IBM Dual Bridge and Memory Controller CPC710-100

User Manual

Version 1.3

June 26, 2000

IBM Microelectronics Division
1580 Route 52, Bldg. 504
Hopewell Junction, NY 12533-6351

The IBM home page can be found at http://www.ibm.com
The IBM Microelectronics Division home page can be found at http://www.chips.ibm.com

6/26/00

Table of Contents

Overview

Features

- Up to 100 MHz PowerPC 60x 64-bit bus
- Supports 100 MHz SDRAM including PC100
- Up to 2 MB flash Boot ROM support
- 32-bit 33 MHz/64-bit 33-66 MHz async dual bus
- Reads two external 32-bit registers
- PreP compliant design
- One-channel DMA controller
- 3.3 volts ± 5%; 5 volt-compliant I/O
- -40 to 105°C junction temperature Industrial
- Power dissipation 2.1 watts typical at 3.3 volts - 100 MHz.
- CBGA package; 625 pins, 32x32mm
- CMOS 5S6, 0.35µm technology (9.0x9.05mm)
- PLL to reduce on-chip system clock skew
- JTAG controller (LSSD design)

60x Bus Interface

- Supports 750 or 7400 or 604e PowerPC
- 100 MHz external bus operation
- Supports two processor or L2 lookaside cache
- Dual 32-byte store back buffers
- High bandwidth 2-way arbiter
- Little Endian mode PowerPC
- Supports SYNC/EIEIO ordering operations
- Supports 60x bus configuration cycles

Memory Controller

- Supports 100 MHz SDRAM including PC100
- Up to 2GB
- 2-way interleaved SDRAM with ECC (external MUX to reduce pin count)
- Supports 16, 64, 128, and 256 Mb SDRAMs
- Programmable timing parameters
- Up to 8 dual bank DIMM
- SDRAM Access command queue with look ahead override option for CPU, PCI's, and DMA
- Access based on 32-byte cache line reload
- Three separate dual 32-byte load buffers (PCI-32, PCI-64, 60x)

PCI-32 and PCI-64 Bus Bridges

- Two independent PCI bus bridges with parking
- PCI revision 2.1 compliant
- 3.3V Compliant with 5.0 V PCI signalling
- Runs async logic to 60x and memory controller
- PCI-64 arbitration can be disabled
- Dual 32-byte buffers in each PCI bus bridge
- Round-robin PCI arbiter
- Coherency for memory access through DMA controller or through PCI master.
- Noncontiguous byte enable transfer to memory
- The CPC710 is single load on all PCI signals

Description

The CPC710 is a highly integrated host bridge device that interfaces a PowerPC 60x bus with SDRAM-based system memory and two PCI ports. It provides arbitration for one or two processors and supports two levels of pipelining per processor along with 64-byte buffers.

The device's memory controller supports SDRAM, allowing the memory to burst data on almost every bus cycle at 100 MHz (1-2-1-1 after initial latency on Read and 1-1-1-1 on write).

For system designs requiring high I/O bandwidth, the device contains two PCI host bus bridges. One bridge supports a standard 32-bit, 33 MHz PCI bus for standard and native I/O. The other bridge supports a 64-bit, 33-66 MHz PCI bus for high data throughput applications such as graphics and high-speed communications.

A DMA controller provides high speed capability for large data transfers between memory and I/O. Store-gathering enhances CPU-to-I/O performance.

Ordering Information

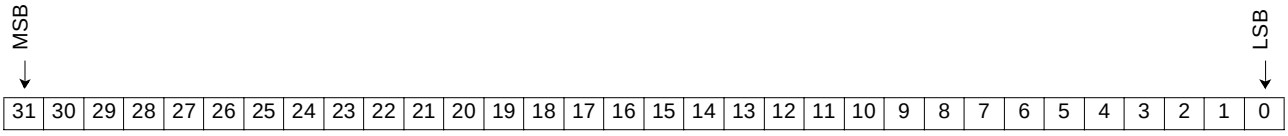
Part Number	Operating Speed	Junct. Temp.	Power Supply	Description
IBM25CPC710AB3A100	100 MHz	-40 +105 C	3.3V +/- 5%	IBM Dual Bridge and Memory Controller

Conventions and Notation

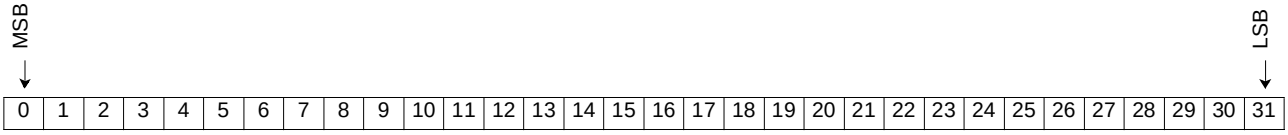
The use of overbars, for example $\overline{\text{RESET}}$, designates signals that are active low. All signals are active high unless shown with an overbar.

Decimal, hexadecimal, and binary numbers are used throughout this document, and are labeled as follows:
Decimal: 1234.56
Hexadecimal: x'ABCD'
Binary: b'0101'

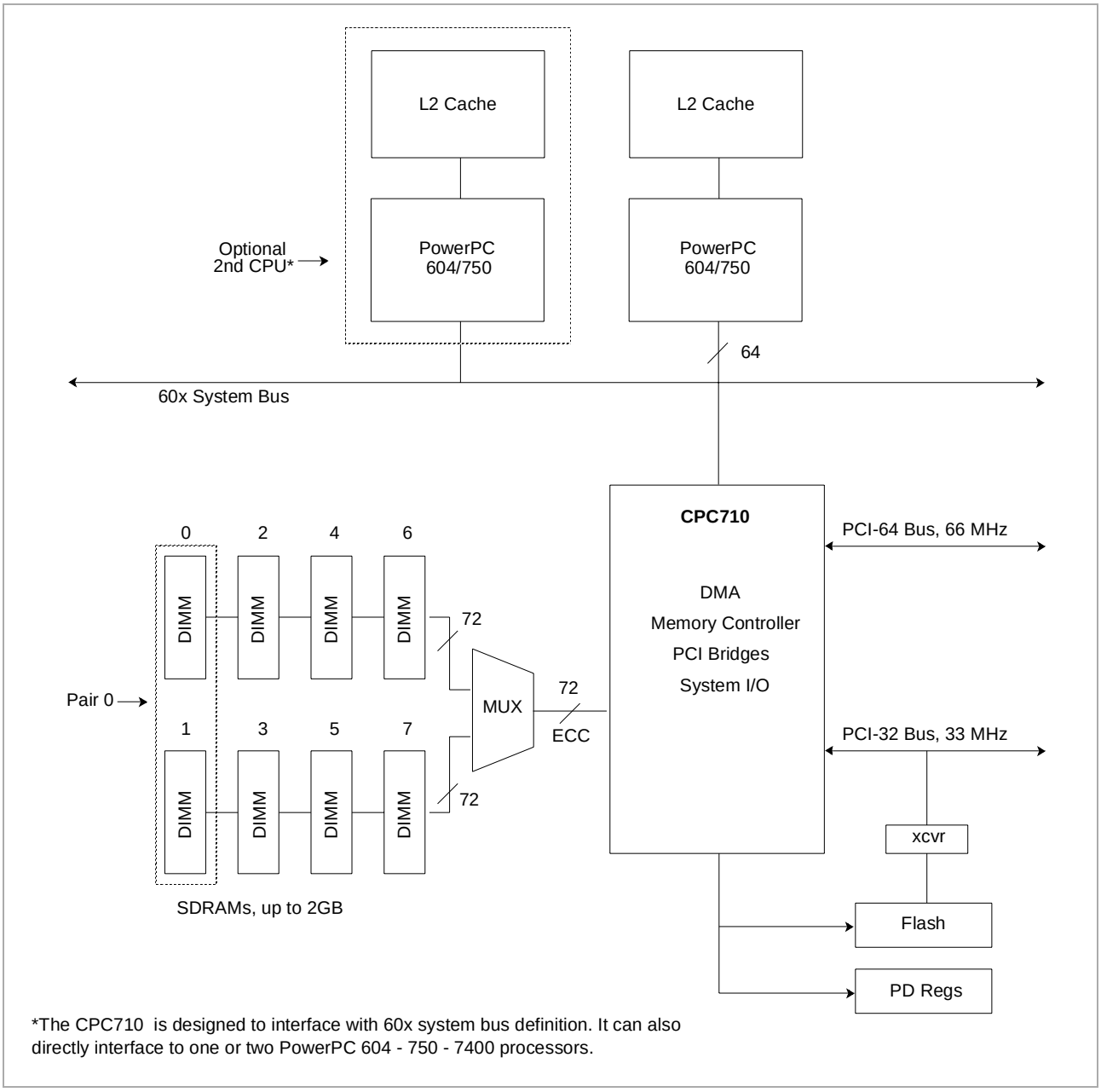
In Little Endian mode, bits and bytes are numbered in descending order from left to right. The most significant bit (MSB) has the highest number and the least significant bit (LSB) has the lowest number:



In Big Endian mode, bits and bytes are numbered in ascending order from left to right. The most significant bit (MSB) has the lowest number and the least significant bit (LSB) has the highest number:

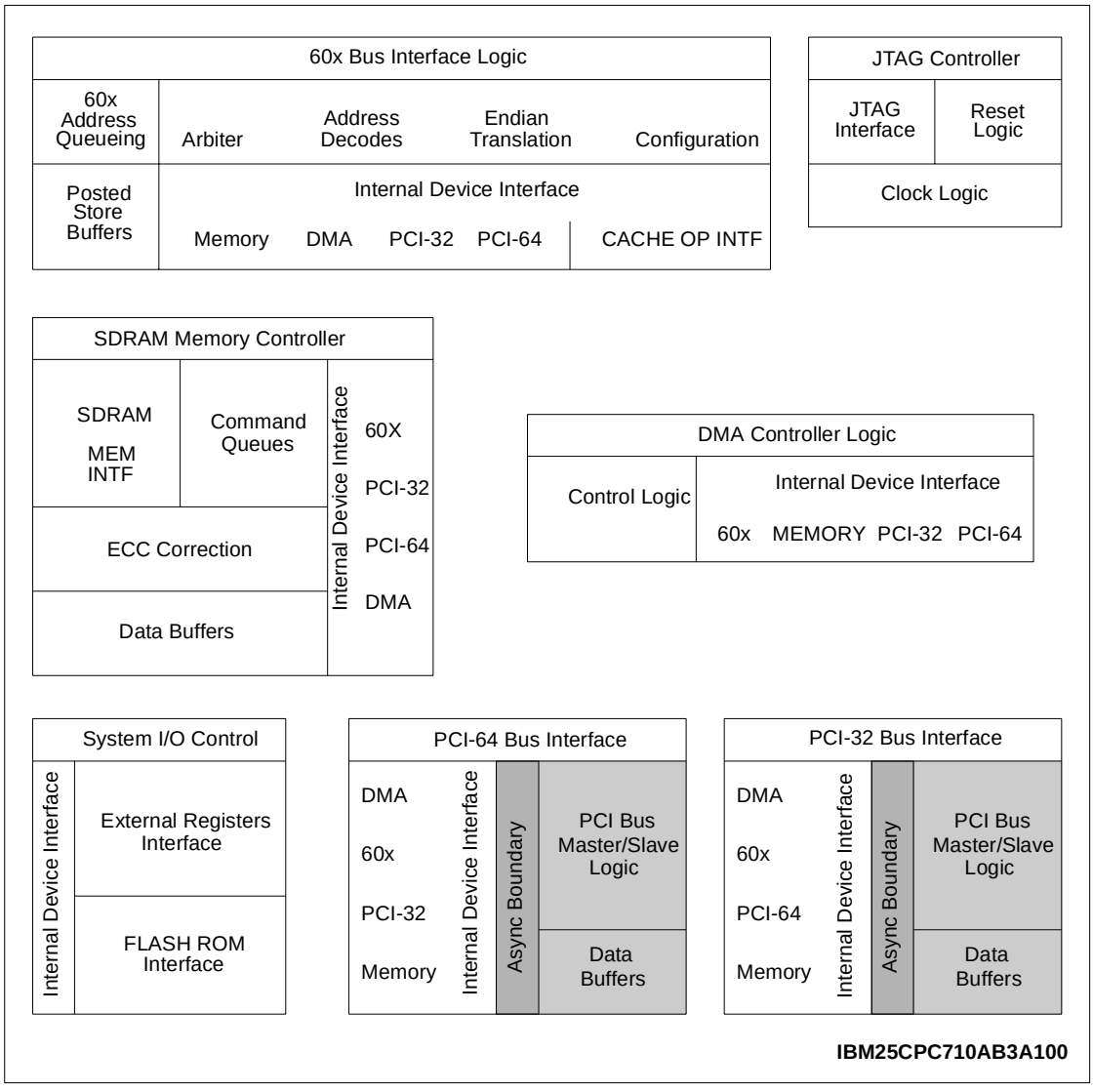


System Level Block Diagram



*The CPC710 is designed to interface with 60x system bus definition. It can also directly interface to one or two PowerPC 604 - 750 - 7400 processors.

Component Block Diagram



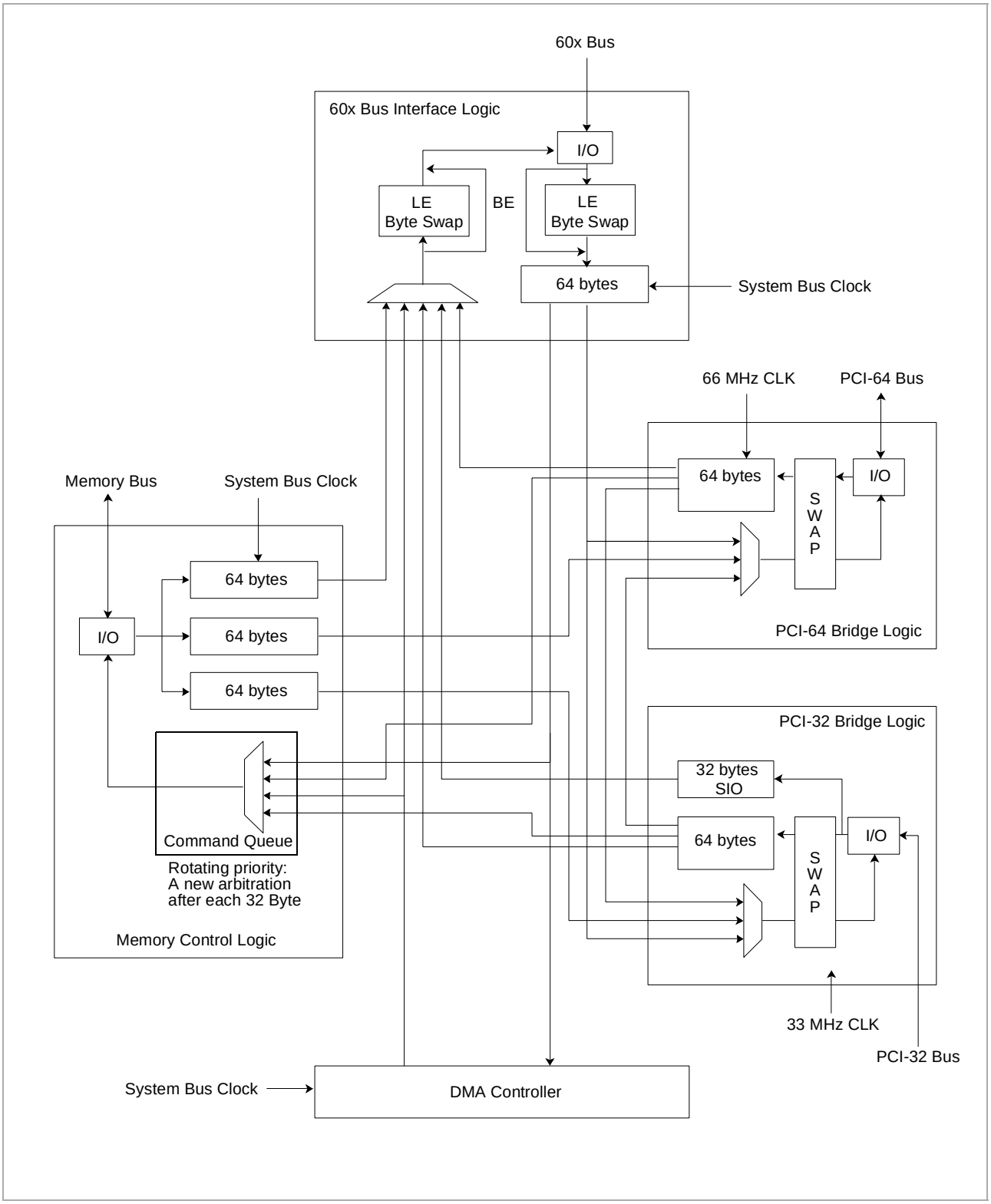
Clocked at System Bus Speed

Clocked at PCI-32 Bus Speed (33 MHz)

Clocked at PCI-64 Bus Speed (33-66 MHz)

Dual Clocked Logic

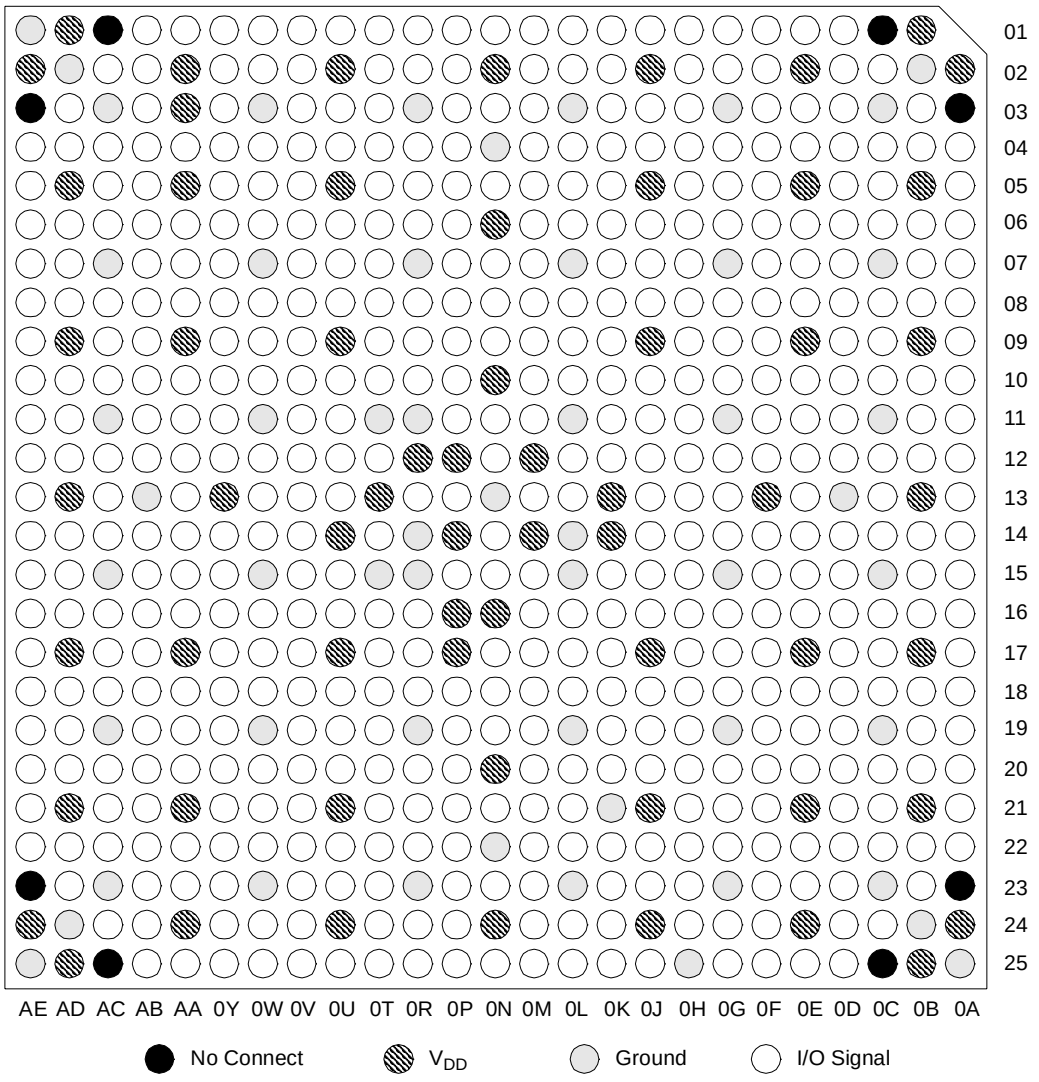
Internal Buffering and Data Flow



Pin Information

Pin assignment (Top View Through Ceramic)

For specific pin assignments, see *Signal Pins, Sorted by Pin Number* on page 14 and *Signal Pins, Sorted by Signal Name* on page 18.



Pin Summary

Pin Type	Number of Pins
V_{DD}	62
Ground	54
Test Pins	20
No Connect	8
Total Module Signal I/Os	480
Total Pins	625

Signal Pins, Sorted by Pin Number (Page 1 of 4)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
0A01	Locator	0B15	G_ADL25	0D04	MUX_CLKENA2	0E18	G_ADH6
0A02	V_{DD}	0B16	TESTIN	0D05	MDATA67	0E19	G_ADH27
0A03	No Connect	0B17	V_{DD}	0D06	G_ARB	0E20	G_ADH17
0A04	WE2	0B18	G_ADH11	0D07	MADDR3	0E21	V_{DD}
0A05	MUX_CLKEN1B	0B19	G_CBE0	0D08	MUX_SEL	0E22	G_ADH19
0A06	MADDR9	0B20	G_ADH10	0D09	MUX_OEA	0E23	INT2
0A07	MADDR7	0B21	V_{DD}	0D10	SCAN_GATE	0E24	V_{DD}
0A08	MUX_CLKEN2B	0B22	G_ADL16	0D11	G_REQ1	0E25	G_PAR64
0A09	MUX_OEB	0B23	G_ADH30	0D12	G_GNT3	0F01	MDATA50
0A10	G_INTB	0B24	Ground	0D13	Ground	0F02	MDATA49
0A11	G_RST	0B25	V_{DD}	0D14	PCG_CLK	0F03	SDCS6
0A12	G_IRDY	0C01	No Connect	0D15	G_ADL26	0F04	MADDR12
0A13	G_GNT0	0C02	SDCKE3	0D16	G_ADL30	0F05	MDATA61
0A14	G_STOP	0C03	Ground	0D17	G_ADL27	0F06	SDCKE5
0A15	G_ADL31	0C04	MADDR11	0D18	G_ADH0	0F07	MDATA70
0A16	G_CBE7	0C05	SYS_HRESET0	0D19	G_CBE2	0F08	G_REQ64
0A17	G_CBE6	0C06	PRES_OE1	0D20	P_ADL20	0F09	SDCKE2
0A18	G_CBE3	0C07	Ground	0D21	G_ADH16	0F10	CHKSTOP
0A19	G_CBE1	0C08	FLASH_OE	0D22	G_ADH28	0F11	G_REQ0
0A20	G_ADL17	0C09	G_REQ4	0D23	G_ADH24	0F12	G_CFG2
0A21	G_ADH12	0C10	MADDR5	0D24	G_ADL15	0F13	V_{DD}
0A22	G_PERR	0C11	Ground	0D25	G_ADH23	0F14	G_ADL20
0A23	No Connect	0C12	G_FRAME	0E01	MDATA48	0F15	G_ADL21
0A24	V_{DD}	0C13	G_INTC	0E02	V_{DD}	0F16	CE0_IO
0A25	Ground	0C14	G_TRDY	0E03	MDATA37	0F17	G_ADH2
0B01	V_{DD}	0C15	Ground	0E04	SDCKE1	0F18	G_GNT5
0B02	Ground	0C16	G_CBE4	0E05	V_{DD}	0F19	G_ADH13
0B03	SDCKE7	0C17	G_ADL19	0E06	SDCKE6	0F20	G_ADH26
0B04	WE1	0C18	G_ADL18	0E07	WE3	0F21	G_ADL4
0B05	V_{DD}	0C19	Ground	0E08	MADDR2	0F22	G_ADH29
0B06	MADDR10	0C20	G_ADL29	0E09	V_{DD}	0F23	G_ADH22
0B07	MADDR8	0C21	G_CBE5	0E10	MADDR4	0F24	G_ADL14
0B08	MADDR6	0C22	G_ADH31	0E11	MUX_CLKENA1	0F25	G_ADL13
0B09	V_{DD}	0C23	Ground	0E12	G_DEVSEL	0G01	MDATA43
0B10	PLL_LOCK	0C24	G_ADH25	0E13	G_GNT1	0G02	MDATA52
0B11	G_GNT2	0C25	No Connect	0E14	G_ADL22	0G03	Ground
0B12	PRES_OE0	0D01	MDATA47	0E15	G_ADH5	0G04	SDCS14
0B13	V_{DD}	0D02	MDATA65	0E16	G_ADH4	0G05	MADDR0_ODD
0B14	G_ADL23	0D03	MDATA66	0E17	V_{DD}	0G06	MDATA45

Signal Pins, Sorted by Pin Number (Page 2 of 4)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
0G07	Ground	0H21	G_ADL6	0K10	SDCAS0	0L24	P_ADL12
0G08	MDATA69	0H22	G_ADL8	0K11	SDCAS1	0L25	P_ADL17
0G09	WE0	0H23	G_PAR	0K12	G_REQ5	0M01	SDCS1
0G10	MADDR0_EVEN	0H24	G_ADL9	0K13	V_{DD}	0M02	CE0_TEST
0G11	Ground	0H25	Ground	0K14	V_{DD}	0M03	SDCS9
0G12	G_REQ3	0J01	MDATA57	0K15	SYS_BR2	0M04	RI
0G13	G_LOCK	0J02	V_{DD}	0K16	G_ADH18	0M05	SDCS8
0G14	G_ACK64	0J03	MDATA54	0K17	SYS_BR3	0M06	CE1_C2
0G15	Ground	0J04	SDCS11	0K18	P_ADL24	0M07	MDATA7
0G16	G_ADH8	0J05	V_{DD}	0K19	FLASH_WE	0M08	CE1_B
0G17	G_ADL0	0J06	MDATA56	0K20	SYS_CLK	0M09	SDRAS0
0G18	G_ADH14	0J07	SDCKE0	0K21	Ground	0M10	SDRAS1
0G19	Ground	0J08	MDATA63	0K22	VDDA	0M11	PLL_RESET
0G20	G_ADL3	0J09	V_{DD}	0K23	P_ADL23	0M12	V_{DD}
0G21	G_ADL5	0J10	MADDR1	0K24	P_ADL22	0M13	RESERVED2
0G22	XADR_LAT	0J11	G_GNT4	0K25	P_ADL18	0M14	V_{DD}
0G23	Ground	0J12	SDCAS2	0L01	SDCS0	0M15	P_CFG0
0G24	G_ADL12	0J13	SDCAS3	0L02	SDCS10	0M16	TESTOUT
0G25	G_ADL11	0J14	G_SERR	0L03	Ground	0M17	P_ADL25
0H01	MDATA42	0J15	G_ADH3	0L04	SDCS12	0M18	P_ADL28
0H02	MDATA41	0J16	G_ADH7	0L05	MDATA40	0M19	P_ADL16
0H03	MDATA36	0J17	V_{DD}	0L06	SDCS13	0M20	TMS
0H04	MDATA55	0J18	G_ADL2	0L07	Ground	0M21	P_REQ5
0H05	MDATA60	0J19	G_ADH20	0L08	SDCS15	0M22	P_ADL15
0H06	SDCS7	0J20	P_ADL21	0L09	MDATA59	0M23	P_REQ4
0H07	MDATA46	0J21	V_{DD}	0L10	SDCKE4	0M24	P_ADL14
0H08	MDATA64	0J22	XCVR_RD	0L11	Ground	0M25	P_REQ3
0H09	MDATA68	0J23	P_RST	0L12	RESERVED3	0N01	SDCS2
0H10	MDATA71	0J24	V_{DD}	0L13	RESERVED5	0N02	V_{DD}
0H11	G_CFG0	0J25	G_ADL7	0L14	Ground	0N03	SDCS5
0H12	G_CFG1	0K01	MDATA58	0L15	Ground	0N04	Ground
0H13	G_ADL24	0K02	CE1_C1	0L16	G_ADL10	0N05	SDCS3
0H14	G_ADL28	0K03	MDATA53	0L17	P_ADL19	0N06	V_{DD}
0H15	G_ADH1	0K04	P_GNT6	0L18	P_ADL27	0N07	MDATA10
0H16	G_ADH9	0K05	MDATA39	0L19	Ground	0N08	MDATA9
0H17	G_ADH15	0K06	CE1_A	0L20	P_ADL29	0N09	MDATA38
0H18	G_ADL1	0K07	MDATA51	0L21	P_ADL30	0N10	V_{DD}
0H19	G_ADH21	0K08	MDATA44	0L22	P_ADL13	0N11	SDRAS2
0H20	G_REQ6	0K09	MDATA62	0L23	Ground	0N12	SYS_CONFIG0

Signal Pins, Sorted by Pin Number (Page 3 of 4)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
0N13	Ground	0R02	MDATA11	0T16	PLL_TUNE1	0V05	MDATA28
0N14	SYS_CONFIG1	0R03	Ground	0T17	P_GNT3	0V06	MDATA30
0N15	P_CFG1	0R04	MDATA12	0T18	P_GNT4	0V07	MDATA1
0N16	V_{DD}	0R05	MDATA22	0T19	P_GNT_0	0V08	SYS_TT3
0N17	P_ADL26	0R06	MDATA35	0T20	P_MEMREQ	0V09	SYS_DATAP2
0N18	P_ADL8	0R07	Ground	0T21	P_GNT1	0V10	SYS_DATAP0
0N19	P_ADL10	0R08	MDATA29	0T22	P_MEMACK	0V11	SYS_DATA29
0N20	V_{DD}	0R09	MDATA33	0T23	G_REQ7	0V12	SYS_DATA37
0N21	G_GNT7	0R10	BS0	0T24	P_ADL0	0V13	SYS_DATA60
0N22	Ground	0R11	Ground	0T25	P_CBE2	0V14	SYS_DATA4
0N23	P_ADL31	0R12	V_{DD}	0U01	MDATA31	0V15	SYS_DATA53
0N24	V_{DD}	0R13	RESERVED4	0U02	V_{DD}	0V16	SYS_ADDR2
0N25	P_ADL11	0R14	Ground	0U03	MDATA32	0V17	SYS_ADDR0
0P01	MDATA13	0R15	Ground	0U04	MDATA8	0V18	SYS_ADDR26
0P02	PCI_CLK	0R16	PLL_TUNE0	0U05	V_{DD}	0V19	SYS_ADDR25
0P03	MDATA14	0R17	P_CBE3	0U06	MDATA18	0V20	P_DEVSEL
0P04	TDI	0R18	P_LOCK	0U07	MDATA0	0V21	G_RESETOUT
0P05	MDATA15	0R19	Ground	0U08	MDATA2	0V22	P_PERR
0P06	TCK	0R20	P_ADL4	0U09	V_{DD}	0V23	SYS_ADDR31
0P07	SYS_HRESET1	0R21	P_GNT2	0U10	SYS_DATA16	0V24	SYS_ADDR16
0P08	TRST	0R22	P_ADL3	0U11	SYS_DATA28	0V25	SYS_ADDR22
0P09	SDCS4	0R23	Ground	0U12	O_GPIO1	0W01	SYS_DBG1
0P10	BS1	0R24	P_ADL7	0U13	O_GPIO0	0W02	SYS_AACK
0P11	SDRAS3	0R25	P_PAR	0U14	V_{DD}	0W03	Ground
0P12	V_{DD}	0T01	MDATA16	0U15	SYS_DATA48	0W04	MDATA6
0P13	RESERVED6	0T02	POWERGOOD	0U16	RESERVED1	0W05	SYS_TSIZ1
0P14	V_{DD}	0T03	MDATA26	0U17	V_{DD}	0W06	SYS_DBG0
0P15	P_CFG2	0T04	DI1	0U18	P_GNT5	0W07	Ground
0P16	V_{DD}	0T05	MDATA25	0U19	G_IDSEL	0W08	SYS_DATAP1
0P17	V_{DD}	0T06	DI2	0U20	SYS_ADDRP1	0W09	SYS_DATAP3
0P18	P_CBE0	0T07	MDATA3	0U21	V_{DD}	0W10	SYS_DATA17
0P19	P_REQ2	0T08	MDATA20	0U22	P_ADL9	0W11	Ground
0P20	P_ADL1	0T09	MDATA19	0U23	P_SERR	0W12	SYS_DATA38
0P21	P_REQ1	0T10	MDATA27	0U24	V_{DD}	0W13	SYS_DATA61
0P22	P_ADL2	0T11	Ground	0U25	P_CBE1	0W14	G_GNT6
0P23	P_REQ0	0T12	P_REQ6	0V01	MDATA24	0W15	Ground
0P24	P_ADL6	0T13	V_{DD}	0V02	MDATA23	0W16	SYS_ADDR3
0P25	P_ISA_MASTER	0T14	CE_TRST	0V03	MDATA4	0W17	SYS_ADDRP3
0R01	MDATA5	0T15	Ground	0V04	MDATA17	0W18	SYS_ADDR9

Signal Pins, Sorted by Pin Number (Page 4 of 4)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
0W19	Ground	AA09	V_{DD}	AB23	SYS_ADDR29	AD12	SYS_DATA62
0W20	SYS_ADDR24	AA10	SYS_DATA10	AB24	SYS_ADDR28	AD13	V_{DD}
0W21	SYS_ADDR10	AA11	SYS_DATA11	AB25	SYS_ADDR27	AD14	SYS_DATA56
0W22	P_STOP	AA12	SYS_ARTRY	AC01	No Connect	AD15	SYS_DATA7
0W23	Ground	AA13	SYS_DATA57	AC02	DLK	AD16	SYS_DATA52
0W24	SYS_ADDR20	AA14	SYS_TS	AC03	Ground	AD17	V_{DD}
0W25	SYS_ADDR21	AA15	SYS_DATA58	AC04	SYS_DATAP6	AD18	SYS_DATA44
0Y01	$\overline{\text{SYS\_BG0}}$	AA16	SYS_DATA1	AC05	$\overline{\text{SYS\_SHD}}$	AD19	SYS_ADDR5
0Y02	$\overline{\text{SYS\_BG1}}$	AA17	V_{DD}	AC06	SYS_DATA34	AD20	SYS_ADDR7
0Y03	MDATA34	AA18	SYS_DATA46	AC07	Ground	AD21	V_{DD}
0Y04	$\overline{\text{SYS\_MACHK0}}$	AA19	SYS_DATA0	AC08	SYS_DATA33	AD22	SYS_ADDR12
0Y05	MDATA21	AA20	$\overline{\text{G\_INTD}}$	AC09	SYS_DATA27	AD23	SYS_ADDR14
0Y06	SYS_TT2	AA21	V_{DD}	AC10	SYS_DATA25	AD24	Ground
0Y07	SYS_DATA19	AA22	PLN_RTC_CLK	AC11	Ground	AD25	V_{DD}
0Y08	O_GPIO2	AA23	$\overline{\text{P\_TRDY}}$	AC12	$\overline{\text{SYS\_L2\_HIT}}$	AE01	Ground
0Y09	RESERVED8	AA24	V_{DD}	AC13	SYS_DATA59	AE02	V_{DD}
0Y10	$\overline{\text{SYS\_SRESET1}}$	AA25	SYS_ADDR17	AC14	SYS_TA	AE03	No Connect
0Y11	SYS_DATA35	AB01	SYS_TSIZ2	AC15	Ground	AE04	$\overline{\text{SYS\_MACHK1}}$
0Y12	TDO	AB02	SYS_TBST	AC16	$\overline{\text{INT1}}$	AE05	SYS_DATA21
0Y13	V_{DD}	AB03	NODLK	AC17	SYS_DATA51	AE06	SYS_DATA22
0Y14	SYS_TBE	AB04	SYS_TT1	AC18	SYS_DATA42	AE07	SYS_DATA14
0Y15	SYS_DATA41	AB05	SYS_DATAP7	AC19	Ground	AE08	SYS_DATA13
0Y16	$\overline{\text{SYS\_SRESET0}}$	AB06	SYS_TEA	AC20	SYS_DATA3	AE09	SYS_DATA30
0Y17	SYS_DATA49	AB07	SYS_DATA12	AC21	$\overline{\text{G\_REQ2}}$	AE10	SYS_DATA31
0Y18	$\overline{\text{P\_FRAME}}$	AB08	RESERVED7	AC22	SYS_ADDR13	AE11	SYS_DATA32
0Y19	SYS_ADDR1	AB09	SYS_DATA36	AC23	Ground	AE12	$\overline{\text{SYS\_BR1}}$
0Y20	SYS_ADDRP2	AB10	SYS_DATA9	AC24	SYS_ADDR30	AE13	$\overline{\text{G\_INTA}}$
0Y21	SYS_ADDR23	AB11	SYS_DATA8	AC25	No Connect	AE14	$\overline{\text{SYS\_BR0}}$
0Y22	SYS_ADDR15	AB12	SYS_DATA39	AD01	V_{DD}	AE15	SYS_DATA43
0Y23	P_ADL5	AB13	Ground	AD02	Ground	AE16	SYS_DATA55
0Y24	SYS_ADDR18	AB14	SYS_DATA40	AD03	SYS_DATAP5	AE17	SYS_DATA50
0Y25	SYS_ADDR19	AB15	SYS_DATA6	AD04	SYS_DATAP4	AE18	SYS_DATA45
AA01	SYS_TSIZ0	AB16	SYS_DATA2	AD05	V_{DD}	AE19	SYS_ADDR4
AA02	V_{DD}	AB17	SYS_DATA5	AD06	SYS_DATA18	AE20	SYS_ADDR6
AA03	V_{DD}	AB18	SYS_DATA54	AD07	SYS_DATA23	AE21	SYS_ADDR8
AA04	SYS_TT4	AB19	SYS_DATA47	AD08	SYS_DATA24	AE22	SYS_ADDR11
AA05	V_{DD}	AB20	$\overline{\text{P\_IRDY}}$	AD09	V_{DD}	AE23	No Connect
AA06	SYS_TT0	AB21	SYS_GBL	AD10	SYS_DATA26	AE24	V_{DD}
AA07	SYS_DATA20	AB22	SYS_ADDRP0	AD11	SYS_DATA63	AE25	Ground
AA08	SYS_DATA15						

Signal Pins, Sorted by Signal Name (Page 1 of 4)

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
BS0	0R10	G_ADH20	0J19	G_ADL27	0D17	G_REQ2	AC21
BS1	0P10	G_ADH21	0H19	G_ADL28	0H14	G_REQ3	0G12
CE_TRST	0T14	G_ADH22	0F23	G_ADL29	0C20	G_REQ4	0C09
CE0_IO	0F16	G_ADH23	0D25	G_ADL30	0D16	G_REQ5	0K12
CE0_TEST	0M02	G_ADH24	0D23	G_ADL31	0A15	G_REQ6	0H20
CE1_A	0K06	G_ADH25	0C24	G_ARB	0D06	G_REQ7	0T23
CE1_B	0M08	G_ADH26	0F20	G_CBE0	0B19	G_REQ64	0F08
CE1_C1	0K02	G_ADH27	0E19	G_CBE1	0A19	G_RESETOUT	0V21
CE1_C2	0M06	G_ADH28	0D22	G_CBE2	0D19	G_RST	0A11
CHKSTOP	0F10	G_ADH29	0F22	G_CBE3	0A18	G_SERR	0J14
DI1	0T04	G_ADH30	0B23	G_CBE4	0C16	G_STOP	0A14
DI2	0T06	G_ADH31	0C22	G_CBE5	0C21	G_TRDY	0C14
DLK	AC02	G_ADL0	0G17	G_CBE6	0A17	Ground	0A25
RESERVED4	0R13	G_ADL1	0H18	G_CBE7	0A16	Ground	0B02
RESERVED5	0L13	G_ADL2	0J18	G_CFG0	0H11	Ground	0B24
RESERVED6	0P13	G_ADL3	0G20	G_CFG1	0H12	Ground	0C03
FLASH_OE	0C08	G_ADL4	0F21	G_CFG2	0F12	Ground	0C07
FLASH_WE	0K19	G_ADL5	0G21	G_DEVSEL	0E12	Ground	0C11
G_ACK64	0G14	G_ADL6	0H21	G_FRAME	0C12	Ground	0C15
G_ADH0	0D18	G_ADL7	0J25	G_GNT0	0A13	Ground	0C19
G_ADH1	0H15	G_ADL8	0H22	G_GNT1	0E13	Ground	0C23
G_ADH2	0F17	G_ADL9	0H24	G_GNT2	0B11	Ground	0D13
G_ADH3	0J15	G_ADL10	0L16	G_GNT3	0D12	Ground	0G03
G_ADH4	0E16	G_ADL11	0G25	G_GNT4	0J11	Ground	0G07
G_ADH5	0E15	G_ADL12	0G24	G_GNT5	0F18	Ground	0G11
G_ADH6	0E18	G_ADL13	0F25	G_GNT6	0W14	Ground	0G15
G_ADH7	0J16	G_ADL14	0F24	G_GNT7	0N21	Ground	0G19
G_ADH8	0G16	G_ADL15	0D24	G_IDSEL	0U19	Ground	0H25
G_ADH9	0H16	G_ADL16	0B22	G_INTA	AE13	Ground	0K21
G_ADH10	0B20	G_ADL17	0A20	G_INTB	0A10	Ground	0L03
G_ADH11	0B18	G_ADL18	0C18	G_INTC	0C13	Ground	0L07
G_ADH12	0A21	G_ADL19	0C17	G_INTD	AA20	Ground	0L11
G_ADH13	0F19	G_ADL20	0F14	G_IRDY	0A12	Ground	0L14
G_ADH14	0G18	G_ADL21	0F15	G_LOCK	0G13	Ground	0L15
G_ADH15	0H17	G_ADL22	0E14	G_PAR	0H23	Ground	0L19
G_ADH16	0D21	G_ADL23	0B14	G_PAR64	0E25	Ground	0L23
G_ADH17	0E20	G_ADL24	0H13	G_PERR	0A22	Ground	0N04
G_ADH18	0K16	G_ADL25	0B15	G_REQ0	0F11	Ground	0N13
G_ADH19	0E22	G_ADL26	0D15	G_REQ1	0D11	Ground	0N22

Signal Pins, Sorted by Signal Name (Page 2 of 4)

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
Ground	0R03	MADDR9	0A06	MDATA34	0Y03	MUX_CLKEN2B	0A08
Ground	0R07	MADDR10	0B06	MDATA35	0R06	MUX_CLKENA1	0E11
Ground	0R11	MADDR11	0C04	MDATA36	0H03	MUX_CLKENA2	0D04
Ground	0R14	MADDR12	0F04	MDATA37	0E03	MUX_OEA	0D09
Ground	0R15	MADDR0_ODD	0G05	MDATA38	0N09	MUX_OEB	0A09
Ground	0R19	MDATA0	0U07	MDATA39	0K05	MUX_SEL	0D08
Ground	0R23	MDATA1	0V07	MDATA40	0L05	No Connect	0A03
Ground	0T11	MDATA2	0U08	MDATA41	0H02	No Connect	0C01
Ground	0T15	MDATA3	0T07	MDATA42	0H01	No Connect	0C25
Ground	0W03	MDATA4	0V03	MDATA43	0G01	No Connect	AC01
Ground	0W07	MDATA5	0R01	MDATA44	0K08	No Connect	AC25
Ground	0W11	MDATA6	0W04	MDATA45	0G06	No Connect	AE03
Ground	0W15	MDATA7	0M07	MDATA46	0H07	No Connect	AE23
Ground	0W19	MDATA8	0U04	MDATA47	0D01	No Connect	0A23
Ground	0W23	MDATA9	0N08	MDATA48	0E01	NODLK	AB03
Ground	AB13	MDATA10	0N07	MDATA49	0F02	O_GPIO0	0U13
Ground	AC03	MDATA11	0R02	MDATA50	0F01	O_GPIO1	0U12
Ground	AC07	MDATA12	0R04	MDATA51	0K07	O_GPIO2	0Y08
Ground	AC11	MDATA13	0P01	MDATA52	0G02	P_ADL0	0T24
Ground	AC15	MDATA14	0P03	MDATA53	0K03	P_ADL1	0P20
Ground	AC19	MDATA15	0P05	MDATA54	0J03	P_ADL2	0P22
Ground	AC23	MDATA16	0T01	MDATA55	0H04	P_ADL3	0R22
Ground	AD02	MDATA17	0V04	MDATA56	0J06	P_ADL4	0R20
Ground	AD24	MDATA18	0U06	MDATA57	0J01	P_ADL5	0Y23
Ground	AE01	MDATA19	0T09	MDATA58	0K01	P_ADL6	0P24
Ground	AE25	MDATA20	0T08	MDATA59	0L09	P_ADL7	0R24
Ground	0G23	MDATA21	0Y05	MDATA60	0H05	P_ADL8	0N18
INT1	AC16	MDATA22	0R05	MDATA61	0F05	P_ADL9	0U22
INT2	0E23	MDATA23	0V02	MDATA62	0K09	P_ADL10	0N19
Locator	0A01	MDATA24	0V01	MDATA63	0J08	P_ADL11	0N25
MADDR0_EVEN	0G10	MDATA25	0T05	MDATA64	0H08	P_ADL12	0L24
MADDR1	0J10	MDATA26	0T03	MDATA65	0D02	P_ADL13	0L22
MADDR2	0E08	MDATA27	0T10	MDATA66	0D03	P_ADL14	0M24
MADDR3	0D07	MDATA28	0V05	MDATA67	0D05	P_ADL15	0M22
MADDR4	0E10	MDATA29	0R08	MDATA68	0H09	P_ADL16	0M19
MADDR5	0C10	MDATA30	0V06	MDATA69	0G08	P_ADL17	0L25
MADDR6	0B08	MDATA31	0U01	MDATA70	0F07	P_ADL18	0K25
MADDR7	0A07	MDATA32	0U03	MDATA71	0H10	P_ADL19	0L17
MADDR8	0B07	MDATA33	0R09	MUX_CLKEN1B	0A05	P_ADL20	0D20

Signal Pins, Sorted by Signal Name (Page 3 of 4)

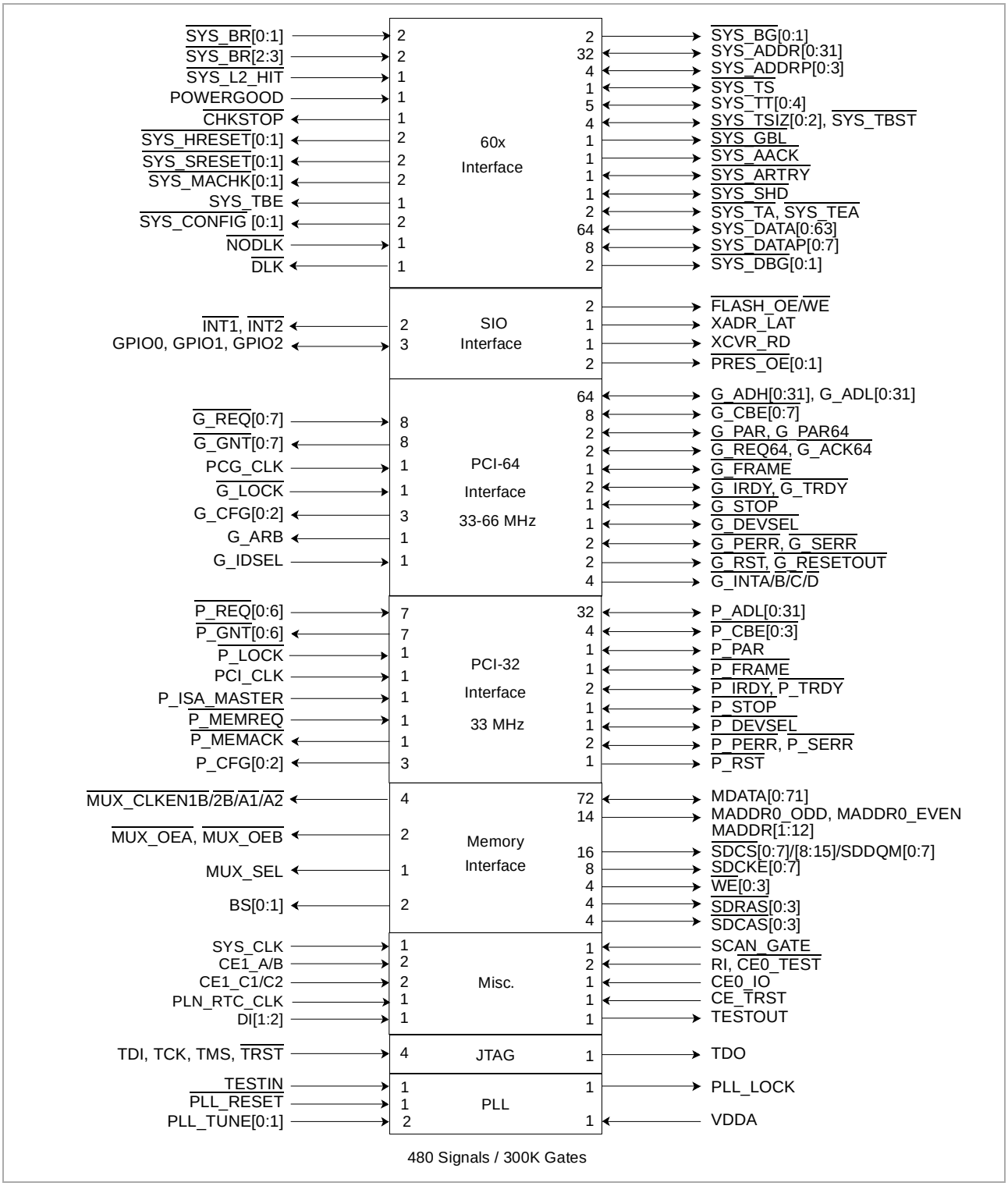
Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
P_ADL21	0J20	P_REQ5	0M21	SDCS4	0P09	SYS_ADDR22	0V25
P_ADL22	0K24	P_REQ6	0T12	SDCS5	0N03	SYS_ADDR23	0Y21
P_ADL23	0K23	P_RST	0J23	SDCS6	0F03	SYS_ADDR24	0W20
P_ADL24	0K18	P_SERR	0U23	SDCS7	0H06	SYS_ADDR25	0V19
P_ADL25	0M17	P_STOP	0W22	SDCS8	0M05	SYS_ADDR26	0V18
P_ADL26	0N17	P_TRDY	AA23	SDCS9	0M03	SYS_ADDR27	AB25
P_ADL27	0L18	PCG_CLK	0D14	SDCS10	0L02	SYS_ADDR28	AB24
P_ADL28	0M18	PCI_CLK	0P02	SDCS11	0J04	SYS_ADDR29	AB23
P_ADL29	0L20	PLL_LOCK	0B10	SDCS12	0L04	SYS_ADDR30	AC24
P_ADL30	0L21	PLL_RESET	0M11	SDCS13	0L06	SYS_ADDR31	0V23
P_ADL31	0N23	PLL_TUNE0	0R16	SDCS14	0G04	SYS_ADDRP0	AB22
P_CBE0	0P18	PLL_TUNE1	0T16	SDCS15	0L08	SYS_ADDRP1	0U20
P_CBE1	0U25	PLN_RTC_CLK	AA22	SDRAS0	0M09	SYS_ADDRP2	0Y20
P_CBE2	0T25	POWERGOOD	0T02	SDRAS1	0M10	SYS_ADDRP3	0W17
P_CBE3	0R17	PRES_OE0	0B12	SDRAS2	0N11	SYS_ARTRY	AA12
P_CFG0	0M15	PRES_OE1	0C06	SDRAS3	0P11	SYS_BG0	0Y01
P_CFG1	0N15	RESERVED1	0U16	SYS_AACK	0W02	SYS_BG1	0Y02
P_CFG2	0P15	RESERVED2	0M13	SYS_ADDR0	0V17	SYS_BR0	AE14
P_DEVSEL	0V20	RESERVED3	0L12	SYS_ADDR1	0Y19	SYS_BR1	AE12
P_FRAME	0Y18	RESERVED7	AB08	SYS_ADDR2	0V16	SYS_BR2	0K15
P_GNT_0	0T19	RESERVED8	0Y09	SYS_ADDR3	0W16	SYS_BR3	0K17
P_GNT1	0T21	RI	0M04	SYS_ADDR4	AE19	SYS_CLK	0K20
P_GNT2	0R21	SCAN_GATE	0D10	SYS_ADDR5	AD19	SYS_CONFIG0	0N12
P_GNT3	0T17	SDCAS0	0K10	SYS_ADDR6	AE20	SYS_CONFIG1	0N14
P_GNT4	0T18	SDCAS1	0K11	SYS_ADDR7	AD20	SYS_DATA0	AA19
P_GNT5	0U18	SDCAS2	0J12	SYS_ADDR8	AE21	SYS_DATA1	AA16
P_GNT6	0K04	SDCAS3	0J13	SYS_ADDR9	0W18	SYS_DATA2	AB16
P_IRDY	AB20	SDCKE0	0J07	SYS_ADDR10	0W21	SYS_DATA3	AC20
P_ISA_MASTER	0P25	SDCKE1	0E04	SYS_ADDR11	AE22	SYS_DATA4	0V14
P_LOCK	0R18	SDCKE2	0F09	SYS_ADDR12	AD22	SYS_DATA5	AB17
P_MEMACK	0T22	SDCKE3	0C02	SYS_ADDR13	AC22	SYS_DATA6	AB15
P_MEMREQ	0T20	SDCKE4	0L10	SYS_ADDR14	AD23	SYS_DATA7	AD15
P_PAR	0R25	SDCKE5	0F06	SYS_ADDR15	0Y22	SYS_DATA8	AB11
P_PERR	0V22	SDCKE6	0E06	SYS_ADDR16	0V24	SYS_DATA9	AB10
P_REQ0	0P23	SDCKE7	0B03	SYS_ADDR17	AA25	SYS_DATA10	AA10
P_REQ1	0P21	SDCS0	0L01	SYS_ADDR18	0Y24	SYS_DATA11	AA11
P_REQ2	0P19	SDCS1	0M01	SYS_ADDR19	0Y25	SYS_DATA12	AB07
P_REQ3	0M25	SDCS2	0N01	SYS_ADDR20	0W24	SYS_DATA13	AE08
P_REQ4	0M23	SDCS3	0N05	SYS_ADDR21	0W25	SYS_DATA14	AE07

