC4372 controller.

Table 27. Error Handling

Cause	Primary Effects
I/O bus timeout during target access to the VRC4372	Target abort signaled on PCI bus Target abort signaled error sent to interrupt controller Timeout signaled to interrupt controller
I/O bus timeout during master access (DMA)	DMA channel terminates abnormally with flags set DMA interrupt signaled to interrupt controller Timeout signaled to interrupt controller
DMA signals master abort on PCI (for example, no response)	PCI flags set Master abort interrupt signaled to interrupt controller DMA interrupt signaled to interrupt controller DMA channel terminates abnormally with flags set
DMA receives target abort on PCI	PCI flags set Target abort received; error signaled to interrupt controller DMA interrupt signaled to interrupt controller DMA channel terminates abnormally with flags set
PCI retry counter expires during DMA	Retry error signaled to interrupt controller DMA interrupt signaled to interrupt controller DMA channel terminates abnormally with flags set
The VRC4372 senses $\overline{\text{PERR}}$ on PCI during DMA	$\overline{\text{PERR}}$ error flag set in DMA status register $\overline{\text{PERR}}$ signaled to interrupt controller DMA channel continues as normal PCI status bits are set
The VRC4372 senses $\overline{\text{PERR}}$ during any valid PCI access by any master	$\overline{\text{PERR}}$ signaled to interrupt controller; PCI status bits are set
The VRC4372 senses $\overline{\text{SERR}}$ during any valid PCI access by any master	$\overline{\text{SERR}}$ signaled to interrupt controller; PCI status bits are set
The VRC4372 senses bad parity on address during a valid PCI access	Claim cycle as though address was correct, but target abort Bad address error signaled to interrupt controller $\overline{\text{SERR}}$ asserted if enabled

9.0 Electrical Specifications

9.1 Absolute Maximum Ratings and Operating Conditions

Table 28. Absolute Maximum Ratings

Parameter	Rating
Storage temperature	–55°C to 125°C
Operating ambient temperature	0°C to 70°C
DC supply voltage (Note)	–0.5 V to 4.6 V
DC voltage on input pins (Note)	–0.5 V to 5.5 V
Voltage discharged between any two pins through 1 KΩ at 100 pF	2000 V
Maximum power dissipation	1 W
Note: DC voltage is measured with respect to ground.	

Table 29. Operating Conditions

Parameter	Symbol	Min.	Max.	Unit
All power pins	V_{DD}	3.0	3.6	V
Supply current (V_{DD})	I_{SS}	—	600	mA
Case temperature	T_C	0	80	°C
Junction temperature	T_J	—	110	°C

9.2 DC Characteristics

Table 30. DC Characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit
Input voltage, low	V_{IL}	–5.5	—	0.8	V
Input voltage, high	V_{IH}	2.0	—	7.6	V
Output voltage, low	V_{OL}	—	—	0.4 0.8	V
Output voltage, high	V_{OH}	2.4	—	—	V
Input leakage current with no pull-down resistor	I_{LI}	—	—	10	µA
Input leakage current with 50 kΩ internal pulldown	I_{LI}	—	66	—	µA

Table 31. Capacitance

Parameter	Symbol	Min.	Typ.	Max.	Unit
Input capacitance	C_{IN}	—	10	20	pF
Output capacitance	C_{OUT}	—	10	20	pF
I/O capacitance	$C_{I/O}$	—	10	20	pF

9.3

AC Specifications

9.3.1 Clock Input

The PCI clock signal (CLK) is required to meet TTL input levels. Figure 19 and Table 32 summarize the clock input requirements (Section R2.1 of the PCI specification).

Figure 19. Clock Input Waveform

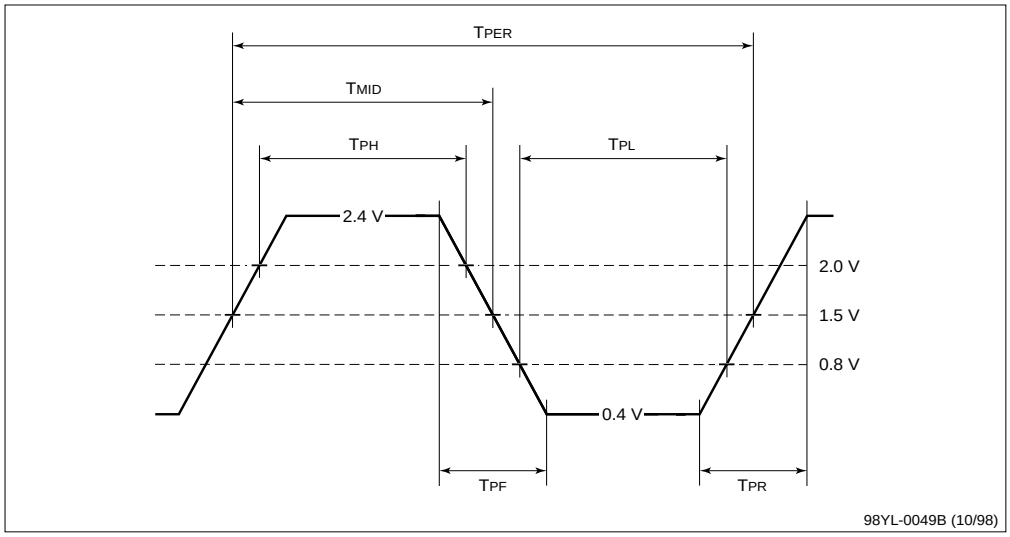


Table 32. Clock Input Timing

Parameter	Symbol	Min.	Max.	Unit
CLK period	TPER	30	—	ns
CLK time, mid	TMID	0.4 TPER	0.6 TPER	ns
CLK time, high	TPH	11	—	ns
CLK time, low	TPL	11	—	ns
CLK rise time	TPR	1	8	ns
CLK fall time	TPF	1	8	ns

9.3.2 Synchronous Inputs, Outputs, and Input/Outputs

The VRC4372 controller is designed to work in both 3.3-volt and 5-volt PCI systems, except for maximum overshoot/undershoot levels. To accommodate both voltages, consideration must be given to switching and signaling levels, as defined in the PCI specification, revision 2.1. Table 33 reproduces Table 4-7 of that document with substitutions made for minimum and maximum VCC levels in the VRC4372 controller.

Timing measurements on the VRC4372 controller for PCI pads are taken with respect to a VTEST voltage of 1.5 volts. Setup and hold input times are measured from the 1.5-volt level on the rising edge of CLK to the 1.5-volt level on the signal. Likewise, minimum/maximum output valid times are measured from the 1.5-volt level on the rising edge of CLK to the 1.5-volt level on the signal.

Table 33. VRC4372 Timing Reference Voltage

Symbol	5-V PCI Signaling	3.3-V PCI Signaling			VRC4372 Universal PCI Reference Values
		PCI R.21	VCC = 3.0	VCC = 3.6	
VTH	2.4 V	0.6 VCC	1.8 V	2.16 V	2.4 V
VTL	0.4 V	0.2 VCC	0.6 V	0.72 V	0.4 V
VTEST	1.5 V	0.4 VCC	1.2 V	1.44 V	1.5 V
VSTEP1	N/A	0.285 VCC	0.86 V	1.03 V	0.285 VCC
VSTEP2	N/A	0.615 VCC	1.85 V	2.15 V	0.615 VCC
VMAX	2.0 V	0.4 VCC	1.2 V	1.4 V	2.0 V

Table 34. PCI Signal Timing

Parameter	Symbol	Min.	Max.	Unit
CLK to signal valid delay; all but $\overline{REQ}$ and $\overline{GNT}$	TVAL	2	11	ns
CLK to signal valid delay; $\overline{REQ}$ and $\overline{GNT}$	TVAL(PTP)	2	12	ns
Float-to-active delay	TON	2	—	ns
Active-to-float delay	TOFF	—	28	ns
Input setup to CLK; all but $\overline{REQ}$ and $\overline{GNT}$	TSU	7	—	ns
Input setup to CLK – $\overline{REQ}$	TSU(PTP1)	12	—	ns
Input setup to CLK – $\overline{GNT}$	TSU(PTP2)	10	—	ns
Input hold time from $\overline{CLK}$	TH	0	—	ns
Reset active to output-float delay (all output drivers)	TRST-OFF	—	40	ns
Reset active time after power stable	TRST	1	—	ms
Reset active time after CLK stable	TRST-CLK	100	—	µs
Test active to test output state (all output drivers)	TTST	—	40	ns

All non-PCI synchronous signals are measured from the 1.5-volt level on the rising edge of CLK to the legal TTL input levels (0.8 volts and 2.0 volts) and to the legal TTL output levels (0.4 volts and 2.4 volts).

All timing is met across the minimum and maximum AC and DC loads for each signal (Tables 34 and 35). The IOBEEP signal is not specified, due to its large load requirements and asynchronous nature.