Signal Pins, Sorted by Signal Name (Page 4 of 4)

Signal	Pin	Signal	Pin	Signal	Pin	Signal	Pin
SYS_DATA15	AA08	SYS_DATA55	AE16	SYS_TT3	0V08	V_{DD}	0Y13
SYS_DATA16	0U10	SYS_DATA56	AD14	SYS_TT4	AA04	V_{DD}	AA02
SYS_DATA17	0W10	SYS_DATA57	AA13	TCK	0P06	V_{DD}	AA03
SYS_DATA18	AD06	SYS_DATA58	AA15	TDI	0P04	V_{DD}	AA05
SYS_DATA19	0Y07	SYS_DATA59	AC13	TDO	0Y12	V_{DD}	AA09
SYS_DATA20	AA07	SYS_DATA60	0V13	TESTIN	0B16	V_{DD}	AA17
SYS_DATA21	AE05	SYS_DATA61	0W13	TESTOUT	0M16	V_{DD}	AA21
SYS_DATA22	AE06	SYS_DATA62	AD12	TMS	0M20	V_{DD}	AA24
SYS_DATA23	AD07	SYS_DATA63	AD11	TRST	0P08	V_{DD}	AD01
SYS_DATA24	AD08	SYS_DATAP0	0V10	V_{DD}	0B01	V_{DD}	AD05
SYS_DATA25	AC10	SYS_DATAP1	0W08	V_{DD}	0B05	V_{DD}	AD09
SYS_DATA26	AD10	SYS_DATAP2	0V09	V_{DD}	0B09	V_{DD}	AD13
SYS_DATA27	AC09	SYS_DATAP3	0W09	V_{DD}	0B17	V_{DD}	AD17
SYS_DATA28	0U11	SYS_DATAP4	AD04	V_{DD}	0B21	V_{DD}	AD21
SYS_DATA29	0V11	SYS_DATAP5	AD03	V_{DD}	0B25	V_{DD}	AD25
SYS_DATA30	AE09	SYS_DATAP6	AC04	V_{DD}	0E02	V_{DD}	AE24
SYS_DATA31	AE10	SYS_DATAP7	AB05	V_{DD}	0E05	V_{DD}	0A02
SYS_DATA32	AE11	$\overline{\text{SYS\_DBG0}}$	0W06	V_{DD}	0M14	V_{DD}	0A24
SYS_DATA33	AC08	$\overline{\text{SYS\_DBG1}}$	0W01	V_{DD}	AE02	V_{DD}	0B13
SYS_DATA34	AC06	SYS_GBL	AB21	V_{DD}	0M12	V_{DD}	0E09
SYS_DATA35	0Y11	SYS_HRESET0	0C05	V_{DD}	0N02	V_{DD}	0E17
SYS_DATA36	AB09	$\overline{\text{SYS\_HRESET1}}$	0P07	V_{DD}	0N06	V_{DD}	0E21
SYS_DATA37	0V12	SYS_L2_HIT	AC12	V_{DD}	0N10	V_{DD}	0E24
SYS_DATA38	0W12	$\overline{\text{SYS\_MACHK0}}$	0Y04	V_{DD}	0N16	V_{DD}	0F13
SYS_DATA39	AB12	$\overline{\text{SYS\_MACHK1}}$	AE04	V_{DD}	0N20	V_{DD}	0J02
SYS_DATA40	AB14	$\overline{\text{SYS\_SHD}}$	AC05	V_{DD}	0N24	V_{DD}	0J05
SYS_DATA41	0Y15	$\overline{\text{SYS\_SRESET0}}$	0Y16	V_{DD}	0P12	V_{DD}	0J09
SYS_DATA42	AC18	$\overline{\text{SYS\_SRESET1}}$	0Y10	V_{DD}	0P14	V_{DD}	0J17
SYS_DATA43	AE15	SYS_TA	AC14	V_{DD}	0P16	V_{DD}	0J21
SYS_DATA44	AD18	SYS_TBE	0Y14	V_{DD}	0P17	V_{DD}	0J24
SYS_DATA45	AE18	SYS_TBST	AB02	V_{DD}	0R12	V_{DD}	0K13
SYS_DATA46	AA18	SYS_TEA	AB06	V_{DD}	0T13	V_{DD}	0K14
SYS_DATA47	AB19	SYS_TS	AA14	V_{DD}	0U02	VDDA	0K22
SYS_DATA48	0U15	SYS_TSIZ0	AA01	V_{DD}	0U05	$\overline{\text{WE0}}$	0G09
SYS_DATA49	0Y17	SYS_TSIZ1	0W05	V_{DD}	0U09	$\overline{\text{WE1}}$	0B04
SYS_DATA50	AE17	SYS_TSIZ2	AB01	V_{DD}	0U14	$\overline{\text{WE2}}$	0A04
SYS_DATA51	AC17	SYS_TT0	AA06	V_{DD}	0U17	$\overline{\text{WE3}}$	0E07
SYS_DATA52	AD16	SYS_TT1	AB04	V_{DD}	0U21	XADR_LAT	0G22
SYS_DATA53	0V15	SYS_TT2	0Y06	V_{DD}	0U24	XCVR_RD	0J22
SYS_DATA54	AB18						

I/O Signals

I/O Signal Diagram



Signal Descriptions

Tri-state driver/receivers interface 3.3 V internal functions with 3.3 V LVTTL off-chip buses. Some receivers interface 3.3 V internal functions with either 3.3 V LVTTL or 5.0 V TTL off-chip bidirectional buses. All drivers are 50 ohm, source-terminated. In the following table, "Pull-up" in the Type column indicates an internal 10K pull-up is built into the device driver/receiver. No additional external device is required.

60x Bus Interface Signals (Page 1 of 3)

Signal Name	I/O	Type	Description
Address Bus Arbitration Signals			
$\overline{\text{SYS\_BR0}}$ $\overline{\text{SYS\_BR1}}$	I	5.0V-tolerant Int. pull-up	**Bus Request.** Indicates the device on the 60x bus associated with this signal is requesting ownership of the address bus.
$\overline{\text{SYS\_BR2}}$ $\overline{\text{SYS\_BR3}}$	I	Int. pull-up	Used for Quad Processor arbitration with external logic. Should be tied to up level (=1) if unused
$\overline{\text{SYS\_BG0}}$ $\overline{\text{SYS\_BG1}}$	O		**Bus Grant.** Indicates the master associated with this signal may, with proper qualification, assume mastership of the address bus.
Address Transfer Start Signals			
$\overline{\text{SYS\_TS}}$	I/O		**Transfer Start.** **Output:** Indicates device has started an address tenure and the address bus and transfer attribute signals are valid. Only “address only operations” and “snoop operation with programmable TT code” are performed. **Input:** Indicates a master on the 60x has started an address tenure and the address bus and transfer attribute signals are valid. For address tenures that require a data transfer, this signal also indicates a request for the data bus.
Address Bus Signals			
SYS_ADDR[0:31]	I/O	Internal Pull-up	**Address Bus.** **Output:** Represents the physical address of a cache operation that should be snooped by devices on the 60x bus. A[0] is the most significant address bit. **Input**: Represents the physical address for the current transaction
SYS_ADDRP[0:3]	I/O	Internal Pull-up	**Address Parity.** **Output:** Represents one bit of odd parity for each of the four bytes of the address bus. Odd parity means that an odd number of bits, including the parity bit, are driven high. The signal assignments correspond to the following: AP[0] - A[0:7] AP[1] - A[8:15] AP[2] - A[16:23] AP[3] - A[24:31] **Input:** Represents one bit of odd parity for each of the four bytes of the address bus. A checkstop is generated if bad parity is detected and bit 8 is ‘1’ in the error control register.
$\overline{\text{SYS\_CONFIG0}}$ $\overline{\text{SYS\_CONFIG1}}$	O		**Configuration.** Indicate the current address tenure is a configuration cycle to the device associated with this signal. The associated device must respond (if present) to addresses in the range ‘FF20 0000’ through ‘FF20 1FFF.’ Other addresses are responded to as normal.
Transfer Attribute Signals			
SYS_TT[0:4]	I/O	Internal Pull-up	**Transfer Type.** **Output:** Indicates the type of transfer in progress. The values are programmable according to the PowerPC type and stored in the ATAS register. **Input**: Indicates the type of transfer in progress.

60x Bus Interface Signals (Page 2 of 3)

Signal Name	I/O	Type	Description
SYS_TSIZ[0:2]	I/O	Internal Pull-up	**Transfer Size.** **Output signals and the TBST signal**: Indicate the data transfer size of the operation. Device sets these signals to a value stored in the ATAS register for the operations it initiates. **Input signals and the TBST signal**: For normal memory accesses, indicate the data transfer size of the operation. For the DMA instructions (eciwx and ecowx), they indicate the 4-bit Resource ID (RID) of the DMA operation (TBST || TSIZ0-TSIZ2).
SYS_TBST	I/O	Internal Pull-up	**Transfer Burst.** **Output signal and the TSIZ signals:** Indicate the data transfer size of the operation. Device sets this signal according to the bit in the ATAS register for operations it initiates. **Input signal:** For normal memory accesses, indicates a burst transfer is in progress. For DMA instructions (eciwx and ecowx), the input signal and the TSIZ signals indicate the 4-bit Resource ID (RID) of the DMA operation (TBST || TSIZ0-TSIZ2).
SYS_GBL	O	Tri-state Int. pull-up	**Global.** Always asserted by the CPC710 for transactions that it initiates to indicate that all devices on the 60x bus must snoop the transaction. Since the CPC710 asserts this signal only when it is PowerPC bus address master, no contention is possible with PowerPC 750 or 7400 Input/output GBL signal connected to SYS_GBL.
Address Transfer Termination Signals			
SYS_AACK	O		**Address Acknowledge.** Indicates the address tenure is complete and the ARTRY sampling window ends on the following bus cycle. Address bus and transfer attribute signals must go to tri-state on the next bus cycle.
SYS_ARTRY	I/O		**Address Retry.** **Output** indicates device detects a condition that requires an address tenure to be retried. **Input**: When asserted in response to a device cache operation, device assumes the cache line is modified and/or present in a CPU or L2 cache. Device then retries the operation on the PCI bus and address tenure is not rerun until the device on the PCI bus reruns its transfer. The pre-charge logic is always signaled to initiate the pre-charge sequence.
SYS_SHD	I/O	Internal Pull-up	**Shared.** **Output**: Not applicable; Device only pre-charges the signal. **Input:** Instructs the pre-charge logic to initiate a pre-charge sequence.
SYS_L2_HIT	I	Internal Pull-up	**L2 Hit.** Indicates an external slave has been addressed by the current master. The device arbiter uses this signal to confirm positive selection of an address tenure on the 60x bus. Warning: This signal is subject to timing constraints.
Data Bus Arbitration Signals			
SYS_DBG0 SYS_DBG1	O		**Data Bus Grant.** Indicates the device associated with this signal may, with the proper qualification, assume mastership of the data bus.
Data Transfer Signals			
SYS_DATA[0:63]	I/O	Internal Pull-up	**Data Bus.** Byte 0: D[0:7] - DH[0:7] Byte 1: D[8:15] - DH[8:15] Byte 2: D[16:23] - DH[16:23] Byte 3: D[24:31] - DH[24:31] Byte 4: D[32:39] - DL[0:7] Byte 5: D[40:47] - DL[8:15] Byte 6: D[48:55] - DL[16:23] Byte 7: D[56:63] - DL[24:31]

60x Bus Interface Signals (Page 3 of 3)

Signal Name	I/O	Type	Description
SYS_DATAP[0:7]	I/O		**Data Parity Bus.** Represents one bit of odd parity for each of the eight bytes of the data bus. Odd parity means that an odd number of bits, including the parity bit, are driven high. The signal assignments correspond to the following: DP[0]: D[0:7] DP[4]: D[32:39] DP[1]: D[8:15]) DP[5]: D[40:47] DP[2]: D[16:23] DP[6]: D[48:55] DP[3]: D[24:31] DP[7]: D[56:63]
Data Transfer Termination Signals			
$\overline{\text{SYS\_TA}}$	I/O	Internal Pull-up	**Transfer Acknowledge.** **Output:** Indicates a single beat of data transfer between device and a master on the 60x bus. For read transfers, indicates the data bus is valid with read data and the master must latch it in. For writes, indicates device has latched in write data from the data bus. Device asserts the signal for each beat in a burst transfer. **Input:** Indicates a single beat of data transfer has occurred. The device arbiter uses this signal and the address transfer attribute signals to determine the end of the data bus tenure.
$\overline{\text{SYS\_TEA}}$	I/O	Internal Pull-up	**Transfer Error Acknowledge.** **Output:** Indicates device has detected an error condition and that a machine check exception is desired. Assertion of this signal terminates the current data bus tenure. Device can be set up to transform any $\overline{\text{SYS\_TEA}}$ to normal $\overline{\text{SYS\_TA}}$ with machine check condition signaling on SYS_MACHK0 or SYS_MACHK1. **Input:** Informs the device's 60x bus arbiter that the current data bus tenure has been terminated.
Miscellaneous Signals			
$\overline{\text{SYS_MACHK}}$0 $\overline{\text{SYS_MACHK}}$1	O		**Machine Check.** Indicates the device has detected an error condition and a machine check exception is desired.
$\overline{\text{CHKSTOP}}$	O	Open Drain	**Checkstop.** Indicates the device has detected a non-recoverable error condition and has entered checkstop state.
$\overline{\text{SYS_HRESET}}$0 $\overline{\text{SYS_HRESET}}$1	O		**Hard Reset [0:1].** Indicates the device or card associated with this signal must initiate a complete hard reset. All outputs should be released to tri-state. Duration of reset, except for device hardware system reset, is controlled by software.
$\overline{\text{SYS_SRESET}}$0 $\overline{\text{SYS_SRESET}}$1	O		**Soft Reset [0:1].** Indicates the processor connected to this signal will take a reset exception. Occurs following a write to the CPU soft reset register (SRST) that has the appropriate bit set.
SYS_TBE	O		**Timebase Enable.** Indicates the processor time bases should continue counting. Reflects bit 12 of the UCTL[12] register 'x FF00 1000'.
POWERGOOD	I	Internal Pull-up	**System Normal operation** when up =1 **General System Reset** when down=0
$\overline{\text{INT}}$1	O		**Interrupt.** Interrupt generated after writing a '1' in the IT_ADD_SET interrupt register. This interrupt can be used by an external interrupt controller. The writing can be made from the CPU in configuration mode or from the PCI-64 bus. Only the PowerPC CPU can reset the interrupt by writing a '1' in the IT_ADD_RESET interrupt reset register.
$\overline{\text{INT}}$2	O		**Interrupt.** Indicates the end of the DMA data transfer. Corresponds to assertion of bit 4 in the GSCR status register.
GPIO0, GPIO1, GPIO2	I/O		**I/O.** General purpose I/O signals.
$\overline{\text{DLK}}$	O		**Deadlock.** Asserted when processor range of address is out of the non-deadlock zone. An address $\overline{\text{SYS\_ARTRY}}$ is sent to the PowerPC when $\overline{\text{DLK}}$ is set.
$\overline{\text{NODLK}}$	I		**Deadlock Disable.** Used only when the deadlock address range checking is programmed. Asserted (=0), it disables checking Tie to high level (=1) the deadlock checking can be performed.

Memory Interface Signals

Signal Name	I/O	Type	Description
MDATA[0:63]	I/O		Memory Data
MDATA[64:71]	I/O		Memory Data ECC bits
MADDR0_ODD	O		Memory Address bit 0 for Odd DIMMs
MADDR0_EVEN	O		Memory Address bit 0 for Even DIMMs
MADDR[1:12]	O		Memory Address bits 1 to 12
$\overline{\text{SDCS}}$[0:7]	O		SDRAM: Chip Select 0 to 7
$\overline{\text{SDCS}}$[8:15] or SDDQM[0:7]	O		Programmable I/Os by setting bit 11 of the MCCR register: 0: SDRAM: Chip Select 8 to 15 1: SDRAM: Chip Data Mask 0 to 7
SDCKE[0: 7]	O		SDRAM Clock Enable (height signals with same shape for buffering issues)
$\overline{\text{WE}}$[0:3]	O		Memory Write Enable: (four signals with same shape for buffering issues)
$\overline{\text{SDRAS}}$[0: 3]	O		SDRAM $\overline{\text{RAS}}$: Row Address Strobe (four signals with same shape for buffering issues)
$\overline{\text{SDCAS}}$[0: 3]	O		SDRAM $\overline{\text{CAS}}$: Column Address Strobe (four signals with same shape for buffering issues)
BS[0:1]	O		SDRAM BS: Internal Bank Select
Signals to be used with Texas Instruments ALVCH162268 MUX: 12 to 24-bit registered bus exchanger (For more information see http://www.ti.com/sc/)			
$\overline{\text{MUX\_OEA}}$ $\overline{\text{MUX\_OEB}}$	O		Output Enable of Data to Device - Port A of the external MUX Output Enable of Data to Memory - Port B
$\overline{\text{MUX\_CLKEN1B}}$ $\overline{\text{MUX\_CLKEN2B}}$	O		Clock Enable of Data sent to the Memory (Two signals with same shape for buffering issues)
$\overline{\text{MUX\_CLKENA1}}$ $\overline{\text{MUX\_CLKENA2}}$	O		Clock Enable of Data sent to Device. Clock A1 the first part of the data is stored in the external MUX controller, and on clock A2 full transfer is made.
$\overline{\text{MUX\_SEL}}$	O		Control the MUX circuit of the external MUX controller

PCI-32 Bus Interface Signals (3.3V - compliant 5V)

Signal Name	I/O	Type	Description
P_ADL[0:31]	I/O		32-bit Multiplexed Address/Data: A write operation is defined as the transfer of data from the PCI bus master to a PCI slave device on the PCI Bus.
P_CBE[0:3]	I/O		Bus Command/Byte Enable
P_DEVSEL	I/O		Device Select
P_FRAME	I/O		Cycle Frame: Driven by the current master to indicate the beginning and duration of an access.
P_IRDY	I/O		Initiator Ready
P_ISA_MASTER	I	Pull-up	ISA Master on the PCI-32 bus: Indicates that the CPC710 must automatically P_DEVSEL the current Memory transfer and must not translate the PCI address before sending the address to memory. **Warning: To become inactive this signal have to be connected to GND = 0**
P_LOCK	I		Reserved for future usage.It is recommended to tie up this signal
P_MEMACK	O		Memory Acknowledge: Indicates device has flushed all CPU to PCI-32 bus buffers and any CPU access to PCI is being SYS_ARTRYed.
P_MEMREQ	I	Pull-up	Memory Request: Indicates a PCI device accessing system memory has a potential deadlock and requests device flush all posted CPU to PCI buffers and ARTRY all PCI-32 bus transfers from the 60x bus.
P_PAR	I/O		Parity Bit
P_GNT[0:6]	O		PCI-32 Bus Grants
P_REQ[0:6]	I	Pull-up	PCI-32 Bus Requests
P_RST	O		PCI-32 Bus Reset
P_PERR	I/O		PCI-32 Data Parity Error
P_SERR	I/O		PCI-32 System Parity Error: Reports parity errors on address, special cycle data, or systems.
P_STOP	I/O		Stop: Asserted by the target to request the master to stop current transaction.
P_TRDY	I/O		Target Ready: Asserted by the target when ready to receive data.
P_CFG[0:2]	O		Configuration Bit Reflect: Reflects PCI-32 configuration bits 13-11 in the CONFIG ADDR register. Set to zero when bits 14 or 15 are on.

PCI-64 Bus Interface Signals (3.3V - compliant 5V)

Signal Name	I/O	Type	Description
G_ADH[0:31]	I/O		32-bit Multiplexed Address/Data Higher Part: In the address phase when G_REQ64 is asserted, these bits are the upper part of 64 bit address AD[63:32]. During data phase an additional 32-bits of data are transferred when G_REQ64 and G_ACK64 are both asserted. High Z when unused
G_ADL[0:31]	I/O		32-bit Multiplexed Address/Data Lower Part: A write operation is defined as the transfer of data from the PCI bus master to a PCI slave device on the PCI Bus.
G_ACK64	I/O		Acknowledge 64-Bit Transfer
G_REQ64	I/O		Request 64-Bit Transfer - External Pull up required
G_PAR64	I/O		Parity Upper Double Word
G_CBE[0:7]	I/O		Bus Command/Byte Enable
G_DEVSEL	I/O		Device Select
G_IDSEL	I	Pull-up	Initialization Device Select: Used as chip select during configuration.
G_FRAME	I/O		Cycle Frame
G_IRDY	I/O		Initiator Ready
G_LOCK	I		Reserved for future usage..It is recommended to tie up this signal
G_PAR	I/O		Parity Bit
G_CFG[0:2]	O		PCI-64 Configuration: Reflects PCI-64 Configuration bits 13-11 in the CONFIG ADDR register. Set to Zero when bit 15 or 14 are on.
G_GNT[0:7]	O		PCI-64 Bus Grant
G_REQ[0:7]	I	Pull-up	PCI-64 Bus Requests: G_REQ[2] is sampled at Reset, to select arbitration on the PCI-64 bus. The arbitration can be made by the device (G_REQ[2]=1) or by external circuit (=0). In case of external arbitration, the request is send to PCI from G_GNT[1] and the grant from the external arbiter is received on pin G_REQ[1].
G_RST	I/O		Reset PCI-64 Bus: External pull-up required. Input: Replicated on G_RESETOUT when programmed Output: Activated by the CPC710 at Powerup or by programming
G_PERR	I/O		PCI-64 Data Parity Error
G_SERR	I/O		PCI-64 System Parity Error
G_STOP	I/O		Stop: Asserted by the target to request the master to stop the current transaction.
G_TRDY	I/O		Target Ready: Asserted by the target when ready to receive data.
G_INTA	O	Open Drain	PCI-64 Interrupt A
G_INTB	O	Open Drain	PCI-64 Interrupt B
G_INTC	O	Open Drain	PCI-64 Interrupt C
G_INTD	O	Open Drain	PCI-64 Interrupt D
G_ARB	O		PCI-64 Arbiter: Asserted (=1) when the device is the PCI-64 arbiter
G_RESETOUT	O		Local Reset: Asserted by PCI-64 reset and special conditions

Test and Clock Signals

Signal Name	I/O	Internal Type	Description
SYS_CLK	I		**System Reference Clock.** It is provided to the CPC710 and used as a clock for the 60X bus and thus the PowerPC processor and for the SDRAM signals that are all synchronous. This clock is not synchronized with the PCI-32 and the PCI-64 independent clocks.
PCI_CLK	I		Main Clock Input for the PCI-32 bit bridge: up to 33 MHz
PCG_CLK	I		Main Clock Input for the PCI-64 bit bridge: up to 66 MHz
PLL_LOCK	I		Output indicating a locked state for the PLL
PLL_TUNE0	I	Pull-down	Loop stability tuning control of the PLL (Default =0)
PLL_TUNE1	I	Pull-up	Loop stability tuning control of the PLL (Default =1)
$\overline{\text{PLL\_RESET}}$	I	Pull-up	Reset and Bypass mode enable of the PLL
PLN_RTC_CLOCK	I		Time base enable for refresh DRAM cycles. It is recommended to have a Clock in the 7.8125 MHz range (128ns period) on input pin PLN_RTC_CLOCK. This clock is necessary for the DRAM refresh cycles and other internal timers such as soft reset time, software power on reset time, bus timeout and SDRAM initialization phase time.
CE1_A	I	Pull-up	Reserved
CE1_B	I	Pull-up	Reserved
CE1_C1	I	Pull-up	Reserved
CE1_C2	I	Pull-up	Reserved
DI1	I	Pull-up	Reserved
DI2	I	Pull-up	Reserved
CE0_IO	I	Pull-down	Reserved
CE0_TEST	I	Pull-down	Reserved
RI	I		Reserved. Must be set to 1.
SCAN_GATE	I	Pull-down	Reserved
TCK	I	Pull-up	JTAG Clock
TDI	I	Pull-up	JTAG Data Input
TDO	O	Pull-up	JTAG Test Data Output
TMS	I	Pull-up	JTAG Test Mode select
$\overline{\text{TRST}}$	I	Pull-up	Asynchronous JTAG Reset
TESTIN	I	Pull-up	Reserved. Must be set to 0.
$\overline{\text{CE\_TRST}}$	I	Pull-up	JTAG Compliance Enable: Down level (=0): Isolate the JTAG from the system reset signal POWERGOOD. Up level (=1): the POWERGOOD going to 0, Resets the JTAG.
TESTOUT	O		Reserved
VDDA	I		Analog V_{DD} for the PLL. Filtering on this 3.3V power supply is necessary to avoid problems with the on-chip PLL.

SIO Signals

Signal Name	I/O	Type	Description
$\overline{\text{FLASH\_OE}}$	O		Output Enable: FLASH ROM.
$\overline{\text{FLASH\_WE}}$	O		Write Enable: FLASH ROM.
$\overline{\text{PRES_OE}}$0	O		Output Enable: Presence detect (PD) buffer 0.
$\overline{\text{PRES_OE}}$1	O		Output Enable: Presence detect (PD) buffer 1.
XADR_LAT	O		Latch Signal: For SIO address register.
XCVR_RD	O		Address Direction: SIO address bus.

Reserved Signals

Signal Name	I/O	Type	Description
RESERVED1	I		External Pull-Up recommended.
RESERVED2	O		External Pull-Up recommended.
RESERVED3	O		External Pull-Up recommended.
RESERVED4	I	Pull-Up	External Pull-Up recommended.
RESERVED5	I	Pull-Up	External Pull-Up recommended.
RESERVED6	O		External Pull-Up recommended.
RESERVED7	I	Pull-Up	External Pull-Up recommended.
RESERVED8	O		External Pull-up recommended

Registers

The registers for the device are specified in three regions. Except for the Standard PCI Configuration Space, which uses indirect addressing, all the registers can be defined in the upper 16 MB of the 4 GB address range.

Standard PCI Configuration Space (register number x‘00’ to x‘68’)

There are two sets of PCI Configuration Space registers; one for each PCI bridge. These registers are accessed by a R/W of the CFGD with the address of the target register of the corresponding PCI bus in the CFGA (Configuration Address register) which specifies the register number and operation to perform.

The table *Standard PCI Configuration Register List* on page 40 describes the Specific PCI Host Bridge Registers supported by the device.

The register PSBAR can be accessed and configured by the CPU or the PCI-64 bus through configuration cycles.

Each of these registers is described in detail on pages 40 through 63.

Specific PCI Host Bridges Space (BAR + x‘000F 6110’ to BAR +x‘000F 9810’)

There are two almost identical sets of registers, one for each PCI bridge that can be placed by the user in the upper 16MB of the System Memory. One BAR value (Base Address) has to be defined first for each PCI bridge; for example as shown in the following figure, BAR_PCI32=FF50 0000 and BAR_PCI64=FF40 0000. The register space for the PCI-32 or PCI-64 bridge can then be accessed by the CPU with the PCI corresponding base value loaded in the Base Address Register (BAR x‘FF20 0018’).
The differentiation between the PCI-64 or PCI-32 is made by enabling the corresponding bit in the Connectivity Configuration Register (CNFR x‘FF00 000C’).

The table *Specific PCI Host Bridge Register List* on page 64 describes the supported Registers.

Each of these registers is described in detail on pages 64 through 92.

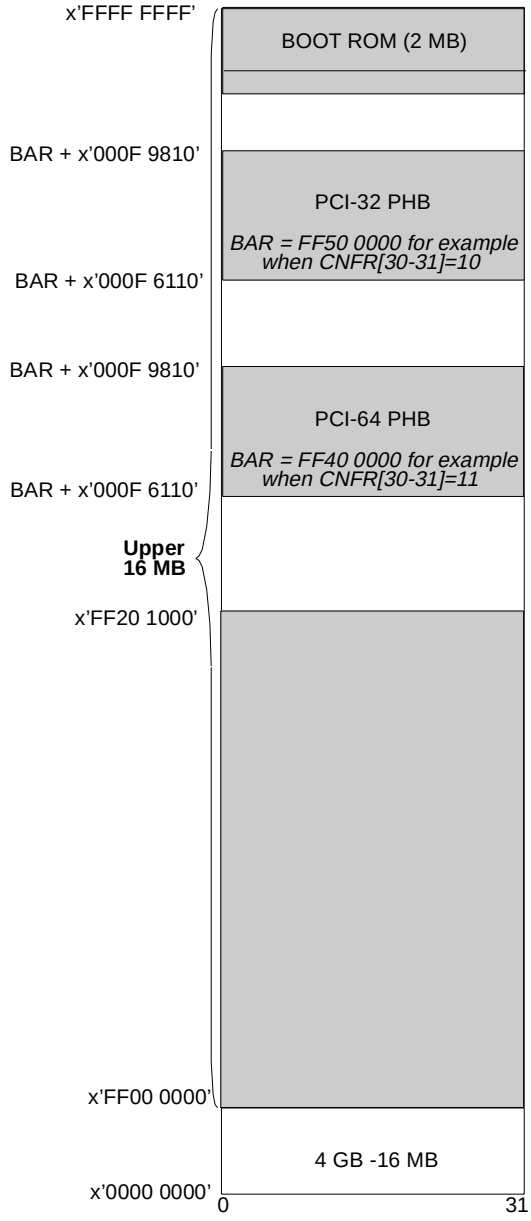
System Register Space (x‘FF00 0000’ to x‘FFFF FFFF’)

The upper 16 MB of the 4 GB address range is reserved for system support functions. The table *System Registers List* on page 93 describes the System Space Registers supported. These registers are defined as Big Endian unless otherwise noted. If the processor is operating in Little Endian mode, software must issue Load & Store reverse instructions to access these registers.

The device responds to all addresses listed in the table *System Registers List* on page 93 with a minimum granularity of 4 K blocks. Accesses to these registers must be single word accesses on word boundaries or unpredictable results may occur.

Each of these registers is described in detail on pages 93 through 130.
Shaded address ranges indicate areas where CPC710 will respond with TEA (addressing error is detected and logged in the System Error Status Register (SESR x’FF00 1060 bit 15 or bit 22)).

Register Map



System Space Address Map

Boot ROM Space

Specific PCI 32 Host Bridge Space

PSEA	CRR	SMBAR	CFGD	PSBAR	TIODLK
PCIDG	PR	SIBAR	PSSIZE		CSR
INTA	ACR	CTLRW		BPMDLK	PLSSR
PIBAR	MSIZE	CTLRO	BARPS	TPMDLK	
PMBAR	IOSIZE	CFGA		BIODLK	

Specific PCI-64 Host Bridge Space

PSEA	CRR	SMBAR	CFGD	BPMDLK	INT_SET
PCIDG	PR	SIBAR	PSSIZE	TPMDLK	CSR
INTA	ACR	CTLRW		BIODLK	PLSSR
PIBAR	MSIZE	CTLRO	BARPS	TIODLK	
PMBAR	IOSIZE	CFGA		IT_AD_RESET	

Device Specific Configuration Space PCIENB

Standard System Configuration Space

DCR	DID	BAR

DMA Space

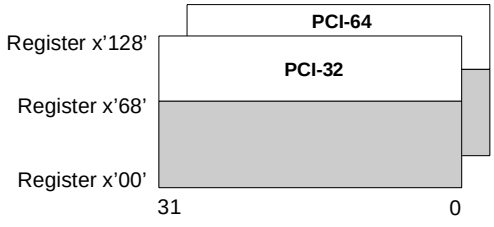
GSCR	XSSR	XTAR	XSCR	XWAR
GSSR	XPAR	GSCR	XSSR	XTAR
XSCR	XWAR	GSSR	XPAR	

Specific System Space

UCTL	ERRC	RGBAN2	AVDG	MCER1	MCER6
MPSR	SESR	GPDIR	MCCR	MCER2	MCER7
SIOC	SEAR	GPIN	MESR	MCER3	SIOR0
60XC	PGCHP	GPOUT	MEAR	MCER4	SIOR1
SRST	RGBAN1	ATAS	MCER0	MCER5	

Standard System Space

PIDR	CNFR	RSTR	SPOR



Standard PCI Configuration Space

Register number is specified in the CFGA (Configuration Address)

VID	SPI	HDRT	MINGNT	RETRY
DEVID	SUBC	**PSBAR**	MAXLT	DLKRETRY
CMND	CLASS		BUSNO	
STAT	CSIZE	INTLN	SUBNO	IT_ADD_SET
RID	LTIM	INTPIN	DISCNT	INT_RESET

Note: BOXED registers are key registers that define PCI bus configuration and register settings.

Standard PCI Configuration Register List

Relative Address	Name	Use	Page	Notes
00 to 01	VID	Vendor ID Register	41	1
02 to 03	DEVID	PCI Device ID Register	41	1
04 to 05	CMND	Command Register	42	
06 to 07	STAT	Status Register	44	2
08	RID	Revision ID	46	1
09	SPI	Standard Programming Interface	46	1
00A	SUBC	Sub-class Code	47	1
0B	CLASS	Base Class Code	47	1
0C	CSIZE	Cache Line Size	48	1
0D	LTIM	Latency Timer	49	
0E	HDRT	Header Type	50	1
0F	BIST	Built In Self Test	51	
10	PSBAR	System Base Address Register for PCI-64	52	4
3C	INTLN	Interrupt Line	53	
3D	INTPIN	Interrupt Pin	54	
3E	MINGNT	Minimum Grant	55	1
3F	MAXLT	Maximum Latency	56	1
40	BUSNO	Bus Number	57	
41	SUBNO	Subordinate Bus Number	58	
42	DISCNT	Disconnect Counter	59	
50	RETRY	Retry Counter	60	
51	DLKRETRY	Auto Retry Counter for access in Peripheral space with potential deadlock	61	3
64	IT_ADD_SET	Set PCI-64 Inter-Processor ($\overline{\text{INT1}}$) Interrupt.	62	4
68	INT_RESET	Reset of INTA,INTB,INTC,INTD on the PCI-64	63	4

1. Read-Only Register, write is ignored
2. Writes will only reset bits in this register; write data interpreted as 1 = reset, 0 = ignore
3. Only for PCI-32
4. Only for PCI-64

Specific PCI Host Bridge registers

Real Address	Name	Use	Page	Note
BAR + x'000F 6110	PSEA	PCI Slave Error Address	65	1
BAR + x'000F 6120'	PCIDG	PCI Diagnostic Register	66	
BAR + x'000F 7700'	INTACK	Interrupt Acknowledge Cycle	67	1
BAR + x'000F 7800'	PIBAR	PCI Base Address for I/O	68	
BAR + x'000F 7810'	PMBAR	PCI Base Address for Memory	69	
BAR + x'000F 7EF0'	CRR	Component Reset Register	70	
BAR + x'000F 7F20'	PR	Personalization Register	71	
BAR + x'000F 7F30'	ACR	Arbiter Control Register	73	
BAR + x'000F 7F40'	MSIZE	PCI Memory Address Space Size	74	
BAR + x'000F 7F60'	IOSIZE	PCI I/O Address Space Size	75	
BAR + x'000F 7F80'	SMBAR	System Base Address for PCI Memory	76	
BAR + x'000F 7FC0'	SIBAR	System Base Address for PCI I/O	77	
BAR + x'000F 7FD0'	CTLRW	Configuration Register R/W	78	
BAR + x'000F 7FE0'	CTLRO	Configuration Register R/O	79	1
BAR + x'000F 8000'	CFGA	CONFIG_ADDR	80	2
BAR + x'000F 8010'	CFGD	CONFIG_DATA	81	2
BAR + x'000F 8100'	PSSIZE	PCI to System CPU Address space Size	82	
BAR + x'000F 8120'	BARPS	CPU Base Address Register	83	
BAR + x'000F 8140'	PSBAR	System Base Address Register for PCI-32	84	3
BAR + x'000F 8200'	BPMDLK	Bottom of Peripheral Memory space with potential deadlock	85	
BAR + x'000F 8210'	TPMDLK	Top of Peripheral Memory space with potential deadlock	86	
BAR + x'000F 8220'	BIODLK	Bottom of Peripheral I/O space with potential deadlock	87	
BAR + x'000F 8230'	TIODLK	Top of Peripheral I/O space with potential deadlock	88	
BAR + x'000F 8300'	IT_ADD_RESET	PCI-64 Reset Interrupt ($\overline{INT}$1) Addressed Register	89	4
BAR + x'000F 8310'	INT_SET	Set of $\overline{G\_INTA}$, $\overline{G\_INTB}$, $\overline{G\_INTC}$, $\overline{G\_INTD}$ on PCI-64	90	4
BAR + x'000F 9800'	CSR	Channel Status Register	91	
BAR + x'000F 9810'	PLSSR	Processor Load/Store Status Register	92	

1. Read-Only Register, write is ignored
2. Little Endian registers
3. Only for PCI-32
4. Only for PCI-64

System Registers List (Page 1 of 2)

Address	Name	Use	Page	Notes
x'FF00 0000' to x'FF00 0007'		Reserved		
Standard System Registers				
x'FF00 0008'	PIDR	Physical Identifier Register	96	
x'FF00 000C'	CNFR	Connectivity Configuration Register	97	
x'FF00 0010'	RSTR	Connectivity Reset Register	98	
x'FF00 00E8'	SPOR	Software POR Register	99	
Specific System Registers				
x'FF00 1000'	UCTL	Universal Control Register	100	
x'FF00 1010'	MPSR	Multi-Processor Semaphore Register	102	
x'FF00 1020'	SIOC	System I/O Control	103	
x'FF00 1030'	ABCNTL	60x Arbiter Control Register	104	
x'FF00 1040'	SRST	CPU Soft Reset Register	106	
x'FF00 1050'	ERRC	Error Control Register	107	
x'FF00 1060'	SESR	System Error Status Register	108	
x'FF00 1070'	SEAR	System Error Address Register	110	
x'FF00 1080'		Reserved		
x'FF00 1100'	PGCHP	Chip program Register	111	
x'FF00 1110'	RGBAN1	Free Register 1	113	
x'FF00 1120'	RGBAN2	Free Register 2	114	
x'FF00 1130'	GPDIR	GPIO Direction Register	115	
x'FF00 1140'	GPIN	GPIO Input Register	116	
x'FF00 1150'	GPOUT	GPIO Output Register	117	
x'FF00 1160'	ATAS	Address Transfer Attribute for Snoop Reg	118	
x'FF00 1170'	AVDG	Device Diagnostic Register	120	
x'FF00 1174' to x'FF00 11FF'		Reserved		
x'FF00 1200'	MCCR	Memory Controller Control Register	122	
x'FF00 1210'		Reserved		
x'FF00 1220'	MESR	Memory Error Status Register	124	
x'FF00 1230'	MEAR	Memory Error Address Register	125	
x'FF00 1300'	MCER0	Memory Configuration Extent Register 0	126	
x'FF00 1310'	MCER1	Memory Configuration Extent Register 1	126	
x'FF00 1320'	MCER2	Memory Configuration Extent Register 2	126	
x'FF00 1330'	MCER3	Memory Configuration Extent Register 3	126	
x'FF00 1340'	MCER4	Memory Configuration Extent Register 4	126	
x'FF00 1350'	MCER5	Memory Configuration Extent Register 5	126	
x'FF00 1360'	MCER6	Memory Configuration Extent Register 6	126	
x'FF00 1370'	MCER7	Memory Configuration Extent Register 7	126	
x'FF00 1400'	SIOR0	SIO Register 0 (DIMM PDs)	128	

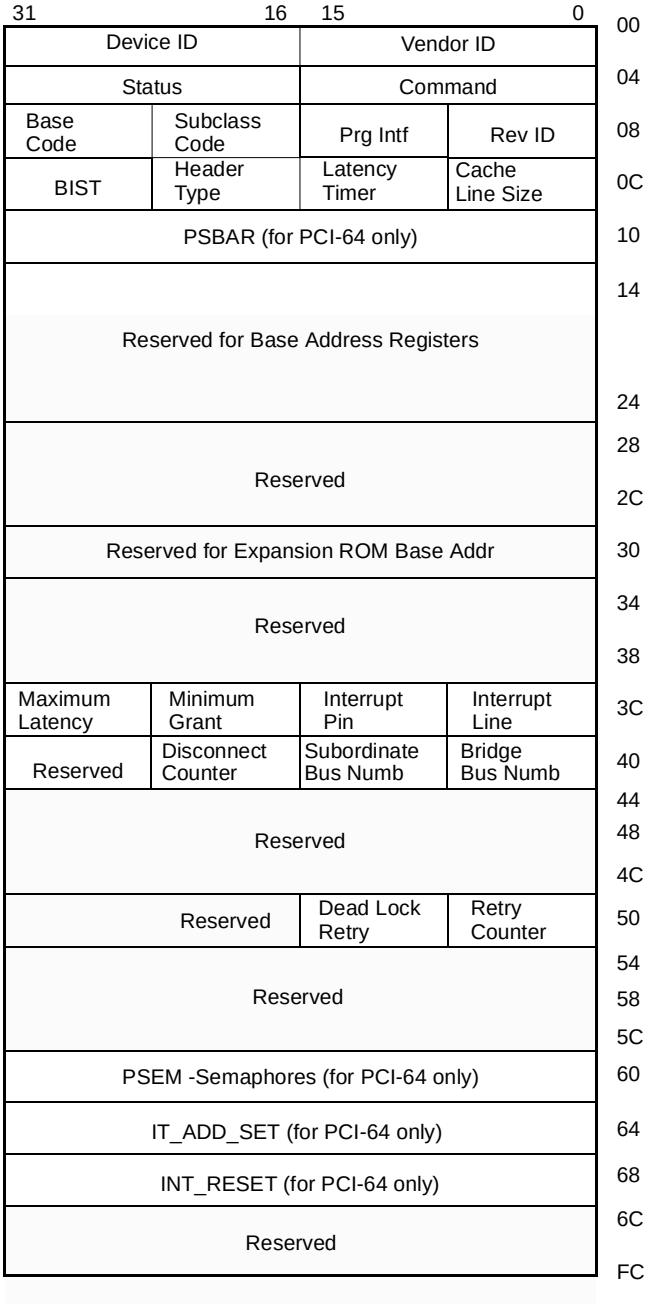
System Registers List (Page 2 of 2)

Address	Name	Use	Page	Notes
x'FF00 1410'		Reserved		
x'FF00 1420'	SIOR1	SIO Register 1 (Planar, DIMM, CPU, etc.)	129	
x'FF00 1424 to x'FF00 1FFF'		Reserved		
x'FF00 2000 to x'FF17 FFFF'		Reserved		
DMA Registers: User Privilege				
x'FF18 0000 to x'FF1C 001F'		Reserved		
x'FF1C 0020'	GSCR	Global Control Register (user)	130	
x'FF1C 0030'	GSSR	Global Status Register (user)	131	
x'FF1C 0040'	XSCR	Transfer Control Register (user)	132	
x'FF1C 0050'	XSSR	Transfer Status Register (user)	133	
x'FF1C 0070'	XPAR	Transfer PCI Address Register (user)	135	
x'FF1C 0090'	XWAR	Transfer Write Back Address Register (user)	136	
x'FF1C 00A0'	XTAR	Transfer Translated Address Register (user)	137	
x'FF1E 0020'	GSCR	Global Control Register (priv)	130	
x'FF1E 0030'	GSSR	Global Status Register (priv)	131	
x'FF1E 0040'	XSCR	Transfer Control Register (priv)	132	
x'FF1E 0050'	XSSR	Transfer Status Register (priv)	133	
x'FF1E 0070'	XPAR	Transfer PCI Address Register (priv)	135	
x'FF1E 0090'	XWAR	Transfer Write Back Address Register (priv)	136	
x'FF1E 00A0'	XTAR	Transfer Translated Address Register (priv)	137	
x'FF1E 00A4' to x'FF1F FFFF'		Reserved		
System Standard Configuration Registers				
x'FF20 0000'	DCR	Device Characteristics Register	138	
x'FF20 0004'	DID	Device ID Register	139	
x'FF20 0008' to x'FF20 0014'	Reserved			
x'FF20 0018'	BAR	Base Address Register	140	
x'FF20 0020' to x'FF20 0FFF'	Reserved			
Device Specific Configuration Space				
x'FF20 1000'	PCIENB	PCI BAR Enable Register	141	
x'FF20 1004' to x'FFDF FFFF'		Reserved		
BOOT ROM				
x'FFE0 0000' to x'FFFF FFFF'	IPLROM	FLASH ROM: Up to 2 MB		

Standard PCI Configuration Registers

The following registers are defined as Little Endian (LE) ordering. Therefore, for software running in Big Endian (BE) mode, any access to these registers (that is not a single byte access) must utilize the load/store byte reversal instructions when accessing these registers. Software running in LE mode can use the normal load and store instructions. There is one set of registers for the PCI 32 bit and one set for the PCI-64 bit. The relative address (or register number) of these registers is specified in the CFGA (Configuration Address).

PCI Configuration Space



Standard PCI Configuration Register List

Relative Address	Name	Use	Page	Notes
00 to 01	VID	Vendor ID Register	41	1
02 to 03	DEVID	PCI Device ID Register	41	1
04 to 05	CMND	Command Register	42	
06 to 07	STAT	Status Register	44	2
08	RID	Revision ID	46	1
09	SPI	Standard Programming Interface	46	1
00A	SUBC	Sub-class Code	47	1
0B	CLASS	Base Class Code	47	1
0C	CSIZE	Cache Line Size	48	1
0D	LTIM	Latency Timer	49	
0E	HDRT	Header Type	50	1
0F	BIST	Built In Self Test	51	
10	PSBAR	System Base Address Register for PCI-64	52	4
3C	INTLN	Interrupt Line	53	
3D	INTPIN	Interrupt Pin	54	
3E	MINGNT	Minimum Grant	55	1
3F	MAXLT	Maximum Latency	56	1
40	BUSNO	Bus Number	57	
41	SUBNO	Subordinate Bus Number	58	
42	DISCNT	Disconnect Counter	59	
50	RETRY	Retry Counter	60	
51	DLKRETRY	Auto Retry Counter for access in Peripheral space with potential deadlock	61	3
64	IT_ADD_SET	Set PCI-64 Inter-Processor ($\overline{\text{INT1}}$) Interrupt.	62	4
68	INT_RESET	Reset of INTA,INTB,INTC,INTD on the PCI-64	63	4

1. Read-Only Register, write is ignored
2. Writes will only reset bits in this register; write data interpreted as 1 = reset, 0 = ignore
3. Only for PCI-32
4. Only for PCI-64

Vendor ID (VID)

This register identifies the device manufacturer.

Reset Value x'1014'

Address x'00'

Access Type Read Only

Bit(s)	Description
15 - 0	**Vendor Identification Number** Value = x'1014'; (x'14' for address 00 and x'10' for address 01)

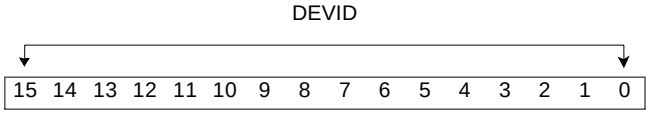
Device ID (DEVID)

This register identifies a particular device.

Reset Value PCI-32 x'0105'
PCI-64 x'00FC'

Address x'02'

Access Type Read Only

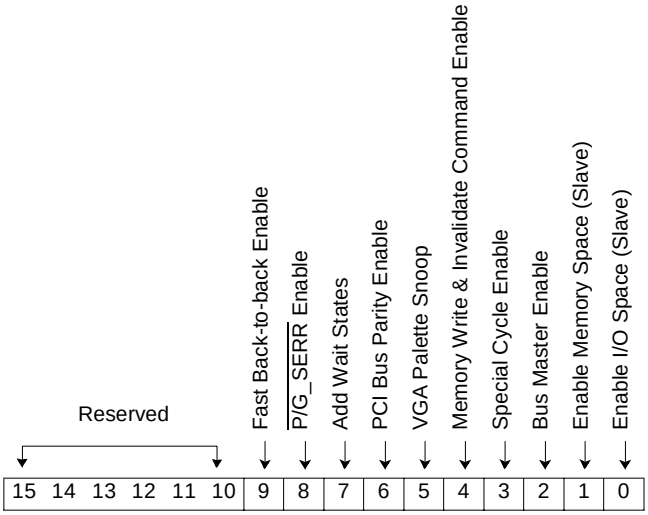


Bit(s)	Description
15 - 0	**Device Identification Number** Value: x'0105' for PCI-32 (x'05' for address 02 and x'01' for address 03) Value: x'00FC' for PCI-64 (x'FC' for address 02 and x'00' for address 03)

PCI Command (CMND)

This register provides control over the bridge's PCI behavior.

Reset Value	PCI-32	x'0000'
	PCI-64	x'0000'
programming example	PCI-32	x'0156'
	PCI-64	x'0356'
Address	x'04'	
Access Type	Read/Write	



Bit(s)	Description
15 - 10	**Reserved**
9	**Fast Back-to-back Enable** 0: Disabled 1: PCI Bridge issues fast back-to-back transfers without regard to which target is being addressed, providing that the previous transaction was a write. Note: This bit should be set if all slaves on the PCI bus support this capability.
8	**SERR Enable** 0: PCI Bridge will not assert $\overline{\text{P/G\_SERR}}$ upon detecting an error. 1: PCI bridge will assert $\overline{\text{P/G\_SERR}}$ for PCI address parity error
7	**Add Wait States**. Read Only. Always returns 0. Device does not support address data stepping.
6	**PCI Bus Parity Enable** 0: Device will disable all parity checking on the PCI bus 1: Device will detect and report parity errors on the PCI bus
5	**VGA Palette Snoop**. Read Only. Always returns 0. Device is not VGA compatible.
4	**Memory Write & Invalidate Command Enable** 0: Device does not generate this type of cycle. 32-byte transfers use the Memory Write command. 1: Device generates this cycle as a master for any 32-byte transfer.
3	**Special Cycle Enable**. Read Only. Always returns 0. Device will not respond to Special Cycle commands.

Bit(s)	Description
2	**Bus Master Enable** 0: PCI Bridge master capability is disabled. 1: PCI Bridge performs as a PCI master for accesses to its address spaces.
1	**Enable Memory Space (Slave)** 0: PCI Bridge will not respond to memory accesses on the PCI bus 1: PCI Bridge will respond to memory accesses on the PCI bus
0	**Enable I/O Space (Slave)** 0: PCI Bridge will not respond to IO accesses on the PCI bus 1: PCI Bridge will respond to IO accesses on the PCI bus except for PREP mode (see note below)

Note: I/O cycles for Slave as defined in bit 0 are not decoded by the CPC710 when the address mapping is in PREP mode. See PREP mode definition.

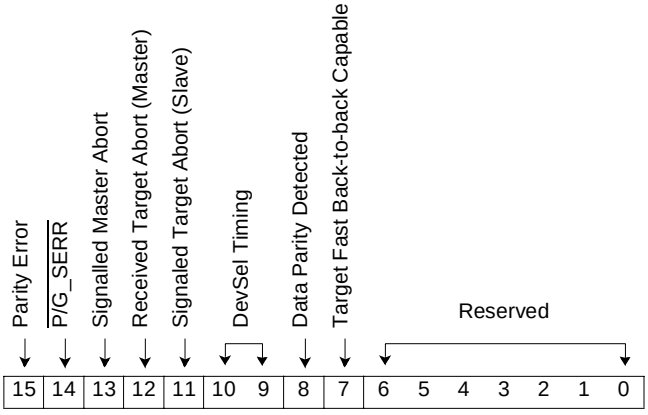
PCI Status (STAT)

This register records status and error information from PCI bus transfers. Reads from this register behave normally. Writes to this register are restricted, in that software cannot set any bit in this register, only reset. Additionally, to reset a bit, software must write a 1 to the corresponding bit location. For example, to reset only bit 14, software must write '0100 0000 0000 0000'b to this register.

Reset Value x'0280'

Address x'06'

Access Type Read/Write



Bit(s)	Description
15	**Parity Error** 0: No Error 1: PCI Bridge has detected a parity error; bit set even if parity checking is disabled.
14	**Signaled System Error (P/G_SERR)** 0: No Error 1: PCI Bridge has asserted SERR due to an address parity error.
13	**Signaled Master Abort** 0: No Error 1: PCI Bridge has issued a master abort.
12	**Received Target Abort (Master)** 0: No Error 1: PCI Bridge has detected a target abort for one of its transactions.
11	**Signaled Target Abort (Slave)** 0: No Error 1: PCI Bridge as a slave has issued a target abort.
10 - 9	**DevSel Timing**. Read Only 01: PCI Bridge responds with Medium timing on P/G_DEVSEL signal.
8	**Data Parity Detected** 0: No Error 1: This bit is set if the following 3 conditions are met: I) PCI Bridge asserted, or observed P/G_PERR signal on PCI bus II) PCI Bridge acting as master III) Bit 6 of Command Register set

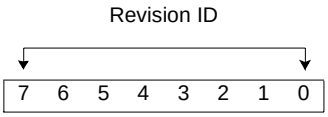
Bit(s)	Description
7	**Target Fast Back-to-back Capable**. Read Only. Always returns a 1 to indicate that the PCI Bridge as a target will accept fast back-to-back transfers when the transfers are not to the same device.
6 - 0	**Reserved**

Revision ID (RID)

Reset Value x'00'

Address x'08'

Access Type Read Only



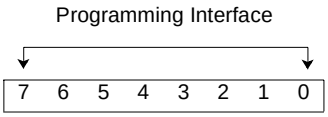
Bit(s)	Description
7 - 0	Provides an extension to the PCI Device ID register. Device always responds with x'00' for reads from this register.

Programming Interface (SPI)

Reset Value x'00'

Address x'09'

Access Type Read Only



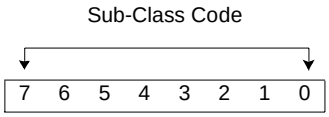
Bit(s)	Description
7 - 0	Defines a specific register-level programming interface. Device always responds with x'00' for reads from this register.

Sub-Class Code (SUBC)

Reset Value x'00'

Address x'0A'

Access Type Read Only



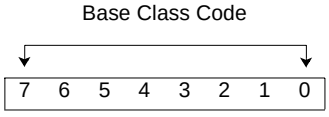
Bit(s)	Description
0 - 15	Specifically identifies a particular function of the Base Class Code register. Device always responds with x'00' for reads to indicate a HOST type of bridge device.

Base Class Code (CLASS)

Reset Value x'06'

Address x'0B'

Access Type Read Only



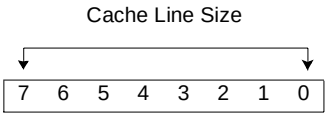
Bit(s)	Description
7 - 0	Classifies the type of function this device performs. Device always responds with x'06' for reads to indicate a Bridge device.

Cache Line Size (CSIZE)

Reset Value x'08'

Address x'0C'

Access Type Read Only



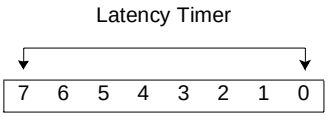
Bit(s)	Description
7 - 0	Specifies the cache line size in units of 32-bit words. Device always responds with x'08' for reads to indicate that device will always disconnect from any PCI master burst operation that crosses a 32-byte boundary.

Latency Timer (LTIM)

Reset Value x'00'

Address x'0D'

Access Type Read/Write



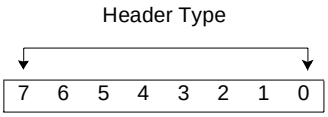
Bit(s)	Description
7 - 0	Provides bus masters with a minimum guaranteed time slice on the PCI bus. The value programmed into this register is the minimum number of PCI bus clocks that a master can own the PCI bus starting from the cycle that FRAME is activated. This register is set to X'00' at reset.

Header Type (HDRT)

Reset Value x'00'

Address x'0E'

Access Type Read Only



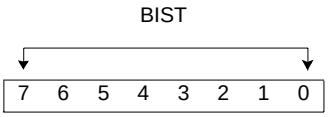
Bit(s)	Description
7 - 0	Specifies the layout of bytes x'10' through x'3F' in the configuration header and whether or not a particular device contains multiple functions. Device always responds with x'00' to reads to indicate Layout 0. Writes to this register are ignored.

Built in Self Test (BIST)

Reset Value x'00'

Address x'0F'

Access Type Read Only



Bit(s)	Description
7 - 0	Provides status and control for a Built-in Self Test which device does not support. Device responds with x'00' to reads from this register and ignores writes.

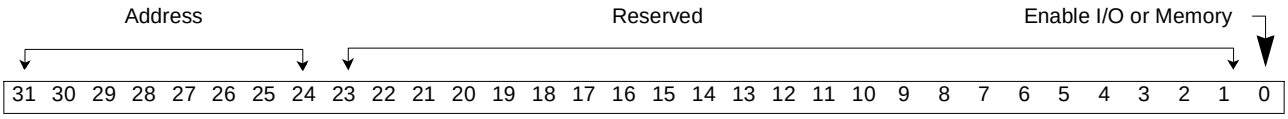
System Base Address Register for PCI-64 (PSBAR)

For PCI-64 only, the function is the same as that for *System Base Address Register for PCI-32 (PSBAR)* on page 84.

Reset Value x'0000 0000'

Address x'10'

Access Type Read/Write



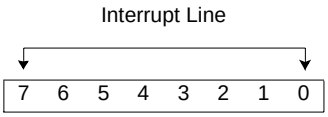
Bit(s)	Description
31 - 24	System Base Address for PCI-64. Contains the upper bits of the System Base address that memory is mapped to.
23 - 1	Reserved
0	**Enable Memory or IO Space (**copy of the bit 7 of the PSSIZE Register) 0: Memory Space 1: IO Space

Interrupt Line (INTLN)

Reset Value x'00'

Address x'3C'

Access Type Read Only



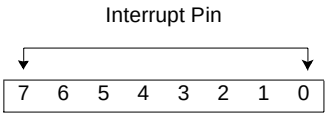
Bit(s)	Description
7 - 0	Indicates interrupt routing information for devices that implement an interrupt. The PCI bridge logic does not generate interrupts and therefore this register is not implemented. Device responds with x'00' to reads from this register and ignores Writes.

Interrupt Pin (INTPIN)

Reset Value x'00'

Address x'3D'

Access Type Read Only



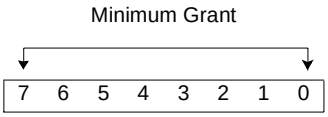
Bit(s)	Description
7 - 0	Specifies which particular interrupt pin, INTA, INTB, INTC, or INTD, is used to generate interrupts. Since the PCI bridge does not generate any interrupts, Device responds with x'00' to reads from this register and ignores writes.

Minimum Grant (MINGNT)

Reset Value x'00'

Address x'3E'

Access Type Read Only



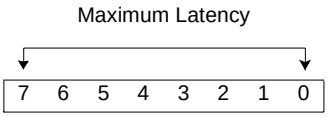
Bit(s)	Description
7 - 0	Specifies the length of a device's burst period in 0.25 µsecs. Device has no specific requirements and therefore always responds with x'00'.

Maximum Latency (MAXLT)

Reset Value x'00'

Address x'3F'

Access Type Read Only



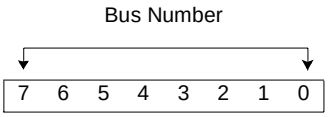
Bit(s)	Description
7 - 0	Specifies how often the device needs to gain access to the PCI bus in 0.25 µs. Device has no specific requirements and therefore always responds with x'00'.

Bus Number (BUSNO)

Reset Value x'00'

Address x'40'

Access Type Read/Write



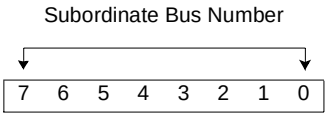
Bit(s)	Description
7 - 0	Contains the assigned bus number for this bridge. Device uses this number to determine what action to take for configuration cycles directed to this bridge. After reset, this register contains a value of x'00'.

Subordinate Bus Number (SUBNO)

Reset Value x'00'

Address x'41'

Access Type Read/Write



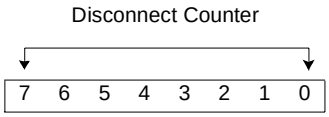
Bit(s)	Description
7 - 0	Specifies the largest bus number beneath this bridge. After reset, this register contains a value of x'00'.

Disconnect Counter (DISCNT)

Reset Value x'00'

Address x'42'

Access Type Read/Write



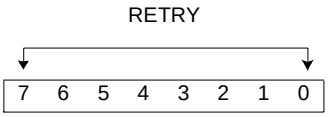
Bit(s)	Description
7 - 0	Device uses this register when acting as a target device as a time-out mechanism in burst operations. The value written to this register is multiplied by four and used to determine when the bridge should assert STOP#. After reset, this register contains x'00' which disables the timer. This counter is enabled only if bit 0 (for PCI 32) or bit 8 (for PCI-64) of AVDG Register is set (see *Diagnostic Register (AVDG)* on page 120) When time-out occurs the bit 9 of the CSR Register is set (see *Channel Status Register (CSR)* on page 91).

Retry Counter (RETRY)

Reset Value x'00'

Address x'50'

Access Type Read/Write



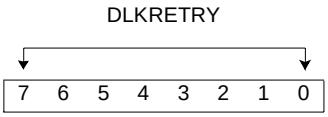
Bit(s)	Description
7 - 0	When the device is a Master on the PCI bus, this register is used as a time-out mechanism for continuous retries on the PCI bus. Whenever a retry occurs for a particular address, the PCI bridge logic increments (decrements) a counter. The 8-bit counter is reset whenever data is transferred. If the count reaches the value specified in this register, the PCI bridge logic will not retry the access and will report the Result by writing bit 5 of the PLSSR Register (see *Processor Load/Store Status Register (PLSSR)* on page 92 After reset, the register contains x'00' which disables the retry counter.

Deadlock Retry Counter (DLKRETRY) for PCI-32 Only

Reset Value x'00'

Address x'51'

Access Type Read/Write



Bit(s)	Description
7 - 0	Available only for the PCI-32, this 8-bit counter is used to limit the number of Retries in the case of an access in a deadlock area space defined with the BPMDLK/TPMDLK or BIODLK/TIODLK registers.

Set Addressed Interrupt (IT_ADD_SET)

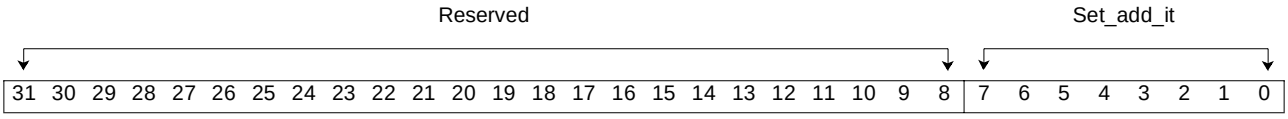
This is a Virtual Register. When addressed, the interrupt signal $\overline{\text{INT}}$1 is set (goes to 0). The SET can be done from the PCI-64 or from the PowerPC CPU in configuration mode.

Only the PowerPC CPU can reset the interrupt $\overline{\text{INT}}$1 by writing a "1" in the IT_ADD_RESET interrupt reset register.

Reset Value x'0000 0000'

Address x'64'

Access Type Write Only



Bit(s)	Description
31 - 8	**Reserved**
7 - 0	**Set_add_it** 1: Writing a 1 in one of these 8 bits SETS the interrupt signal $\overline{\text{INT}}$1 0: No action

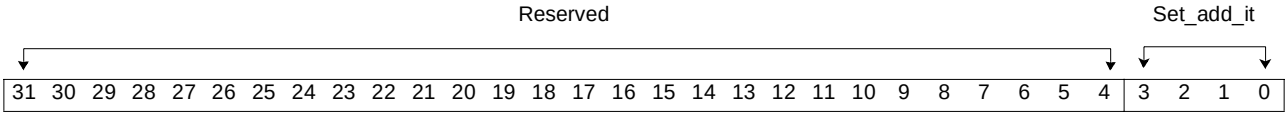
Reset PCI-64 Interrupt (INT_RESET)

Resets one of the posted interrupt $\overline{\text{G\_INTA}}$, $\overline{\text{G\_INTB}}$, $\overline{\text{G\_INTC}}$, $\overline{\text{G\_INTD}}$ on the PCI-64bit bus. Reset can be done from the PCI-64 or from the CPU in configuration mode. The CPU can only execute the SET of INTA, INTB, INTC, INTD when writing in Register INT_SET at address BAR + x'000F 8310'.

Reset Value x'0000 0000'

Address x'68'

Access Type Read/Write



Bit(s)	Description
31 - 4	**Reserved**
3 - 0	**RESET_Interrupts** 1: Resets the bit corresponding to one PCI-64 interrupt 0: No action Bit 0: $\overline{\text{G\_INTA}}$ Bit 1: $\overline{\text{G\_INTB}}$ Bit 2: $\overline{\text{G\_INTC}}$ Bit 3: $\overline{\text{G\_INTD}}$

Specific PCI Host Bridge Registers

Specific PCI Host Bridge Register List

Real Address	Name	Use	Page	Note
BAR + x'000F 6110	PSEA	PCI Slave Error Address	65	1
BAR + x'000F 6120'	PCIDG	PCI Diagnostic Register	66	
BAR + x'000F 7700'	INTACK	Interrupt Acknowledge Cycle	67	1
BAR + x'000F 7800'	PIBAR	PCI Base Address for I/O	68	
BAR + x'000F 7810'	PMBAR	PCI Base Address for Memory	69	
BAR + x'000F 7EF0'	CRR	Component Reset Register	70	
BAR + x'000F 7F20'	PR	Personalization Register	71	
BAR + x'000F 7F30'	ACR	Arbiter Control Register	73	
BAR + x'000F 7F40'	MSIZE	PCI Memory Address Space Size	74	
BAR + x'000F 7F60'	IOSIZE	PCI I/O Address Space Size	75	
BAR + x'000F 7F80'	SMBAR	System Base Address for PCI Memory	76	
BAR + x'000F 7FC0'	SIBAR	System Base Address for PCI I/O	77	
BAR + x'000F 7FD0'	CTLRW	Configuration Register R/W	78	
BAR + x'000F 7FE0'	CTLRO	Configuration Register R/O	79	1
BAR + x'000F 8000'	CFGA	CONFIG_ADDR	80	2
BAR + x'000F 8010'	CFGD	CONFIG_DATA	81	2
BAR + x'000F 8100'	PSSIZE	PCI to System CPU Address space Size	82	
BAR + x'000F 8120'	BARPS	CPU Base Address Register	83	
BAR + x'000F 8140'	PSBAR	System Base Address Register for PCI-32	84	3
BAR + x'000F 8200'	BPMDLK	Bottom of Peripheral Memory space with potential dead-lock	85	
BAR + x'000F 8210'	TPMDLK	Top of Peripheral Memory space with potential deadlock	86	
BAR + x'000F 8220'	BIODLK	Bottom of Peripheral I/O space with potential deadlock	87	
BAR + x'000F 8230'	TIODLK	Top of Peripheral I/O space with potential deadlock	88	
BAR + x'000F 8300'	IT_ADD_RESET	PCI-64 Reset Interrupt ($\overline{INT}$1) Addressed Register	89	4
BAR + x'000F 8310'	INT_SET	Set of $\overline{G\_INTA}$, $\overline{G\_INTB}$, $\overline{G\_INTC}$, $\overline{G\_INTD}$ on PCI-64	90	4
BAR + x'000F 9800'	CSR	Channel Status Register	91	
BAR + x'000F 9810'	PLSSR	Processor Load/Store Status Register	92	

1. Read-Only Register, write is ignored
2. Little Endian registers
3. Only for PCI-32
4. Only for PCI-64

PCI Slave Error Address Register (PSEA)

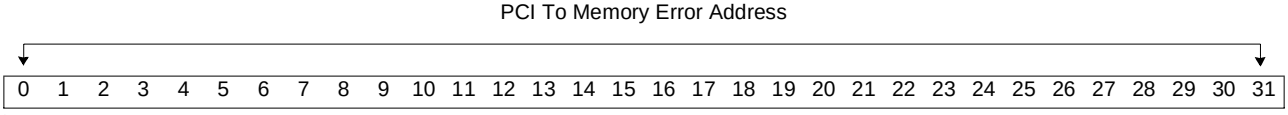
This register is used to log the PCI address when an error occurs during Device PCI slave transfer. See *PCI Master Error Handling* on page 195 for additional details.

This register is reset to zero after a POWERGOOD or when one of the bit RSTR[2] of the , “Connectivity Reset Register (RSTR)” Page 98 for PCI 32 or bit RSTR[3] for PCI 64 is forced to zero or from a, “Software Power On Reset Control Register (SPOR)” Page 99 reset.

Reset Value x’0000 0000’

Address BAR + x’000F 6110’

Access Type Read Only



Bit(s)	Description
0 - 31	**PCI To Memory Error Address.** Contains the address present on the PCI bus when an error occurs during a PCI transfer.

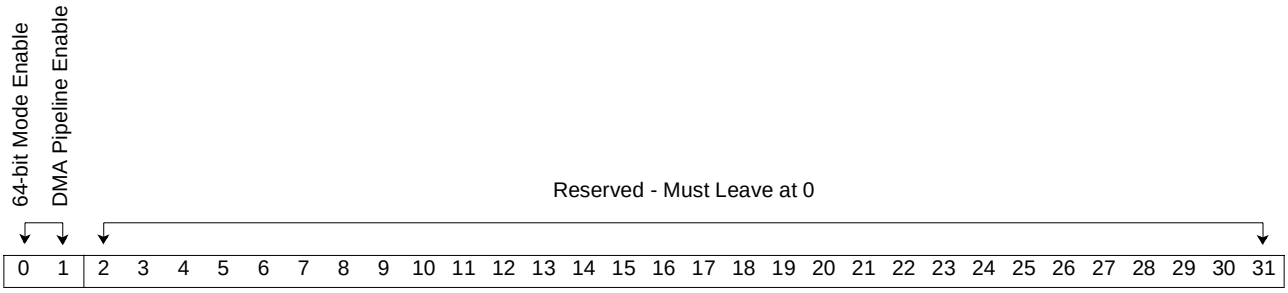
PCI Diagnostic Register (PCIDG)

This register contains two mode bits that are used for special modes of operation.

Reset Value PCI-32 x'4000 0000'
PCI-64 x'C000 0000'

Address BAR + x'000F 6120'

Access Type Read/Write



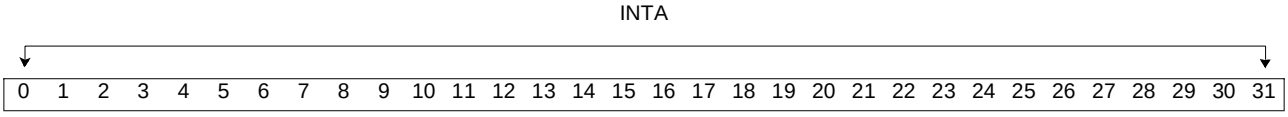
Bit(s)	Description
0	**64-bit Mode Enable (only for PCI 64)** 0: Operates as a 32-bit bridge. $\overline{\text{G\_REQ64}}$ and $\overline{\text{G\_ACK64}}$ never activated by device. 1: Operates as a 64-bit bridge. $\overline{\text{G\_REQ64}}$ always activated for device initiated transfers.
1	**DMA Pipeline Enable** 0: DMA transfers are NOT pipelined internal to device. Results in significantly lower bandwidth to PCI bus. 1: DMA transfers are pipelined internal to device for maximum bandwidth.
2 - 31	**Reserved**. Must be set to 0

Interrupt Acknowledge Cycle (INTACK)

A read to the INTACK register generates an Interrupt Acknowledge Cycle on the PCI bus. An Interrupt Acknowledge Transaction has no addressing mechanism and is implicitly targeted to the interrupt controller in the system. The vector is returned by the interrupt controller when TRDY is asserted.

Address BAR + x'000F 7700'

Access Type Read Only



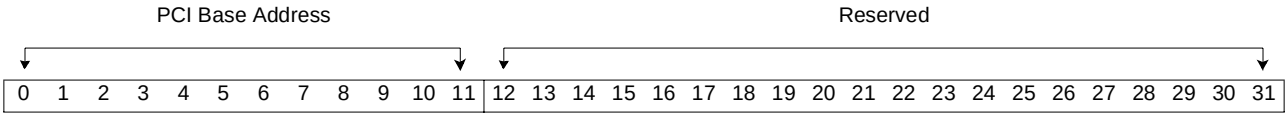
Bit(s)	Description
0 - 31	This register is a port through to the PCI bus. Writes to this register are ignored.

PCI Base Address for I/O (PIBAR)

Reset Value x'0000 0000'

Address BAR + x'000F 7800'

Access Type Read/Write



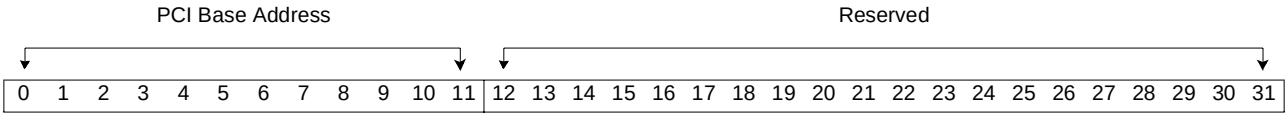
Bit(s)	Description
0 - 11	**PCI Base Address.** Contains the upper bits of the PCI base address that PCI I/O is mapped to. Note: Address must be aligned on boundary equal to size specified in PCI I/O Size register
12 - 31	**Reserved**

PCI Base Address for Memory (PMBAR)

Reset Value x'0000 0000'

Address BAR + x'000F 7810'

Access Type Read/Write



Bit(s)	Description
0 - 11	**PCI Base Address.** Contains the upper bits of the PCI base address that PCI Memory is mapped to. **Note**: Address must be aligned on boundary equal to size specified in PCI Memory Size register.
12 - 31	**Reserved**

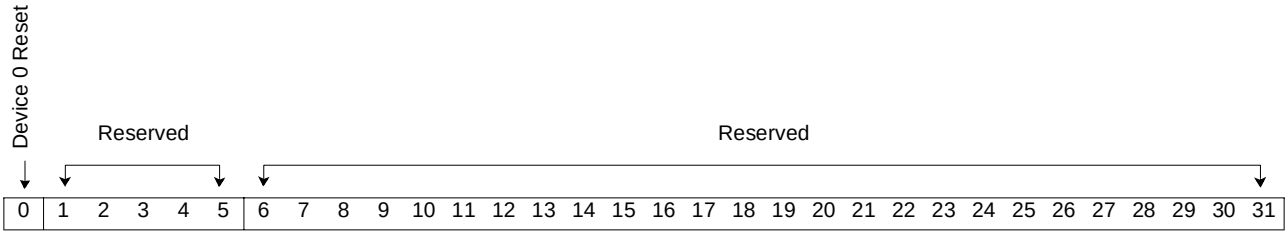
Component Reset Register (CRR)

This register provides software with a means to disable all devices on the PCI bus by writing a zero in bit 0.

Reset Value x'0000 0000'

Address BAR + x'000F 7EF0'

Access Type Read/Write



Bit(s)	Description
0	**Device 0 Reset** 0: Reset signal $\overline{\text{P\_RST}}$ or $\overline{\text{G\_RST}}$ is active 1: Reset signal inactive
1 - 5	**Reserved**. Must be left to 1
6 - 31	**Reserved**.

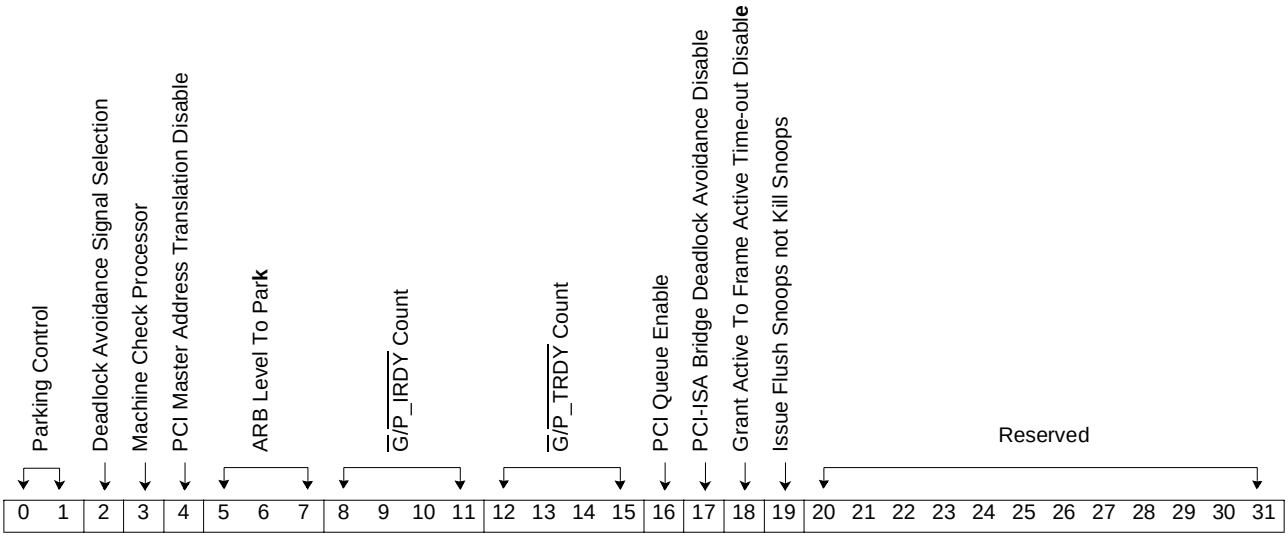
Personalization Register (PR)

This register provides additional programmability of the PCI Bridge logic.

Reset Value x'0000 0000'

Address BAR + x'000F 7F20'

Access Type Read/Write



Bit(s)	Description
0 - 1	**Parking Control** 0x: Bus is parked on device 10: MRU algorithm for parking 11: Park specified ARB level below in bits 5-7
2	**Deadlock Avoidance Signal Selection** (**PCI-32 Bridge Only**) To avoid deadlocks with PCI-ISA bridges on the PCI 32-bit bridge, the bridge must indicate to the device that a PCI access is about to occur before the P_GNT signal is activated. Any posted PCI 32-bit bus transfers must be flushed prior to activating the P_GNT signal and any accesses to the PCI 32-bit bus must be disabled after the GNT is given and continue disabled until the PCI access is complete and the P_GNT signal is removed. The PCI-ISA bridge must not grant the secondary ISA bridge until device has activated the P_GNT signal. Device provides two input signals for this purpose that are selectable with this bit. 0: Selects the P_REQ[5] signal. P_GNT[5] indicates buffers flushed and any PCI. Transfers will be disabled on the 60x bus until the P_REQ[5] signal is deactivated. 1: Selects the P_REQ signal. P_REQ indicates buffers flushed and any PCI. Transfers will be disabled on the 60x bus until the P_REQ signal is deactivated
3	**Machine Check Processor.** If an error is detected as a target during a PCI access operation, the device generates a Machine Check to the processor specified by the value of this register. 0: PCI bridge logic machine checks processor 0 1: PCI bridge logic machine checks processor 1
4	**PCI Master Address Translation Disable** 0: PCI Master addresses are always translated before being presented to system memory (see *Base Address Register (BAR)* on page 140) 1 PCI Master addresses are NOT translated and sent directly to system memory

Bit(s)	Description
5 - 7	**ARB Level To Park.** Contains the encoded arbitration level to park when bus is idle: level 000 is for agent **0**, level 001 for agent **1** and so on.
8 - 11	**IRDY Count.** Contains the number of PCI clocks times 8 that device waits before detecting a time-out condition. A value of zero disables the time-out check.
12 - 15	**TRDY Count.** Contains the number of PCI clocks times 8 that device waits before detecting a time-out condition. A value of zero disables the time-out check.
16	**PCI Queue Enable** 0: PCI logic does not queue requests 1: PCI logic queues up to two operations
17	**PCI-ISA Bridge Deadlock Avoidance Disable** 0: PCI-ISA Bridge is present in the system. Therefore, device will: 1) NOT deactivate $\overline{\text{P\_GNT}}$[5] even if other $\overline{\text{REQ}}$s become active (other REQs internally gated) 2) not activate $\overline{\text{P\_GNT}}$[5] until 60x bus has flushed all posted PCI 32-bit bus transfers. 1: PCI-ISA Bridge is NOT present in system, so device treats the $\overline{\text{P\_REQ}}$[5] signal like any other PCI bus $\overline{\text{REQ}}$ signal.
18	**Grant Active To Frame Active Time-out Disable** 0: If device grants the PCI bus to a PCI master and other $\overline{\text{REQ}}$s are outstanding, the PCI master must activate the FRAME signal within 20 cycles or device will deactivate its GNT signal. 1: Once device has granted the bus to a PCI device, Device waits until it sees FRAME active from that device before deactivating its grant signal. **Note:** The 20 cycle count is not guaranteed. The timer runs continuously and therefore device could remove the grant at any time.
19	**Issue Flush Snoops Instead Of Kill Snoops** 0: PCI bridge requests the 60x logic to perform Kill snoops on 60x bus for DMAs as normal. 1: PCI bridge substitutes Flush snoops instead of Kill snoops to the 60x logic. This is to avoid a 604 coherency problem that exists for Kill snoop operations.
20-31	**Reserved**

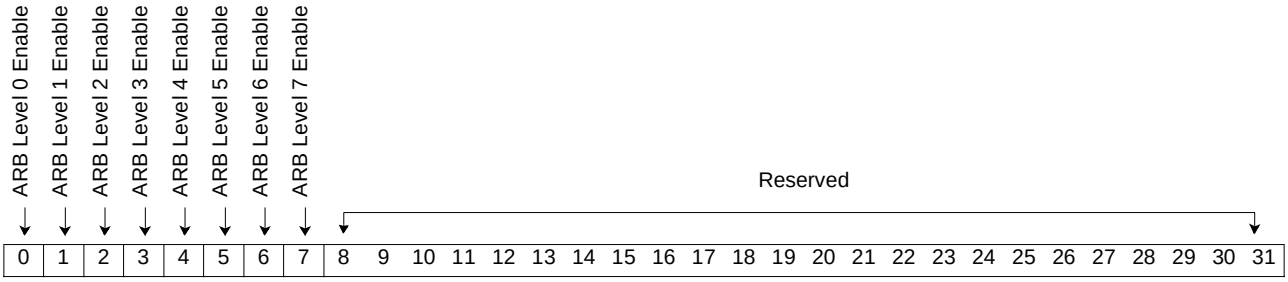
Arbiter Control Register (ACR)

This register provides software with a means to disable individual devices on the PCI bus from generating master bus operations.

Reset Value x'0000 0000'

Address BAR + x'000F 7F30'

Access Type Read/Write



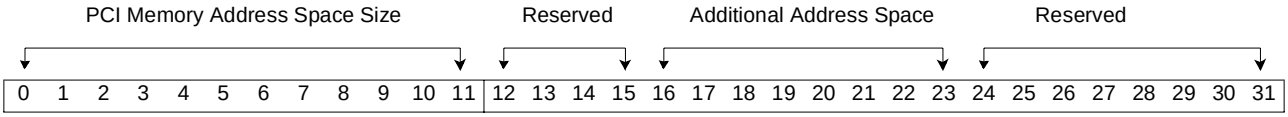
Bit(s)	Description
0	**ARB Level 0 Enable** 0: ARB level is ignored 1: ARB level is enabled
1	**ARB Level 1 Enable**
2	**ARB Level 2 Enable**
3	**ARB Level 3 Enable**
4	**ARB Level 4 Enable**
5	**ARB Level 5 Enable**
6	**ARB Level 6 Enable**
7	**ARB Level 7 Enable** (Not supported in 32-bit PCI bridge)
8-31	**Reserved**

PCI Memory Address Space Size (MSIZE)

Reset Value x'FFF0 0000'

Address BAR + x'000F 7F40'

Access Type Read/Write



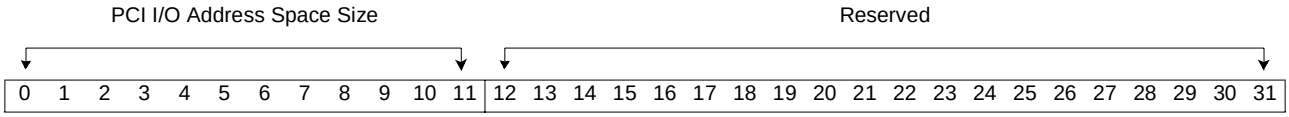
Bit(s)	Description
0-11	**PCI Memory Address Space Size** x'FFF' 1 MB x'FFE' 2 MB x'FFC' 4 MB x'FF8' 8 MB x'FF0' 16 MB x'FE0' 32 MB x'FC0' 64 MB x'F80' 128 MB x'F00' 256 MB x'E00' 512 MB x'C00' 1 GB x'800' 2 GB x'000' 4 GB
12-31	**Reserved**

PCI I/O Address Space Size (IOSIZE)

Reset Value x'FFF0 0000'

Address BAR + x'000F 7F60'

Access Type Read/Write



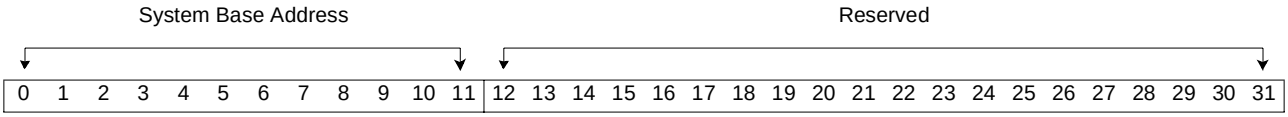
Bit(s)	Description
0-11	**PCI I/O Address Space Size** x'FFF' 1 MB x'FFE' 2 MB x'FFC' 4 MB x'FF8' 8 MB x'FF0' 16 MB x'FE0' 32 MB x'FC0' 64 MB x'F80' 128 MB x'F00' 256 MB x'E00' 512 MB x'C00' 1 GB x'800' 2 GB x'000' 4 GB
12-31	**Reserved**

System Base Address for PCI Memory (SMBAR)

Reset Value x'A000 0000'

Address BAR + x'000F 7F80'

Access Type Read/Write



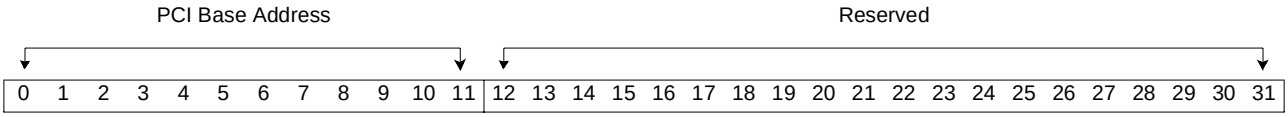
Bit(s)	Description
0 - 11	**System Base Address.** This register contains the upper bits of the SYSTEM address that PCI Memory is mapped to. Note: Address must be aligned on a boundary equal to the size specified in PCI Memory Size register.
12 - 31	**Reserved**
Note: Address is decoded only if the Master Enable bit in the PCI Command Register is on.	

System Base Address for PCI I/O (SIBAR)

Reset Value x'8000 0000'

Address BAR + x'000F 7FC0'

Access Type Read/Write



Bit(s)	Description
0 - 11	**System Base Address** This register contains the upper bits of the SYSTEM address that PCI I/O is mapped to. Note: Address must be aligned on boundary equal to size specified in PCI I/O Size register.
12 - 31	**Reserved**
Note: Address is decoded only if the Master Enable bit in the PCI Command Register is on.	

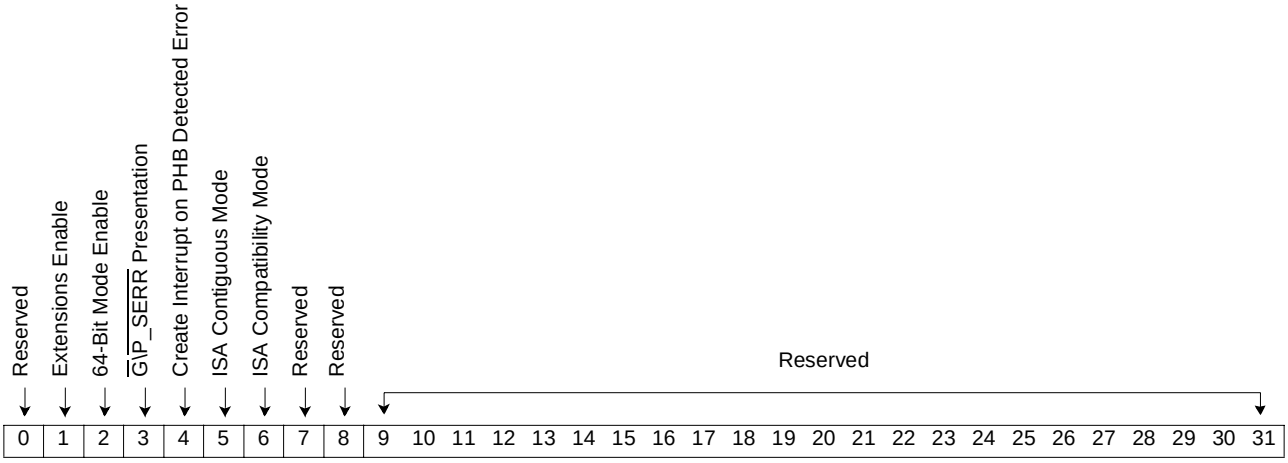
PHB Configuration Register (CTLRW)

This register is primarily used by software to program device for a particular address translation mode.

Reset Value x'0200 0000'

Address BAR + x'000F 7FD0'

Access Type Read/Write



Bit(s)	Description
0	**Reserved**
1	**Extensions Enable.** Read Only. 0: Device does not perform extended error recovery
2	**64-Bit Mode Enable.** Read Only. 0: Device does not support 64-bit addresses
3	**SERR Presentation.** Read Only. 0: PCI Bridge always generates Machine Check if $\overline{G}/\overline{P\_SERR}$ driven active
4	**Create Interrupt On PHB Detected Error.** Read Only. 0: PCI Bridge always generates Machine Check for PHB detected error
5	**ISA Contiguous Mode** This bit programs how device translates the first 8 MB of PCI I/O space. See *Noncontiguous I/O Address Mode Enabled* on page 145 for additional details. 0: ISA space is contiguous 1: ISA space is not-contiguous
6	**ISA Compatibility Mode.** Read Only. 1: Device contains an external pin for this function (P_ISA_MASTER).
7	**Reserved**
8	**Reserved**
9-31	**Reserved**

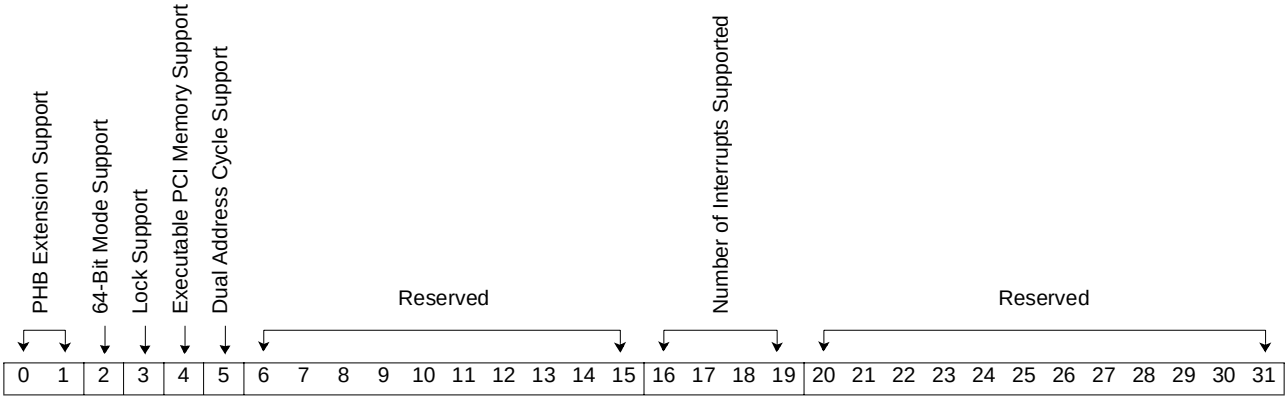
PHB Configuration Register (CTLRO)

This register is the primary indicator to software that device contains the PCI extensions specified in the IBM PHB architecture. It also provides bridge capability information to software.

Reset Value PCI-32 x'9006 0000'
PCI-64 x'9004 0000'

Address BAR + x'F000 7FE0'

Access Type Read Only



Bit(s)	Description
0 - 1	**PHB Extension Support** 10: Device supports error detection extensions but NOT error recovery
2	**64-Bit Mode Support** 0: NOT supported
3	**Lock Support** 1: Full support of PCI lock bus cycles
4	**Executable PCI Memory Support** 0: NOT supported
5	**Dual Address Cycle Support** 0: NOT supported
6 - 15	**Reserved**
16 - 19	**Number Of Interrupts Supported** 0000: No interrupts supported
20 - 31	**Reserved**

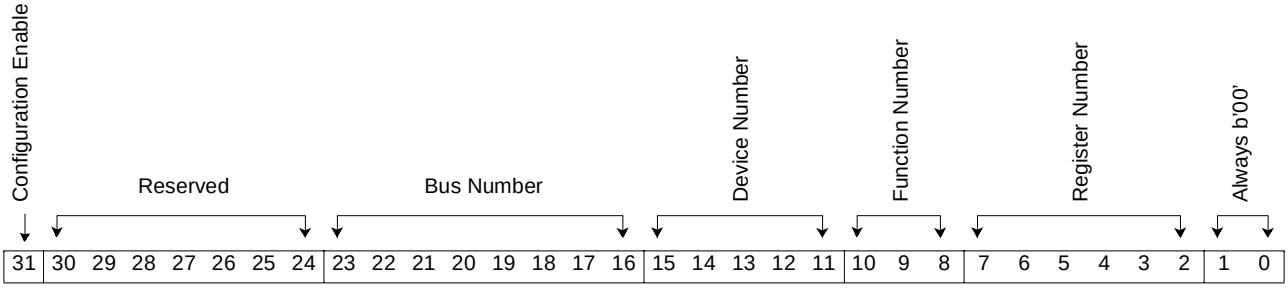
CONFIG_ADDRESS Register (CFGA)

This Little Endian register, along with the CONFIG_DATA register, provides software with a means to configure the PCI bus. Device implements Configuration Mechanism #1 as specified in the PCI Local Bus Specification. See heading *Configuration Cycles* on page 192 for additional details.

Reset Value x'0000 0000'

Address BAR + x'000F 8000'

Access Type Read/Write



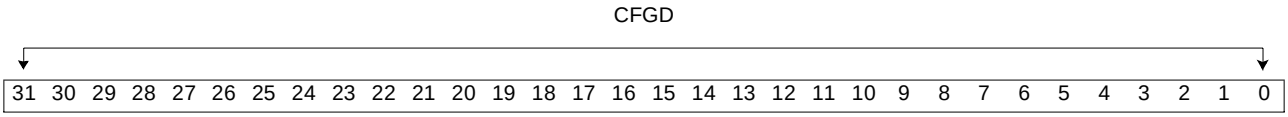
Bit(s)	Description
31	**Configuration Enable** 0: Disabled 1: Enabled, accesses to the CONFIG_DATA register result in device executing a configuration access to itself or to the PCI bus.
30 - 24	**Reserved**
23 - 16	**Bus Number.** Specifies which PCI bus is being configured. Device checks this field to determine the appropriate configuration action.
15-11	**Device Number**. Selects a particular device to be configured on a bus.
10-8	**Function Number.** For devices that implement more than one function, this field specifies which function to configure within a device.
7-2	**Register Number**. Specifies which register out of the 256-byte PCI Configuration header to access
1-0	**Always b'00'**

CONFIG_DATA Register (CFGD)

Reset Value x'0000 0000'

Address BAR + x'000F 8010'

Access Type Read/Write



Bit(s)	Description
31 - 0	**Virtual 32-bit Register**. When this Little Endian register is accessed in Read or Write, the device initiates a PCI Configuration Read or Write cycle of external PCI devices, the address of which is provided by the PCI CONFIG_ADDRESS Register.

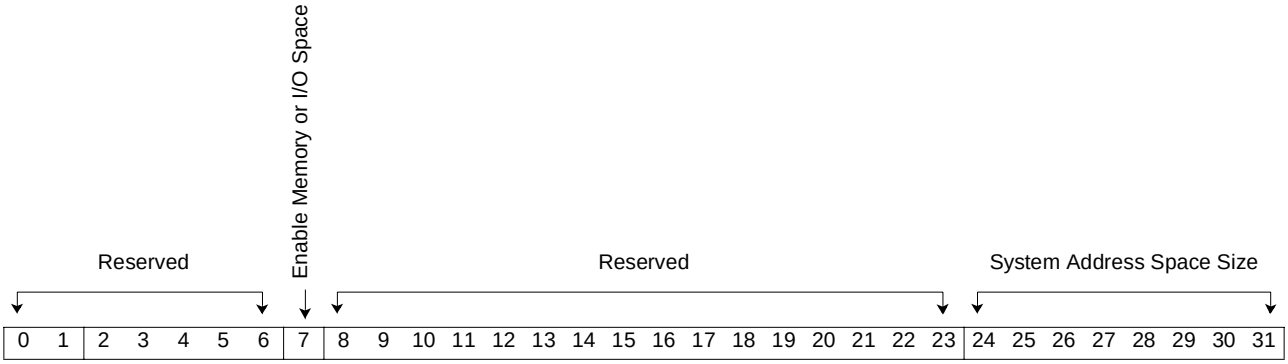
System Address Space SIZE for PCI (PSSIZE)

This is the same definition for 32-bit PCI and 64-bit PCI.