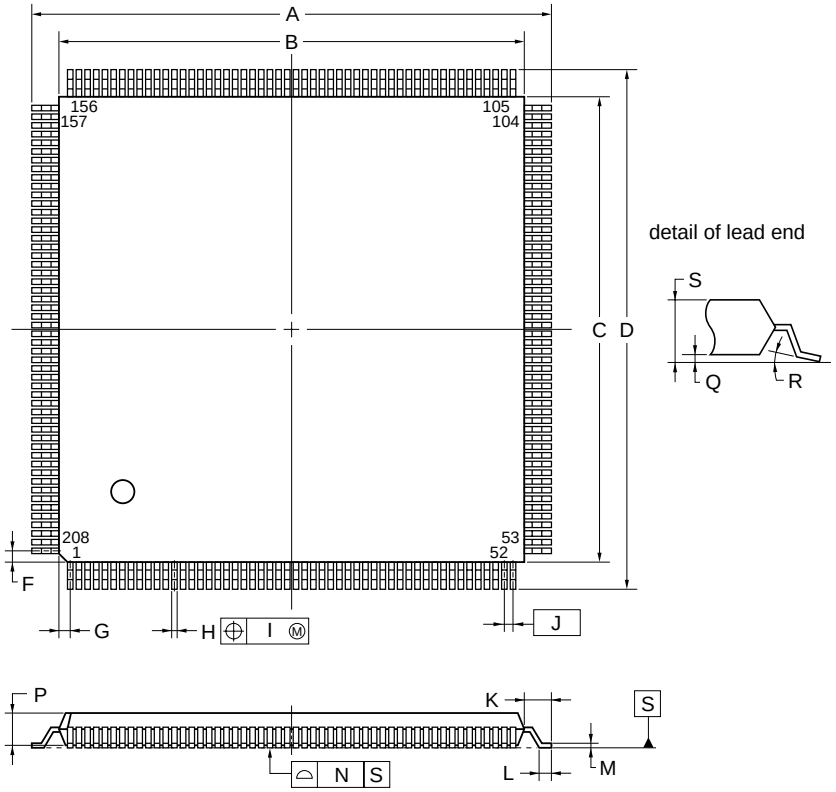
Table 35. Non-PCI Signal Timing

Parameter	Symbol	TTL Min.	TTL Max.	1.5-V Max.	Unit
CLK to output valid delay on: IOD[15:0], $\overline{HBE}$, $\overline{LBE}$, $\overline{RD[1:0]}$, $\overline{WR[1:0]}$, $\overline{CPRD[1:0]}$, $\overline{CPWR[1:0]}$, $\overline{CS[7:0]}$, $\overline{DACK[3:0]}$	T_{OV0}	3	22	19	ns
CLK to output valid delay on: $\overline{IOFRAME}$, $\overline{W\_R}$, $\overline{CPRD[2]}$, $\overline{CPWR[2]}$, $\overline{CS[9:8]}$, $\overline{EOP\_TC}$, GPIO[10:0]	T_{OV1}	3	16	15	ns
CLK to output valid delay for $\overline{BUFOE}$	T_{OV2}	T_{MID} + 3	T_{MID} + 16	T_{MID} + 14.6	ns
CLK to output valid delay for $\overline{ALE}$	T_{OV3}	6	30	—	ns
CLK to output valid delay for IPL[2:0]	T_{OV4}	1.5	7 (Note 1)	—	ns
CLK to output valid delay for IOA[7:0]	T_{OV5}	3	24	19	ns
IOD[15:0] valid to $\overline{ALE}$ asserted	T_{REL0}	2	—	4.3 (Note 2)	ns
CLK to IOD[15:0] float delay	T_{OFF0}	3	22	—	ns
CLK to IOD[15:0] driven delay	T_{ON0}	3	22	—	ns
Reset active to reset output state; all output drivers	$T_{RST-OFF}$	—	40	—	ns
Test active to test output state; all output drivers	T_{TST}	—	40	—	ns
Input setup to CLK	T_{IS0}	7	—	—	—
Input hold from CLK	T_{IH0}	0	—	—	—

Notes:
1. The IPL lines are driven fast to allow CPU interrupts to meet the setup time of 3.5 ns and hold time of 1.5 ns.
2. The IOD lines should not be greater than 7.5 inches longer than $\overline{ALE}$. This allows a round trip delay (15 inches) for IOD setup time for the F373 of 4.3 ns (15 inches x 0.22 ns/ inch = 3.3 ns + 1 ns setup = 4.3 ns).

10.0 Package Description

208-PIN PLASTIC FP (FINE PITCH) (28x28)



NOTE

Each lead centerline is located within 0.10 mm of its true position (T.P.) at maximum material condition.

ITEM	MILLIMETERS
A	30.6±0.2
B	28.0±0.2
C	28.0±0.2
D	30.6±0.2
F	1.25
G	1.25
H	$0.22^{+0.05}_{-0.04}$
I	0.10
J	0.5 (T.P.)
K	1.3±0.2
L	0.5±0.2
M	$0.17^{+0.03}_{-0.07}$
N	0.10
P	3.2±0.1
Q	0.4±0.1
R	5°±5°
S	3.8 MAX.

P208GD-50-LML, MML, SML-6

Some of the information contained in this document may vary from country to country. Before using any NEC product in your application, please contact a representative from the NEC office in your country to obtain a list of authorized representatives and distributors who can verify the following:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics Inc. (U.S.)
Santa Clara
Tel: 800-366-9782
Fax: 800-729-9288

NEC Electronics (Germany) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

NEC Electronics (UK) Ltd.
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Italiana s.r.l.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH
Benelux Office
Eindhoven, the Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.
Velizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.
Spain Office
Madrid, Spain
Tel: 01-504-2787
Fax: 01-504-2860

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
United Square, Singapore 1130
Tel: 253-8311
Fax: 250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-719-2377
Fax: 02-719-5951

NEC do Brasil S.A.
Sao Paulo-SP, Brasil
Tel: 011-889-1680
Fax: 011-889-1689

NEC

U13504EU1V1DS00