Reset Value x'0000 0000'

Address BAR + x'000F 8100'

Access Type Read/Write



Bit(s)	Description
0 - 6	**Reserved**
7	**Enable Memory Or IO Space** 0: Memory Space 1: IO Space
8 - 23	**Reserved**
24 - 31	**System Address Space Size** x'FF': 16 MB x'FE': 32 MB x'FC': 64 MB x'F8': 128 MB x'F0': 256 MB x'E0': 512 MB x'C0': 1 GB x'80': 2 GB x'00': 4 GB

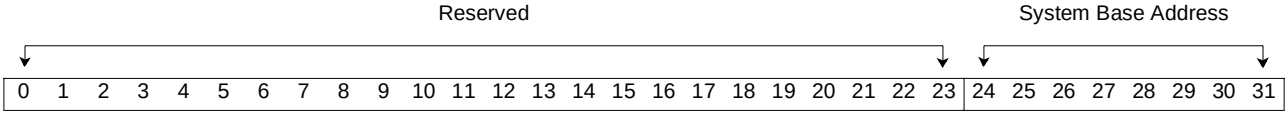
System Base Address Register (BARPS)

The definition is the same for 32-bit PCI and 64-bit PCI.

Reset Value x'0000 0000'

Address BAR + x'000F 8120'

Access Type Read/Write



Bit(s)	Description
0 - 23	**Reserved**
24 - 31	**System Base Address.** Contains the upper bits of the system Base address that Memory is mapped to.

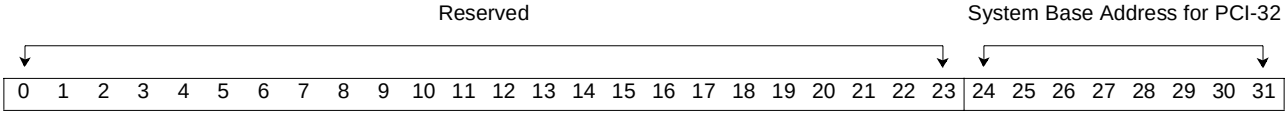
System Base Address Register for PCI-32 (PSBAR)

Only the PCI-32 bit bridge has this register.

Reset Value x'0000 0000'

Address BAR + x'000F 8140'

Access Type Read/Write



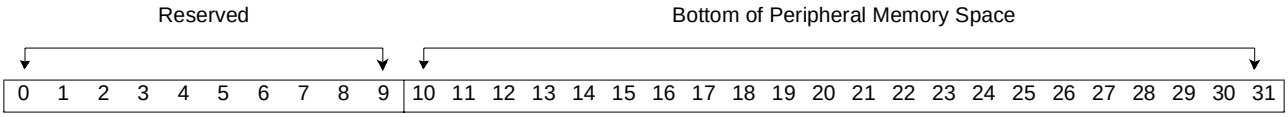
Bit(s)	Description
0 - 23	**Reserved**
24 - 31	**System Base Address for PCI-32**. Contains the upper bits of the System Base address that memory is mapped to.

Bottom of Peripheral Memory Space With Potential Deadlock (BPMDLK)

Reset Value x'0000 0000'

Address BAR + x'000F 8200'

Access Type Read/Write



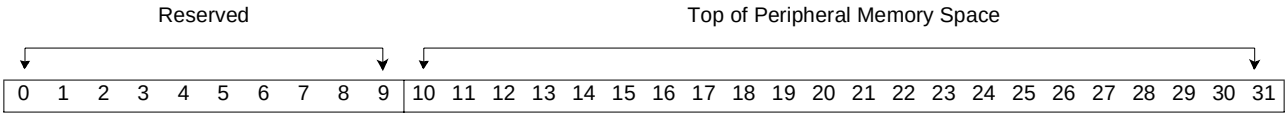
Bit(s)	Description
0 - 9	**Bottom of Peripheral Memory Space**. Contains the bottom address for the CPU to PCI MEMORY access with potential deadlock
10 - 31	**Reserved**

Top of Peripheral Memory Space With Potential Deadlock (TPMDLK)

Reset Value x'0000 0000'

Address BAR + x'000F 8210'

Access Type Read/Write



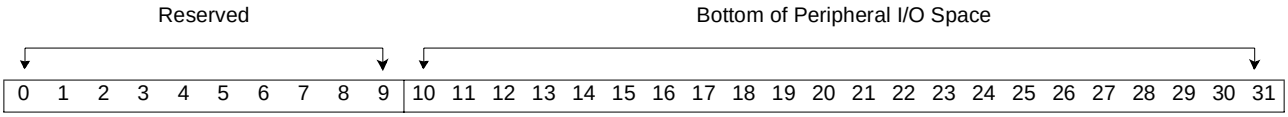
Bit(s)	Description
0 - 9	**Top of Peripheral Memory Space**. Contains the top address for the CPU to PCI MEMORY access with potential deadlock
10 - 31	**Reserved**

Bottom of Peripheral I/O Space With Potential Deadlock (BIODLK)

Reset Value x'0000 0000'

Address BAR + x'000F 8220'

Access Type Read/Write



Bit(s)	Description
0 - 9	**Bottom of Peripheral IO Space**. Contains the bottom address for the CPU to PCI IO access with potential deadlock
10 - 31	**Reserved**

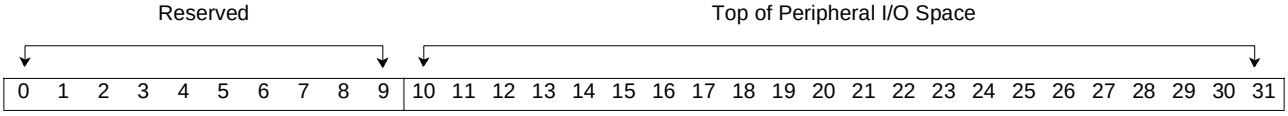
Top of Peripheral I/O Space With Potential Deadlock (TIODLK)

This register exists on PCI-32 and PCI-64.

Reset Value x'0000 0000'

Address BAR + x'000F 8230'

Access Type Read/Write



Bit(s)	Description
0 - 9	**Top of Peripheral I/O Space**. Contains the top address for the CPU to PCI IO access with potential deadlock
10 - 31	**Reserved**

Potential Deadlock management:

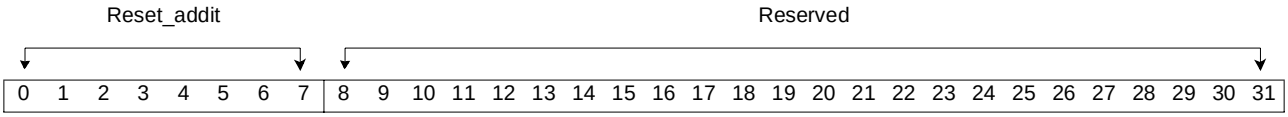
The normal mode of burst transfer from a PCI Master to the Memory is 32 Bytes. The setting of bit 0 permit to have Long Burst of up to 4KBytes with no Disconnect RETRY during the Burst.

Reset Addressed Interrupt Register (IT_ADD_RESET)

This Virtual register exists only for PCI-64 bridge. Only the CPU can write to this register and reset the $\overline{IT}1$ output interrupt signal.

Address BAR + ’000F 8300’

Access Type Write Only



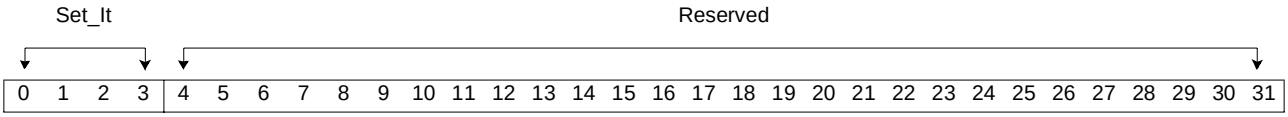
Bit(s)	Description
0 - 7	**Reset_addit** 1: Writing a 1 in one of these 8 bits resets the interrupt signal $\overline{INT}1$ 0: No action
8 - 31	**Reserved**

Set PCI-64 Interrupt Register (INT_SET)

This register exits only for PCI-64, Interrupt can be set only by the CPU.

Address BAR + ’000F 8310’

Access Type Write Only



Bit(s)	Description
0 - 3	**Set_It** Writing 1 by the CPU set the bit corresponding Writing 0 has no action Bit 0: $\overline{\text{INTA}}$ Bit 1: $\overline{\text{INTB}}$ Bit 2: $\overline{\text{INTC}}$ Bit 3: $\overline{\text{INTD}}$
4 - 31	**Reserved**

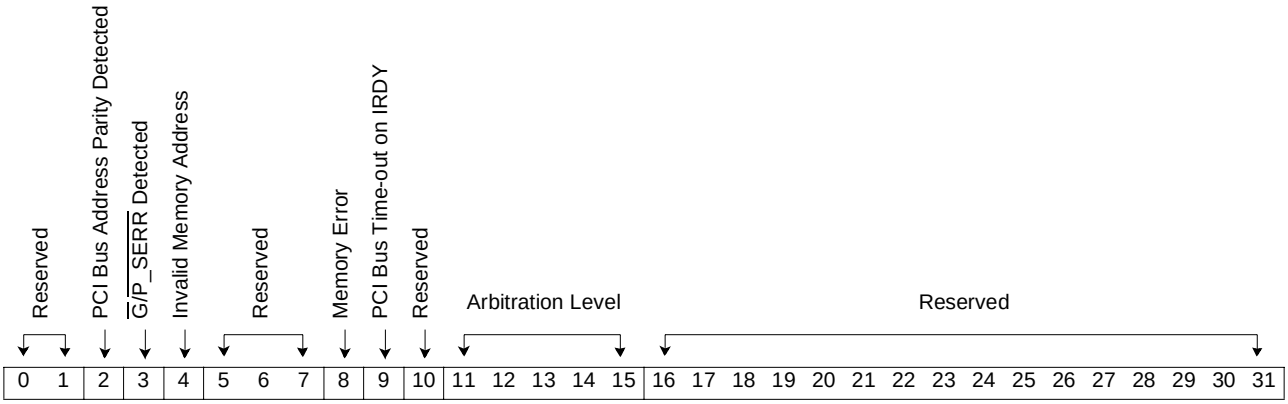
Channel Status Register (CSR)

This register is used to log errors during PCI Master to system transfers. Please see *PCI Master Error Handling* on page 195 for additional details.

Reset Value x'0000 0000'

Address BAR + '000F 9800'

Access Type Read/Write



Bit(s)	Description
0 - 1	**Reserved**
2	**PCI Bus Address Parity Detected** 0: No Error 1: PCI Bridge detected address parity error
3	**SERR Detected** 0: No Error 1: PCI Bridge detected $\overline{G}/\overline{P\_SERR}$ during transaction
4	**Invalid Memory Address** 0: No Error 1: PCI access occurred to invalid system memory address
5 - 7	**Reserved**
8	**Memory Error** 0: No Error 1: Double bit ECC error occurred during memory access
9	**Bus Time-out** 0: No Error 1: PCI Bridge detected bus time-out; no IRDY detected (see *Disconnect Counter (DISCNT)* on page 59)
10	**Reserved**
11 - 15	**Arbitration Level.** Encoded arbitration level of PCI device when error occurred
16 - 31	**Reserved**

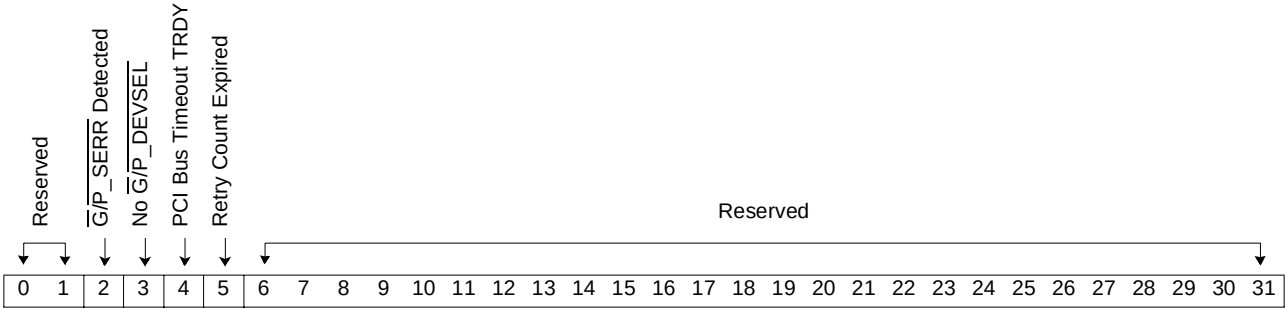
Processor Load/Store Status Register (PLSSR)

This register provides error status information for all transfers initiated by the CPU, a PCI master, or the other PCI Bridge logic. Please see *Error Handling for CPU-Initiated Transactions* on page 164 for additional details on this register.

Reset Value x'0000 0000'

Address BAR + x'000F 9810'

Access Type Read/Write



Bit(s)	Description
0 - 1	**Reserved**
2	**SERR Detected** 0: No Error 1: PCI Bridge detected G/P_SERR active during master operation
3	**No Devsel** 0: No Error 1: PCI Bridge did not receive G/P_DEVSEL during master operation
4	**PCI Bus Timeout** 0: No Error 1: PCI Bridge detected bus time-out; no G/P_TRDY detected
5	**Retry Count Expired** 0: No Error 1: PCI Bridge detected bus time-out, too many retry's (see *Retry Counter (RETRY)* on page 60)
6 - 31	**Reserved**

System Registers Space

System Registers List (Page 1 of 3)

Address	Name	Use	Page	Notes
x'FF00 0000' to x'FF00 0007'		Reserved		
Standard System Registers				
x'FF00 0008'	PIDR	Physical Identifier Register	96	1
x'FF00 000C'	CNFR	Connectivity Configuration Register	97	see p 33
x'FF00 0010'	RSTR	Connectivity Reset Register	98	
x'FF00 00E8'	SPOR	Software POR Register	99	7
Specific System Registers				
x'FF00 1000'	UCTL	Universal Control Register	100	
x'FF00 1010'	MPSR	Multi-Processor Semaphore Register	102	
x'FF00 1020'	SIOC	System I/O Control	103	
x'FF00 1030'	ABCNTL	60x Arbiter Control Register	104	
x'FF00 1040'	SRST	CPU Soft Reset Register	106	7
x'FF00 1050'	ERRC	Error Control Register	107	
x'FF00 1060'	SESR	System Error Status Register	108	
x'FF00 1070'	SEAR	System Error Address Register	110	
x'FF00 1080'		Reserved		
x'FF00 1100'	PGCHP	Chip program Register	111	
x'FF00 1110'	RGBAN1	Free Register 1	113	
x'FF00 1120'	RGBAN2	Free Register 2	114	
x'FF00 1130'	GPDIR	GPIO Direction Register	115	
x'FF00 1140'	GPIN	GPIO Input Register	116	
x'FF00 1150'	GPOUT	GPIO Output Register	117	
x'FF00 1160'	ATAS	Address Transfer Attribute for Snoop Reg	118	
x'FF00 1170'	AVDG	Device Diagnostic Register	120	
x'FF00 1174' to x'FF00 11FF'		Reserved		
x'FF00 1200'	MCCR	Memory Controller Control Register	122	
x'FF00 1210'		Reserved		
x'FF00 1220'	MESR	Memory Error Status Register	124	
x'FF00 1230'	MEAR	Memory Error Address Register	125	
x'FF00 1300'	MCER0	Memory Configuration Extent Register 0	126	

1. RO: Read Only Register
2. All bits can be read. Only bits [4:31] can be written
3. All bits can be read. Only bits [0:3] can be written
4. Four beat burst read operations allowed to this address space; Single byte writes only
5. Not decoded by system logic
6. Byte accesses allowed
7. WO: Write Only Register
8. Range that IBM Dual Bridge and Memory Controller responds to is programmable

System Registers List (Page 2 of 3)

Address	Name	Use	Page	Notes
x'FF00 1310'	MCER1	Memory Configuration Extent Register 1	126	
x'FF00 1320'	MCER2	Memory Configuration Extent Register 2	126	
x'FF00 1330'	MCER3	Memory Configuration Extent Register 3	126	
x'FF00 1340'	MCER4	Memory Configuration Extent Register 4	126	
x'FF00 1350'	MCER5	Memory Configuration Extent Register 5	126	
x'FF00 1360'	MCER6	Memory Configuration Extent Register 6	126	
x'FF00 1370'	MCER7	Memory Configuration Extent Register 7	126	
x'FF00 1400'	SIOR0	SIO Register 0 (DIMM PDs)	128	1
x'FF00 1410'		Reserved		
x'FF00 1420'	SIOR1	SIO Register 1 (Planar, DIMM, CPU, etc.)	129	1
x'FF00 1424 to x'FF00 1FFF'		Reserved		
x'FF00 2000 to 'FF17 FFFF'		Reserved		
DMA Registers: User Privilege				
x'FF18 0000 to x'FF1C 001F'		Reserved		
x'FF1C 0020'	GSCR	Global Control Register (user)	130	1
x'FF1C 0030'	GSSR	Global Status Register (user)	131	1
x'FF1C 0040'	XSCR	Transfer Control Register (user)	132	
x'FF1C 0050'	XSSR	Transfer Status Register (user)	133	1
x'FF1C 0070'	XPAR	Transfer PCI Address Register (user)	135	2
x'FF1C 0090'	XWAR	Transfer Write Back Address Register (user)	136	1
x'FF1C 00A0'	XTAR	Transfer Translated Address Register (user)	137	1
x'FF1E 0020'	GSCR	Global Control Register (priv)	130	
x'FF1E 0030'	GSSR	Global Status Register (priv)	131	1
x'FF1E 0040'	XSCR	Transfer Control Register (priv)	132	
x'FF1E 0050'	XSSR	Transfer Status Register (priv)	133	1
x'FF1E 0070'	XPAR	Transfer PCI Address Register (priv)	135	3
x'FF1E 0090'	XWAR	Transfer Write Back Address Register (priv)	136	
x'FF1E 00A0'	XTAR	Transfer Translated Address Register (priv)	137	1
x'FF1E 00A4' to x'FF1F FFFF'		Reserved		
System Standard Configuration Registers				
x'FF20 0000'	DCR	Device Characteristics Register	138	1
x'FF20 0004'	DID	Device ID Register	139	1
x'FF20 0008' to x'FF20 0014'	Reserved			

1. RO: Read Only Register
2. All bits can be read. Only bits [4:31] can be written
3. All bits can be read. Only bits [0:3] can be written
4. Four beat burst read operations allowed to this address space; Single byte writes only
5. Not decoded by system logic
6. Byte accesses allowed
7. WO: Write Only Register
8. Range that IBM Dual Bridge and Memory Controller responds to is programmable

System Registers List (Page 3 of 3)

Address	Name	Use	Page	Notes
x'FF20 0018'	BAR	Base Address Register	140	see p. 33
x'FF20 0020' to x'FF20 0FFF'	Reserved			
Device Specific Configuration Space				
x'FF20 1000'	PCIENB	PCI BAR Enable Register	141	
x'FF20 1004' to x'FFDF FFFF'		Reserved		
BOOT ROM				
x'FFE0 0000' to x'FFFF FFFF'	IPLROM	FLASH ROM: Up to 2 MB		4, 6, 8

1. RO: Read Only Register
2. All bits can be read. Only bits [4:31] can be written
3. All bits can be read. Only bits [0:3] can be written
4. Four beat burst read operations allowed to this address space; Single byte writes only
5. Not decoded by system logic
6. Byte accesses allowed
7. WO: Write Only Register
8. Range that IBM Dual Bridge and Memory Controller responds to is programmable

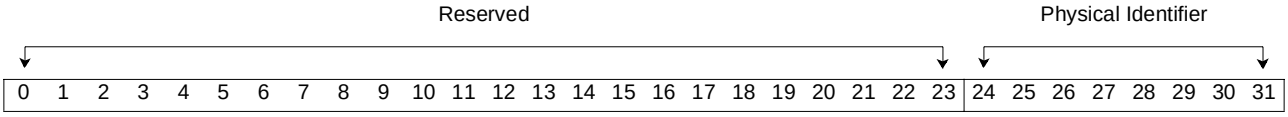
Physical Identifier Register (PIDR)

This register provides a unique number for each processor (or any 60x bus master) reading this location. It is primarily used by processors to differentiate themselves in multiprocessor configurations. When this register is read, device latches the current processor's SYS_BR/SYS_BG pair into this register which physically identifies the processor. Each processor has a unique SYS_BR/SYS_BG pair connected to it.

Reset Value x'0000 0000'

Address x'FF00 0008'

Access Type Read/Write



Bit(s)	Description
0 - 23	**Reserved**
24 - 31	**Physical Identifier.** Device responds with two values for this field: x'00': Indicates processor associated with $\overline{BR}$0 and $\overline{BG}$0 pins. x'01': Indicates processor associated with $\overline{BR}$1 and $\overline{BG}$1 pins

Connectivity Configuration Register (CNFR)

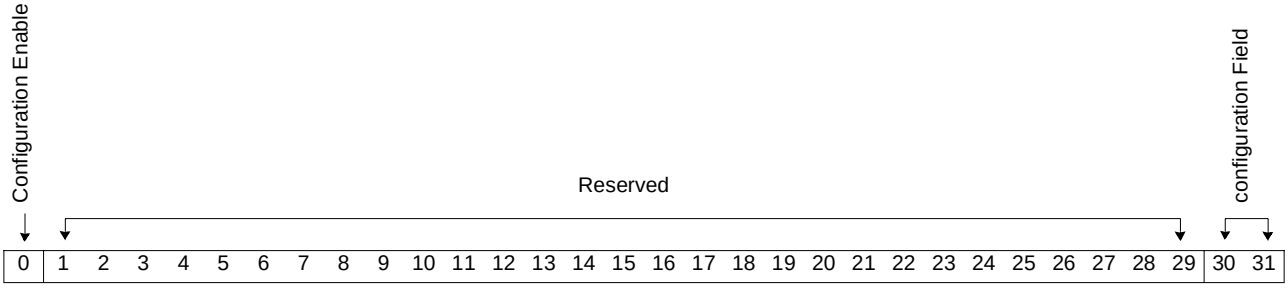
The CNFR register described below is used to support the initialization and configuration of devices on the 60x bus. This register provides the unique setup signal required to insure that only one device will respond to configuration addresses at a time. Software must adhere to the following restrictions for configuration:

- A write to the CNFR register must be followed by a SYNC operation or a read of the register.
- Software must issue a read to the Device Characteristics Register (DCR) at x'FF20 0000' to determine if a device is present.
- If software receives a x'F000 0000' response from the DCR read, this indicates that no device is present and therefore no other configuration registers should be accessed. Access to other configurations registers will result in a bus time-out condition.

Reset Value x'0000 0000'

Address x'FF00 000C'

Access Type Read/Write



Bit(s)	Description
0	**Configuration Enable** 0: Inactivate all $\overline{\text{SYS\_CONFIG}}$[n] signals 1: Activate appropriate $\overline{\text{SYS\_CONFIG}}$[n] as described in bits 30 - 31
1 - 29	**Reserved**
30 - 31	**Configuration Field** 00: $\overline{\text{SYS_CONFIG}}$0. $\overline{\text{SYS_CONFIG}}$0 signal driven active 01: $\overline{\text{SYS_CONFIG}}$1. $\overline{\text{SYS_CONFIG}}$1 signal driven active 10: $\overline{\text{SYS_CONFIG}}$2. Configuration access directed to PCI-32 bus 11: $\overline{\text{SYS_CONFIG}}$3. Configuration access directed to PCI-64 bus

See *60x Bus Configuration* on page 163 for details on configuration bus cycles and procedures.

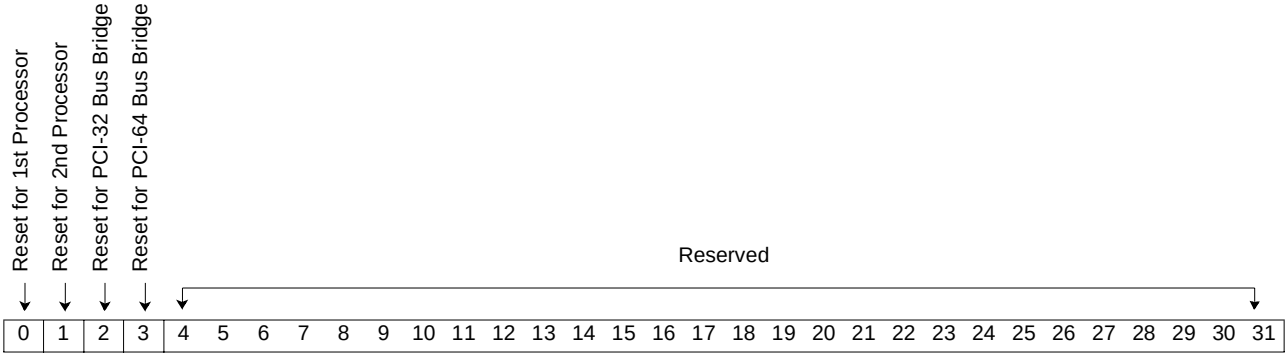
Connectivity Reset Register (RSTR)

This register provides a means to individually reset devices on the 60x bus. Bits 0 and 1 directly control SYS_HRESET0 and SYS_HRESET1 respectively. The remaining two bits control reset signals that are internal to device.

Reset Value x'C000 0000'

Address x'FF00 0010'

Access Type Read/Write



Bit(s)	Description
0	**Reset For 1st Processor** 0: SYS_HRESET0 signal is active 1: SYS_HRESET0 signal is inactive
1	**Reset For 2nd Processor** 0: SYS_HRESET1 signal is active 1: SYS_HRESET1 signal is inactive
2	**Reset for PCI 32 Bus Bridge** 0: Reset signal active 1: Reset signal is inactive
3	**Reset for PCI-64 Bus Bridge** 0: Reset signal active 1: Reset signal is inactive
4 - 31	**Reserved**

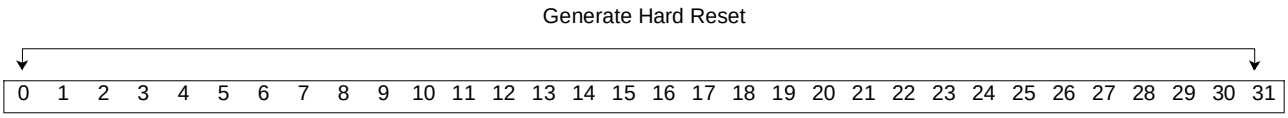
Software Power On Reset Control Register (SPOR)

This register provides a mechanism for software to initiate a hard reset to the system. The device will activate resets to all processors and I/O devices.

Reset Value x'0000 0000'

Address x'FF00 00E8'

Access Type Write Only



Bit(s)	Definition
0 - 31	**Generate Hard Reset** A write to this register will initiate a power on reset.

Specific System Registers Space

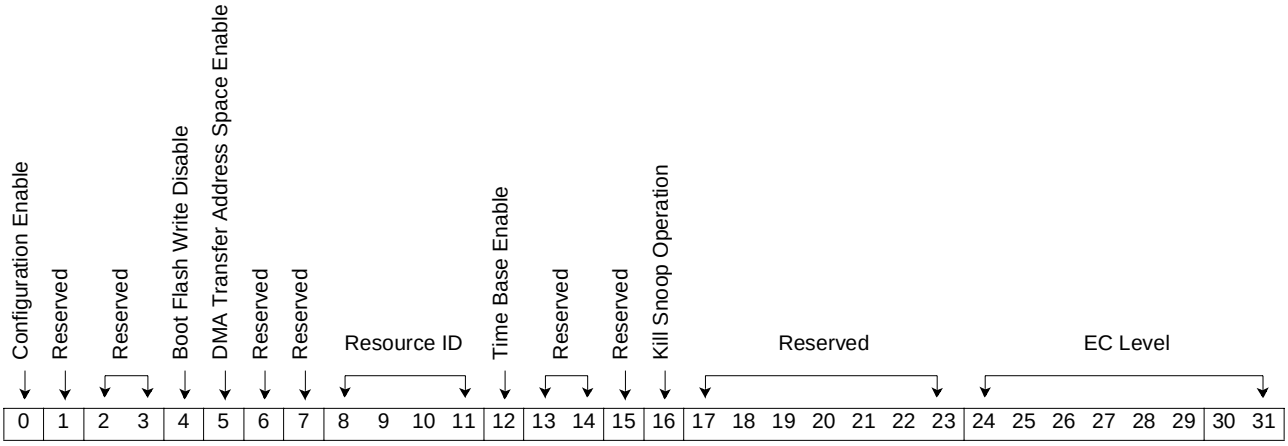
Universal System Control (UCTL)

This register is used to enable address ranges to be decoded by the CPC710 and processor related operations.

Reset Value x'0008 0080

Address x'FF00 1000'

Access Type Read/Write



Bit(s)	Description
0	**Reserved**
1	**Reserved**. R/W
2-3	**Reserved**. R/W Must be left to 0
4	**Boot Flash Write Disable** 0: Indicates that writes are allowed to Boot FLASH space 1: Writes to Boot FLASH space are inhibited
5	**DMA Transfer Address Space Enable** 0: Accesses to DMA Address Range allowed 1: Accesses to DMA Address Range inhibited
6	**Reserved**. R/W Must be left to 0
7	**Reserved**
8 - 11	**Resource ID.** This 4-bit field contains the Resource ID that device uses to determine whether or not it is the target of a DMA transfer operation**.**
12	**Time Base Enable** 0: The Time Base Enable signal to the CPU is deactivated. CPU real time clocks halted 1: The Time Base Enable signal SYS_TBE to the CPU is activated. CPU real time clocks enabled
13 - 14	**Reserved**
15	**Reserved** R/W

Bit(s)	Description
16	**Kill Snoop Operation (Must set to 0 for the PowerPC 750)** 0: Device issues Kill address only transaction types for full cache line invalidates 1: Device issues Flush address only transaction type for full cache line invalidates (Workaround of the 604 errata "Kill snoop bug")
17 - 23	**Reserved**
24 - 31	**CPC710 - EC LEVEL**. Read only Bit 24 always at one Bits 25 to 27 main Engineering changes Bits 28 to 31 for sub Engineering changes 1000 0000 for CPC710_100

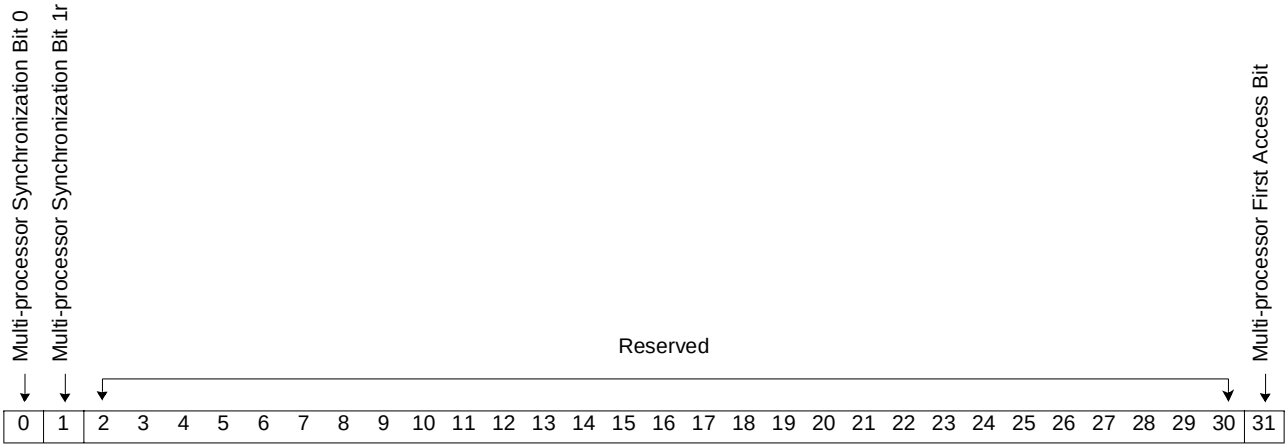
Multi-Processor Semaphore (MPSR)

This register is used by the IPL boot code to facilitate bring-up of processors in an MP environment. It provides a first access bit, BIT 31, that allows a method for processors to determine which processor is the master, since both processors are active after power on. BIT 31 contains a value of 0 after power on reset. When the first processor read occurs to this register, BIT 31 returns a value of 0. All subsequent reads of this register return a value of 1 for BIT 31. In addition to the First Access Bit, bits 0 and 1 provide semaphores for use by the firmware during boot time and are utilized until system memory has been initialized and tested.

Reset Value x'0000 0000'

Address x'FF00 1010'

Access Type Read/Write



Bit(s)	Description
0	**Multi-processor Synchronization Bit 0.** Used for communication between processors at IPL time
1	**Multi-processor Synchronization Bit 1.** used for communication between processors at IPL time
2 - 30	**Reserved**
31	**Multi-processor First Access Bit**. Read Only; Set after read 0: Initial power on value; Indicates first read of this register. 1: Indicates that this register has been read at least once previously.

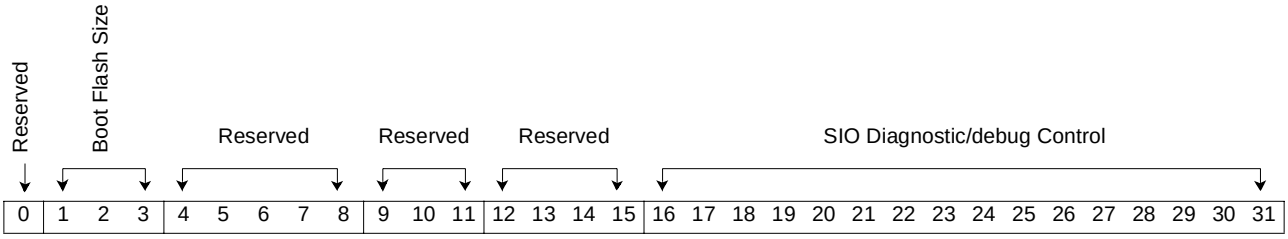
System I/O Control (SIOC)

This register provides initialization and control of the Boot FLASH devices to which device interfaces.

Reset Value x'0000 0000'

Address x'FF00 1020'

Access Type Read/Write



Bit(s)	Description
0	**Reserved**. R/W
1 - 3	**Boot Flash Size** 000: 2.0 MB - Device initiates FLASH access for addresses x'FFE0 0000' to x'FFFF FFFF' 001: 1.0 MB - Device initiates FLASH access for addresses x'FFF0 0000' to x'FFFF FFFF' 011: 0.5 MB - Device initiates FLASH access for addresses x'FFF0 0000' to x'FF7F FFFF' 111: Reserved
4 - 8	**Reserved**
9 - 11	**Reserved**. These bit should be left to zero.
12 - 15	**Reserved**
16 - 31	**SIO Diagnostic/debug Control.** Must Leave At Zero; Do Not Change 16-17 Controls SIO_TT1 signal 18-23 Controls PROC ID output 24-27 Controls asynchronous boundary 28-29 Controls SIO_TS signal 30-31 Controls SIO_TA signal

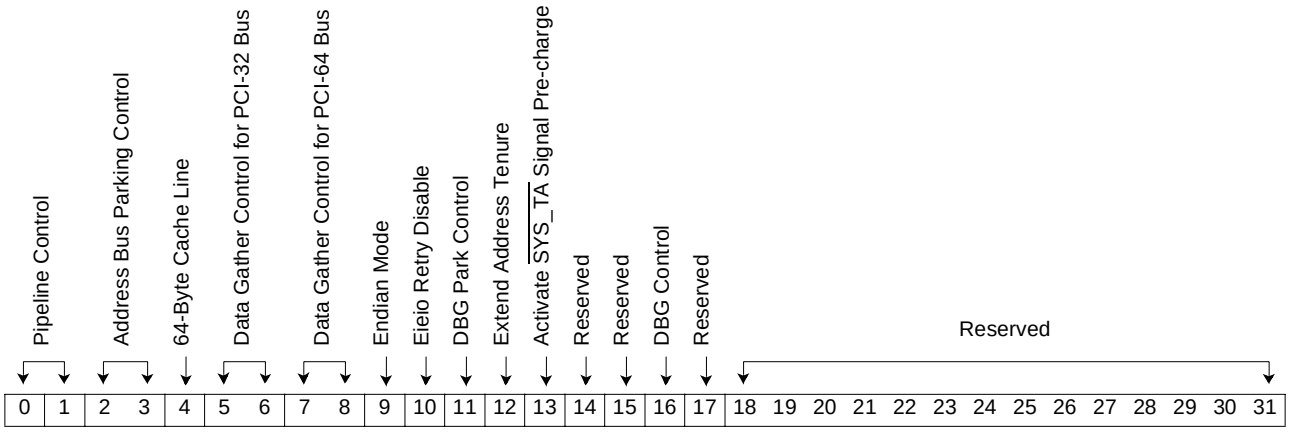
-60x Arbiter Control Register (ABCNTL)

This register provides extensive control over the 60x bus arbiter operation. For a detailed description of the 60x bus arbiter, see section *60x Bus Arbiter Description* on page 154.

Reset Value x'0000 0000'

Address x'FF00 1030'

Access Type Read/Write



Bit(s)	Description
0 - 1	**Pipeline Control** 00: Pipelining Disabled 01: One level pipelining per arbitration level is enabled (two outstanding addresses allowed) 10: Two level pipelining per arbitration level is enabled (three outstanding addresses allowed) 11: One level pipeline enabled across both arbitration levels; Selected this mode for operation with an L2 look aside controller.
2 - 3	**Address Bus Parking Control** 00: Parking Disabled 01: Parking enabled for Arbitration level 0 only 10: Parking enabled for Arbitration level 1 only 11: MRU parking enabled; Last arbitration level active is parked.
4	**64-Byte Cache Line** 0: Arbiter will grant the address bus as normal 1: Arbiter will grant a second address bus tenure to the current arbitration level, if the current arbitration level is again requesting the address bus and if first access is a burst transaction. Normal round robin grant sequence will resume after each pair of grants.
5 - 6	**Data Gather Control for PCI-32 bus** 0x: Not enabled 10: Enabled for accesses to incrementing addresses only 11: Enabled for accesses to incrementing and same addresses - NOT RECOMMENDED
7 - 8	**Data Gather Control for PCI-64 bus** 0x: Not enabled 10: Enabled for accesses to incrementing addresses only 11: Enabled for accesses to incrementing and same addresses - NOT RECOMMENDED

Bit(s)	Description
9	**Endian Mode of the PowerPC CPU** 0: 60x logic interprets data from 60x in Big Endian mode 1: 60x logic interprets data from 60x in Little Endian mode
10	**Eieio Retry Disable** 0: Device will always $\overline{\text{SYS\_ARTRY}}$ an EIEIO operation until every command in 60x queues has been dispatched to the logic units inside device. 1: Device will not $\overline{\text{SYS\_ARTRY}}$ an EIEIO operation
11	**DBG Park Control** 0: DBG signals are not parked when bus is idle 1: DBG signals are parked when bus is idle; mode to use for 0 wait state L2 look aside Bit 13 must be set to zero or this bit is ignored.
12	**Reserved.** R/W **Must be left to 0**
13	**Activate TA Signal Pre-charge** 0: The TA signal is precharged by device after a data bus tenure 1: The TA signal is not precharged by device at the completion of a data bus tenure
14	**Reserved**. These bit should be left to zero.
15	**Reserved.** R/W
16	**DBG Control** 0: DBG0 and DBG1 signals are driven separately 1: DBG0 and DBG1 signals are effectively the same, they are logically ORed; mode to use for L2 lookaside
17	**Reserved.** R/W
18 - 31	**Reserved**

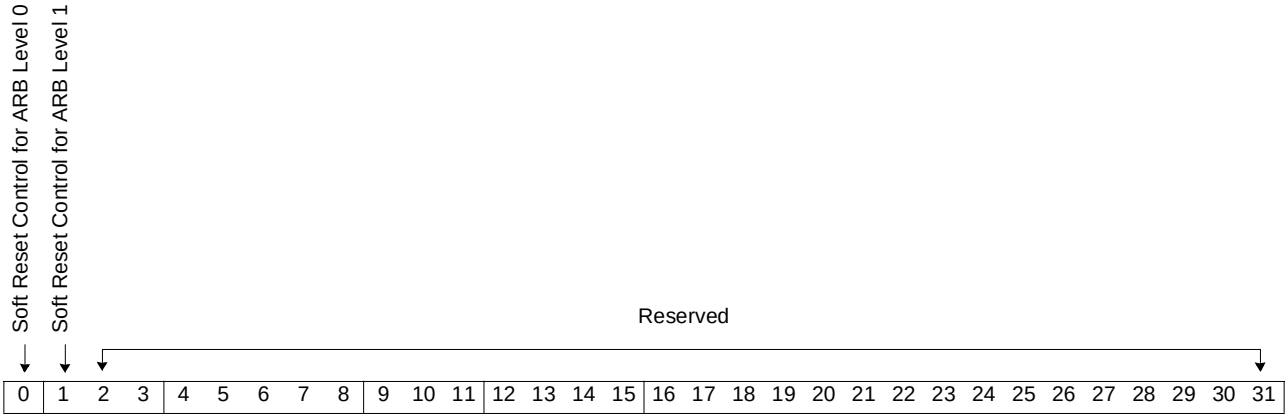
CPU Soft Reset Register (SRST)

This register provides software with a mechanism to issue soft resets to each of the processors. When device detects a write to this register, the corresponding $\overline{\text{SYS\_SRESET}}$ 0 or 1 signal is driven active for a minimum of eight RTC_CLK clocks .

Reset Value x'0000 0000'

Address x'FF00 1040'

Access Type Write Only



Bit(s)	Description
0	**Soft Reset Control for ARB Level 0** 0: Writing 0 to this bit has no effect 1: Writing 1 to this bit will initiate a pulse on the $\overline{\text{SYS_SRESET}}$0 signal.
1	**Soft Reset Control for ARB Level 1** 0: Writing 0 to this bit has no effect 1: Writing 1 to this bit will initiate a pulse on the $\overline{\text{SYS_SRESET}}$1 signal.
2 - 31	**Reserved**

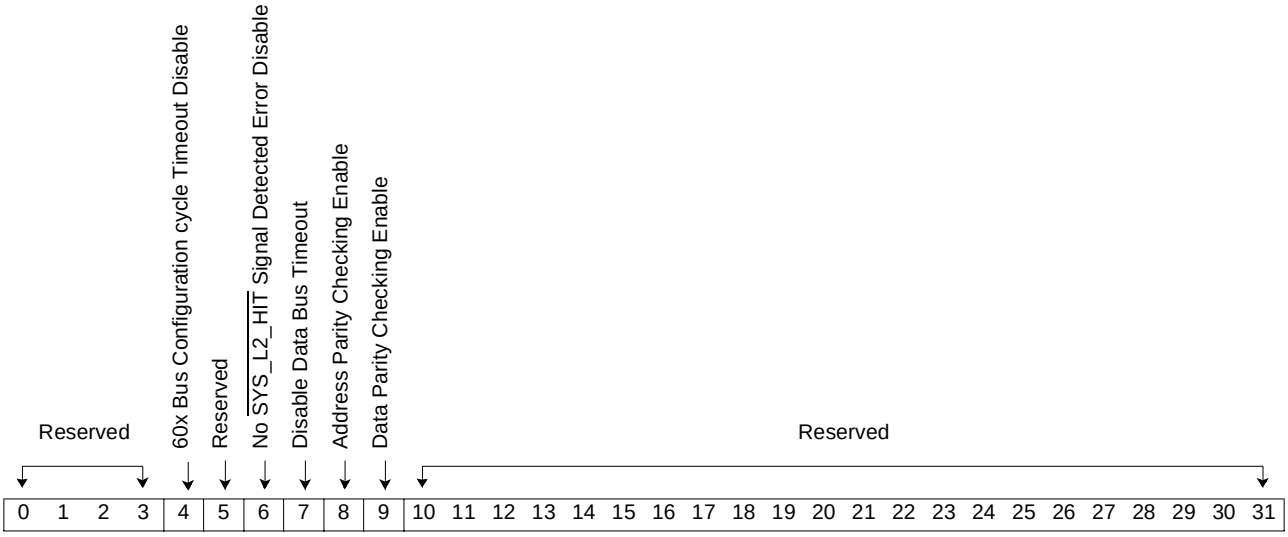
Error Control Register (ERRC)

This register controls how the 60x interface logic responds when detecting an error.

Reset Value x'0000 0000'

Address x'FF00 1050'

Access Type Read/Write



Bit(s)	Description
0 - 3	**Reserved**
4	**60x Bus Configuration Cycle Timeout Disable** 0: 60x logic will respond with x'F000 0000' for a read from x'FF20 0000' if $\overline{\text{SYS\_L2\_HIT}}$ signal not driven active 1: No configuration response performed
5	**Reserved**
6	**No $\overline{\text{SYS\_L2\_HIT}}$ Signal Detected Error Disable** 0: 60x logic will generate TEA on the system bus if $\overline{\text{SYS\_L2\_HIT}}$ signal not driven active after AACK 1: No action if $\overline{\text{SYS\_L2\_HIT}}$ detected inactive
7	**Disable Data Bus Timeout** In the case of timeout, the CPC710 activates the $\overline{\text{CHKSTOP}}$ and set bit 20 of the SESR Register *(see System Error Status Register (SESR)* on page 108). 0: Device will signal error if 8ms time-out detected from DBG to TA 1: Device will not signal an error for this condition
8	**Address Parity Checking Enable** 0: 60x logic will not check address parity on the system bus 1: 60x logic will check address parity on the system bus for CPU to the CPC710 access only. In case of parity error the $\overline{\text{CHKSTOP}}$ signal is activated.
9	**Data Parity Checking Enable** 0: 60x logic will not check data parity on the system bus 1: 60x logic will check data parity on the system bus for CPU to the CPC710 access only. In case of parity error the $\overline{\text{CHKSTOP}}$ signal is activated.
10 - 31	**Reserved**

System Error Status Register (SESR)

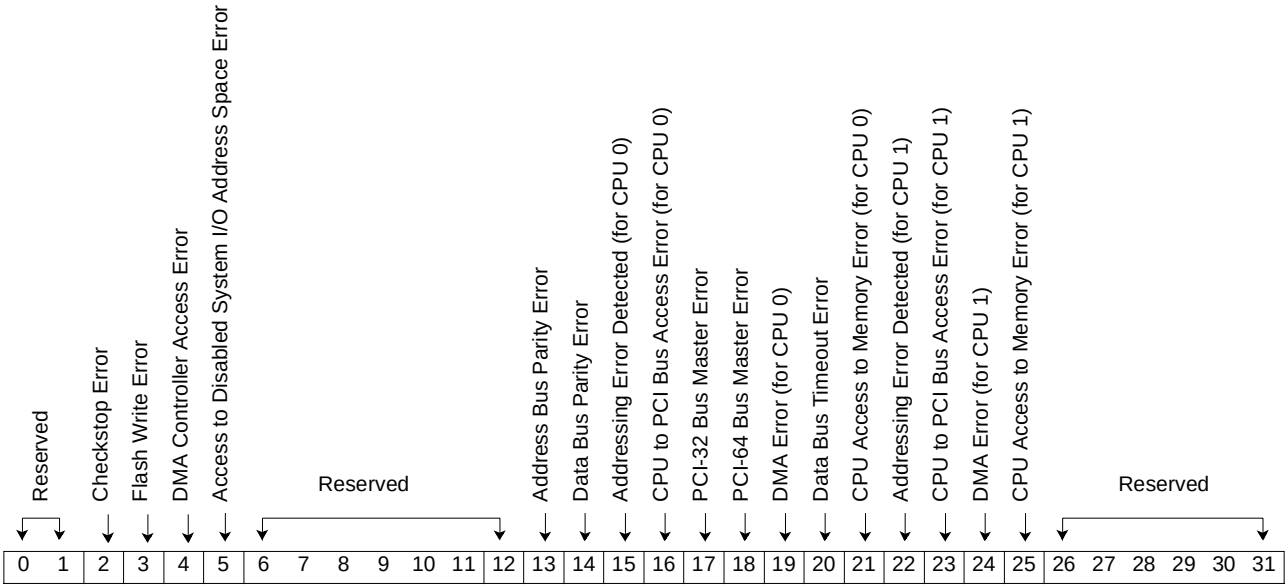
This register is the primary error status register for device and should be read first after a Machine Check interrupt occurs (SYS_MACHK0 or SYS_MACHK1 activated by the CPC710) . All errors that result from CPU initiated transfers are logged in this register. Errors resulting from transfers initiated by a PCI Master or by the DMA controller will result in BITs 17, 18, or 19 being set and require software to interrogate additional error registers in the PCI bridge logic and the DMA controller logic. BIT 16, CPU to PCI Bus error, will also require software to interrogate additional error registers in the PCI bridge logic.

The bits 22, 23, 24, 25 that are available for read after a CPU1 Machine Check interrupt have the same meaning as errors reported on bits 15, 16, 19, 21 for CPU0. Software is responsible for writing zeros to this register in order to clear the bits that are set.

Reset Value x'0000 0000'

Address x'FF00 1060'

Access Type Read/Write



Bit(s)	Description
0 - 1	**Reserved**
2	**Checkstop Error** 0: No Error 1: Device initiated checkstop occurred
3	**Flash Write Error** 0: No Error 1: Write to flash occurred when not enabled
4	**DMA Controller Access Error** 0: No Error 1: Access performed to DMA Controller when not enabled (see *DMA Global Control (GSCR)* on page 130*)*

Bit(s)	Description
5	**Access to Disabled System I/O Address Space Error** 0: No Error 1: Access performed to System I/O address space that is not enabled
6 - 12	**Reserved**
13	**Address Bus Parity Error** 0: No Error 1: 60x bus address parity error detected by device
14	**Data Bus Parity Error** 0: No Error 1: 60x bus data parity error detected by device
15	**Addressing Error Detected (for CPU 0)** 0: No Error 1: Addressing error
16	**CPU to PCI Bus Access Error (for CPU 0)** 0: No Error 1: Error occurred on PCI bus while servicing processor load/store request
17	**PCI-32 Bus Master Error** 0: No Error 1: Error occurred during PCI master initiated operation
18	**PCI-64 Bus Master Error** 0: No Error 1: Error occurred during PCI master initiated operation
19	**DMA Error (for CPU 0)** 0: No Error 1: Error occurred during DMA transfer
20	**Data Bus Timeout Error** 0: No Error 1: Indicates that the CPC710 has detected a 8ms time-out between DBG to last $\overline{\text{SYS\_TA}}$ or $\overline{\text{SYS\_TEA}}$. In this case of error the CPC710 activates also the $\overline{\text{CHKSTOP}}$ signal.
21	**CPU Access to Memory Error (for CPU 0)** 0: No Error 1: Error occurred during an access by the CPU to memory; Error logged in MESR and MEAR
22	**Addressing Error Detected (FOR CPU 1)** 0: No Error 1: Addressing error.
23	**CPU to PCI Bus Access Error (for CPU 1)** 0: No Error 1: Error occurred on PCI bus while servicing processor load/store request
24	**DMA Error (for CPU 1)** 0: No Error 1: Error occurred during DMA transfer
25	**CPU Access to Memory Error (for CPU 1)** 0: No Error 1: Error occurred during an access by the CPU to memory;
26 - 31	**Reserved**

System Error Address Register (SEAR)

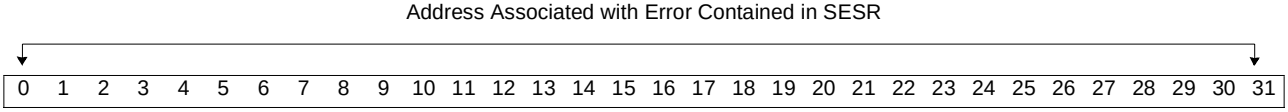
This register contains the CPU address associated with the error that is logged in the SESR register described previously. This register is only updated for errors that are due to CPU initiated transfers. The address for errors that result from transfers initiated by PCI masters or DMA controller are located in error registers contained in the PCI bridge logic or the DMA controller logic.

In the case of dual-processor implementation, this register will contain only the address of the first error detected.

Reset Value x'0000 0000'

Address x'FF00 1070'

Access Type Read/Write



Bit(s)	Description
0 - 31	**Address Associated with Error Contained in SESR**

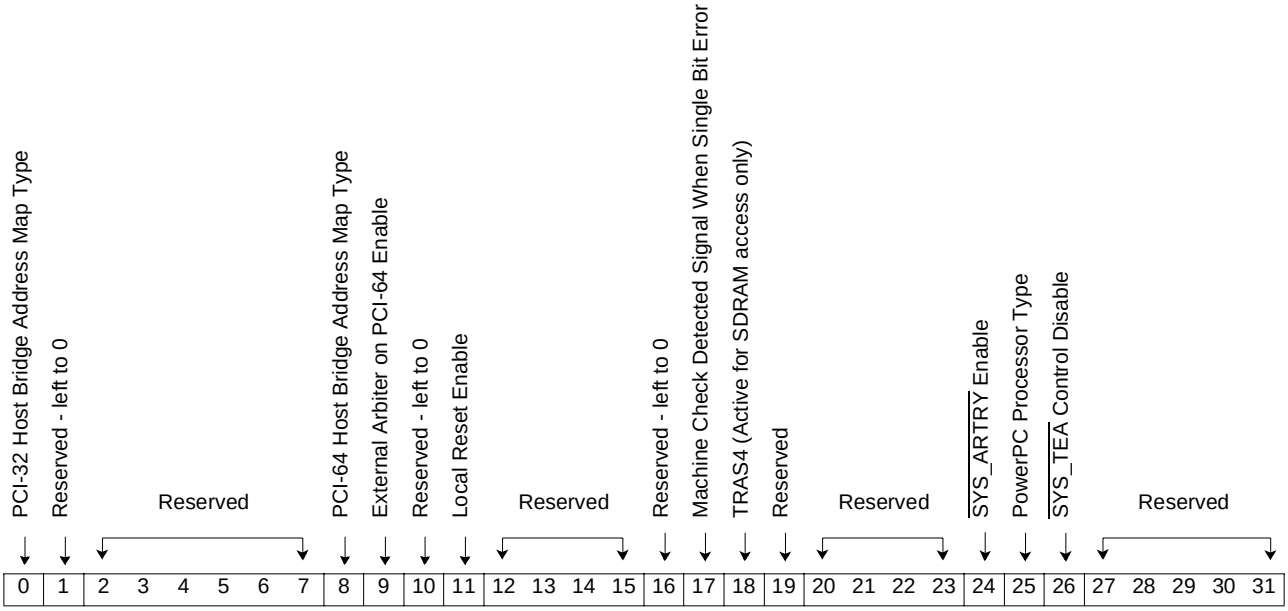
Chip Programmability Register (PGCHP)

This register allows the device to be programmed in order to provide additional functions.

Reset Value x'0000 0000'

Address x'FF00 1100'

Access Type Read/Write



Bit(s)	Description
0	**PCI-32 Host Bridge Address Map Type.** Controls which address map is used. See *Address Maps* on page 143. 0: PREP mode 1: FPHB mode (Specific Base address on PCI-32 is used)
1	**Reserved.** Must be left to 0.
2 - 7	**Reserved**
8	**PCI-64 Host Bridge Address Map Type.** Controls which address map is used. See *PCI to System Memory Addressing Model (FPHB Mode)* on page 148 0: PREP mode 1: FPHB mode (Specific Base address on PCI-64 is used)
9	**External Arbiter on PCI-64 Enable.** Read only status bit. 0: Internal arbiter is activated 1: Internal arbiter is deactivated
10	**Reserved.** Must be left to 0.
11	**Local Reset Enable** 0: PCI-64 Signal $\overline{\text{G\_RESETOUT}}$ is not driven 1: If external arbiter then $\overline{\text{G\_RESETOUT}}$ is driven.
12 - 15	**Reserved**

Bit(s)	Description
16	**Reserved.** R/W - Must be left to 0
17	**Machine Check Detected Signal When Single Bit Error** 0: $\overline{\text{SYS\_MACHK}}$ signal not driven 1: $\overline{\text{SYS\_MACHK}}$ signal is driven
18	**TRAS4 (Active for SDRAM access only)** 0: t_{RASmin} = 5 * clock 1: t_{RASmin} = 4 * clock
19	**Reserved**
20 - 23	**Reserved.** R/W
24	**Auto Retry Enable** 0: $\overline{\text{SYS\_ARTRY}}$ is not driven 1: $\overline{\text{SYS\_ARTRY}}$ is driven when the access is in Peripheral Memory or I/O space with potential deadlock
25	**PowerPC Processor Type** (see *Address Transfer Attribute for Snoop Register (ATAS)* on page 118). 0: 604 1: 750 or later version
26	**TEA Control Disable** 0: $\overline{\text{SYS\_TEA}}$ is driven 1: $\overline{\text{SYS\_TEA}}$ is not driven but Machine Check Signal is.
27 - 31	**Reserved**

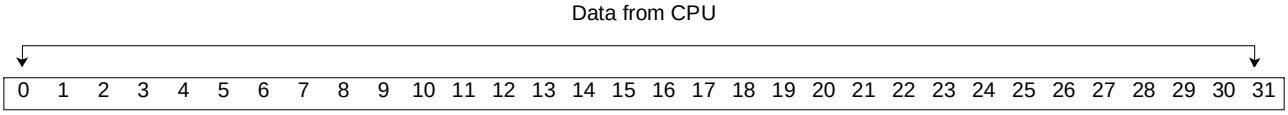
Free Register 1 (RGBAN1)

This register contains Data coming from the CPU.

Reset Value x'0000 0000'

Address x'FF00 1110'

Access Type Read/Write



Bit(s)	Description
0 - 31	**Data from CPU**

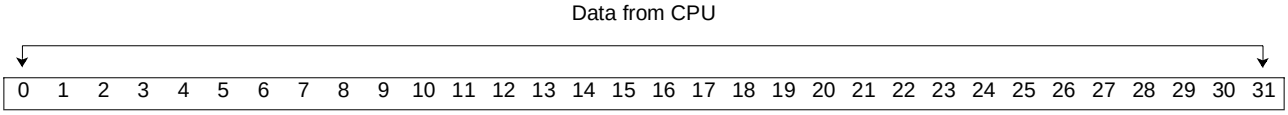
Free Register 2 (RGBAN2)

This register contains Data coming from the CPU.

Reset Value x'0000 0000

Address x'FF00 1120'

Access Type Read/Write



Bit(s)	Description
0 - 31	**Data from CPU**

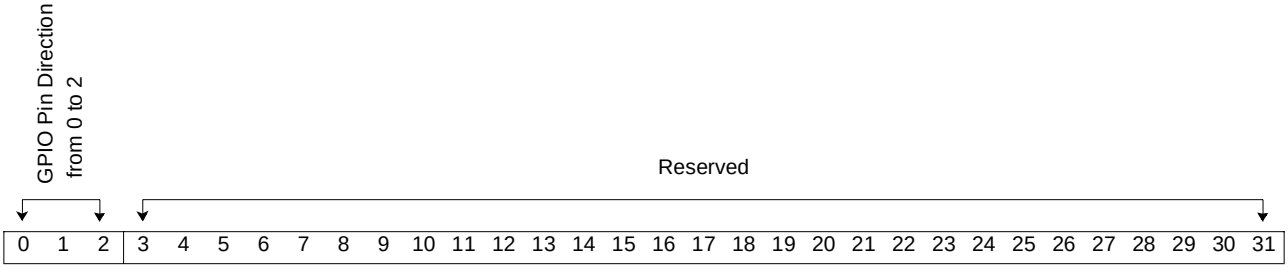
GPIO Direction Register (GPDIR)

This register sets the direction of signals (input or output) on pins GPIO0, GPIO1 and GPIO2.

Reset Value x'0000 0000

Address x'FF00 1130'

Access Type Read/Write



Bit(s)	Description0
0	**GPIO 0 Pin Direction** 0: Input 1: Output
1	**GPIO 1 Pin Direction** 0: Input 1: Output
2	**GPIO 2 Pin Direction** 0: Input 1: Output
3 - 31	**Reserved**

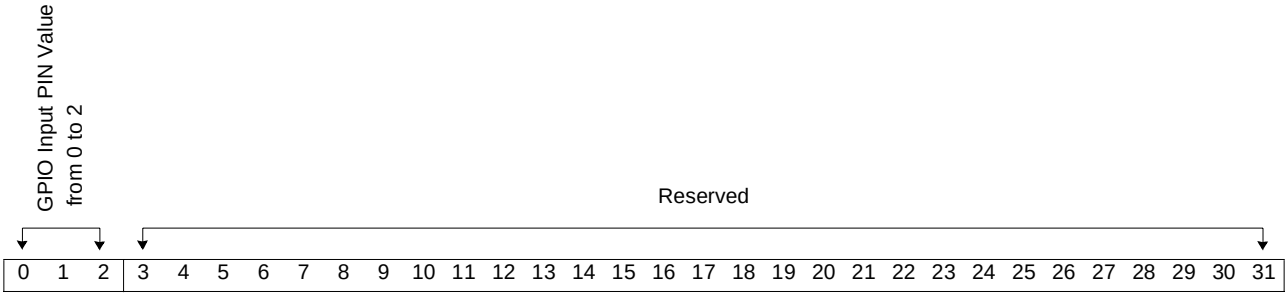
GPIO Input Value Register (GPIN)

This register stores values of the signal on pins GPIO0, GPIO1, GPIO2 if it is defined as input.

Reset Value x'0000 0000

Address x'FF00 1140'

Access Type Read Only



Bit(s)	Description
0	**GPIOO Input Pin Value**
1	**GPIO1 Input Pin Value**
2	**GPIO2 Input Pin Value**
3 - 31	**Reserved**

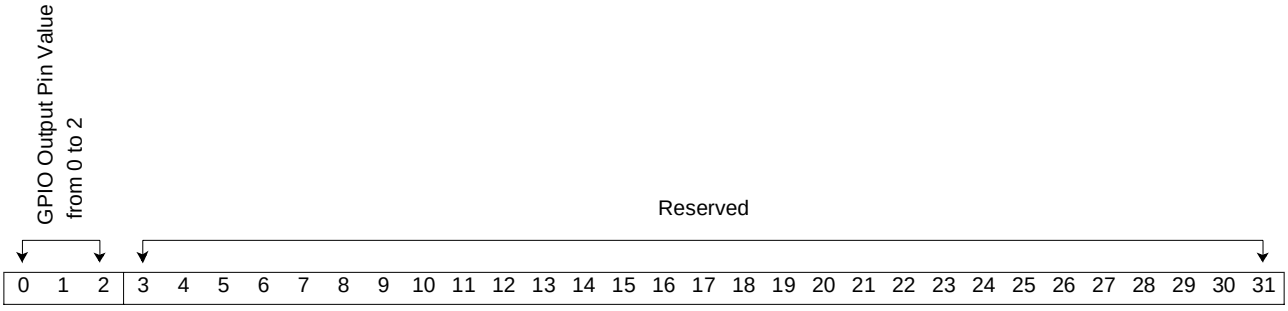
GPIO Output Value Register (GPOUT)

This register stores values of signal on pins GPIO0, GPIO1, GPIO2 if defined as output.

Reset Value x'0000 0000

Address x'FF00 1150'

Access Type Read/Write



Bit(s)	Description
0	**GPIOO Output Pin Value**
1	**GPIO1 Output Pin Value**
2	**GPIO2 Output Pin Value**
3 - 31	**Reserved**

Address Transfer Attribute for Snoop Register (ATAS)

This register contains SYS_TT, SYS_TSIZ and $\overline{\text{SYS\_TBST}}$ values that are used during a snoop transaction. These values can be changed according to the type of PowerPC processor. This register must be set if bit 25 of the PGCHP register is programmed to 1. See *Chip Programmability Register (PGCHP)* on page 111.

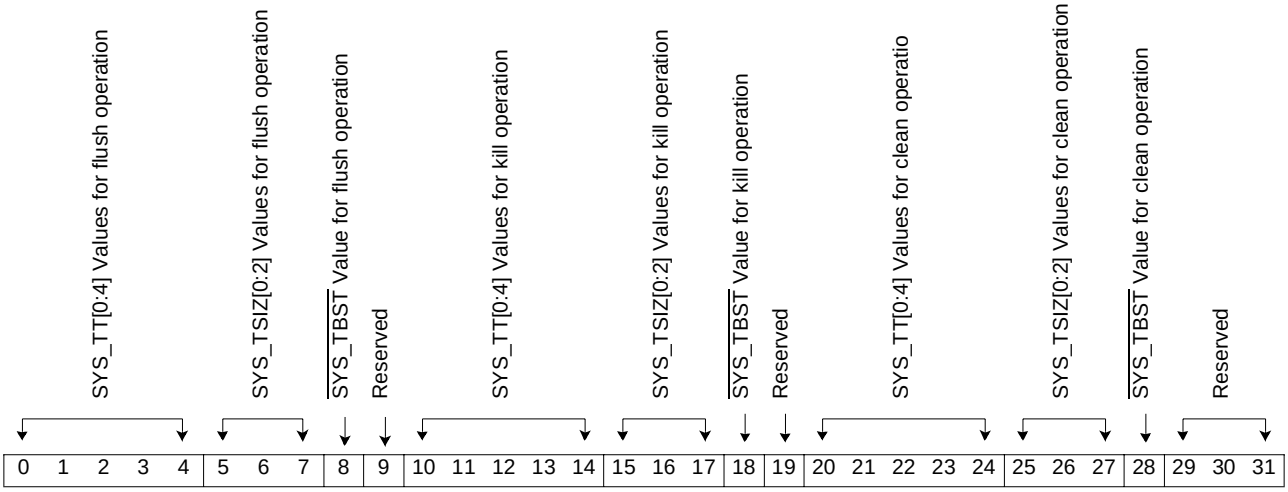
Reset Value x'0000 0000'

Address x'FF00 1160'

Access Type Read/Write

Programming Value x'709C 2508' b'0111 0000 1001 1100 0010 0101 0000 1000'

This setting is recommended for the PowerPC750 which is not able to perform Cache/memory coherency with Kill and Flush operation as the PowerPC 604.



Bit(s)	Description
0 - 4	**SYS_TT[0:4] Values for flush operation**
5 -7	**SYS_TSIZ[0:2] Values for flush operation**
8	**$\overline{\text{SYS\_TBST}}$ Value for flush operation**
9	**Reserved**
10 - 14	**SYS_TT[0:4] Values for kill operation**
15 - 17	**SYS_TSIZ[0:2] Values for kill operation**
18	**$\overline{\text{SYS\_TBST}}$ Value for kill operation**
19	**Reserved**
20 - 24	**SYS_TT[0:4] Values for clean operation**
25 - 27	**SYS_TSIZ[0:2] Values for clean operation**
28	**$\overline{\text{SYS\_TBST}}$ Value for clean operation**
29 - 31	**Reserved**

programming the ATAS (Address Transfer Attribute for Snoop Register) register:

When the CPC710-100 generates the following snoop cycle with TT signal on the 60x bus, the PowerPC 750 takes no action. At the difference with the PowerPC604, the PowerPC 750 does not handle cache/system memory coherency.

TT[0:4]OperationAnswer from the 750

TT[0:4]	Operation	Answer from the 750
00000	Clean Sector	No action
00100	Flush Sector	No action
01100	Kill sector	No action

To verify the coherency between Cache and System memory, with a PowerPC 750, it is necessary for the CPC710 bridge chip to modify the TT[0:4] and thus oblige the PowerPC750 to react on snoop operations with the Address only cycles on the 60x bus.

It is possible to program the ATAS register such that the Clean, Flush, Kill code are modified in a "Snoop" code for PowerPC750

Typical changes of TT[0:4] code for the PowerPC750

Clean	TT[0:4]= 00000	-> Read	TT[0:4]	= 01010
Flush	TT[0:4]= 00100	-> RWITM	TT[0:4]	= 01110
Kill	TT[0:4]= 01100	-> RWITM	TT[0:4]	= 01110

After modification, to perform Cache/Memory coherency, the new Address only cycles are:

TT[0:4]OperationAnswer from the 750

TT[0:4]	Operation	Answer from the 750
01010	Read	Flush or Kill
01110	RWITM	Flush or Kill
01110	RWITM	Flush or Kill

Typical ATAS programming: ATAS[0:31]= 0x709C2508

TSIZ[0:2] and TBST can be programmed on the Address only cycles to the following recommended values TSIZ[0:2]= 000 et TBST = 1

Flush modification to RWITM:
ATAS[0:4] <= 01110
ATAS[5:7] <= 000
ATAS[8] <= 1

Kill modification to RWITM
ATAS[10:14] <= 01110
ATAS[15:17] <= 000
ATAS[18] <= 1

Clean modification to READ
ATAS[20:24] <= 01010
ATAS[25:27] <= 000
ATAS[28] <= 1

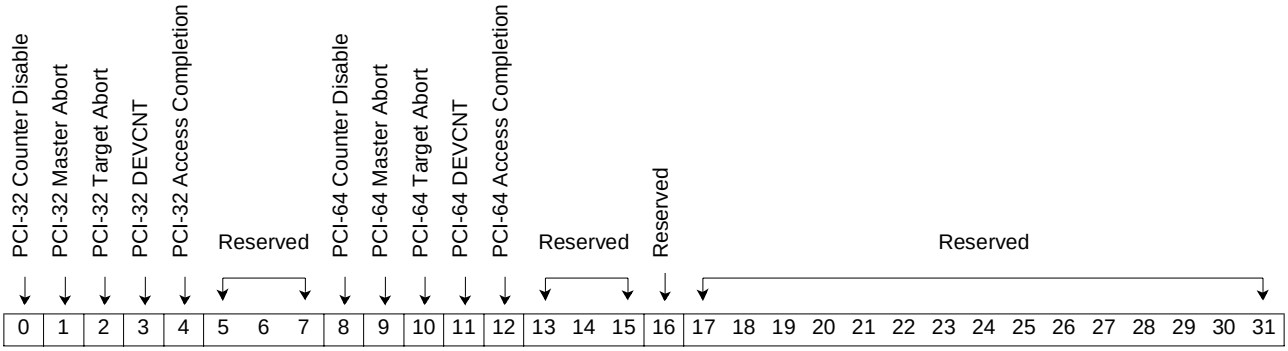
The modification is active only if bit 25 of the PGCHP is set to 1
PGCHP[25]= 1 - @FF001100 (Processor type 750 on).

Diagnostic Register (AVDG)

Reset Value x'0000 0000'

Address x'FF00 1170'

Access Type Read/Write



Bit(s)	Description
0	**PCI-32 Counter Disable** 0: TRDY, IRDY, DISCNT Counters are not activated (The MSB is always at 1) (he DISCNT counter is programmable - see *Disconnect Counter (DISCNT)* on page 59 1: TRDY, IRDY, DISCNT Counters are ACTIVATED
1	**PCI-32 Master Abort** 0: Window of Master Abort is reduced to one cycle (avoid parasitic master abort detection) 1: Window of Master Abort is not reduced
2	**PCI-32 Target Abort** 0: Device detects Target abort (The Frame output is taken) 1: Device never detects Target Abort but retry indefinitely accesses
3	**PCI-32 DEVCNT** 0: Stop the counter down when devsel is detected 1: Do not stop the counter down
4	**PCI-32 Access Completion** 0: The completion is activated when device is master and not during external exchange 1: The completion appears when the data is not the last one.
5-7	**Reserved**
8	**PCI-64 Counter Disable** 0: TRDY, IRDY, DISCNT Counters are not activated (the MSB is always at 1) DISCNT counter is programmable - see *Disconnect Counter (DISCNT)* on page 59 1: TRDY, IRDY, DISCNT Counters are ACTIVATED
9	**PCI-64 Master Abort** 0: Window of Master Abort is reduced to one cycle (avoid parasitic master abort detection) 1: Window of Master Abort is not reduced
10	**PCI-64 Target Abort** 0: Device detects Target abort (The Frame output is taken) 1: Device never detects Target Abort but retry indefinitely accesses

Bit(s)	Description
11	**PCI-64 DEVCNT** 0: Stop the counter down when devsel is detected 1: No stop the counter down
12	**PCI-64 Access Completion** 0: The completion is activated when device is master and not during external exchange 1: The completion appears when the data is not a last.
13 - 15	**Reserved**
16	**Reserved -** Must be left to 0
17 - 31	**Reserved**

Memory Controller Control Register (MCCR)

This register provides the primary control for the memory controller logic.

Reset Value x'0000 0000'

Address x'FF00 1200'

Access Type Read/Write

Bit(s)	Description
0	**Global System Memory Address Space Enable** 0: Device will not respond to addresses specified in Memory Configuration Extent Register (MCERx) 1: System memory address space enabled.
1	**Diagnostic Mode** 0: Normal Mode: Multi-bit ECC error will generate Machine Check 1: Diagnostic Mode: Multi-bit ECC does NOT generate Machine Check; logged in MEAR & MESR
2	**SDRAM Initialization Status (read-only)** 0: SDRAM initialization is not completed. 1: SDRAM initialization is completed.
3 - 4	**ECC Mode** 00: Normal generation and checking of ECC codes 01: ECC check disabled; Byte lane 0 routed to/from ECC check field. Data byte 0 forced to all zeros 10: ECC check disabled; Normal routing of data and normal ECC code generation 11: Reserved
5 - 7	**Row Cycle Time for SDRAM Auto-refresh (t_{RC})** 000: 5 bus cycles 001: 6 bus cycles 010: 7 bus cycles 011: 8 bus cycles 100: 9 bus cycles 101: 10 bus cycles 110: 11 bus cycles 111: 12 bus cycles
8 - 9	**DRAM Type** **Must be set to 10 for SDRAM**
10	**Data Pacing Mode** (Must be set to 1 for SDRAM)
11	**Data Mask Mode** (SDRAM only) 0: 16 $\overline{\text{SDCS}}$ are available. DQM pin of SDRAM devices must be grounded. 1: Only eight $\overline{\text{SDCS}}$ are available. DQM signal is present on the eight other pins. Activate to prevent tRDL violation during "Write interrupted by pre-charge" operations if the device does not guarantee that the data presented in the same time as pre-charge is not properly ignored.
12 - 15	**Reserved.** Must be left to 0
16	**Reserved.** Must be set to 0
17	**Reserved.** Must be set to 1
18	**Reserved.** Must be set to 1
19	**Reserved.** Must be set to 0
20	**Reserved.** Must be set to 0

Bit(s)	Description
21 - 22	**Reserved.** Must be set to b'00'
23	**Reserved.** Must be set to 0
24 - 29	**Reserved**
30	**Disable Page Mode** 0: Memory controller will perform fast page accesses for back to back operations if appropriate 1: Memory controller will perform fast page access only within a burst operation. It will NOT perform fast page accesses for back to back bursts even if they occur to the same $\overline{RAS}$ page.
31	**Disable Queue Same Page Override** 0: Memory queue ordering can be overidden if an operation is to the same page. 1: Memory queue always processed in order received.

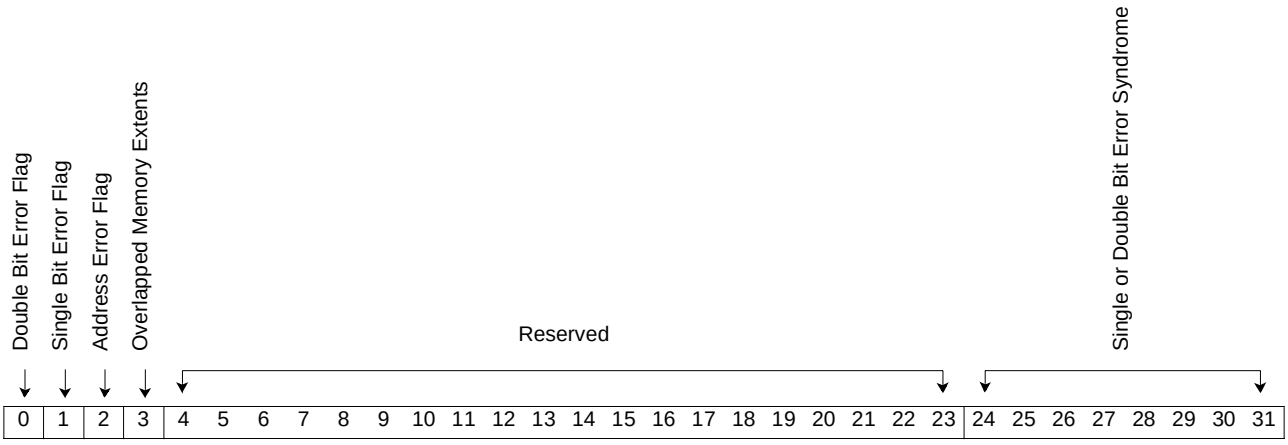
Memory Error Status Register (MESR)

This register provides error status information on memory errors. In order to log additional errors, software must clear the register by writing zeros throughout.

Reset Value x'0000 0000'

Address x'FF00 1220'

Access Type Read/Write



Bit(s)	Description
0	**Double Bit Error Flag** 0: No Error 1: Double Bit ECC error occurred
1	**Single Bit Error Flag** 0: No Error 1: Single bit ECC error occurred
2	**Address Error Flag** 0: No Error 1: Address error occurred
3	**Overlapped Memory Extents** 0: No Error 1: An access occurred to an address that is mapped in two different memory configuration extents.
4 - 23	**Reserved**
24 - 31	**Single or Double Bit Error Syndrome.** Used to determine the failing DIMM

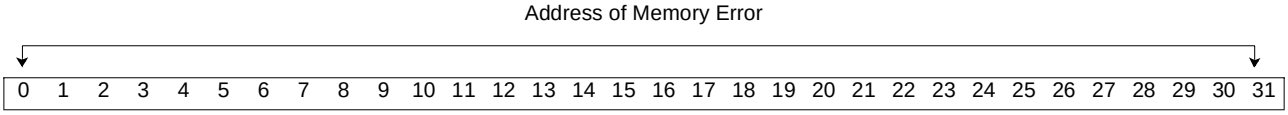
Memory Error Address Register (MEAR)

This register contains the address associated with the error logged in the MESR.

Reset Value x'0000 0000'

Address x'FF00 1230'

Access Type Read/Write



Bit(s)	Description
0 - 31	**Address of Memory Error**

Memory Configuration Extent Registers (MCER[0:7])

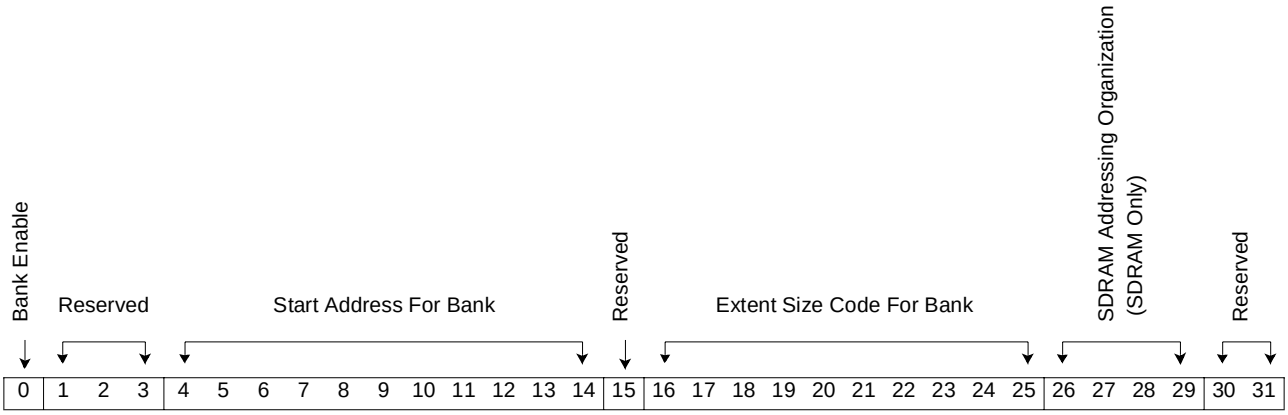
Each one of the eight registers MCER0 to MCER7 defines one of up to eight banks of memory (Bank 0 to 7) supported. All registers have the same definition, and each defines the size and location for the particular bank of memory.

Reset Value x'0000 0000'

Address

MCER0	x'FF00 1300'
MCER1	x'FF00 1310'
MCER2	x'FF00 1320'
MCER3	x'FF00 1330'
MCER4	x'FF00 1340'
MCER5	x'FF00 1350'
MCER6	x'FF00 1360'
MCER7	x'FF00 1370'

Access Type Read/Write



Bit(s)	Description
0	**Bank Enable** 0: Bank is not present or register initialization is not complete 1: Bank is present and decoded by device
1 - 3	**Reserved**. R/W
4 - 14	**Start Address For Bank.** (Bits 0 - 10) - Defines the beginning address of this bank. Contains bits 0 - 10 of the 32-bit real address. Address restricted to a boundary equal to the size of the bank.
15	**Reserved**. R/W

Bit(s)	Description
16 - 25	**Extent Size Code For Bank** x'3FF' Reserved x'3FB' Reserved x'3F3' 4 MB x'3E3' 8 MB x'3C3' 16 MB x'383' 32 MB x'303' 64 MB x'203' 128 MB x'003' 256 MB x'002' 512 MB x'000' 1 GB
26 - 29	**SDRAM Addressing Organization** b'0001': 11/ 8/2 (Row/Col/Bank select) Address lines b'0010': 11/ 9/1 b'0011': 11/10/1 b'0100': 12/ 8/2 b'0101': 12/10/2 b'0110': 13/ 8/1 b'0111': 13/ 8/2 b'1000': 13/ 9/1 b'1001': 13/10/1 b'1010': 11/ 8/1 b'1011': 12/ 8/1 b'1100': 12/9/1 b'0000': All other supported organizations (see *Supported SDRAM Organizations* on page 177)
30 - 31	**Reserved**

System I/O Register 0 (SIOR0)

This register is user defined.

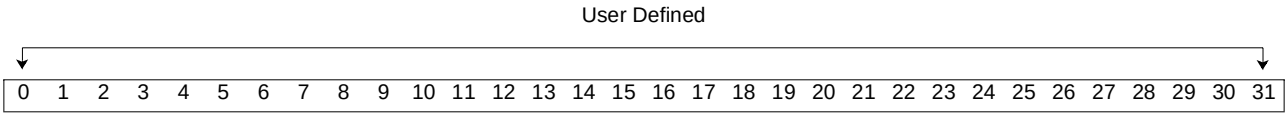
However it has been introduced in the PowerPC chip support to provide the memory DIMM presence detect pins for all four pairs of DIMM sockets. Device supports a maximum of four pairs, or eight DIMMs. The DIMM pairs must be of exactly the same type and therefore only one DIMM presence detect pin of each pair are read in from this register.

The read of this register results in the assertion of the $\overline{\text{PRES\_OE0}}$ signal and a Read cycle through the PCI32 A/D lines. That permits a read of the outside buffers containing the presence detect bits.
Bit 0 of this register correspond to bit 31 on the PCI A/D lines.

Reset Value x'0000 0000'

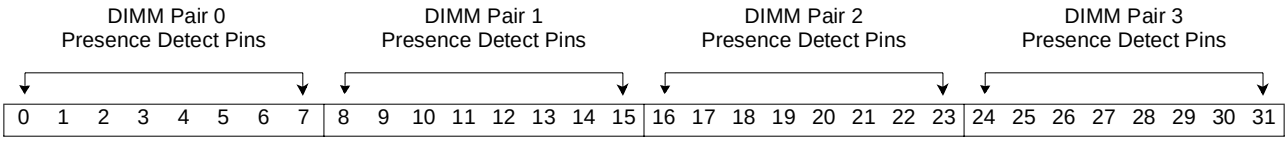
Address x'FF00 1400'

Access Type Read Only



Bit(s)	Description
0 - 31	**User-defined**

Example of usage:



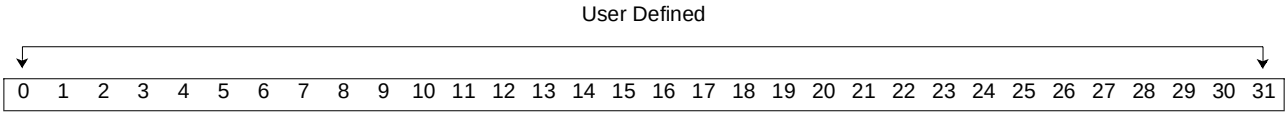
Bit(s)	Description
0 - 7	**DIMM Pair 0 Presence Detect Pins: PD1-PD8**
8 - 15	**DIMM Pair 1 Presence Detect Pins: PD1-PD8**
16 - 23	**DIMM Pair 2 Presence Detect Pins: PD1-PD8**
24 - 31	**DIMM Pair 3 Presence Detect Pins: PD1-PD8**

Please see *MCER Register Initialization* on page 182 for PD definition and device supported values.

System I/O Register 1 (SIOR1)

Address x‘FF00 1420’

Access Type Read Only



Bit(s)	Description
0 - 31	**User-defined**

The read of this register results in the assertion of the $\overline{\text{PRES\_OE1}}$ signal and a Read cycle through the PCI32 A/D lines. That permits a read of the outside buffers containing the presence detect bits.

Bit 0 of this register correspond to bit 31 on the PCI A/D lines.

DMA Registers Space

DMA Global Control (GSCR)

Reset Value x'0000 0000'

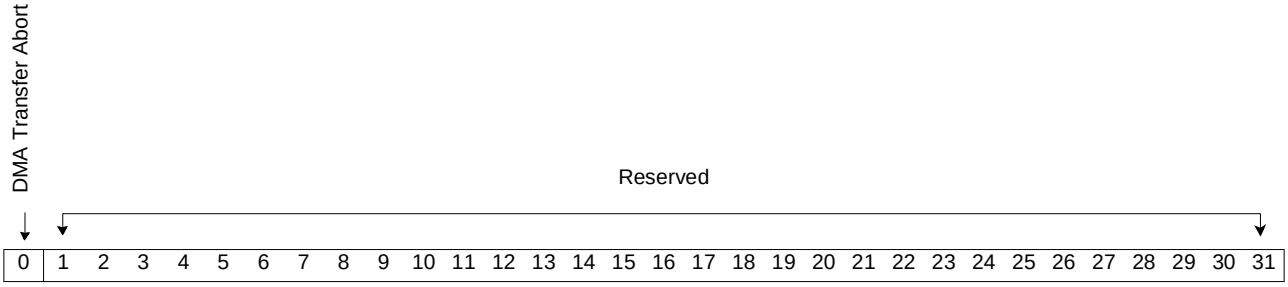
Address	User	x'FF1C 0020'
	Privileged	x'FF1E 0020'
Access Type	User	Read Only
	Privileged	Read/Write

Bit(s)	Description
0 - 1	**DMA Transfer Enable** 00: Reset DMA Controller to default power up mode. 10: DMA Controller disabled 11: DMA Controller is enabled
2	**Reserved.** R/W
3	**Interrupt Enable** When set, generates an interrupt at the completion of a DMA transfer 0: IT2 disabled 1: IT2 enabled
4	I**nterrupt Status** 0: End of DMA transfer interrupt IT2 not asserted 1: End of DMA transfer interrupt IT2 asserted Software must write a 0 to Reset the IT2 Interrupt
5	**Extended DMA Transfer Enable** 0: Single DMA transfer 1: Extended DMA
6	**Direction for Extended DMA** 0: PCI to MEMORY ('ECOWX') 1: MEMORY to PCI ('ECIWX')
7	**Reserved.** R/W - Must be left to 0
8 - 15	**Reserved.** RO
16 - 31	**Number of DMA Transfer Loops to Do** (if Extended DMA is enabled). During an Extended DMA, contains the number of loops remaining. If bit 5 is enabled and this register is set to 0, the mode is a single DMA transfer.

DMA Global Status (GSSR)

Reset Value x'0000 0000

Address	User	x'FF1C 0030'
	Privileged	x'FF1E 0030'
Access Type	User	Read Only
	Privileged	Read Only



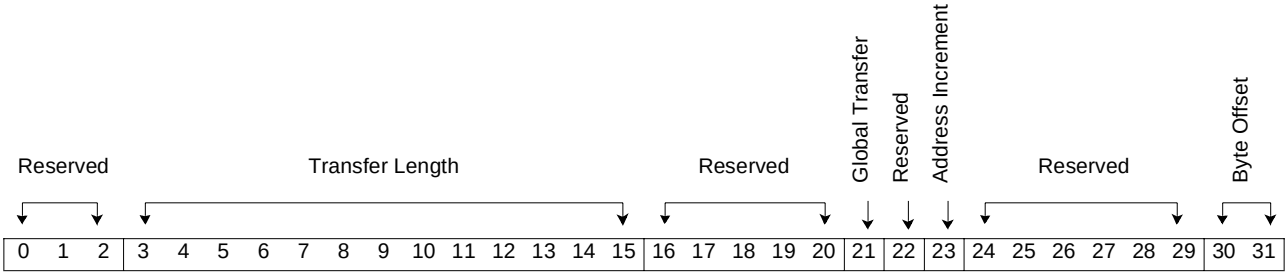
Bit(s)	Description
0	**DMA Transfer Aborted** 0: No Error 1: DMA transfer aborted
1 - 31	**Reserved**

DMA Transfer Control (XSCR)

Reset Value x'0000 0000'

Address User x'FF1C 0040'
Privileged x'FF1E 0040'

Access Type Read/Write
(User and Privileged)



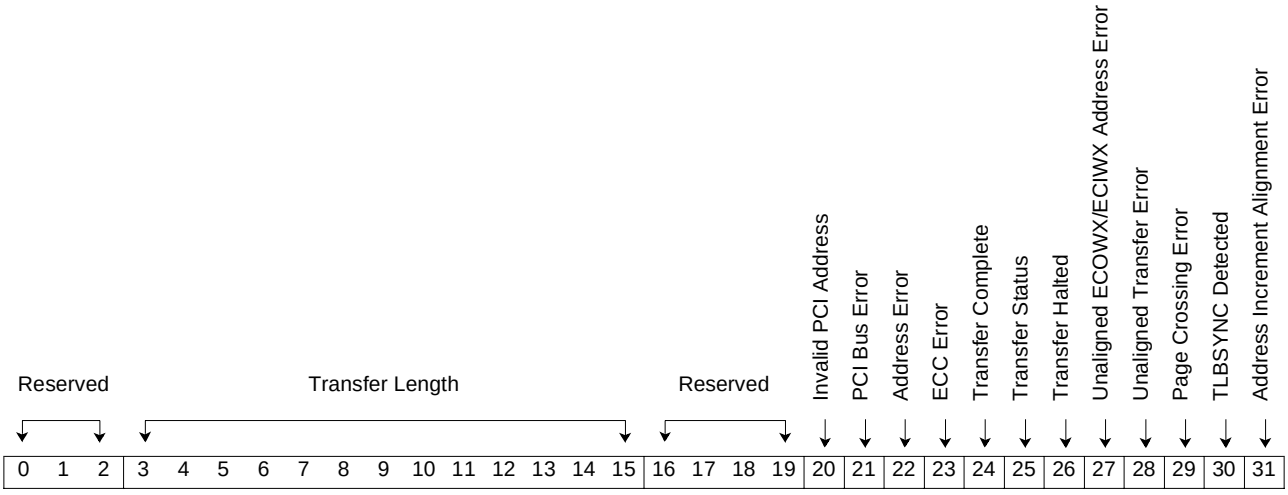
Bit(s)	Description
0 - 2	**Reserved**
3 - 15	**Transfer Length.** Contains the number of Bytes to be transferred in a Loop (maximum is 4 K). A value of 0 will transfer 0 bytes.
16 - 20	**Reserved**
21	**Global Transfer** 0: No snoop operations required for accesses to system memory 1: Accesses to system must be coherent
22	**Reserved**
23	**Address Increment** 0: Do NOT increment I/O address during DMA transfer 1: Increment I/O address during DMA transfer
24 - 29	**Reserved**
30 - 31	**Byte Offset**. Specifies the byte offset associated with the DMA transfer real address

DMA Transfer Status (XSSR)

Reset Value x'0000 0000'

Address User x'FF1C 0050'
Privileged x'FF1E 0050'

Access Type Read Only
(User and Privileged)



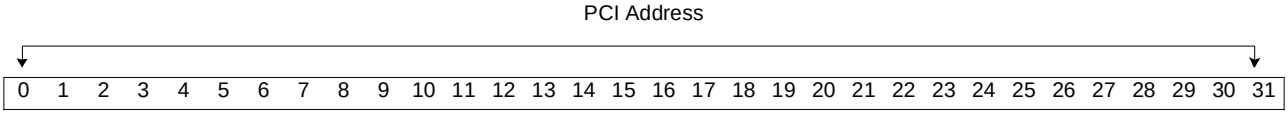
Bit(s)	Description
0 - 2	**Reserved**
3 - 15	**Transfer Length**. Contains the number of bytes remaining when the transfer was completed or aborted.
16 - 19	**Reserved**
20	**Invalid PCI Address** 1: XPAR did not match any PCI extents
21	**PCI Bus Error** 1: Error detected during PCI bus transaction. No Interrupt generated, only MACHK (Machine Check Signal) is activated.
22	**Address Error** 1: Invalid memory address detected (no Interrupt generated, only MACHK is activated)
23	**ECC Error** 1: Double-bit ECC error detected in memory (no Interrupt gen., only MACHK is activated)
24	**Transfer Complete** 0: Transfer is not complete 1: Transfer is complete for Memory Interface
25	**Transfer Status** 0: No DMA transfer in progress 1: DMA transfer operation is underway
26	**Transfer Halted** 1: DMA transfer operation in progress was halted due to start of second DMA transfer operation

Bit(s)	Description
27	**Unaligned ECOWX/ECIWX Address Error** 1: Address associated with ECOWX/ECIWX is not word aligned
28	**Unaligned Transfer Error** 1: Address alignment error
29	**Page Crossing Error** 1: Page Crossing detected during DMA transfer
30	**TLBSYNC Detected** 0: No TLBSYNC Detected 1: TLBSYNC detected during DMA transfer Transfer
31	**Address Increment Alignment Error** 1: Improper alignment of addresses when Address Increment bit is off

DMA Transfer PCI Address (XPAR)

Reset Value x'0000 0000'

Address	User	x'FF1C 0070'
	Privileged	x'FF1E 0070'
Access Type	User	Read Only, Read/Write
	Privileged	Read/Write, Read Only



Bit(s)	Description
0 - 31	**PCI Address**. Contains the adapter I/O address for the DMA transfer operation.

DMA Transfer Write Back Address (XWAR)

Reset Value x'0000 0000'

Address	User	x'FF1C 0090'
	Privileged	x'FF1E 0090'
Access Type	User	Read Only
	Privileged	Read/Write

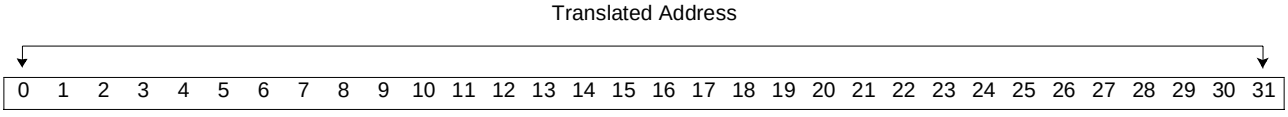
Bit(s)	Description
0 - 25	**Writeback Address.** Contains the real address used by the device at the end of the DMA transfer operation to which the completion status is written.
26 - 31	**Reserved** (assumed to be zero).

DMA Transfer Translated Address (XTAR)

Reset Value x'0000 0000'

Address User x'FF1C 00A0'
Privileged x'FF1E 00A0'

Access Type Read Only
(User and Privileged)



Bit(s)	Description
0 - 31	**Translated Address.** Contains the 32-bit real address presented on the processor bus during the ecowx/eciwx transfer.

System Standard Configuration Registers Space

The registers listed in this section can only be accessed from the PowerPC processor. Access through the 60x bus directed to the specific PCI bridge permits the PCI configuration. Both PCI bridges inside device must be configured before any PCI configuration cycles can be issued. The primary purpose of these registers is to provide a mechanism for firmware to identify the PCI bridge and the DCR and DID registers, and to assign a 1 MB address space in the system memory map for the location of the PCI bridge facilities (BAR register).

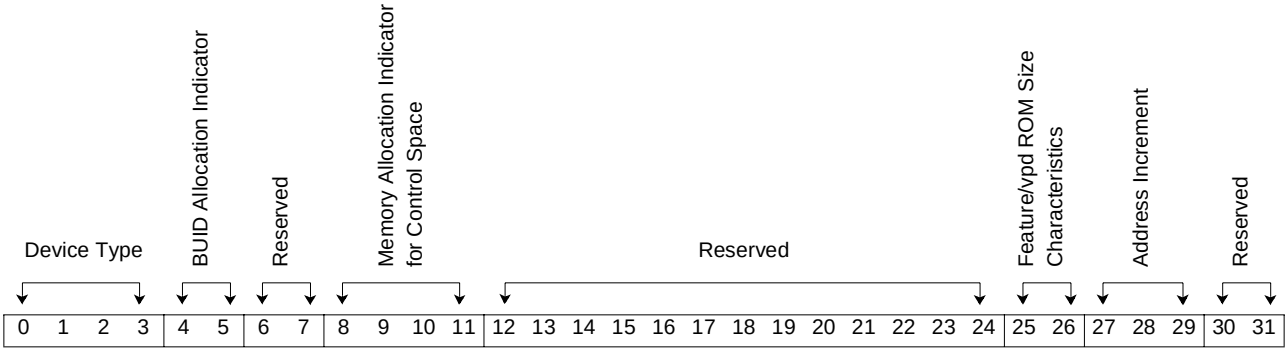
Device Characteristics Register (DCR)

This register identifies the type of device present.

Reset Value x'3010 0000'

Address x'FF20 0000'

Access Type Read Only



Bit(s)	Description
0 - 3	**Device Type** 0011: I/O device
4 - 5	**BUID Allocation Indicator** 00: NO BUID Required
6 - 7	**Reserved**
8 - 11	**Memory Allocation Indicator for Control Space** 0001: 1 MB
12 - 24	**Reserved**
25 - 26	**Feature/vpd ROM Size Characteristics** 00: No Feature/VPD ROM present
27 - 29	**Address Increment** 000: 4-byte Increment
30 - 31	**Reserved**

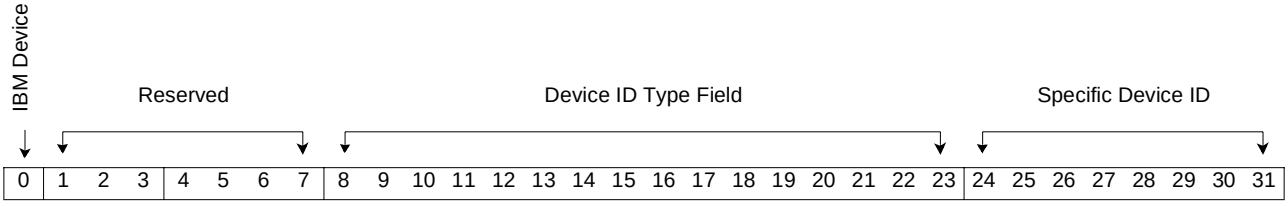
Device ID Register (DID)

This register provides specific device type information.

Reset Value x'08020 1100'

Address x'FF20 0004'

Access Type Read Only



Bit(s)	Description
0	**IBM Device** 1: IBM Device
1 - 7	**Reserved**
8 - 23	**Device ID Type Field**: x'2011': Device PCI Bridge
24 - 31	**Specific Device ID**: x'00': PCI-32 bridge x'01': PCI-64 bridge

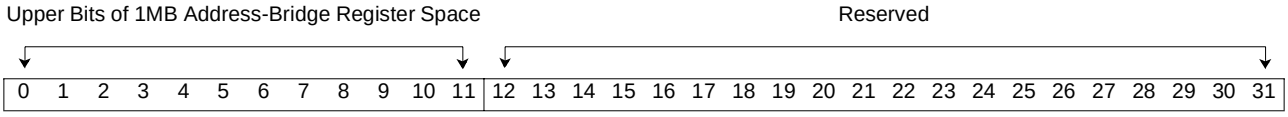
Base Address Register (BAR)

This register is written by software to indicate to the PCI bridge where its register space is located in the 4 GB system addressing space. There are no restrictions placed on the value of this register other than it must not overlap other extents defined for the system.

Reset Value x'0000 0000

Address x'FF20 0018'

Access Type Read/Write



Bit(s)	Description
0 - 11	**Upper Bits of 1 MB Address for Bridge Register Space**
12 - 31	**Reserved.** (Assumed to be X'0 0000')
Note: The start address is assumed to be on a 1 MB boundary.	

Device Specific Configuration Space

The registers listed in this section can only be accessed with configuration cycles on the 60x bus directed to the specific PCI bridge.

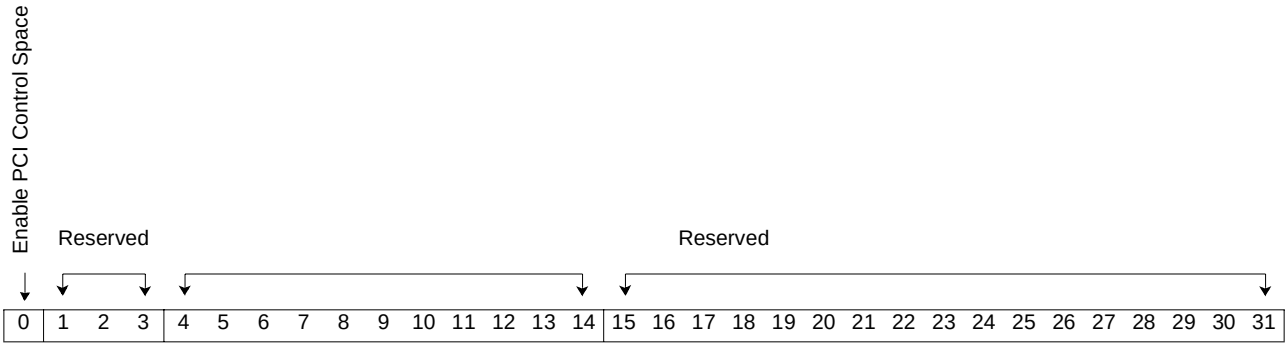
PCI BAR Enable Register (PCIENB)

This register provides a mechanism for software to disable the PCI bridge logic from decoding the address space pointed to by the BAR. This register is primarily used at power on time when the BAR has not been initialized.

Reset Value x'0000 0000'

Address x'FF20 1000'

Access Type Read/Write



Bit(s)	Description
0	**Enable PCI Control Space** 0: PCI Bridge only responds to configuration cycles 1: PCI Bridge responds to address space specified in the BAR register
1 - 3	**Reserved**
4 - 31	**Reserved**

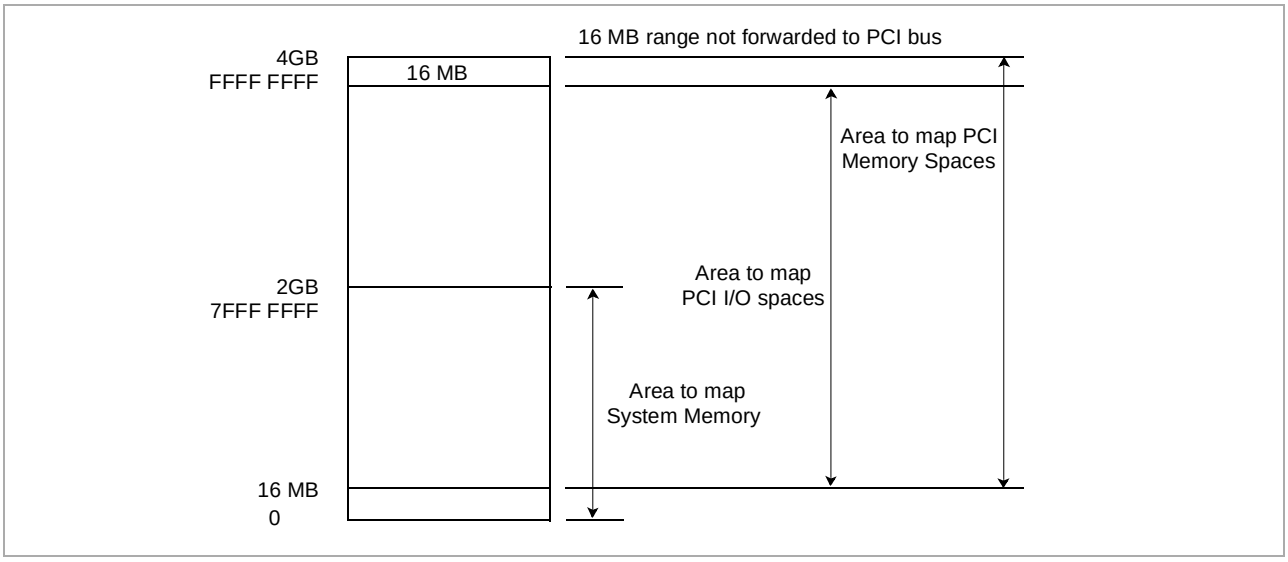
Addressing Model

Address Maps

Device address spaces can be programmed using the PowerPC REference Platform specification (PREP) Mode. Also a highly Flexible PCI Host Bridge (FPHB) Mode is available.The address map is highly programmable in either mode.
The following restrictions must be observed when programming the device:

- The upper 16 MB is reserved for ROM, system configuration, DMA controller, etc. See *System Register Space (x'FF00 0000' to x'FFFF FFFF')* on page 33 for the definition of this address space. Only PCI Memory spaces are allowed to overlap this area, however, they are not forwarded to the PCI bus.
- At least 1MB of system memory must be available at address 0. Minimum granularity of DIMMs is 16MB.
- System memory cannot be located above 2GB. Access in the upper 2GB is not checked by the CPC710 and result is unpredictable.
- Avoid overlapping system memory extents with PCI extents. Hang conditions and unpredictable results can occur if a processor accesses an address contained in two different extents.

Memory Map

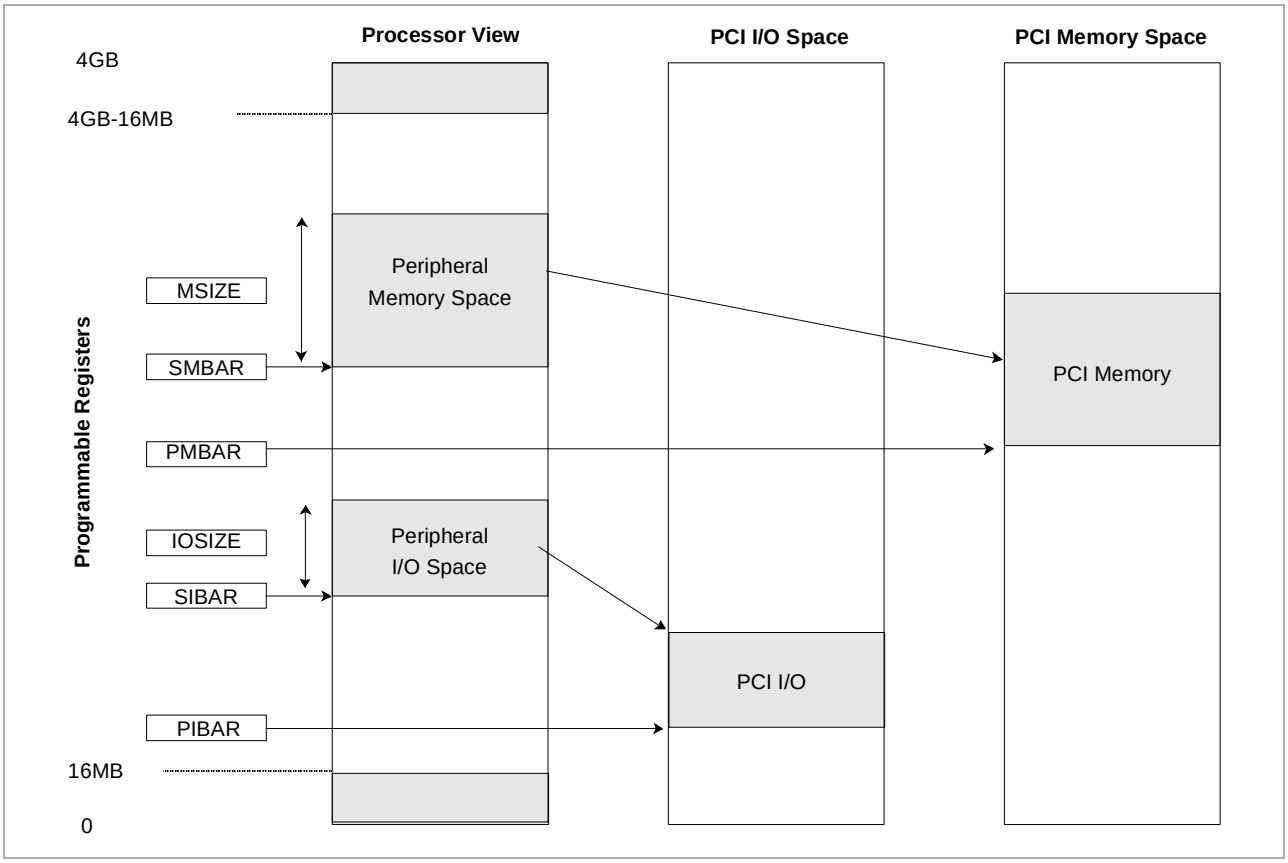


CPU to PCI Addressing Model

PREP and FPHB Modes

Programmable registers in the Specific PCI Host Bridge Space (*Specific PCI Host Bridges Space (BAR + x'000F 6110' to BAR +x'000F 9810')* on page 33) map PCI Memory and PCI I/O address spaces into the 4 GB System address space. Each PCI bridge in the device contains a set of these registers, allowing firmware to program PCI address spaces anywhere in memory rather than at fixed PCI address spaces. For example, the registers can be configured to have PCI address spaces in system memory follow the fixed Sandalfoot PCI addressing model.

CPU to PCI Addressing Model (PREP and FPHB Modes)



As the figure above shows, the device monitors addresses on the processor bus to determine whether a CPU address falls within the ranges specified by the following SMBAR/MSIZE and SIBAR/IOSIZE registers:

- *PCI Memory Address Space Size (MSIZE)* on page 74
- *System Base Address for PCI Memory (SMBAR)* on page 76
- *PCI I/O Address Space Size (IOSIZE)* on page 75
- *System Base Address for PCI I/O (SIBAR)* on page 77

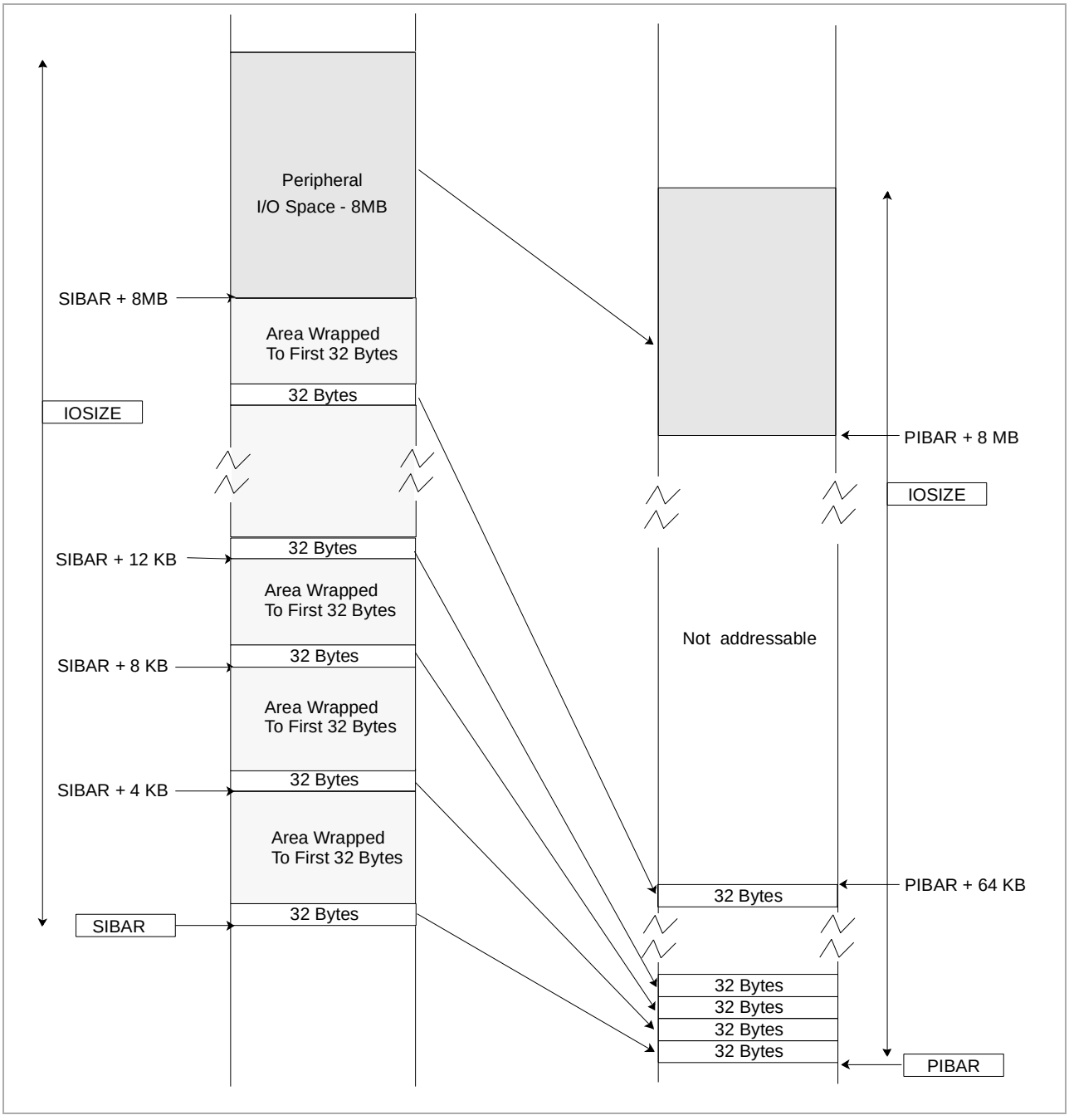
If the address falls within one of these ranges, the 60x interface logic passes the address and command to the appropriate PCI bridge logic for execution using the translation specified by the following PMBAR or PIBAR registers:

- *PCI Base Address for I/O (PIBAR)* on page 68
- *PCI Base Address for Memory (PMBAR)* on page 69

Peripheral I/O Address Translation

The first 8MB of Peripheral I/O space requires additional translation. To prevent 32-byte granularity accesses to ISA addresses, the device supports a noncontiguous I/O address mode in which the first 64KB of PCI bus I/O space is divided into 32byte segments spaced at 4K intervals within system memory. This mode is selected by bit 5 of *PHB Configuration Register (CTLRW)* on page 78).

Noncontiguous I/O Address Mode Enabled



PCI to System Memory

Two types of address mapping modes are available: PowerPC Reference Platform (PREP) Mode and Flexible PCI Host Bridge (FPHB) Mode. To select a mode, program bit 0 (for PCI-32) or bit 8 (for PCI-64) in the *Chip Programmability Register (PGCHP)* on page 111.

PowerPC Reference Platform (PREP) Mode

In PREP Mode, access from the PCI to the system can be performed with or without PCI address translation. When translation is used, the most common method is to translate addresses by complementing the upper 12 bits. PCI addresses ranging from x'8000 0000' to x'FFFF FFFF' are translated to system memory addresses x'0000 0000' to x'7FFF FFFF'.

In this mode only PCI access to Memory are decoded by the CPC710; Configuration and I/O are not decoded.

PCI Master Address Operation

Whenever the PCI bridge logic identifies addresses coming from ISA Masters (when the P_ISA_MASTER signal is active =1), they are passed directly to system memory. Otherwise, the untranslated addresses are checked to determine whether they fall within a bridge's PCI memory address range by comparing the PCI address to the following registers:

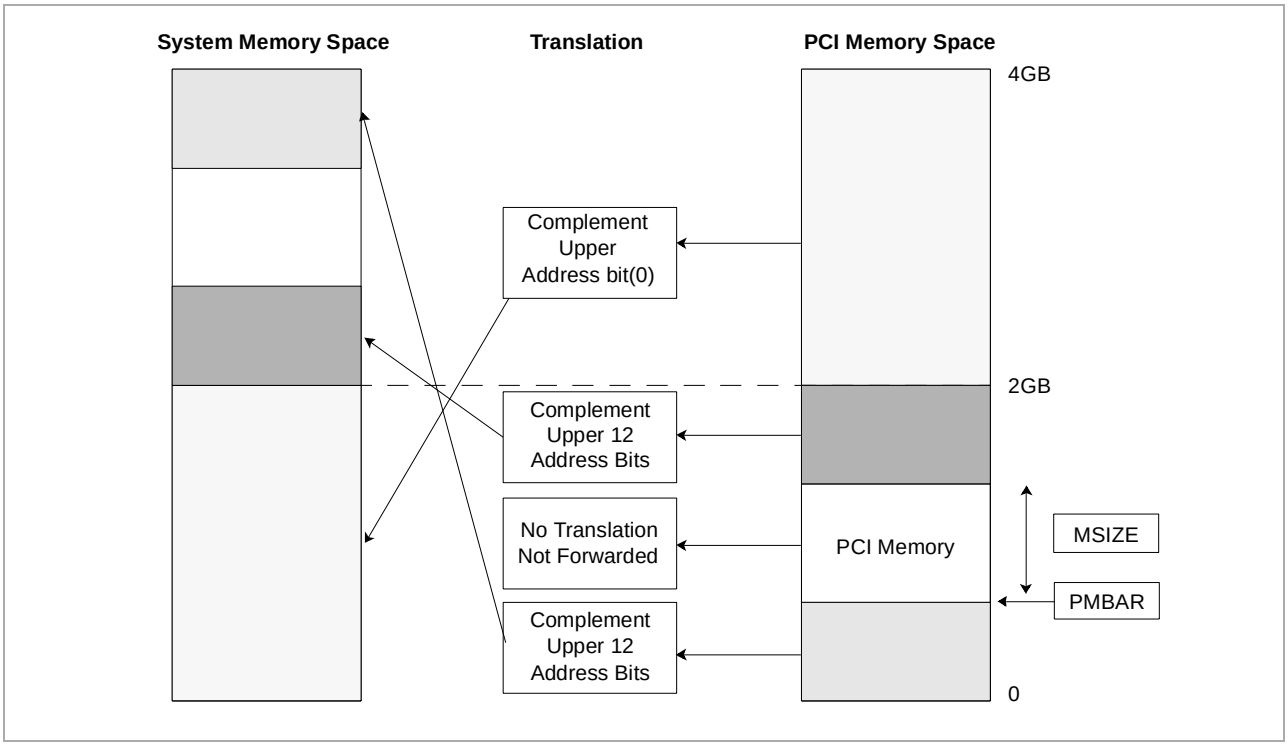
- *PCI Base Address for Memory (PMBAR)* on page 69
- *PCI Memory Address Space Size (MSIZE)* on page 74

If there is no match and if translation is enabled by software, the PCI address is translated to a system address (bit 4 - *Personalization Register (PR)* on page 71). A series of checks is performed to determine whether the access is back to the same bridge. If it is, the PCI bridge will not respond to the PCI master.

The PCI bridge logic also forwards the access to system memory. If this address does not match a memory configuration extent, the memory controller logic returns an invalid address error, thus ensuring that PCI masters do not access system facilities.

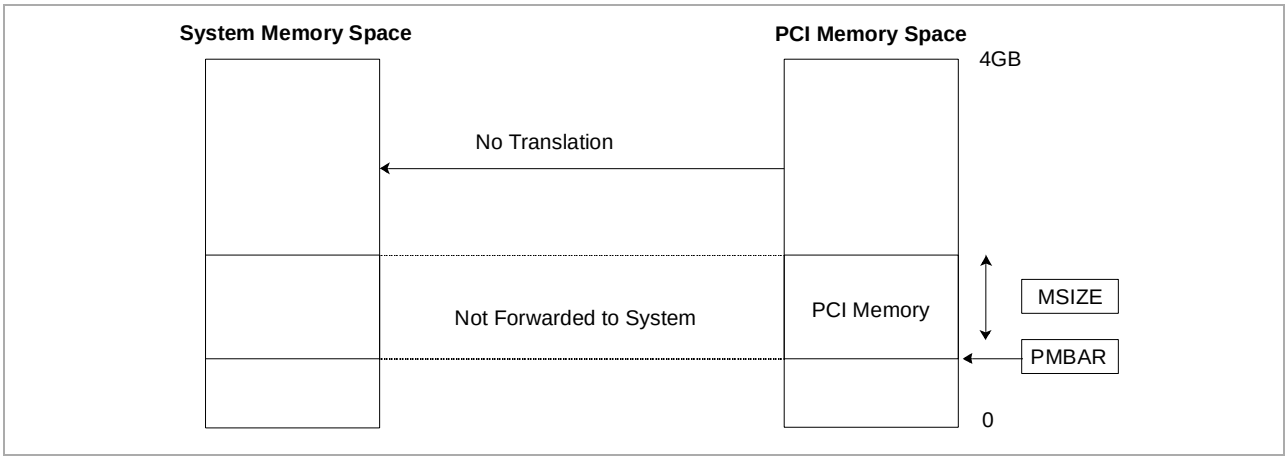
Translation Enabled

If translation is enabled, the PCI bridge logic translates addresses before presenting them to system memory, as shown in the following figure. However, not all addresses are presented.



Translation Disabled

If translation is disabled, the PCI bridge does not translate addresses before presenting them to the system, as shown in the following figure.



Note: Translation can be disabled for CPU-to-PCI transfers if the values stored in the PMBAR and SMBAR registers are the same.

Flexible PCI Host Bridge (FPHB) Mode

In FPHB Mode, External Masters on the 32-bit and 64-bit PCI buses address system memory using the address translation model shown in the following figure. The model uses several programmable registers in the Specific PCI Host Bridge Space (*Specific PCI Host Bridge Registers* on page 64).

Note: Each PCI bridge contains a set of programmable registers.

The device monitors addresses on the PCI bus to determine whether a PCI address falls within the range specified by the following PSBAR/PSSIZE registers:

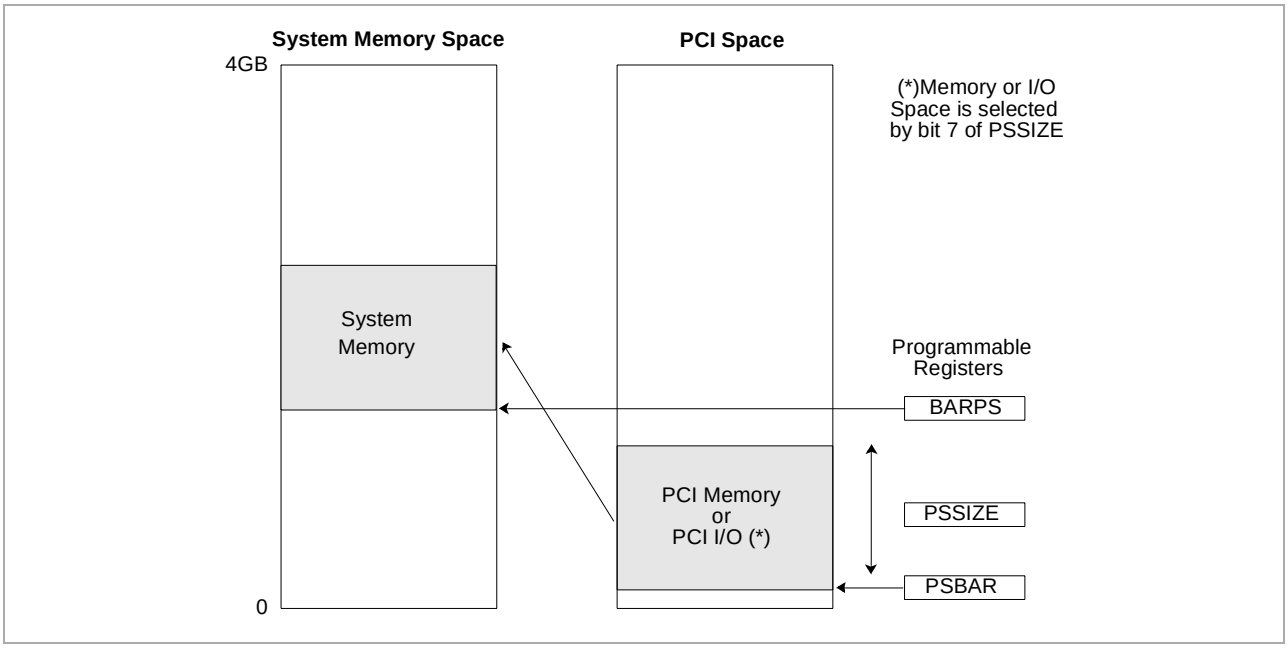
- *System Base Address Register for PCI-32 (PSBAR)* on page 84
- *System Address Space SIZE for PCI (PSSIZE)* on page 82

If an address falls within this range, the PCI interface logic passes the address to the 60x bridge logic for execution using the translation specified by the *System Base Address Register (BARPS)* on page 83.

PCI64 Configuration by external PCI Agent:

This flexible FPHB mode allows the configuration by an external PCI agent of some of the CPC710 registers of the PCI 64 bus bridge such as the PSBAR, PSEM, IT_ADD_SET and INT_RESET registers. See *Standard PCI Configuration Space (register number x'00' to x'68')* on page 33

PCI to System Memory Addressing Model (FPHB Mode)



60x Interface

Overview

The 60x interface ties the CPC710 to the PowerPC 60x system bus. It performs the following functions:

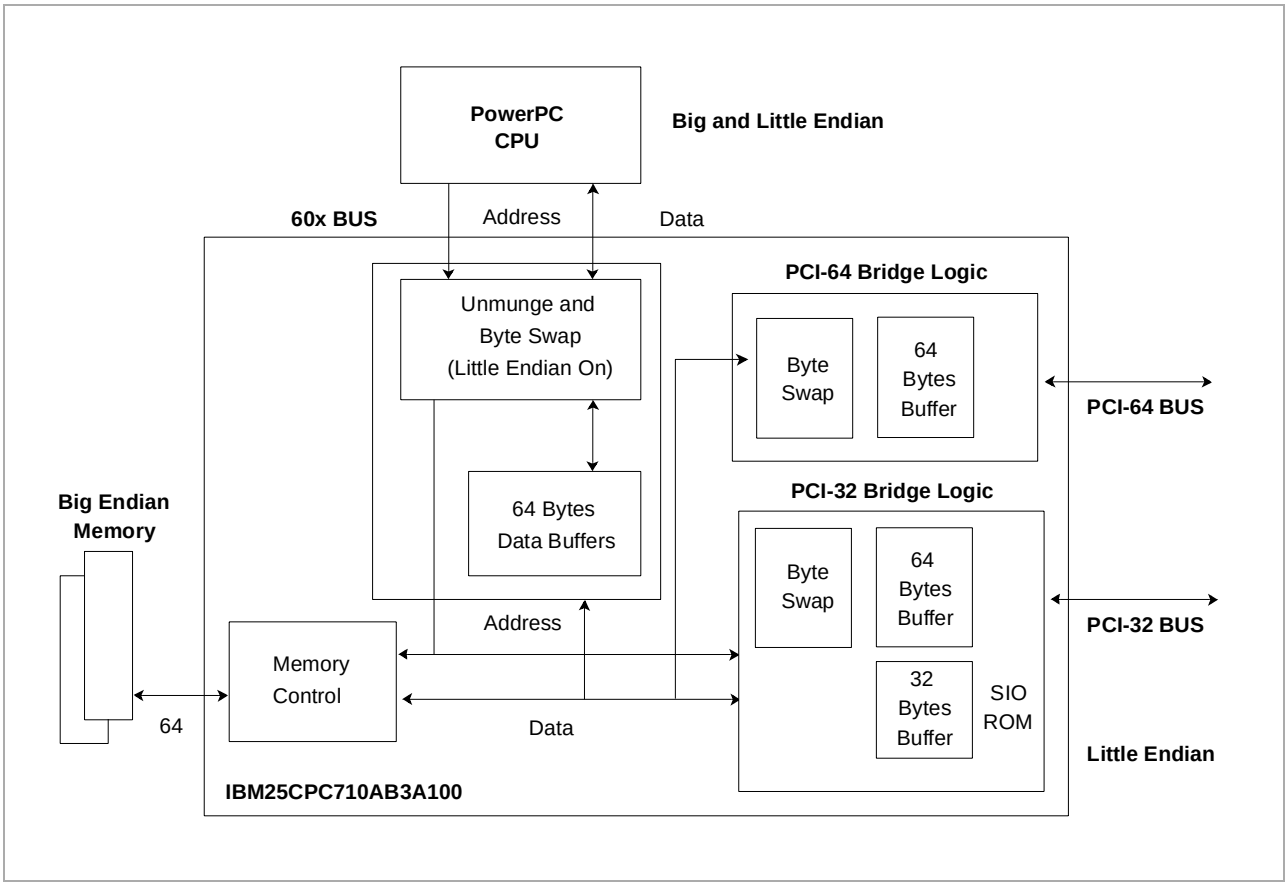
- Arbitration
- Configuration
- Processor load/store address decoding
- PCI to Memory access Snoop operations
- Sync/EIEIO processing
- Endian translation
- Reset logic operations
- Time base functions

Endian Support

The Data in a system built with the CPC710 are in the following mode:

- System Memory: Big Endian
- PCI space: Little Endian (Bytes are always swapped inside the CPC710)
- PowerPC Processor Big Endian.
  However, the Little Endian mode is also supported for the processor but the CPC710 internally swap bytes and unmundge address before sending it to the memory or the PCI bus.

CPC710 Endian Logic



The following table shows how the Data are transmitted from the CPU to the PCI-32 bus for various size of bytes.

Access CPU to PCI-32 in Write

CPU Addr	data[0:63]	PCI Addr	data[31:0]	BE
TSIZE = 1 Byte				
0	11 00 00 00 00 00 00 00	0	00 00 00 11	1110
1	00 11 00 00 00 00 00 00	0	00 00 11 00	1101
2	00 00 11 00 00 00 00 00	0	00 11 00 00	1011
3	00 00 00 11 00 00 00 00	0	11 00 00 00	0111
4	00 00 00 00 11 00 00 00	4	00 00 00 11	1110
5	00 00 00 00 00 11 00 00	4	00 00 11 00	1101
6	00 00 00 00 00 00 11 00	4	00 11 00 00	1011
7	00 00 00 00 00 00 00 11	4	11 00 00 00	0111
TSIZE = 2 Bytes				
0	11 22 00 00 00 00 00 00	0	00 00 22 11	1100
1	00 11 22 00 00 00 00 00	0	00 22 11 00	1001
2	00 00 11 22 00 00 00 00	0	22 11 00 00	0011
3	00 00 00 11 22 00 00 00 BURST	0	11 00 00 00	0111
		4	00 00 00 22	1110
4	00 00 00 00 11 22 00 00	4	00 00 22 11	1100
5	00 00 00 00 00 11 22 00	4	00 22 11 00	1001
6	00 00 00 00 00 00 11 22	4	22 11 00 00	0011
TSIZE = 3 Bytes				
0	11 22 33 00 00 00 00 00	0	00 33 22 11	1000
1	00 11 22 33 00 00 00 00	0	33 22 11 00	0001
2	00 00 11 22 33 00 00 00 BURST	0	22 11 00 00	0011
		4	00 00 00 33	1110
3	00 00 00 11 22 33 00 00 BURST	0	11 00 00 00	0111
		4	00 00 33 22	1100
4	00 00 00 00 11 22 33 00	4	00 33 22 11	1000
5	00 00 00 00 00 11 22 33	4	33 22 11 00	0001
TSIZE = 4 Bytes				
0	11 22 33 44 00 00 00 00	0	44 33 22 11	0000
1	00 11 22 33 44 00 00 00 BURST	0	33 22 11 00	0001
		4	00 00 00 44	1110
2	00 00 11 22 33 44 00 00 BURST	0	22 11 00 00	0011
		4	00 00 44 33	1100
3	00 00 00 11 22 33 44 00 BURST	0	11 00 00 00	0111
		4	00 44 33 22	1000
4	00 00 00 00 11 22 33 44	4	44 33 22 11	0000
TSIZE = 5 Bytes				
0	11 22 33 44 55 00 00 00 BURST	0	44 33 22 11	0000
		4	00 00 00 55	1110
1	00 11 22 33 44 55 00 00 BURST	0	33 22 11 00	0001
		4	00 00 55 44	1100
2	00 00 11 22 33 44 55 00 BURST	0	22 11 00 00	0011
		4	00 55 44 33	1000
3	00 00 00 11 22 33 44 55 BURST	0	11 00 00 00	0111
		4	55 44 33 22	1000
TSIZE = 6 Bytes				
0	11 22 33 44 55 66 00 00 BURST	0	44 33 22 11	0000
		4	00 00 66 55	1100
1	00 11 22 33 44 55 66 00 BURST	0	33 22 11 00	0001
		4	00 66 55 44	1000
2	00 00 11 22 33 44 55 66 BURST	0	22 11 00 00	0011
		4	66 55 44 33	0000
TSIZE = 7 Bytes				
0	11 22 33 44 55 66 77 00 BURST	0	44 33 22 11	0000
		4	00 77 66 55	1000
1	00 11 22 33 44 55 66 77 BURST	0	33 22 11 00	0001
		4	77 66 55 44	0000
TSIZE = 8 Bytes				
0	11 22 33 44 55 66 77 88 BURST	0	44 33 22 11	0000
		4	88 77 66 55	0000

Processor Behavior Mode

The CPC710 supports PowerPC 604 and 750 processors operating in Big Endian (BE) and Little Endian (LE) modes. The mode determines the order in which a multibyte scalar is stored in memory or I/O. In BE mode, the specified address contains the scalar's most significant byte (MSB), the next sequential address contains the second MSB, and so on. In LE mode, the specified address contains the scalar's least significant byte (LSB), the next sequential address contains the second LSB, and so on.

Processor Behavior in LE Mode

PowerPC 604 and 750 processors normally operate in BE mode. To operate in LE mode, the processors generate an LE address internally and then modify, or "munge," the three low-order address bits to create a BE address equivalent. The processors do not issue unaligned LE transfers on the bus. Instead, they take an alignment interrupt. However, the PowerPC 604+ processor does issue unaligned LE transfers as long as they do not cross word boundaries. The following table describes the addresses generated by the processor for LE transfers.

Endian Behavior

PREP architecture requires data to be stored in the same Endian mode as the processor. Therefore, the device implements logic to "unmunge" the address and byte swap the data bus as it comes from the processor before sending it to memory or to the PCI bridges. See Device Endian Logic below.

Processor Little Endian Address Modification

Processor's Internally Generated LE Effective Address [29:31]	Transfer Size (bytes)							
	1	2	3[1]	4	5[2]	6[2]	7[2]	8
	Resulting Processor Big-endian Address [29:31]							
0	7	6	5[3]	4				0
1	6	5[3]	4[3]					
2	5	4						
3	4							
4	3	2	1[3]	0				
5	2	1[3]	0[3]					
6	1	0						
7	0							

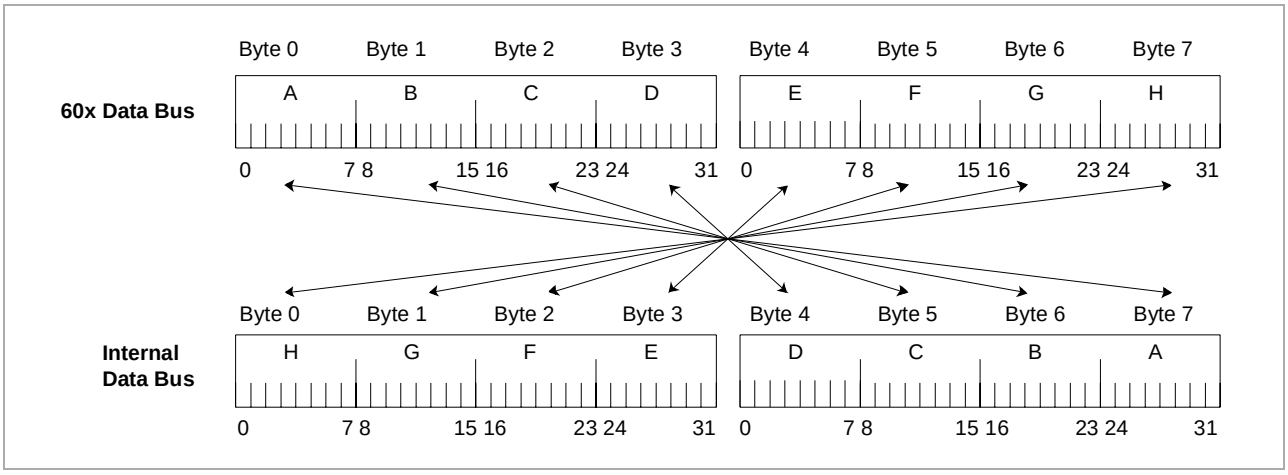
1. The PowerPC 604+ does not support 3-byte transfers in LE mode, however, these transfer sizes will result from an unaligned 4-byte access to an odd address
2. These transfer sizes are not supported by any of the processors.
3. These cells apply only to the PowerPC 604+ which performs unaligned LE transfers.

Because the device cannot determine the processor's Endian state, software must write to the Arbiter Control Register (bit 9) at the same time the processor HID register bit is updated. If the processor is operating in BE mode, bit 9 must be set to 0 to prevent the device from unmunging or byte swapping the processor's data. If the processor is operating in LE mode, bit 9 must be set to 1 to unmunge the address as specified in *Little Endian Address Unmunge Equations* below, and to swap the data bus bytes as specified in *Data Bus Byte Swap for Little Endian* below.

Little Endian Address Unmunge Equations

Transfer Size	Equation to Convert to Address
1 Byte	ADDR[29:31] XOR ‘111’
2 Byte	ADDR[29:31] XOR ‘110’ and ‘1($\overline{31}$)1’
3 Byte	ADDR[29:31] XOR ‘101’
4 Byte	ADDR[29:31] XOR ‘100’
8 Byte	none

Data Bus Byte Swap for Little Endian



60x Bus Arbiter Description

The arbiter in the device has the following characteristics:

- Arbitration for three devices; two levels for external masters and one for internal device requests
- No half-cycle precharge required for SYS_TA, SYS_TEA, ABB, and DBB
- Highly programmable address pipeline control
- Data streaming capability for external devices
- Programmable address bus parking capability
- Programmable timing on SYS_AACK
- Rotating address bus request priority scheme

Rotating Priority Resolution

The device's 60x arbiter implements an algorithm that rotates priorities when the address bus is granted to a master. When multiple masters request the bus, the arbiter grants the bus to the master with the highest priority during the arbitration period, then downgrades that priority to the lowest level for the next period. The arbitration period occurs during the SYS_AACK assertion cycle.

If two masters continuously request the bus, they receive alternate control. This logic is satisfactory unless a master implements a 64-byte cache line and needs to issue two 32-byte burst transfers to fill the cache. In this case, the device has a programmable mode whereby the arbiter allows one bus master to perform a pair of back-to-back address tenures even if another master requests the bus. This mode allows the device's memory controller to remain in page mode for these accesses. Without this mode, another master could insert a memory transaction to take the memory controller out of page mode.

Address Bus Pipelining

Pipelining is controlled by bits 0 and 1 of the 60x Arbitration Control register (ABCNTL).

Bit	Description	
0-1	'10'	If enabled by software, the arbiter maintains up to a two level pipeline **per master** . The arbiter continues to grant the address bus to a specific master until there are as many as three outstanding address tenures waiting for a data bus tenure to complete or begin. Since the device supports two masters on the system bus, there can be as many as six address tenures on the 60x system bus that have not completed or begun a data bus tenure. The arbiter stops granting the address bus to a particular master after its third address tenure. The device can also drive a seventh, address-only, tenure onto the bus to satisfy a DMA snoop operation.
	'01'	The arbiter maintains a one level pipeline per master. The device stops granting the address bus to a master after it has two outstanding address tenures waiting for a data bus tenure to complete . With two masters in the system, there could be as many as four outstanding address tenures waiting for a data bus tenure to complete or begin, and a fifth device generated address-only tenure.
	'00'	Pipelining is completely disabled. Even with two masters in the system, there will only be one address tenure waiting for a data tenure to complete.
	'11'	Implemented to accommodate slave devices like an L2 lookaside that can only support one level pipeline regardless of the number of masters on the 60x bus.

Arbiter Requirements

Internal ABB

All devices on the 60x bus must generate an internal ABB. Because the arbiter may grant the address bus to a requesting device while another master is active, the requesting master must generate an ABB based on SYS_TS and SYS_AACK. The current master does not provide an ABB.

Qualified SYS_BG Equation

Use the following equation to detect a qualified bus grant using positive logic:

QBG = SYS_BG + ABB + SYS_ARTRY

where ABB represents the interval between SYS_TS and SYS_AACK active

Note: Bus Request (SYS_BR) need not be active to detect a qualified bus grant (parked case).

SYS_TS Assertion

All master devices must drive SYS_TS active in the cycle immediately following a qualified address bus. Otherwise, the address tenure is aborted and another master is free to drive the address bus.

SYS_BR Negation

All master devices must negate SYS_BR for at least one bus cycle immediately after receiving a qualified bus grant.

Qualified SYS_DBG Equation

The equation for qualified SYS_DBG using positive logic is:

QDBG = SYS_DBG + ARTRY

DBB is unused because the arbiter does not issue a SYS_DBG when DBB is active. The arbiter monitors transaction sizes to determine the end of a data bus tenure and waits until the previous data tenure is complete before issuing a SYS_DBG to the next master.

Note: QDGB can only be negated by an SYS_ARTRY of the address tenure associated with the QDBG data bus tenure. Therefore, once the SYS_ARTRY window has passed for an address tenure, the data bus tenure associated with that address tenure cannot be negated by SYS_ARTRY from a subsequent address tenure.

High Impedance After SYS_TEA

Masters and slaves must execute all data bus signals as high impedance within two bus clocks from SYS_TEA assertion.

SYS_DRTRY Assertion

Slaves are not allowed to drive SYS_DRTRY active. The device arbiter does not receive SYS_DRTRY.

Slave Data Bus Determination

To determine whether the data bus is currently in use by a previous address tenure, a slave must sample DBB from its master during the TS active cycle. If DBB is active, the slave must wait for DBB to go inactive in a one-level pipeline mode for at least one cycle before providing read data or accepting write data.

SYS_L2_Hit Assertion

For the device to determine whether an addressing error has occurred, all slaves on the 60x bus must assert SYS_L2_HIT when selected by an address on the 60x bus.

Warning: The SYS_L2_HIT signal is subject to timing constraints.

Bus Enhancements

$\overline{\text{DBB}}$ not Required by Masters

Masters do not require $\overline{\text{DBB}}$. The arbiter does not grant the data bus to a requesting master if the bus is currently in use. The device does not drive $\overline{\text{DBB}}$ since it acts as an address-only bus master.

Half-Cycle Precharge not Required on $\overline{\text{SYS\_TA}}$

The device can be programmed so the precharge of $\overline{\text{SYS\_TA}}$ is not required. This requires all slaves to initially drive $\overline{\text{SYS\_TA}}$ active or inactive immediately following a data bus grant. At the end of a data bus tenure, a slave does not perform a precharge, which requires a slave in the next data bus tenure to drive $\overline{\text{SYS\_TA}}$ in the first cycle of the tenure.

The ABCNTL[13] mode bit “-60x Arbiter Control Register (ABCNTL)” Page 104 - forces the device to do a precharge if a slave on the 60x bus does not support this function.

Half-Cycle Precharge not Required on $\overline{\text{SYS\_TEA}}$

The arbiter does not grant the data bus for two bus cycles following assertion of $\overline{\text{SYS_TEA}}$s. This allows a slave to perform a full cycle precharge on $\overline{\text{SYS_TEA}}$s.

$\overline{\text{SYS\_ARTRY\_PREV}}$ in QDBG Equation Eliminated

When $\overline{\text{SYS\_ARTRY}}$ is asserted, the arbiter negates all bus grants in the cycle following $\overline{\text{SYS\_ARTRY}}$. This supplants the requirement for masters to qualify associated bus grants by asserting $\overline{\text{SYS\_ARTRY}}$ in a previous cycle.

60x Bus Transfer Types and Sizes

The following tables describe the transaction types supported by the device on the processor bus. The device supports the PowerPC 604 critical double word burst transactions.

Non-Burst Transactions ($\overline{\text{SYS\_TBST}}$ = 1)

SYS_TSIZ[0:2]	A[29:31]	Definition	Note
0 0 0	000 - 111	8-byte transfer	1
0 0 1	000 - 111	1-byte transfer	
0 1 0	000 - 111	2-byte transfer	
0 1 1	000 - 111	3-byte transfer	
1 0 0	000 - 111	4-byte transfer	
1 0 1	000 - 111	5-byte transfer	
1 1 0	000 - 111	6-byte transfer	
1 1 1	000 - 111	7-byte transfer	

1. For transfers where the number of bytes to transfer cross a doubleword boundary, the device will truncate the transfer size to avoid crossing a doubleword boundary.

Burst Transactions ($\overline{\text{SYS\_TBST}}$ = 0)

SYS_TSIZ[0:2]	A[27:31]	Definition	Support	Notes
0 0 0	x x x x x	8 Byte transfer	No	2
0 0 1	0 0 x x x	16-byte transfer beginning on 32-byte boundary	TBD	
	0 1 x x x	16-byte transfer beginning on odd doubleword boundary	No	
	1 0 x x x	16-byte transfer beginning on odd 16-byte boundary	TBD	
	1 1 x x x	16-byte transfer beginning on odd doubleword boundary	No	
0 1 0	0 0 x x x	32-byte transfer beginning on 32-byte boundary	Yes	
	0 1 x x x	32-byte transfer beginning on odd doubleword boundary		1,3
	1 0 x x x	32-byte transfer beginning on odd 16-byte boundary		1,3
	1 1 x x x	32-byte transfer beginning on odd doubleword boundary		1,3

1. For transfers that cross a 32-byte boundary, the device will wrap to the beginning of the 32-byte block to satisfy the data transfer.
2. Unpredictable results will occur if this transfer size is attempted on the processor bus.
3. Not supported on store operations

Transfer Types

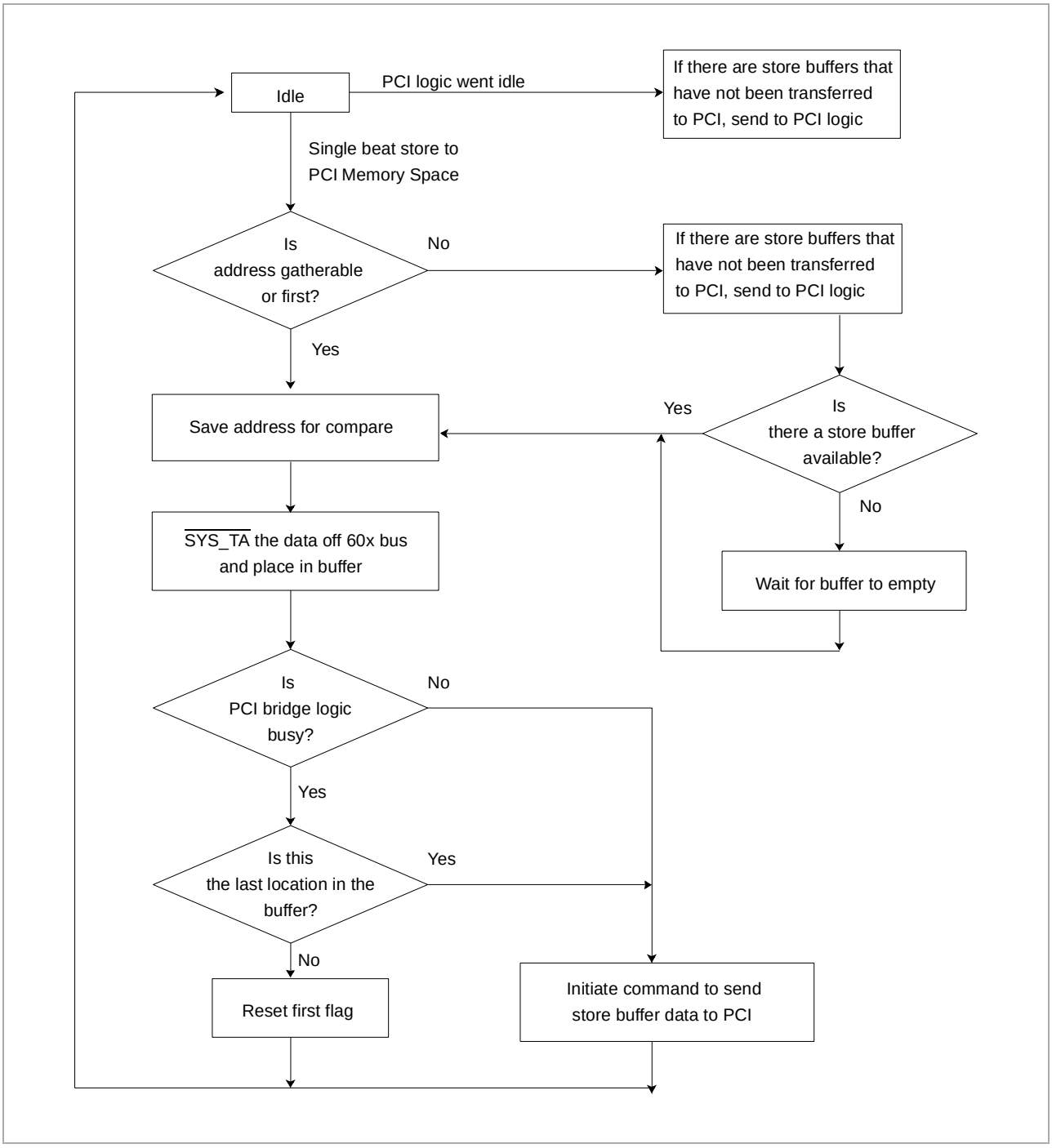
TT[0:4]	Operation	Transaction	Support as Master	Support as Slave
0 0 0 0 0	Clean Sector	Address-only	Yes	NOP
0 0 0 0 1	LARX Reservation Set	Address-only	No	NOP
0 0 0 1 0	Write with Flush	SBW or Burst	No	Yes
0 0 0 1 1	Reserved (arbiter will assume address-only transaction)			
0 0 1 0 0	Flush Sector	Address-only	Yes	NOP
0 0 1 0 1	Reserved (arbiter will assume address-only transaction)			
0 0 1 1 0	Write with Kill	Burst	No	Yes (treated as 00010)
0 0 1 1 1	Reserved (arbiter will assume address-only transaction)			
0 1 0 0 0	SYNC	Address-only	No	Yes
0 1 0 0 1	TLBSYNC	Address-only	No	Yes
0 1 0 1 0	Read	SBR or Burst	No	Yes
0 1 0 1 1	RWNITC - Read with no Intent to Cache	SBR or Burst	Yes (See ATAS Register)	Yes (treated as 01010)
0 1 1 0 0	Kill Sector	Address-only	Yes	NOP
0 1 1 0 1	ICBI	Address-only	No	NOP
0 1 1 1 0	RWITM - Read with Intent to Modify	Burst	Yes (See ATAS Register)	Yes (treated as 01010)
0 1 1 1 1	Reserved (arbiter will assume address-only transaction)			
1 0 0 0 0	EIEIO	Address-only	No	Yes
1 0 0 0 1	Reserved (arbiter will assume address-only transaction)			
1 0 0 1 0	Write with Flush Atomic	SBW	No	Yes (treated as 00010)
1 0 0 1 1	Reserved (arbiter will assume address-only transaction)			
1 0 1 0 0	ECOWX - Graphics Write	SBW	No	Yes
1 0 1 0 1	Reserved (arbiter will assume address-only transaction)			
1 0 1 1 x				
1 1 0 0 0	TLB Invalidate	Address-only	No	No
1 1 0 0 1	Reserved (arbiter will assume address-only transaction)			
1 1 0 1 0	Read Atomic	SBR or Burst	No	Yes (treated as 01010)
1 1 0 1 1	Reserved (arbiter will assume address-only transaction)			
1 1 1 0 0	ECIWX - Graphics Read	SBR	No	Yes
1 1 1 0 1	Reserved (arbiter will assume address-only transaction)			
1 1 1 1 0	RWITM Atomic	Burst	No	Yes (treated as 01010)
1 1 1 1 1	Reserved (arbiter will assume address-only transaction)			

Note: SBW = Single Beat Write, SBR = Single Beat Read

Data Gathering

The 60x logic gathers data for CPU store-transfers to the PCI bus bridges. During data gathering, single beat stores of up to 32bytes from the CPU are gathered before being sent to the PCI bus bridge unit. Data gathering reduces asynchronous boundary crossings and facilitates data bursting on the PCI bus.

Data Gathering Algorithm



SYNC and EIEIO

When a processor executes a SYNC instruction, a SYNC address-only tenure is broadcast on the 60x bus to notify the system that a software-placed barrier is present. The system is responsible for ensuring all previously executed load and store operations are complete and all resultant actions are visible to the system. The device satisfies this requirement by SYS_ARTRYing the SYNC operation until all of its store buffers are empty, all reads have been executed, and all data have been placed in internal device buffers for requests issued by the same processor issuing SYNC.

When a processor broadcasts an EIEIO on the 60x bus, the system is responsible for ensuring all previous transactions are complete before executing operations. The device does not SYS_ARTRY the EIEIO because the 60x logic dispatches bus transactions to the logic units in the order in which they occur on the system bus and each logic unit executes its commands in the order received. For diagnostic purposes, the device can be programmed to SYS_ARTRY the EIEIO in the same manner as SYNC (see bit 10 of the ABCNTL Register "-60x Arbiter Control Register (ABCNTL)" Page 104).

The logic units are system memory, PCI-32 bus bridge, PCI-64 bus bridge, system I/O logic, and DMA controller logic. EIEIO operations are valid for transfers to and from the same logic unit, but execution order of load and store operations to different logic units cannot be guaranteed. For example, a store to the PCI-32 bus bridge followed by a PCI-64 store could be presented to the respective PCI buses in reverse order if a bus is busy. To preserve the order among logic units, software must issue a SYNC instead of an EIEIO.

Address Retry (SYS_ARTRY)

Precharging SYS_ARTRY and SYS_SHD

The IBM25CPC710AB3A100 device always precharges SYS_ARTRY and SYS_SHD. All other devices on the 60x bus must disable their precharges of these signals. The device negates SYS_ARTRY and SYS_SHD for a half bus cycle during the second cycle following the SYS_ARTRY window's last cycle.

CPC710SYS_ARTRY Assertions

The device asserts SYS_ARTRY for:

- SYNC operations as described in the previous section
- EIEIO operations as described in the previous section
- XFERDATA when more than two transfers have been initiated
- a processor access to the PCI bus when a PCI-ISA bus bridge requests the same PCI bus
- a processor access to system memory when a DMA occurs to the same cache line
- a processor access to system memory when a DMA operation occurs to the same line
- a processor access into a range of PCI-32 or PCI-64 addresses defined as potential deadlock

Recommended SYS_ARTRY Procedure

- A master that has had its address tenure retried should negate its SYS_BR[n] for at least one bus cycle in the cycle immediately following detection of an active SYS_ARTRY.
- A master that has retried an address tenure due to a snoop hit should activate its SYS_BR[n] in the cycle immediately following the detection of an active SYS_ARTRY. This ensures the master that retries is serviced before the master that was retried.

Locking Signal $\overline{\text{DLK}}$

The device can set the $\overline{\text{DLK}}$ output for CPU access to the PCI-32 or PCI-64 bus when:

- input $\overline{\text{NODLK}}$ is not active (='1')
- the access is within the address range defined by the selected bridge's registers SMDLK1/2 or SIDLK1/2
- bit 24 of the PGCHP register is '1'.

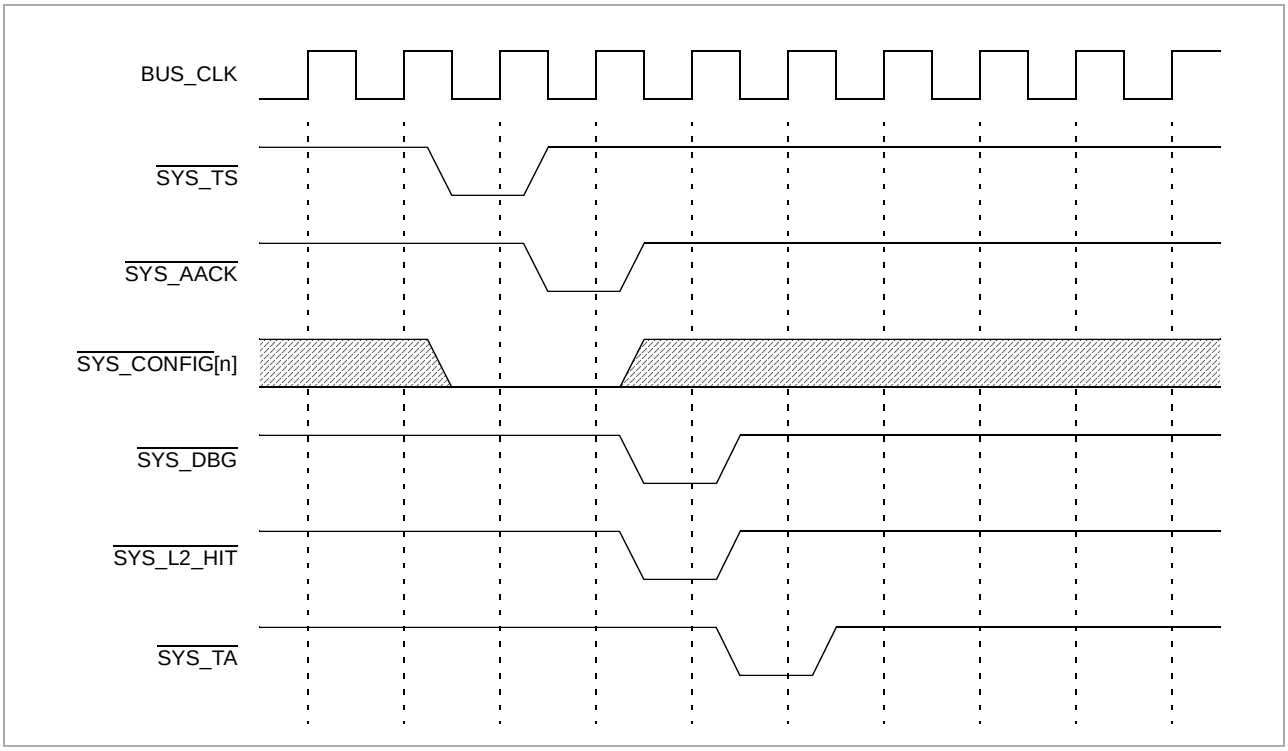
The 60x CPU can then receive an $\overline{\text{SYS\_ARTRY}}$ when the $\overline{\text{NODLK}}$ signal becomes active ('0').

60x Bus Configuration

The system uses configuration cycles to identify and configure BUCs on the 60x bus. The following rules apply to configuration cycles:

- All configuration cycles must use the Basic Transfer Protocol.
- The BUC responds to addresses in the configuration space only when its $\overline{\text{SYS\_CONFIG}}$[n] is active.
- All configuration cycles must be single beat, word-aligned word accesses except for accesses to the feature ROMs that are configured for byte access.
- BUCs must respond to DCR reads at $\overline{\text{SYS\_L2\_HIT}}$ assertion as shown in the following figure. Otherwise, the device assumes no device is present and $\overline{\text{SYS_TA}}$s the CPU with data indicating no device present.

60x Bus Configuration Cycle



Note: For in-line L2 caches, the L2 cache controller only supports configurations from its local processor. Software should not configure the controller from a processor on the system bus side of the controller. This makes it unnecessary for this kind of L2 cache controller to drive $\overline{\text{SYS\_L2\_HIT}}$.

Error Handling for CPU-Initiated Transactions

The devices uses Machine Checks to indicate errors. This allows software to log errors before the system is shut down. In an MP environment, the device activates the Machine Check pin that corresponds to the CPU initiating the transaction. Because the PowerPC 601 does not provide this pin, the checkstop pin is used.

Checkstop Errors

The device generates a checkstop when the following are detected:

- Address parity error on the 60x system bus (if enabled)
- Data parity error on 60x system bus (if enabled)
- Internal timeout due to no response from slave on load

The 60x logic performs the following when generating a checkstop:

1. Sets appropriate bit(s) in SESR
2. Drives CHKSTOP active until power on reset

The following table describes the error handling performed for CPU initiated transactions. The 60x logic drives SYS_MACHK signals, not the PCI bridge logic or the memory control logic.

Error Handling for CPU Initiated Transactions (Page 1 of 4)

Operation	Error	Mode	Action	Notes
Access not directed to device	Addressing Error (SYS_L2_HIT not driven active)	Disabled	No action taken	
		Enabled	Set No Select error bit in SESR Set error address in SEAR PGCHP[26] = 0: Signal Machine Check with SYS_TEA PGCHP[26] ≠ 0: Signal Machine Check with SYS_MACHK	
Access to device	Bus Time-out: Time expired from SYS_AACK active to first SYS_TA	Disabled	Inhibit timer; no action taken	
		Enabled	Set bus time-out error bit in SESR Set checkstop generated bit in SESR Set error address in SEAR Signal Checkstop	
Access to internal device facilities	Access to a reserved or non- implemented address		Terminate CPU transaction normally	1
	Alignment or size			
	Store to read-only register			
	Load from write-only register			

1. A dummy 0 is returned for read operation. For write, data is ignored.

Error Handling for CPU Initiated Transactions (Page 2 of 4)

Operation	Error	Mode	Action	Notes
Access to system memory	Single bit error	Don't care	Set single-bit error and syndrome in MESR Set error address in MEAR Return corrected data to CPU If PGCHP[17] = 1 and PGCHP[26] = 1: Set memory error bit in SESR Set memory error address in SEAR Signal Machine Check with SYS_MACHK	
	Double-bit error	Normal	Set error in MESR Set error address in MEAR Set memory error bit in SESR Set memory error address in SEAR if PGCHP[26] = 0: Loads: Signal Machine Check with SYS_TEA Stores: Signal Machine Check with SYS_MACHK If PGCHP[26] = 1, signal Machine Check with SYS_MACHK	
		Diagnostic	Set double-bit error in MESR Set error address in MEAR Return uncorrected data to CPU Signal Machine Check with SYS_MACHK if write less than eight bytes	
Access to PCI bus	No DEVSEL received		Master-abort the PCI transaction Set master-aborted bit 13 in PCI Status register Set device error bit in PLSSR register Set PCI error bit in SESR Set error address in SEAR register If PGCHP[26] = 0: Loads: Signal Machine Check with SYS_TEA Stores: Signal Machine Check with SYS_MACHK If PGCHP[26] = 1: Signal Machine Check with SYS_MACHK Terminate CPU transaction normally	1
	Detected SERR active during PCI transaction		Master-abort the PCI transaction Set master aborted bit 13 in PCI Status register Set SERR detected error in PLSSR register Set PCI error bit in SESR Set error address in SEAR register If PGCHP[26] = 0: Loads: Signal Machine Check with SYS_TEA Stores: Signal Machine Check with SYS_MACHK If PGCHP[26] = 1: Signal Machine Check with SYS_MACHK Terminate CPU transaction normally	1

1. A dummy 0 is returned for read operation. For write, data is ignored.

Error Handling for CPU Initiated Transactions (Page 3 of 4)

Operation	Error	Mode	Action	Notes
Access to PCI bus (cont'd)	Detected PCI bus data parity error on load	Enabled by bit 6 in PCI CMND register	Continue transfer on PCI bus to completion Activate the PERR signal Set data parity error bit 8 in PCI Status register Set data parity error bit 15 in PCI Status register Set PCI error bit in SESR Set error address in SEAR register If PGCHP[26] = 0: Loads: Signal Machine Check with SYS_TEA Stores: Signal Machine Check with SYS_MACHK If PGCHP[26] = 1: Signal Machine Check with $\overline{\text{SYS\_MACHK}}$ Terminate CPU transaction normally	1
		Disabled by bit 6 in PCI CMND register.	Set parity error bit 15 in PCI Status register Proceed normally with PCI transaction Proceed normally with CPU transaction	
	Detected PERR on store	Enabled by bit 6 in PCI CMND register	Continue transfer on PCI bus to completion Set data parity error bit 8 in PCI Status register Set data parity error bit 15 in PCI Status register Set PCI error bit in SESR Set error address in SEAR register If PGCHP[26] = 0: Loads: Signal Machine Check with SYS_TEA Stores: Signal Machine Check with SYS_MACHK If PGCHP[26] = 1: Signal Machine Check with $\overline{\text{SYS\_MACHK}}$ Terminate CPU transaction normally	1
		Disabled by bit 6 in PCI CMND register	Set data parity error bit 15 in PCI Status register Proceed normally with PCI transaction Proceed normally with CPU transaction	
	Detected target abort		Set received target abort bit in PCI Status register Set PCI error bit in SESR Set error address in SEAR register If PGCHP[26] = 0: Loads: Signal Machine Check with $\overline{\text{SYS\_TEA}}$ Stores: Signal Machine Check with SYS_MACHK If PGCHP[26] = 1: Signal Machine Check with $\overline{\text{SYS\_MACHK}}$ Terminate CPU transaction normally	1

1. A dummy 0 is returned for read operation. For write, data is ignored.

Error Handling for CPU Initiated Transactions (Page 4 of 4)

Operation	Error	Mode	Action	Notes
Access to PCI bus (cont'd)	PCI Bus timeout: P/G_TRDY count expired	Enabled	Master-abort the PCI transaction Set master aborted bit 13 PCI Status register Set PCI bus time-out error in PLSSR register Set PCI error bit in SESR Set error address in SEAR register If PGCHP[26] = 0: Loads: Signal Machine Check with $\overline{\text{SYS\_TEA}}$ Stores: Signal Machine Check with SYS_MACHK If PGCHP[26] = 1: Signal Machine Check with $\overline{\text{SYS\_MACHK}}$ Terminate CPU transaction normally	1
	Retry count expired	Enabled	Stop retrying PCI transfer Set retry count expired bit in PLSSR register Set PCI error bit is SESR Set error address in SEAR register If PGCHP[26] = 0: Loads: Signal Machine Check with $\overline{\text{SYS\_TEA}}$ Stores: Signal Machine Check with SYS_MACHK If PGCHP[26] = 1: Signal Machine Check with $\overline{\text{SYS\_MACHK}}$ Terminate CPU transaction normally	1

1. A dummy 0 is returned for read operation. For write, data is ignored.

Memory Controller

Overview

The device's memory controller controls processor and I/O interactions with the memory system.

The memory controller supports SDRAM and is 2-way interleaved to allow the memory to burst data on every CPU bus cycle at 100 MHz (1-1-2-1 after initial latency) using only one memory address bus. To handle critical word load, individual control of the LSB column address bits is required for the DIMM pair.

The controller supports up to eight dual DIMMs banks of interleaved 72-bit memory (64-bit Data + 8-bit ECC). To reduce pin count, the controller requires a Texas Instruments (TI) ALVCH162268 MUX to externally multiplex the 144-bit data to 72-bits for device input. Programmable parameters allow a variety of memory organizations and timings.

ECC protection is provided for all 64 bits of the data bus, detecting and correcting single- and double-bit errors. Different SDRAM organizations can be mixed (*Supported SDRAM Organizations* on page 177). 60x bus operation is limited to 100 MHz.

Programmable parameters allow for a variety of memory organizations (See "Supported SDRAM Organizations" on page 177.).

The SDRAM must comply with the following requirements (compatible with the PC100 Specification [1]).

- CAS Latency = 2
- Burst length = 2
- Maximum tRCDmin allowed is 2 Clock cycles.
- Maximum tRPmin allowed is 2 Clock cycles.
- Maximum tRASmin allowed is 5 Clock cycles.

It is possible to use Extended Data Out/Hyper-Page DRAM (EDO/HPM DRAM) with IBM Dual Bridge and Memory Controller, but this application is not fully supported: contact your support for more informations.

Only one type of RAM can be used in a system (it is not possible to mix SDRAM with DRAM devices). However, different kind of SDRAM organizations can be mixed. 60x bus operation is limited to 75 MHz when using EDO DRAM and to 100 MHz when using SDRAM.

Memory Performance for Cache Line Operations (ECC Active)

Pipeline Levels	Operation			CAS Latency = 2			
				100 MHz			
0	Read Burst	Initial		160 MB/s 16-1-2-1			
		Sus-tained	Page Miss	160 MB/s 16-1-2-1			
			Page Hit	160 MB/s 16-1-2-1			
	Write Burst	Initial		533 MB/s 3-1-1-1			
		Sus-tained	Page Miss				
			Page Hit				
2	Read Burst	Initial		160 MB/s 16-1-2-1			
		Sus-tained	Page Miss	228 MB/s 10-1-2-1			
			Page Hit	400 MB/s 4-1-2-1			
	Write Burst	Initial		564 MB/s 3-1-1-1-2-1-1-1-2-1-1-2			
		Sus-tained	Page Miss	246 MB/s 10-1-1-1			
			Page Hit	400 MB/s 5-1-1-1			

Bank Definitions

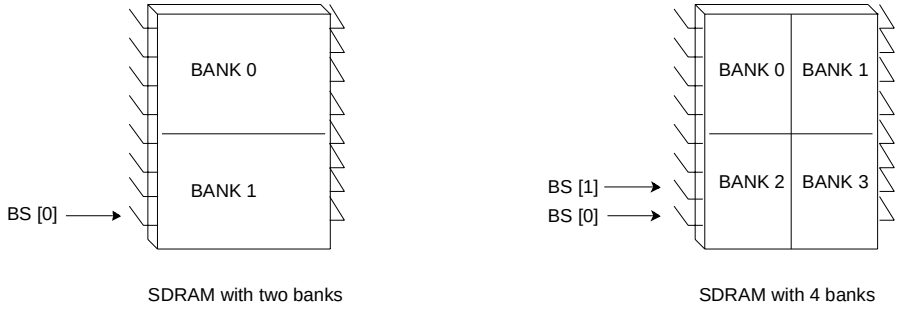
The word "Bank" covers a couple of different meanings, depending on the point-of-view:

1. SDRAM Banks
2. DIMM Banks
3. Interleaved Banks

SDRAM Banks

As shown in the following diagram, SDRAMs contain memory arranged in two or four banks. The Memory Controller selects these banks using Bank Select (BS) address pins.

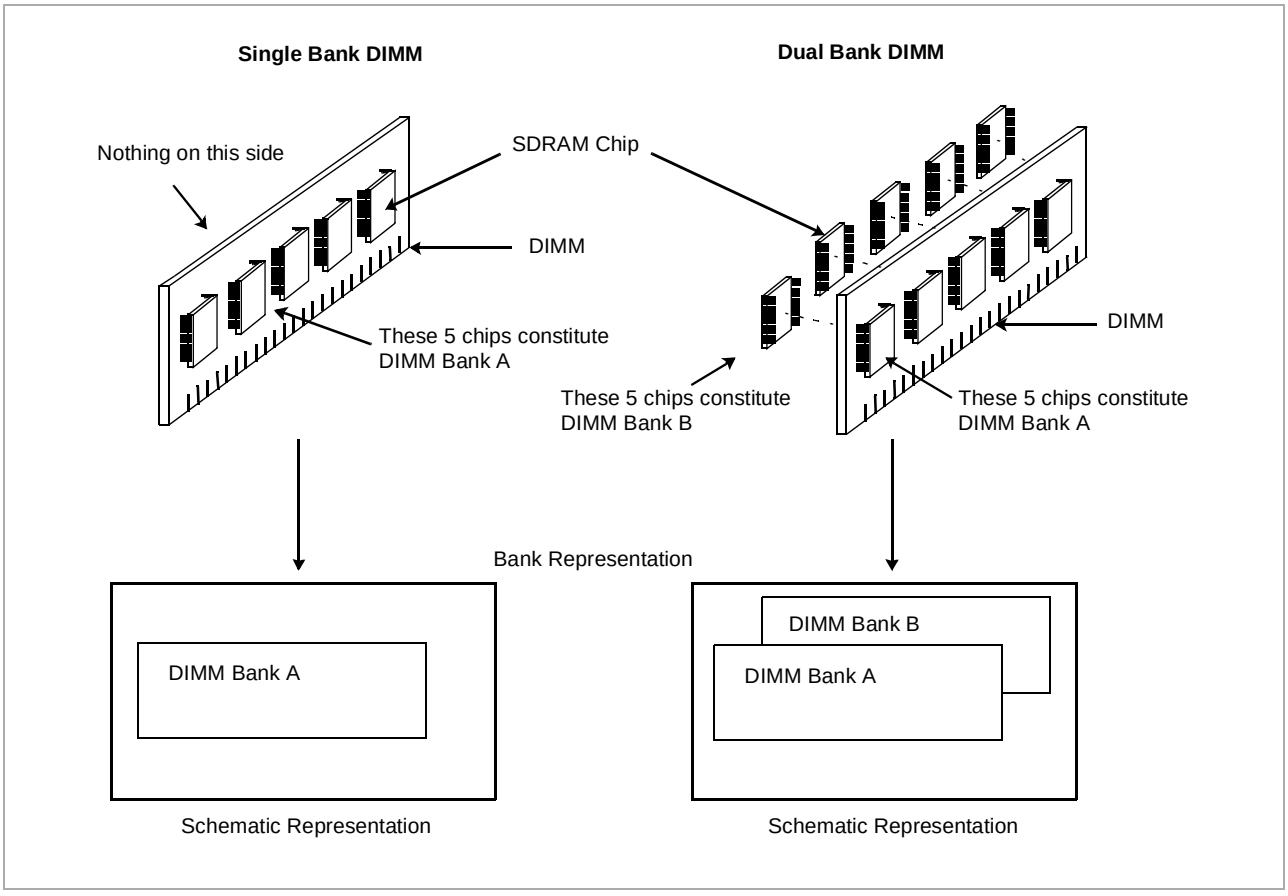
SDRAM Bank Configuration



DIMM Banks

As shown in the following diagram, DIMMs are available in single bank and dual bank configurations.

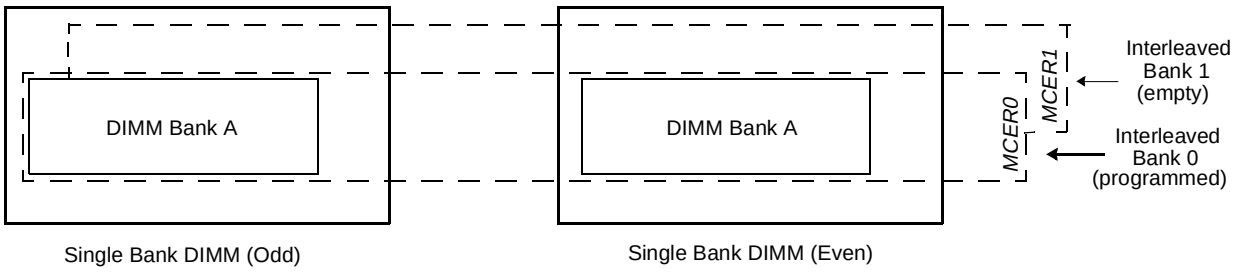
DIMM Bank Configuration



Interleaved Banks

An Interleaved Bank consists of two interleaved DIMM Banks. The two DIMM Banks are called Odd and Even. As shown in the following figures, MCER registers must be programmed according to the DIMM Bank configuration used.
Since the device works in an interleaved way, the minimum equipment required is two Single or Dual DIMM Banks.

Programming with Single Bank DIMMs

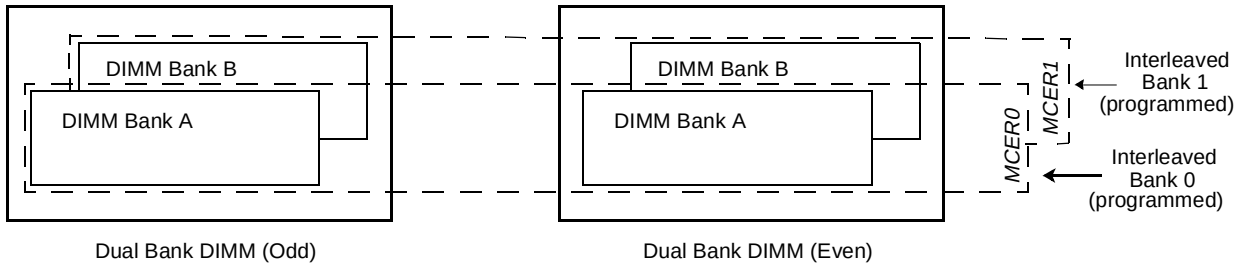


MCER0 (for Interleaved Bank 0) must be programmed.

MCER0 (for Interleaved Bank 1) must be empty, or bit '0' set to '0'.

All unused MCERx locations must be empty.

Programming with Dual Bank DIMMs



MCER0 and MCER1 (for Interleaved Banks 0 and 1) must be programmed.

Memory Signal Connections

SDRAM Common Signals

Signal Name	Type
MDATA[0:71]	72-bit Data
MADDR[1:12]	Address
MADDR0_ODD	
MADDR0_EVEN	

External MUX Controller for Memory Data

Signal Name	Toggle for Reads	Toggle for Writes
$\overline{\text{MUXCLKEN1B}}$	Yes	No (’1’b)
$\overline{\text{MUXCLKEN2B}}$	Yes	No (’1’b)
$\overline{\text{MUXCLKENA1}}$	No (’1’b)	Yes
$\overline{\text{MUXCLKENA2}}$	No (’1’b)	Yes
$\overline{\text{MUX\_OEA}}$	No (’1’b)	No (’0’b)
$\overline{\text{MUX\_OEB}}$	No (’0’b)	No (’1’b)
$\overline{\text{MUX\_SEL}}$	High for EVEN (DH) Low for ODD (DL)	No (’1’b)

\

Memory Address Bit Definition for Non-Row Column Addressing Bits

Address Bit	Definition
0-1	’00’: Base address of memory
28	Interleaving Bit 0: Even DIMMs (0, 2, 4, or 6) 1: Odd DIMMs (1, 3, 5, or 7)

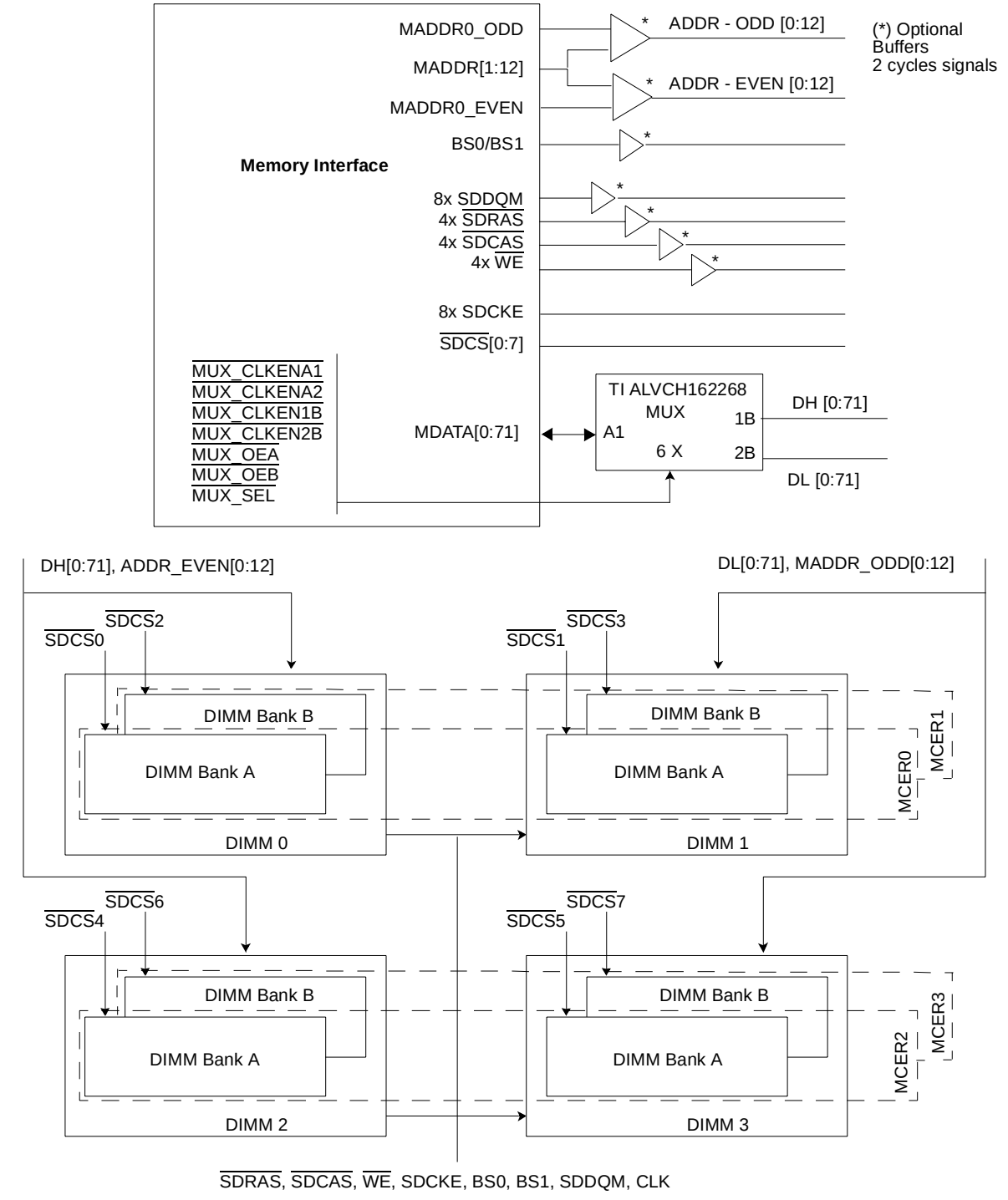
SDRAM Subsystem Signals

Signal Name	Type	Comments
BS0	SDRAM Bank Select	
BS1		
$\overline{\text{SDCS}}$[0:7]	Chip Select	See the following table for connections
SDDQM[0:7]	Data Mask	8 pins for load purposes
$\overline{\text{SDRAS}}$[0:3]	Row Address Strobe	4 pins for load purposes
$\overline{\text{SDCAS}}$[0:3]	Column Address Strobe	4 pins for load purposes
$\overline{\text{WE}}$[0:3]	Write Enable	4 pins for load purposes
SDCKE[0:7]	Clock Enable	8 pins for load purposes

SDRAM DIMM Chip Select Connections

Signal Name	DIMM and DIMM Bank
$\overline{\text{SDCS}}$[0]	DIMM 0, Bank A
$\overline{\text{SDCS}}$[1]	DIMM 1, Bank A
$\overline{\text{SDCS}}$[2]	DIMM 0, Bank B
$\overline{\text{SDCS}}$[3]	DIMM 1, Bank B
$\overline{\text{SDCS}}$[4]	DIMM 2, Bank A
$\overline{\text{SDCS}}$[5]	DIMM 3, Bank A
$\overline{\text{SDCS}}$[6]	DIMM 2, Bank B
$\overline{\text{SDCS}}$[7]	DIMM 3, Bank B

SDRAM Subsystem Overview



Note: The input clock for the SDRAM is the 60x bus clock, which is not driven by the device.

Supported SDRAM Organizations

The CPC710 is fully compatible with the JEDEC Standard.
It supports the addressing listed in the following table.

Supported DIMMs

DIMM Size (MByte)	SDRAM Addressing bit (Row/Col/Bank)	SDRAM (Mbits x I/Os)	Number of Chips per Bank with ECC	DIMM Size (MByte)	SDRAM Addressing bit (Row/Col/Bank)	SDRAM (MB x I/Os)	Number of Chips per Bank with ECC
8M Single	11/8/1	1Mx16	4 + 1				
16M Single	11/9/1 11/9/1 11/8/2 12/8/1	2Mx8 2Mx32 2Mx32 2Mx8	8 + 1 2 + 1 2 + 1 8 + 1	16M Dual	11/8/1	1Mx16	4 + 1
32M Single	11/10/1 11/10/1 12/9/1 12/8/2 13/8/1	4Mx4 4Mx16 4Mx4 4Mx16 4Mx16	16 + 2 4 + 1 16 + 2 4 + 1 4 + 1	32M Dual	11/9/1 11/9/1 11/8/2 12/8/1	2Mx8 2Mx32 2Mx32 2Mx8	8 + 1 2 + 1 2 + 1 8 + 1
64M Single	12/9/2 12/9/2 13/8/2 13/9/1	8Mx8 8Mx32 8Mx32 8Mx8	8 + 1 2 + 1 2 + 1 8 + 1	64M Dual	11/10/1 11/10/1 12/8/2 12/9/1 13/8/1	4Mx4 4Mx16 4Mx16 4Mx4 4Mx16	16 + 2 4 + 1 4 + 1 16 + 2 4 + 1
128M Single	12/10/2 12/10/2 13/10/1 13/9/2	16Mx4 16Mx16 16Mx4 16Mx16	16 + 2 4 + 1 16 + 2 4 + 1	128M Dual	12/9/2 12/9/2 13/8/2 13/9/1 12/9/2	8Mx8 8Mx32 8Mx32 8Mx8 8Mx16	8 + 1 2 + 1 2 + 1 8 + 1 4 + 1
256M Single	13/10/2	32Mx8	8 + 1	256M Dual	12/10/2 12/10/2 12/10/2 13/10/1 13/9/2	16Mx4 16Mx8 16Mx16 16Mx4 16Mx16	16 + 2 8 + 1 4 + 1 16 + 2 4 + 1
512M Single	13/11/2	64Mx4	16 + 2	512M Dual	13/10/2 13/10/2	32Mx8 32Mx4	8 + 1 16 + 2
				1024M Dual	13/11/2	64Mx4	16 + 2

Note: The number of chips per MCER is double the number of chips per DIMM bank.

SDRAM Buffering Requirements

The SDRAM interface is designed to run in a 100 MHz environment. Because signal loading is critical, some outputs connect to four or eight pins. The following table lists loads and running frequencies for all SDRAM signals that use the 60x bus clock.

SDRAM Input Signal Frequencies

Signal Name	Running Frequency	Maximum Input Capacitance		Note
		SDRAM (Note 1)	Unbuffered DIMM (Note 2)	
$\overline{\text{SDCS}}$	BUS_CLK	5pF	30pF	3
SDCKE	BUS_CLK	5pF	50pF	3
MDATA0/1	BUS_CLK/2	7pF	15pF	
MADDR0/1	BUS_CLK/2	5pF	50pF	
BS	BUS_CLK/2	5pF	50pF	
$\overline{\text{SDRAS}}$	BUS_CLK/2	5pF	50pF	
$\overline{\text{SDCAS}}$	BUS_CLK/2	5pF	50pF	
$\overline{\text{WE}}$	BUS_CLK/2	5pF	50pF	
SDDQM	BUS_CLK/2	5pF	50pF	

1. These are usual values for a single SDRAM chip (V=3.3V, T=25C, f=1MHz)
2. These are usual values for an unbuffered DIMM (8 x 1M x 16) (V=3.3V, T=25C, f=1MHz)
3. Signal is critical, runs at full speed.

DIMM Row Address Derivation for SDRAM x72 Width

DIMM Addressing	DIMM Row Address												
	12	11	10	9	8	7	6	5	4	3	2	1	0
13/11/2	4	7	9	10	11	12	13	14	15	16	17	18	19
13/10/2	4	7	9	10	11	12	13	14	15	16	17	18	19
13/9/2	4	7	9	10	11	12	13	14	15	16	17	18	19
13/10/1	4	7	9	10	11	12	13	14	15	16	17	18	19
13/9/1	5	7	9	10	11	12	13	14	15	16	17	18	19
13/8/2	6	7	9	10	11	12	13	14	15	16	17	18	19
13/8/1	6	7	9	10	11	12	13	14	15	16	17	18	19
12/10/2	-	7	9	10	11	12	13	14	15	16	17	18	19
12/9/2	-	7	9	10	11	12	13	14	15	16	17	18	19
12/8/2	-	7	9	10	11	12	13	14	15	16	17	18	19
12/9/1	-	7	9	10	11	12	13	14	15	16	17	18	19
12/8/1	-	7	9	10	11	12	13	14	15	16	17	18	19
11/10/1	-	-	9	10	11	12	13	14	15	16	17	18	19
11/9/1	-	-	9	10	11	12	13	14	15	16	17	18	19
11/8/2	-	-	9	10	11	12	13	14	15	16	17	18	19
11/8/1	-	-	9	10	11	12	13	14	15	16	17	18	19

DIMM Column and Bank Address Derivation for SDRAM x72 Width

DIMM Addressing	DIMM Column Address												DIMM Bank Address	
	11	10[2]	9	8	7	6	5	4	3	2	1	0[1]	1	0
13/11/2	2	-	3	6	20	21	22	23	24	25	26	27	5	8
13/10/2	-	-	3	6	20	21	22	23	24	25	26	27	5	8
13/9/2	-	-	-	6	20	21	22	23	24	25	26	27	5	8
13/10/1	-	-	5	6	20	21	22	23	24	25	26	27	-	8
13/9/1	-	-	-	6	20	21	22	23	24	25	26	27	-	8
13/8/2	-	-	-	-	20	21	22	23	24	25	26	27	5	8
13/8/1	-	-	-	-	20	21	22	23	24	25	26	27	-	8
12/10/2	-	-	4	6	20	21	22	23	24	25	26	27	5	8
12/9/2	-	-	-	6	20	21	22	23	24	25	26	27	5	8
12/8/2	-	-	-	-	20	21	22	23	24	25	26	27	6	8
12/9/1	-	-	-	6	20	21	22	23	24	25	26	27	-	8
12/8/1	-	-	-	-	20	21	22	23	24	25	26	27	-	8
11/10/1	-	-	6	7	20	21	22	23	24	25	26	27	-	8
11/9/1	-	-	-	7	20	21	22	23	24	25	26	27	-	8
11/8/2	-	-	-	-	20	21	22	23	24	25	26	27	7	8
11/8/1	-	-	-	-	20	21	22	23	24	25	26	27	-	8

1. The Memory Controller interleaves with only one memory address bus. To handle critical word load, individual control of the LSB column address bits is required for the DIMMs. MADDR0_ODD is used for the LSB address of the even and odd DIMMs.
2. Bit 10 is never used as address during $\overline{\text{CAS}}$ phase.

Memory Controller Registers

MCCR Register

The Memory Controller Control Register contains all the parameters to fit the Memory Controller to the Synchronous DRAM components used. The following table describes how to program the MCCR Register (*Memory Controller Control Register (MCCR)* on page 122.

Bit(s)	Description
0	**Global System Memory Address Space Enable** 0: Device will not respond to addresses specified in Memory Configuration Extent Register (MCERx) 1: System memory address space enabled.
1	**Diagnostic Mode** This bit is used to control presentation of double-bit ECC errors to the system. This bit is primarily intended for use in memory testing at power on time. Software can use this bit when testing memory and or ECC logic in order to avoid the hardware generating a machine check for double-bit ECC errors. The error however, is still logged into the MEAR 0: Normal Mode: Multi-bit ECC error will generate Machine Check 1: Diagnostic Mode: Multi-bit ECC does NOT generate Machine Check; logged in MEAR & MESR
2	**SDRAM Initialization Status (read-only)** 0: SDRAM initialization is not completed. 1: SDRAM initialization is completed.
3 - 4	**ECC Mode** This field provides software with a means to control ECC generation and checking. b'01' is provided to allow software direct read/write access to the ECC byte that is associated with every doubleword of data stored in memory and also provide a mechanism to verify the memory controller's ECC generation and checking logic. In this mode, byte lane 0 data (MSB of a double word) is written to the ECC byte instead of the normal ECC code byte. Data byte 0 will be forced to all zeros. For reads, byte 0 will contain the byte stored in the ECC byte, not the data at byte 0. ECC checking is not enabled for reads in this mode. This mode also allows firmware write single-bit and multi-bit errors into memory to allow for ECC logic testing. 00: Normal generation and checking of ECC codes. The device will generate the normal ECC code when writing to memory and check ECC when reading. 01: ECC check disabled; Byte lane 0 routed to/from ECC check field. Data byte 0 forced to all zeros This mode is provided to allow software direct read/write access to the ECC byte that is associated with every doubleword of data stored in memory and also provide a mechanism to verify the memory controller's ECC generation and checking logic. In this mode, byte lane 0 data (MSB of a double word) is written to the ECC byte instead of the normal ECC code byte. Data byte 0 will be forced to all zeros. For reads, byte 0 will contain the byte stored in the ECC byte, not the data at byte 0. ECC checking is not enabled for reads in this mode. This mode also allows firmware write single-bit and multi-bit errors into memory to allow for ECC logic testing. The device will still generate normal ECC codes when writing to memory. 10: ECC check disabled; Normal routing of data and normal ECC code generation The device will still generate normal ECC codes when writing to memory. 11: Reserved
5 - 7	**Row Cycle Time for SDRAM Auto-refresh (t_{RC})** Allows to fit the delay between the Refresh Command and the next Activation. This delay has to be at least the tRCmin value specified in the SDRAM datasheet. 000: 5 bus cycles 001: 6 bus cycles 010: 7 bus cycles 011: 8 bus cycles 100: 9 bus cycles 101: 10 bus cycles 110: 11 bus cycles 111: 12 bus cycles

Bit(s)	Description
8 - 9	**DRAM Type** **Must be set to 10 for SDRAM**
10	**Data Pacing Mode** (Must be set to 1 for SDRAM)
11	**Data Mask Mode** (SDRAM only) 0: 16 $\overline{\text{SDCS}}$ are available. DQM pin of SDRAM devices must be grounded. 1: Only eight $\overline{\text{SDCS}}$ are available. DQM signal is present on the eight other pins. Activate to prevent tRDL violation during “Write interrupted by pre-charge” operations if the device does not guarantee that the data presented in the same time as pre-charge is not properly ignored.
12 - 15	**Reserved.** Must be left to 0
16	**Reserved.** Must be set to 0
17	**Reserved.** Must be set to 1
18	**Reserved.** Must be set to 1
19	**Reserved.** Must be set to 0
20	**Reserved.** Must be set to 0
21 - 22	**Reserved.** Must be set to b’00’
23	**Reserved.** Must be set to 0
24 - 29	**Reserved**
30	**Disable Page Mode** 0: Memory controller will perform fast page accesses for back to back operations if appropriate 1: Memory controller will perform fast page access only within a burst operation. It will NOT perform fast page accesses for back to back bursts even if they occur to the same $\overline{\text{RAS}}$ page.
31	**Disable Queue Same Page Override** 0: Memory queue ordering can be overridden if an operation is to the same page. 1: Memory queue always processed in order received.

MCCR Register Settings

Configuration	Register Bits																								
	0	1	2	3-4	5-7	8-9	10	11	12-15	16	17	18	19	20	21 22	23	24	25	26	27	28	29	30	31
SDRAM with t_{RC}=8 cycles	1	0		0 0	011	**1 0**	**1**	1	**0 0 0 0**	**0**	**1**	**1**	**0**	**0**	**0 0**								0	0

Note: Values written in bold font are mandatory for the specified configuration.

MCER Register

The Memory Configuration Extent Registers (MCER [0:7]) program the start address and size of each bank. The following table shows the relationship between the DIMMs and the MCER registers.

MCER to Program Functions of DIMMs

Bank Definition (DIMMs equipped)	Corresponding MCER	Note
DIMM0-Bank1 and DIMM1-Bank1	MCER0	
DIMM0-Bank2 and DIMM1-Bank2	MCER1	
DIMM2-Bank1 and DIMM3-Bank1	MCER2	
DIMM2-Bank2 and DIMM3-Bank2	MCER3	
DIMM4-Bank1 and DIMM5-Bank1	MCER4	1
DIMM4-Bank2 and DIMM5-Bank2	MCER5	1
DIMM6-Bank1 and DIMM7-Bank1	MCER6	1
DIMM6-Bank2 and DIMM7-Bank2	MCER7	1

1. When using SDRAM and Data Mask, Mode is active (see MCCR, bit 11) and device can support only up to four bank. MCER [4:7] must be off.

To configure contiguous address spaces with different bank sizes, software must put the largest bank sizes at the lowest addresses and continue in order to the smallest bank sizes. To set up the MCER registers, software must read the PD bits and the ID bits for each DIMM. These bits are located in the System I/O registers (see *System I/O Register 0 (SIOR0)* on page 128). The following table describes how to initialize these registers.

MCER Register Initialization (see notes 1-4)

DIMM Description		Device Bank Size (MB)		MCER(x) Bits[16:25]	MCER(x+1) Bits[16:25]
DIMM Size (MB)	Number of Banks per DIMM	Bank x	Bank x+1		
2	1	4	Not equipped	x'3F3'	Off
4	2	4	4	x'3F3'	x'3F3'
4	1	8	Not equipped	x'3E3'	Off
8	2	8	8	x'3E3'	x'3E3'
8	1	16	Not equipped	x'3C3'	Off
16	2	16	16	x'3C3'	x'3C3'
16	1	32	Not equipped	x'383'	Off
32	2	32	32	x'383'	x'383'

1. "DIMM size" is the size in MB of one DIMM (including Bank A and Bank B if dual bank DIMM).
2. "Number of banks per DIMM": One for single bank DIMM (i.e., DIMM equipped with Bank A only); two for dual bank DIMM (i.e., DIMM equipped with Bank A and Bank B).
3. x in MCER(x) = 0, 2, 4, or 6
4. a setting of “off” indicates that the bank must be disabled by setting MCER(x) Bit 0 = 0.

MCER Register Initialization (see notes 1-4)

DIMM Description		Device Bank Size (MB)		MCER(x) Bits[16:25]	MCER(x+1) Bits[16:25]
DIMM Size (MB)	Number of Banks per DIMM	Bank x	Bank x+1		
32	1	64	Not equipped	x'303'	Off
64	2	64	64	x'303'	x'303'
64	1	128	Not equipped	x'203'	Off
128	2	128	128	x'203'	x'203'
128	1	256	Not equipped	x'003'	Off
256	2	256	256	x'003'	x'003'
256	1	512	Not equipped	x'002'	Off
512	2	512	512	x'002'	x'002'
512	1	1024	Not equipped	x'000'	Off
1024	2	1024	1024	x'000'	x'000'

1. "DIMM size" is the size in MB of one DIMM (including Bank A and Bank B if dual bank DIMM).
2. "Number of banks per DIMM": One for single bank DIMM (i.e., DIMM equipped with Bank A only); two for dual bank DIMM (i.e., DIMM equipped with Bank A and Bank B).
3. x in MCER(x) = 0, 2, 4, or 6
4. a setting of "off" indicates that the bank must be disabled by setting MCER(x) Bit 0 = 0.

Error Handling

The memory controller detects four errors:

1. Single-bit ECC
2. Multi-bit ECC
3. Invalid address
4. Overlapping memory extents

Errors 2, 3, and 4 are considered hard errors. If one occurs, it is logged into MESR and MEAR and cannot be overwritten with a subsequent hard error. Single-bit ECC errors are considered soft and once logged into the MEAR and MESR, can be overwritten with a subsequent hard error.

Single-Bit ECC Error, General Case

The hardware procedure for this error is:

1. Set the single-bit error bit in the MESR register.
2. If neither a double-bit error nor an address error is present, store the syndrome in the MESR and the address in MEAR.
3. Corrected data is not written back to memory but forwarded to the requesting logic.
4. When Chip Programmability Register (PGCHP) bit 17 = 1, a Machine Check is performed to signal the processor that it could rewrite correct data to memory.

Software must write zeros to the MESR to clear this error. If more than one single-bit ECC error occurs before the MESR clears, only the first error is recorded. When a double-bit ECC error or an address error occurs, the software overwrites the MESR and MEAR.

Single-Bit ECC Error, Special Case

For non-burst write transactions that do not span an entire aligned double-word, the Memory Controller performs a read-modify-write sequence to memory. If the read portion of the sequence results in a single-bit ECC error, the error is not logged into the MESR and MEAR for both the diagnostic and normal modes. However, the memory controller automatically writes corrected data to memory.

Invalid Address Error

An Invalid Address error is detected by the Memory Controller when an address does not match one of the eight configuration extents. The hardware procedure for this error is:

1. If no hard errors are in the MESR register, set the invalid address error bit.
2. If no hard errors are in the MEAR register, store the address.
3. In diagnostic mode, the Memory Controller responds with dummy data and indicates an Invalid Address error to the requesting logic. To enable further error logging, the software writes zeros into the MESR.

When more than one address error occurs before the MESR clears, only the first error is recorded. No Single- or Double-Bit ECC errors are logged into the MESR and MEAR if they occur after the Invalid Address error.

Double-Bit ECC Error, General Case

The hardware procedure for this error is:

1. Set the Double-Bit error bit in MESR if no hard errors are present.
2. Store the syndrome in MESR and the address in MEAR if no hard errors are present.
3. In normal mode, indicate the error to the requesting logic with DAT_ERR for the appropriate double word that failed in memory.
4. In diagnostic mode, do not indicate Double-Bit errors with DAT_ERR.
5. Software must write zeros to the MESR to clear errors.
6. If more than one Double-Bit error occurs before the MESR clears, only the first error is recorded.
7. A Single-Bit error is not logged into MESR and MEAR when it occurs after a Double-Bit error.
8. If an Address error occurs after a Double-Bit error, it is not be logged into MESR and MEAR.

Double-Bit ECC Error, Special Case

For non-burst write transactions that do not span an entire aligned double-word, the Memory Controller performs a read-modify-write sequence to memory. If the read portion of this sequence results in a Double-Bit error, the controller indicates the error to the requesting logic through the response bus instead of using DAT_ERR. If this occurs in diagnostic mode, the error is logged into MESR and MEAR, but not reported through the response bus.

Overlapping Memory Extents

Overlapping Memory Extents are not detected until an access occurs to an address mapped to two different configuration extents. When an overlap condition is detected, the hardware follows the following procedure:

1. Set the Overlapping Memory Extent error bit in MESR if no hard errors exist.
2. Store the address in MEAR if no hard errors exist.
3. The Memory Controller responds with dummy data for reads, ignores write data, and indicates an Invalid Address error to the requesting logic. To enable further error logging, the software writes zeros into the MESR.

When a Single-bit or a hard error occurs after an Overlapping Memory Extent, the error is not logged into MESR and MEAR.

PCI Bridges

Overview

The device's PCI Bridge function executes load and store operations from the CPU to the PCI buses. It also provides an interface for PCI devices to access system memory. The PCI Bridge logic fully supports the PCI Local Bus Specification [2]. The following table describes the physical connections for PCI devices on the PCI-32 bus in a desktop system.

PCI-32 Bus Device Physical Connection Example

Device	ARB Level	RESET Signal	IDSEL Signal[1]
PCI SLOT 0	$\overline{\text{P\_REQ0}}$/$\overline{\text{P\_GNT0}}$	$\overline{\text{P\_RST}}$	IDSEL1
PCI SLOT 1	$\overline{\text{P\_REQ1}}$/$\overline{\text{P\_GNT1}}$	$\overline{\text{P\_RST}}$	IDSEL2
PCI SLOT 2	$\overline{\text{P\_REQ2}}$/$\overline{\text{P\_GNT2}}$	$\overline{\text{P\_RST}}$	IDSEL3
ETHERNET CHIP	$\overline{\text{P\_REQ3}}$/$\overline{\text{P\_GNT3}}$	$\overline{\text{P\_RST}}$	IDSEL4
SCSI CHIP	$\overline{\text{P\_REQ4}}$/$\overline{\text{P\_GNT4}}$	$\overline{\text{P\_RST}}$	IDSEL5
ISA BRIDGE CHIP	$\overline{\text{P\_REQ5}}$/$\overline{\text{P\_GNT5}}$	$\overline{\text{P\_RST}}$	IDSEL6
MPIC	Not required	$\overline{\text{POWERGOOD}}$	IDSEL7

1. See *External IDSEL Signal Logic for PCI-32* on page 193 for description of these signals

Address Map

There are two PCI bus bridges in the device. Both implement the register maps listed in the following table. The PCI Host Bridge Standard configuration space is Little Endian.

PCI Bus Bridge Configuration Address Map

Area	Real Address	Name	Note	Use	Page
System Standard Configuration Space	x'FF20 0000'	DCR	1	Device Characteristics Register	138
	x'FF20 0004'	DID	1	Device ID Register	139
	x'FF20 0018'	BAR		Base Address Reg. for Bridge Registers	140
	x'FF20 001C' to x'0FFF'			Reserved	
Device Specific Configuration Space	x'FF20 1000'	PCIENB		PCI BAR Enable Register	141
	x'FF20 1004' to x'1FFF'			Reserved	

1. Read Only Register, write is ignored

System Standard Configuration Registers

System Standard Configuration Registers can only be accessed with 60x bus configuration cycles directed to a specific PCI bridge. Both of the device's PCI bridges must be configured before any PCI configuration cycles can be issued. The registers provide a mechanism for firmware to identify the PCI bridge and the DCR and DID registers, and assign a 1 MB address space in the system memory map for the location of the PCI bridge facilities (BAR register). For detailed descriptions of these registers, refer to the following:

- *Device Characteristics Register (DCR)* on page 138
- *Device ID Register (DID)* on page 139
- *Base Address Register (BAR)* on page 140
- *PCI BAR Enable Register (PCIENB)* on page 141

System PHB Registers

The PCI bridge logic follows the PowerPC PCI Host Bridge (PHB) Architecture, including the enhanced error detection and error reporting features. The logic deviates from PHB Architecture only in its ability to recover from PCI errors.

PCI Bus Commands

The following table describes the subset of PCI bus commands supported by the device.

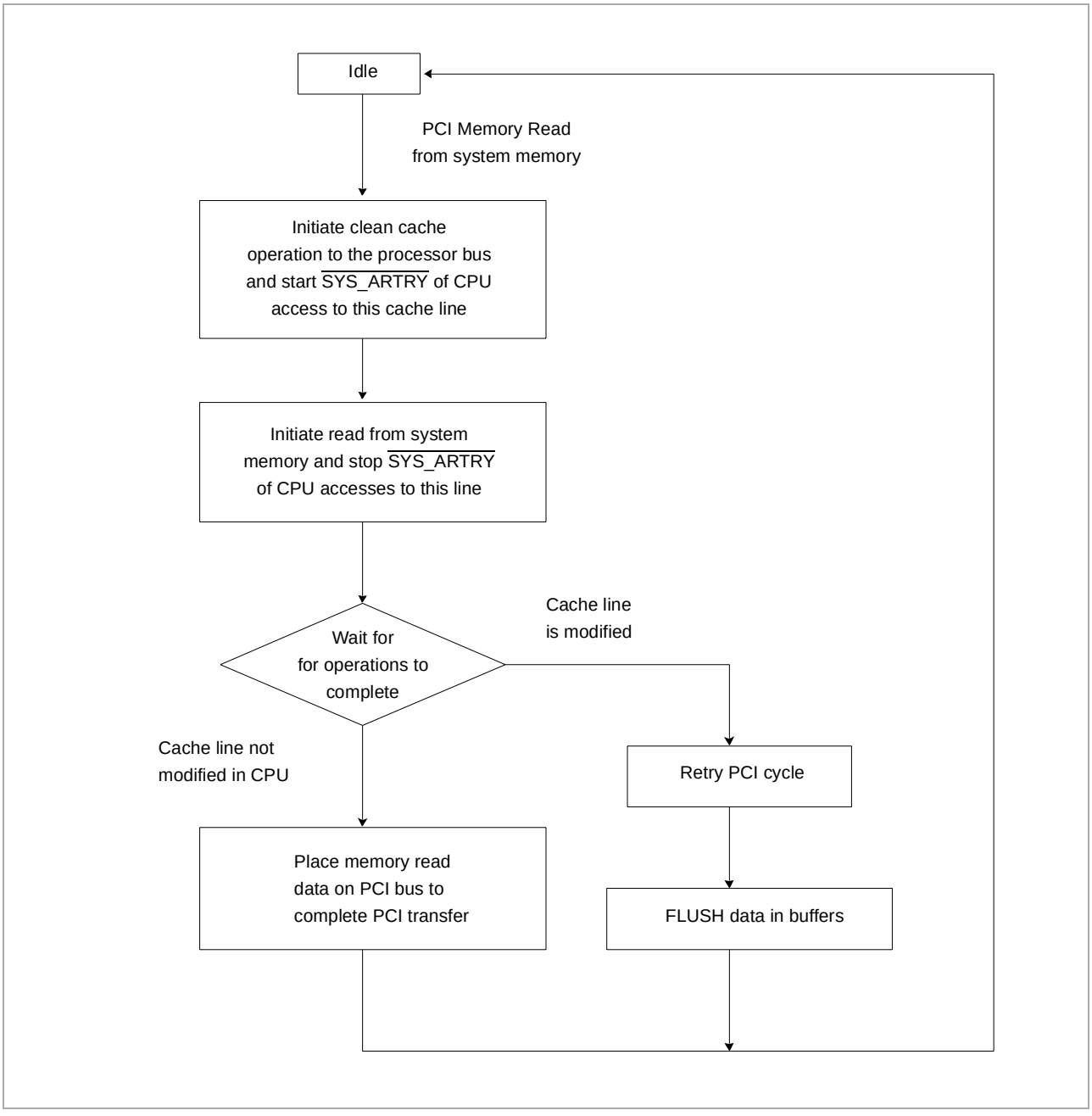
Supported PCI Commands

C/BE[3:0]	Command	Support as Initiator	Support as Target
0000	Interrupt Acknowledge	Yes	No
0001	Special Cycle	Yes	No
0010	I/O Read Cycle	Yes	No
0011	I/O Write Cycle	Yes	No
0100	Reserved		
0101	Reserved		
0110	Memory Read	Yes	Yes
0111	Memory Write	Yes	Yes
1000	Reserved		
1001	Reserved		
1010	Configuration Read	Yes	Yes (PCI-64 only)
1011	Configuration Write	Yes	Yes (PCI-64 only)
1100	Memory Read Multiple	No	Yes
1101	Dual Address Cycle	No	No
1110	Memory Read Line	Yes	Yes
1111	Memory Write and Invalidate	Yes	Yes

PCI Master Memory Read Cycles

When the device receives a memory read bus cycle from system memory, it first initiates a CLEAN cache operation to the processor bus. Processor accesses to this cache line are SYS_ARTRYed until the memory read is finished. If the cache line is determined to be stale in memory, the PCI bus cycle is retried. The following figure shows the states the device follows when executing a PCI memory read cycle.

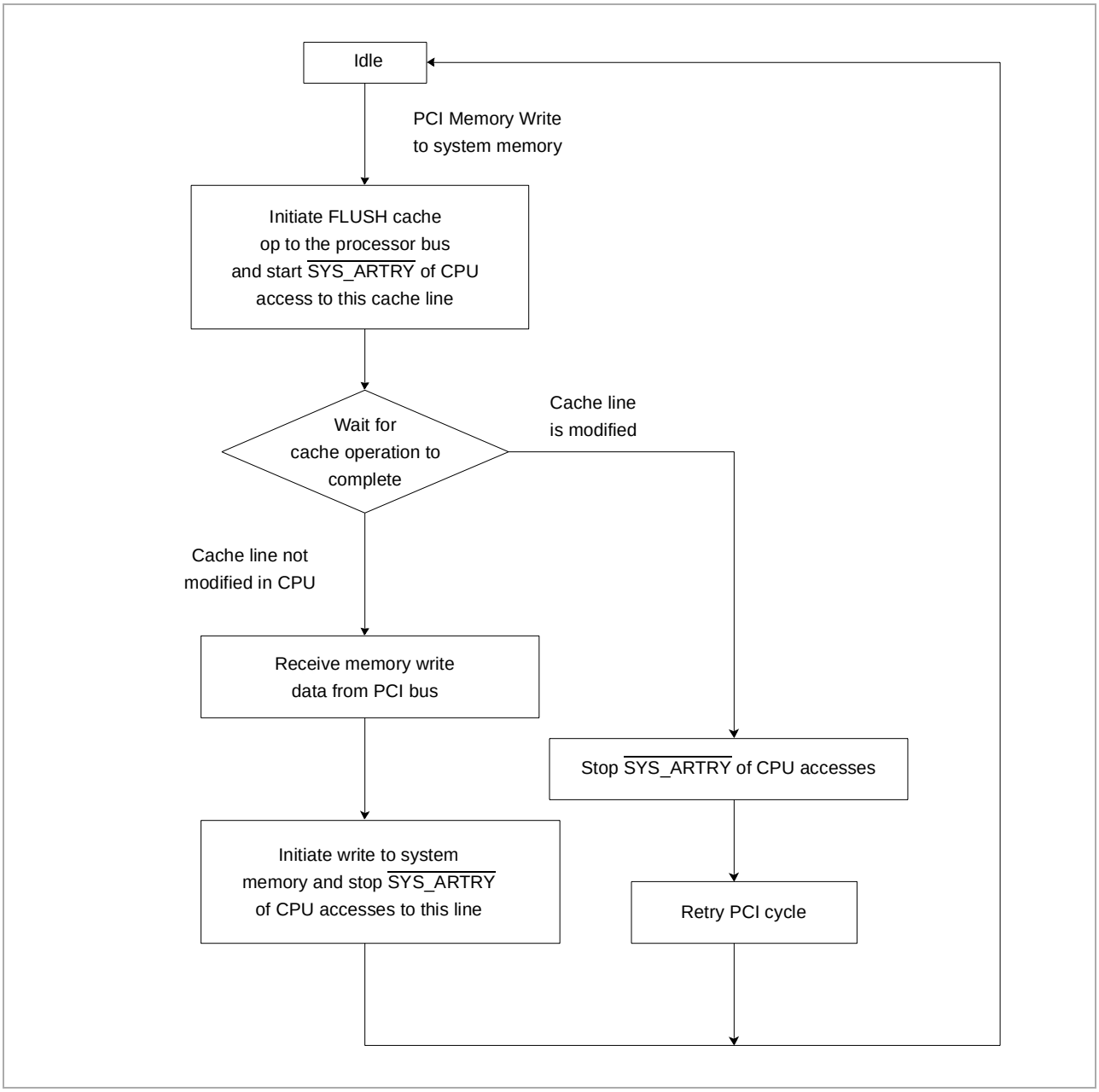
PCI Memory Read State Diagram



PCI Master Memory Write Cycles

When the device receives a memory write bus cycle to system memory, it first initiates a FLUSH cache operation to the processor bus. Processor accesses to this cache line will be SYS_ARTRYed until the memory write is finished. If the cache line is determined to be stale in memory, the PCI bus cycle is retried. The following sequence describes the states the device follows when executing a PCI memory write cycle.

PCI Memory Write State Diagram



Configuration Cycles

The device implements **Configuration Mechanism #1** as specified in the PCI Local Bus Specification [2]. This mechanism uses an indirect addressing model with the CONFIG_ADDRESS and CONFIG_DATA registers. The configuration target address is first written into CONFIG_ADDRESS and then an access is made to CONFIG_DATA to generate a configuration transfer. Each PCI bridge has a separate set of these registers. When each decodes an access to its CONFIG_DATA register, it performs different operations depending on the values stored in CONFIG_ADDRESS.

PCI Configuration Cycle Matrix

CONFIG_ADDRESS Register Fields					Action	Notes
Enable	Bus#	Device#	Function#	Register#		
0	x	x	x	x	Configuration not enabled. Returns 0's on loads and ignores write data.	1
1	BUS# < BUSNO	x	x	x	Invalid Bus# in CONFIG_ADDR. Returns 1's on loads and ignores store data. No access made to PCI Bus.	1, 2
	BUS# = BUSNO	0	0	x	Access to PCI Bridge configuration space. Read/Write to PCI Bridge configuration registers.	1, 2
		1-21	x	x	Configuration access to device on PCI Bus. TYPE 0 configuration cycle on PCI bus.	
		22-30	x	x	Not supported. TYPE 0 configuration cycle with no IDSELs on. Returns 1s on loads and ignores store data.	
		31	7	0	Special cycle command. Special cycle command issued to PCI Bus.	
	BUS# > BUSNO BUS# < SUBNO	x	x	x	Configuration access to bridge on PCI Bus. TYPE 1 configuration cycle on PCI Bus	1, 2, 3
	BUS# > BUSNO BUS# > SUBNO	x	x	x	Invalid bus# CONFIG_ADDR. Returns 1s on loads and ignores store data. No access made to PCI Bus.	

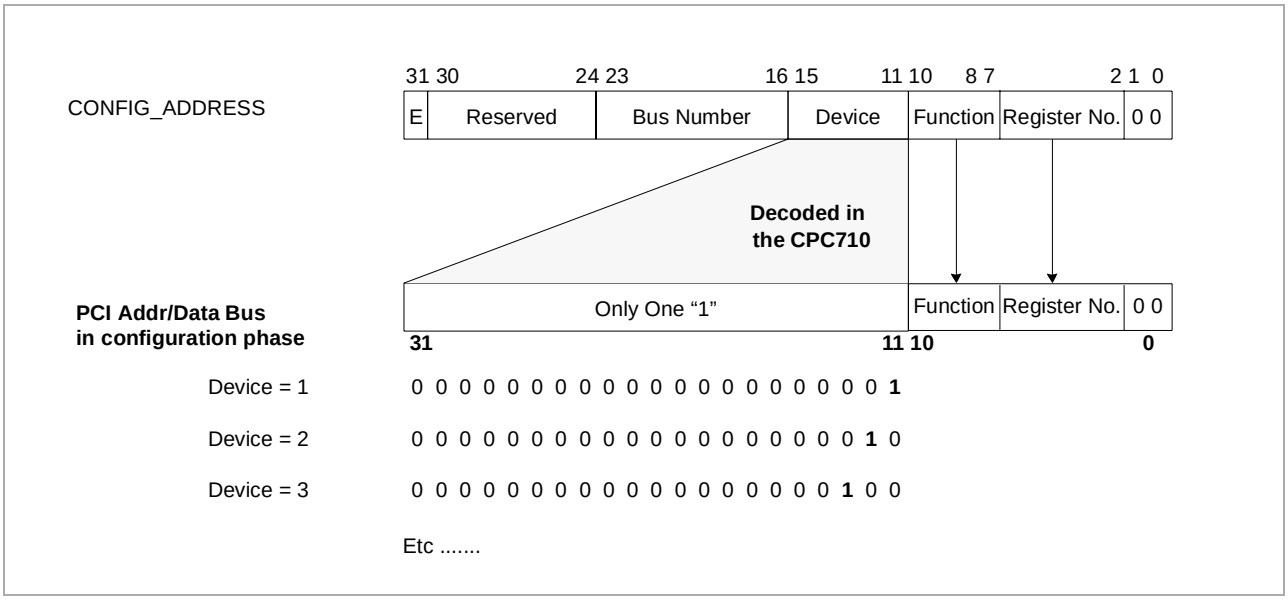
1. Firmware must insure the SUBNO register in the PCI header is greater than or equal to the BUSNO register in the PCI header. Unpredictable results can occur if this is not true.
2. The PCI Bridge performs a compare of the BUS NUMBER field in the CONFIG_ADDRESS register and the BUS NUMBER field in the bridge's 256-byte PCI header.
3. The PCI Bridge performs a compare of the BUS NUMBER field in the CONFIG_ADDRESS register and the SUBORDINATE BUS NUMBER field in the bridge's 256-byte PCI header.

If there is no response to a configuration cycle (no DEVSEL# detected), the device Master-Aborts the cycle, sets the Master Abort bit in the PCI Status register, and completes the processor cycle normally by returning all ones on reads and ignoring data on writes.

TYPE 0 Configuration Cycles

During a TYPE 0 configuration cycles, the CPC710 provides on the AD[11-31] lines the IDSEL of the device to be configured on the PCI-32 or PCI-64 bus as described in the PCI 2.1 specification.

PCI Address/Data Bus for TYPE 0 Configuration Cycles

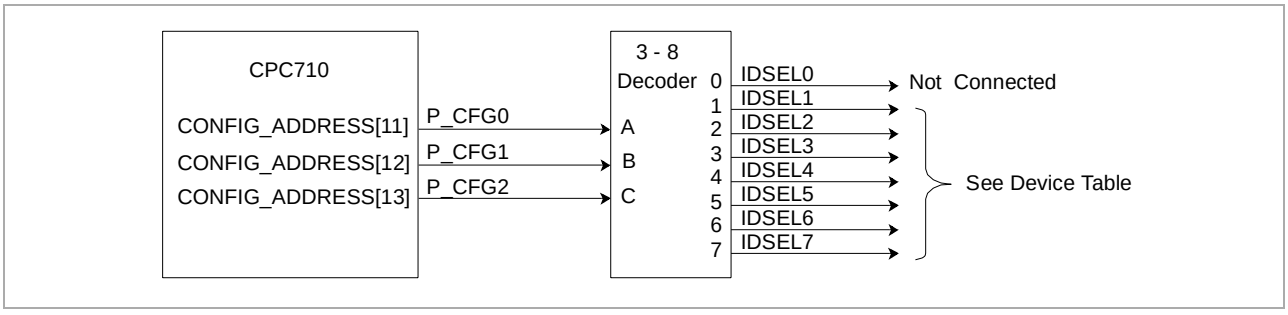


External IDSEL Signals Logic on PCI-32 and PCI-64: (CONFIG_ADDRESS[14-15]=00 only):

In order to decrease loading of the AD lines, the CPC710 also indirectly supports up to eight separate IDSEL lines. It relies on external 3-8 decoders to provide a unique signal for each device on the bus. The CPC710 drives external address bits 13 through 11 (Little Endian) using the three P/G_CFG signals. It also drives PCI address/data bus bits 31 through 11 as shown in the above table.

The recommended external connection for the 3-8 decoders is shown below.

External IDSEL Signal Logic for PCI-32



Type 1 Configuration Cycles

For TYPE 1 configuration cycles, the CPC710 directly copies the contents of the CONFIG_ADDRESS register to the Address/Data signals on the PCI bus. However, Address/Data[1:0] contains '01' to indicate a TYPE 1 configuration cycle.

PCI Performance Estimates

PCI to Memory Sustained Throughput

	Read	Write	Units
PCI-64bit @ 66 MHz	96	132	MByte/sec.
	18-1-1-1-PARL	12-1-1-1-PARL	66MHz PCI Cycles
PCI-32bit @ 33 MHz	53	58	MByte/sec.
	12-1-1-1-1-1-1-1-PRAL	12-1-1-1-1-1-1-1-PRAL11-1-1-1-1-1-1-1-PRAL	33MHz PCI Cycles

Assumptions:
- 4KBytes Burst
- PCI Master parked on PCI bus
- No other activity present
- Adapter supports fast back-back transfers for stores to memory
- No L1 or L2 cache hits

PARL (PCI Rearbitration Access Latency) min = 1 cycle

CPU to PCI Sustained Throughput

Operation	Loads @ 100 MHz Bus		Stores @ 100 MHz Bus		Units
	PCI-32bit @ 33 MHz	PCI-64bit @ 66 MHz	PCI-32bit @ 33 MHz	PCI-64bit @ 66 MHz	
Burst 32 bytes	71	194	71	194	MB/s
Single 8 bytes	30	67	30	67	MB/s
Single 4 bytes	17	33	17	33	MB/s

Assumptions:
- CPU is parked on 60x bus
- 1 Level Pipeline
- IBM25CPC710AB3A100 parked on PCI bus
- No other activity present

PCI Master Error Handling

For PCI bus errors detected on CPU initiated transfers, refer to *Error Handling for CPU-Initiated Transactions* on page 164. The following table describes the error handling performed for PCI master errors.

PCI Master Error Handling (Page 1 of 2)

Operation	Error	Mode	Action	Notes
Any PCI Bus Transfer	Address Parity Error	Enabled by PCI CMND register bit 6	Save encoded arb level in CSR register Set Address parity error detected bit in CSR register Set Parity error detected bit 15 in PCI status register Place PCI address in PSEA register Activate SERR signal if enabled by bit 8 PCI CMND register Set Signalled SERR bit in PCI Status register if enabled Target abort PCI transfer if address matches Set Signaled target abort bit in PCI status register Signal Machine Check with $\overline{\text{SYS\_MACHK}}$	
		Disabled	Set Parity error detected bit 15 in PCI status register Complete PCI transfer normally if address matches	1
	Detected SERR Active PCI Bridge Logic Idle		Set Detected SERR active bit in CSR register Save encoded ARB level in CSR register Signal Machine Check with $\overline{\text{SYS\_MACHK}}$	
Access to System Memory	Single Bit Error		Set single-bit error and syndrome in MESR Set error address in MEAR Return corrected data to PCI device Proceed normally with PCI transaction	1
	Double Bit Error	Normal	Set double-bit error in MESR Set error address in MEAR Set memory error bit in CSR register Loads: - Target abort PCI transfer - Set signaled target abort bit in PCI status register - Signal Machine Check with $\overline{\text{SYS\_MACHK}}$ Stores: - Signal Machine Check with $\overline{\text{SYS\_MACHK}}$	
		Diagnostic	Set double-bit error in MESR Set error address in MEAR Return uncorrected data to PCI device Proceed normally with PCI transaction	1

Notes:

1. “Normally” means that dummy zeros are returned for loads and write data is ignored.

PCI Master Error Handling (Page 2 of 2)

Operation	Error	Mode	Action	Notes
Access to System Memory (cont'd)	Invalid Address		Set invalid address error in MESR Set error address in MEAR Set invalid memory address bit in CSR Loads: - Target abort PCI transfer - Set signaled target abort bit in PCI status register - Signal Machine Check with $\overline{\text{MACHK}}$ Stores: - Signal Machine Check with $\overline{\text{MACHK}}$	
	Detected SERR Active		Set SERR detected error bit in CSR register Save encoded ARB level in CSR register Target abort PCI transfer Set signaled target abort bit in PCI status register Signal Machine Check with $\overline{\text{MACHK}}$	
	Detected PCI Bus Data Parity Error during PCI Master Store	Enabled by PCI CMND register bit 6	Activate the PERR signal Set parity error bit 15 in PCI Status register Complete PCI transfer, however, Flush store data; do not write to memory	
		Disabled	Set parity error bit 15 in PCI Status register Proceed normally with PCI transaction	1
	Detected PERR during PCI Master Load		Proceed normally with PCI transaction	1
	Received Master Abort		Proceed normally with PCI transaction	1
	PCI Bus Timeout: IRDY Count Expired		Target abort PCI transfer Set signaled target abort bit in PCI status register Set PCI bus time-out error in CSR register Save encoded ARB level in CSR register Signal Machine Check with $\overline{\text{MACHK}}$	
Access to Device on 2nd PCI Bus	Internal Response Bus Contains "PCI Error" Status		2nd PCI bridge logs errors same as CPU initiated 2nd PCI bridge does NOT drive $\overline{\text{MACHK}}$ pin Set PCI - PCI error bit in CSR register Save encoded ARB level in CSR register Save PCI address in PSEA register Loads: - Target abort PCI transfer - Set signaled target abort bit in PCI status register - Signal Machine Check with $\overline{\text{MACHK}}$ Stores: - Signal Machine Check with $\overline{\text{MACHK}}$	

Notes:

1. "Normally" means that dummy zeros are returned for loads and write data is ignored.

System I/O Interface

The device implements a 2 MB ROM space from address 4G-2M to 4 GB .

Configuration

There is no configuration requirement for SIO logic. These areas are hard wired in the upper 16 MB of real memory.

System I/O Registers: Application Presence Detect Bits

The device provides Output Enables signals and read cycles for two external 32-bit registers. The read of the SIOR0 or SIOR1 results in a read of bits 0 to 31 of these register which correspond respectively to the data present on the line 31 and 0 of the PCI 32 bit A/D during the read cycle.

For descriptions of these registers, refer to:

- *System I/O Register 0 (SIOR0)* on page 128, controls PRES_OE0 signal
- *System I/O Register 1 (SIOR1)* on page 129, controls PRES_OE1 signal
- *MCCR Register Settings* on page 181 (for PD definition and the device's supported values)

Flash Interface

Boot Rom

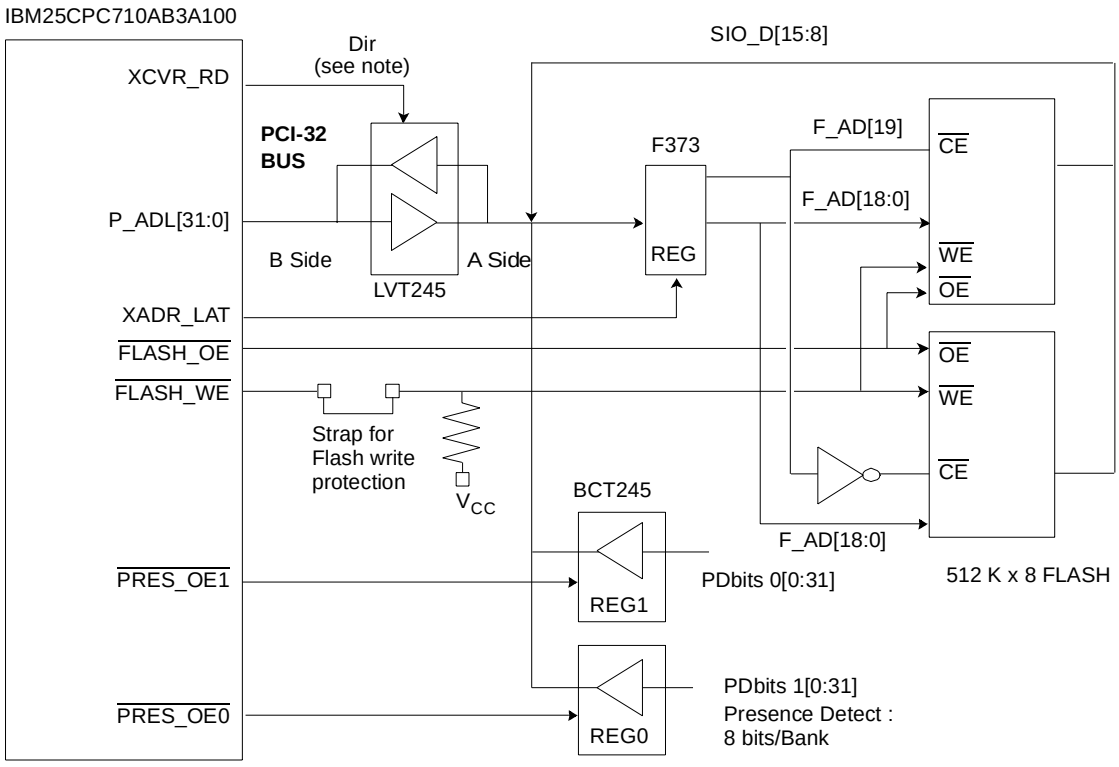
The Device's Boot ROM base address is fixed at x'FFF0 0000'. Accesses to the architected Boot ROM space within the size limit defined in the System I/O Control Register (*System I/O Control (SIOC)* on page 103) are decoded as valid Boot ROM accesses. If the ROM Size parameter is larger than the actual amount of installed Boot ROM, the data will wrap. An access within the architected Boot ROM space but outside the size limit (SIOC x'FF00 1020') results in a bus timeout Machine Check error. The Boot ROM interface logic satisfies burst read requests from the processor by concatenating multiple bytes from the Boot ROM.

The device is designed to interface with 512 K, 1 Mb, 2Mb (x8) 3.3 V Flash memory with 80 to 120 ns access time. The following figure shows Boot Flash with the bits used for Address and Data on the PCI-32 bus AD lines. PCI AD bits 20:0 are used for Flash Address (LSB starts at bit 0).

Bits [15:8] of the PCI-32 bus AD lines are used for the 8-bit data. The Boot Flash is accessed under control of the device's PCI-32 controller to generate non-PCI cycles with FRAME not asserted. Flash is read and written by setting bit 4 (R/W) in the UCTL Register.During the Flash access the PCI bus is clocked by the System Clock.

The address on the PCI bus is defined from 0 Lsb to 28 Msb.

Connection of Boot ROM and System I/O Registers (PD) to Device



Note: There is no output enable control for the LVT245 drivers. All control is done by the Direction control bit on signal XCVR_RD.
For a Read of the Boot Flash, the XCVR_RD signal is at the Up level =1, such that the data is transfered through teh 245 buffer from the A side (Flash) to the B side(PCI-32 A/D).

DMA Controller

Introduction

The data transfers between the system memory and the PCI buses can be performed either by the DMA controller or by a PCI master on one of the PCI busses which can access in Read or Write the System memory (See the PCI section).The DMA is initiated either by a PowerPC specific instruction or by writing to the XTAR register. The DMA is defined with one channel, and with several type of mode of operations. To signal the end of the DMA operation, the External interrupt $\overline{INT2}$ is raised.

Mode of operation of the DMA

A complete DMA transfer can be done in the following modes that can be programmed in the DMA Global Control Register (GSCR):

- Elementary
- Extended Mode

The DMA controller runs with an elementary block of up to 4 KB of data to transfer.

In the Extended Mode, an automatic address increment is performed at the end of each elementary DMA transfers. Up to 65,000 iterations (or loops) of elementary DMA's can be programmed with address increments to transfer up to 256 MB of data in a single DMA. The end-of-transfer DMA interrupt $\overline{INT2}$ is raised only after completion of the multiple elementary DMAs loops.

Starting the DMA

Write in the XTAR register

The write in the XTAR register results in the start of a DMA operation.

eciwx or ecowx instruction

DMAs are initiated by either a eciwx (read: Data from Memory to PCI) or ecowx (write: PCI to Memory) instruction from the processor and ended by an External Interrupt command. The controller uses an elementary burst of 32 Bytes on the PCI bus to facilitate interleaved PCI bus operations. The eciwx and ecowx instructions use the processor's internal address translation logic to present real addresses on the system bus. This eliminates the need for external hardware to translate virtual addresses and for software to calculate real addresses. Because the DMA is virtual, no software overhead is required for pinning system memory that would otherwise be needed if the DMA operated in real address mode.

Execution of an eciwx or ecowx instruction involves the same sequence as a normal cache inhibited load and store with a few exceptions. The processor calculates an effective address, translates it, and presents the resulting real address to the system bus as normal. However, this address bus does not select the slave. The address is passed to the slave to be used on a subsequent transfer. The slave is selected by a 4-bit Resource ID (RID) that is placed on the $\overline{SYS\_TBST}$ and SYS_TSIZ[0:2] signals by the processor.

The device is selected for these transactions when the RID on the bus matches Configuration Register bits 8-11 in the device's System Control Register. The bus transaction is always a single beat regardless of the SYS_TBST signal setting. While the DMA is occurring, the device monitors the bus for a TLB Sync (resulting from normal page maintenance by the OS kernel) to terminate the transfer. Software can then restart the transfer at the faulting address.

The DMA Controller transfers data between system memory and PCI only. It cannot perform memory-to-memory transfers. DMA operation is transparent to the PCI adapter, which behaves as a PIO slave device. Although eciwx and ecowx both initiate DMA, the preferred instruction is ecowx because it writes to the system bus. eciwx is provided to avoid access violation errors on pages marked read-only.

Software ensures proper implementation of the DMA operation, including address alignments and page boundaries. The device aborts a DMA transfer when any of the following conditions are detected:

- TLBSYNC operation detected (internal commands are completed before termination).
- Improper DMA transfer setup.
- Second DMA transfer initiated when one is already in progress.
- The transfer crosses a page boundary.

DMA Transfer Registers

Several registers support the DMA transfer process. They are mapped to two different address spaces so the software can mark the x'FF1C xxxx' range as user space and the 'FF1E xxxx' range as privileged space. This provides protection needed to allow the eciwx and ecowx instructions to be executed by application level software. The registers are listed in the following table and are described in *DMA Registers Space* on page 130.

DMA Transfer Register Summary

Register	User			Privileged			Description
	Address	Bits	Mode	Address	Bits	Mode	
GSCR	FF1C 0020	[0:31]	R	FF1E 0020	[0:31]	R/W	Global Control Register
GSSR	FF1C 0030	[0:31]	R	FF1E 0030	[0:31]	R	Global Status Register
XSCR	FF1C 0040	[0:31]	R/W	FF1E 0040	[0:31]	R/W	DMA Transfer Control Register
XSSR	FF1C 0050	[0:31]	R	FF1E 0050	[0:31]	R	DMA Transfer Status Register
XPAR	FF1C 0070	[0:3]	R	FF1E 0070	[0:3]	R/W	PCI Address Register
		[4:31]	R/W		[4:31]	R	
XWAR	FF1C 0090	[0:31]	R	FF1E 0090	[0:31]	R/W	Writeback Address Register
XTAR	FF1C 00A0	[0:31]	R	FF1E 00A0	[0:31]	R	Translated Address Register

The steps for executing a DMA transfer with software are:

1. Initialize XSCR to indicate length and direction of transfer.
2. Initialize XPAR with the PCI address. The PCI logic takes the address in the XPAR register and applies the translation as described in *CPU to PCI Addressing Model (PREP and FPHB Modes)* on page 144.
3. Initialize XWAR with the address to which the device writes to indicate status following transfer.
4. Clear cache line status in memory at address specified in XWAR.
5. Execute the ecowx instruction (or eciwx if read only page) to start transfer.
6. Wait until an End-of-DMA transfer interrupt (IT) occurs, then read the status on the memory address specified in XWAR. Reset bit 4 of the GSCR register to acknowledge the IT. Alternatively, perform cache polling to the memory address specified in XWAR and wait until the cache status flag changes from x'00' to x'FF'.

DMA Transfer Status Cache Line

The following table shows the definition of the 64-bit of status stored in main memory at the address defined by the XWAR register. Only bits 32-63 of the second double-word of the Write Back Status cache line are valid. All other bytes in the cache line must be ignored.

DMA Transfer Status Cache Line Definition

Bit(s)	Description
Status Double-word 0	
0-63	Reserved
Status Double-word 1	
0-63	Undefined
Status Double-word 2	
0-63	Undefined
Status Double-word 3	
0-31	x'0000 0000'
32 - 39	Poll Status Cache Line Valid Flag x'00' - Initial value set by software. Indicates status cache line is not valid. x'FF' - Written by hardware to indicate that the status cache line has been updated and is valid
40	Transfer Complete 0 - Transfer is not complete 1 - Transfer is complete
41	TLBSYNC Detected 0 - No TLBSYNC Detected 1 - TLBSYNC detected during DMA transfer Transfer
42	Reserved
43	Page Crossing Error 1 - Page Crossing detected during DMA transfer
44	Second DMA Transfer Halt 1 - DMA transfer operation in progress was halted due to start of second DMA transfer operation
45	Unaligned ecowx/eciwx Address 1 - Address associated with ECOWX/ECIWX is not word aligned
46	Unaligned Transfer Error 1 - Address alignment error
47	Address Increment Alignment Error 1 - Improper alignment of addresses when Address Increment bit is off
48	Invalid PCI Address 1- XPAR did not match any PCI extents
49 - 50	Reserved
51 - 63	Transfer Length This field contains the number of bytes remaining when the transfer was completed or aborted

DMA Procedure

The DMA transfer process begins when the 60x logic detects an ecowx or eciwx transaction on the processor bus. If the RID bits in the IBM25CPC710AB3A100 and System Control register match the RID bits on the SYS_TBST and SYS_TSIZ[0:2] lines, the 60x logic accepts the transfer. If the instruction is an ecowx, the 60x logic SYS_TAs the bus for dummy write data and sends a DMA Transfer Write command to the DMA Controller.

The internal address bus associated with the Transfer Write command contains the address from the processor bus. This address is placed in the XTAR register by the DMA Controller. During the processor address tenure, the 60x logic sets an internal flag to indicate special handling of TLBSYNC operations on the processor bus. If the flag is not set, the 60x logic ignores all TLBSYNC operations on the processor bus. If the flag is set, a TLBSYNC operation on the bus causes the 60x logic to place a one cycle pulse on the UX6_TLB_SYNC line to the DMA Controller. The 60x logic continuously SYS_ARTRYs the TLBSYNC bus operation until it receives a one cycle pulse on the internal UXI_XFER_DONE line from the DMA Controller. This pulse also resets the 60x logic's internal flag to perform special handling of the TLBSYNC operations.

Note: Since the PowerPC601 processor does not issue TLBSYNC operations, the 60x logic must treat any SYNCs following a TLBI as a TLBSYNC operation when operating with a PowerPC601 processor.

When the eciwx instruction is used, the 60x logic performs the same steps except that the 60x logic internally sends a DMA Transfer Read command to the DMA Controller and waits for a dummy read data response. The dummy read data is then placed on the processor bus to complete the eciwx transfer on the processor bus. The internal flag for special handling of TLBSYNC is set during the eciwx address bus tenure on the processor bus.

After the DMA Controller receives the DMA Transfer command, it issues a Load Pointer command on the internal command bus to the appropriate PCI bus bridge logic unit. This transfers the address in XPAR to the PCI bus bridge pointer register. The DMA Controller then issues a series of Blit commands, or internal Elementary Commands from the DMA Controller to the PCI logic, to the same PCI bus bridge logic unit that transfers the data. The first Blit command contains the memory address stored in the XTAR register.

The PCI bus bridge logic receives the Blit commands and then executes the transfer. For Blit Reads, the DMA Controller first determines whether the read from memory requires a snoop transaction. If the read is coherent, the controller issues a snoop command to the 60x logic. If the snoop fails, the controller retries the snoop until it passes. Once the snoop passes, a Blit Read command is transmitted to the PCI bus bridge logic. The PCI Bridge logic executes the command and then increments the value in its pointer register by the size of the transfer unless the Address Increment field in the Load Pointer command is set to No Increment. Blit Write commands are handled in same way except the transfer is from I/O to System Memory.

Note: The DMA Controller should wait a minimum of eight cycles before reissuing snoop commands after a snoop fail response.

After the transfer is complete, the controller signals the 60x logic by activating UXI_XFER_DONE for one cycle. The controller then issues a Write with Kill to the address specified in XWAR register to indicate to software that the transfer is complete. The controller issues a Kill Cache to the 60x logic, and upon receiving a clean response, issues a Write command to system memory. The write to memory need only be a single beat write to the bytes reserved for DMA transfer status.

Special Boundary Conditions

Due to queueing in the 60x logic, a pulse could be placed on the TLBSYNC line to the DMA Controller before the controller receives an ecowx or eciwx. In this case, the controller waits until it receives an ecowx or eciwx and then immediately terminates the DMA transfer. When two DMA transfers overlap, the controller ignores the TLBSYNC pulse if a DMA transfer is nearly complete. However, because the 60x logic could have an eciwx or ecowx queued, the controller would have to remember the TLBSYNC pulse to terminate the second DMA transfer properly. To do this, the 60x logic indicates the presence of an eciwx or ecowx instruction in its queue to the controller.

Initialization

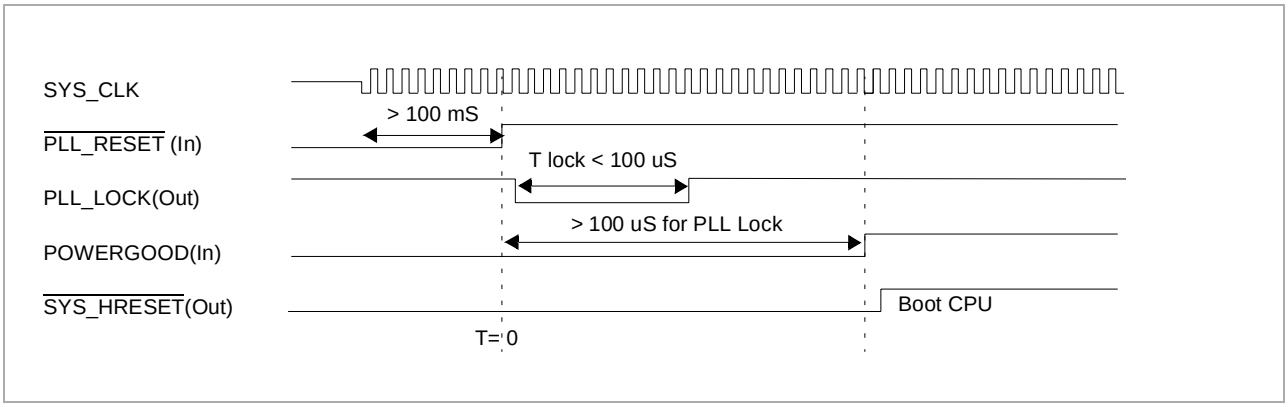
Power Up Sequence

The power up sequence for the CPC710 is:

- At t= - 100 mS
  - All PLL inputs are stable and at their final values:
    PLL_TUNE0=0 , PLL_TUNE1=1 and TESTIN=0
    SYS_CLK is stable at or below the target frequency
    VDDA and Vdd (supply) are at their final values
  - POWERGOOD input is de-asserted Low for Reset.
  - $\overline{\text{PLL\_RESET}}$ input is asserted active (Low).
  - PCI clocks inputs (PCI_CLK and PCG_CLK) are stable at the target frequency
- At t= 0
  - $\overline{\text{PLL\_RESET}}$ input is de-asserted inactive (High).
  - PLL_LOCK output goes down up to the time that clock are locked to the PLL, then is asserted active (High), indicating the PLL is locked.
- At t=100 uS minimum
  - POWERGOOD input is asserted (High).
  - $\overline{\text{HRESET}}$ output de-asserted by the CPC710 (High)
  - Bus transactions may begin.
  - Boot can begin.

Note: Chip reset is only controlled by the SYS_CLK.
TESTIN is a manufacturing test input for the PLL.

PowerUp Sequence



POWERGOOD Power-On Reset

Using the system Power-On Reset POWERGOOD signal, the device resets internally and generates a reset signal to all CPUs and I/O devices. All device I/O pins go to tri-state. After a POWERGOOD cycle, outputs on all interfaces are either floating or driven to their inactive state, except for the reset signals sent to the board as described below.

1. PowerPC bus: $\overline{\text{HRESET}}$0 and $\overline{\text{HRESET}}$1 are driven Low for the same duration as the POWERGOOD active pulse (low level).
2. PCI-64 bus: $\overline{\text{G\_RST}}$ is driven Low from the beginning of the POWERGOOD assertion and remains active after POWERGOOD is deasserted. $\overline{\text{G\_RST}}$ is deactivated when the processor writes a 1 into bit 0 of the, “Component Reset Register (CRR)” Page 70 - BAR + x’000F 7EF0’ for PCI-64. $\overline{\text{G\_RST}}$ is deactivated within a period that complies with the PCI Specification [2] for the 64-bit interface.
3. PCI-32 bus: $\overline{\text{P\_RST}}$ is driven Low from the beginning of the POWERGOOD assertion and remains active after POWERGOOD is deasserted. $\overline{\text{P\_RST}}$ is deactivated when the processor writes a 1 into bit 0 of the, “Component Reset Register (CRR)” Page 70 - BAR + x’000F 7EF0’ for PCI-32 after several PCI clocks

Reset individual devices

The Connectivity Reset Register (RSTR)at the address x‘FF00 0010’ provides a means to individually reset devices on the 60x bus. Bits 0 and 1 directly control $\overline{\text{SYS_HRESET}}$0 and $\overline{\text{SYS_HRESET}}$1 respectively. The remaining two bits control PCI-32 and PCI-64 reset signals that are outputs of the CPC710.

PCI32 bus example:

When bit 2 of register RSTR is asserted low, the PCI32 bus goes to reset.

After the reset, when the bit 2 is deasserted (Returns to high level =1) it takes 250ns before the PCI 32 bus can be used for normal accesses.

Reset in Multiprocessor mode

The sequence of Power-On Reset in Multiprocessor is the same as for a Single CPU on the 60X bus.

Simultaneously the $\overline{\text{HRESET}}$0 and $\overline{\text{HRESET}}$1 signal goes up after the POWERGOOD signal goes up. One of the two CPU get the PowerPC bus through $\overline{\text{SYS\_BR0}}$ or $\overline{\text{SYS\_BR1}}$ and get granted to access the Boot ROM at address FFFF 0100.

It can be decided for example that the CPU 0 is the Master and the CPU 1 the slave, with the CPU 0 in charge of running the code to configure the CPC710 bridge.

The Master/slave configuration is defined with the help of registers PIDR & RSTR

- **PIDR Physical Identifier Register**: When
  BR0-BG0 signal pair is set, bit 31 is set to 0
  BR1-BG1 signal pair is set, bit 31 is set to 1
- **RSTR Connectivity Reset Register**;
  permit to reset CPU0 or CPU1

The first action of the boot code is to permit to the connected CPU to read the PIDR register such that this CPU identifies if he is a Master or a slave.

In the case the CPU 1 (slave) get access first, the boot code can put him in a pooling mode until the Master complete the I/O and Memory initialization. One way is to write in the Register RSTR.

4 Way Multiprocessor

The CPC710 has four (4) PowerPC Bus Requests but internally the arbitration is done on 2 requests. In the case of 4 Way multiprocessing support, it is necessary to add an external logic to handle the additional 2 processors. (See scheme below).

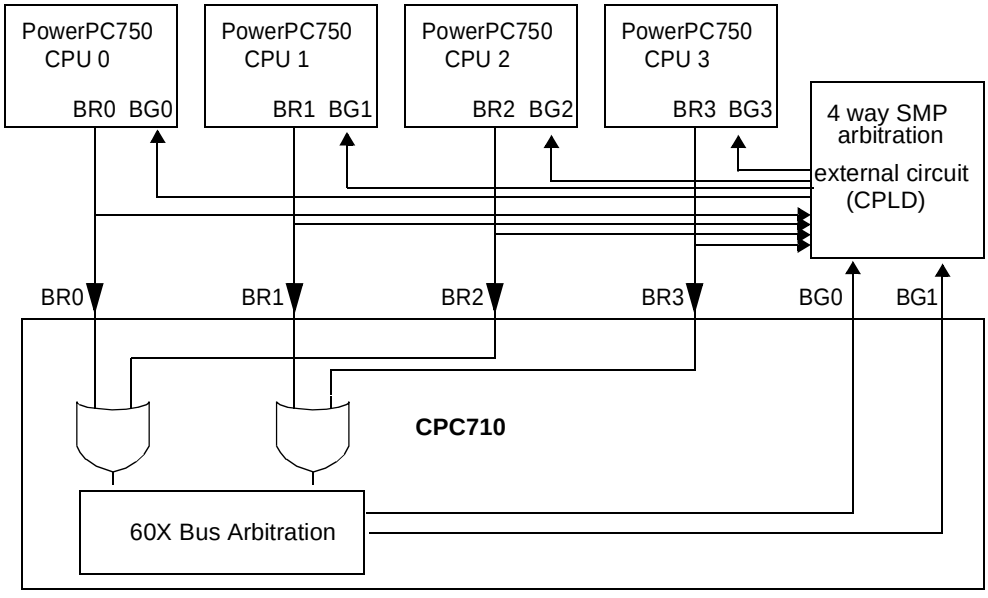


Figure 10 : Abitration to support of 4 Way Multiprocessing with the CPC710

Typical Register setup sequence

Many deviations from the proposed following example of set up are possible. However it is important to keep the basic operations in the same sequence order as described below.

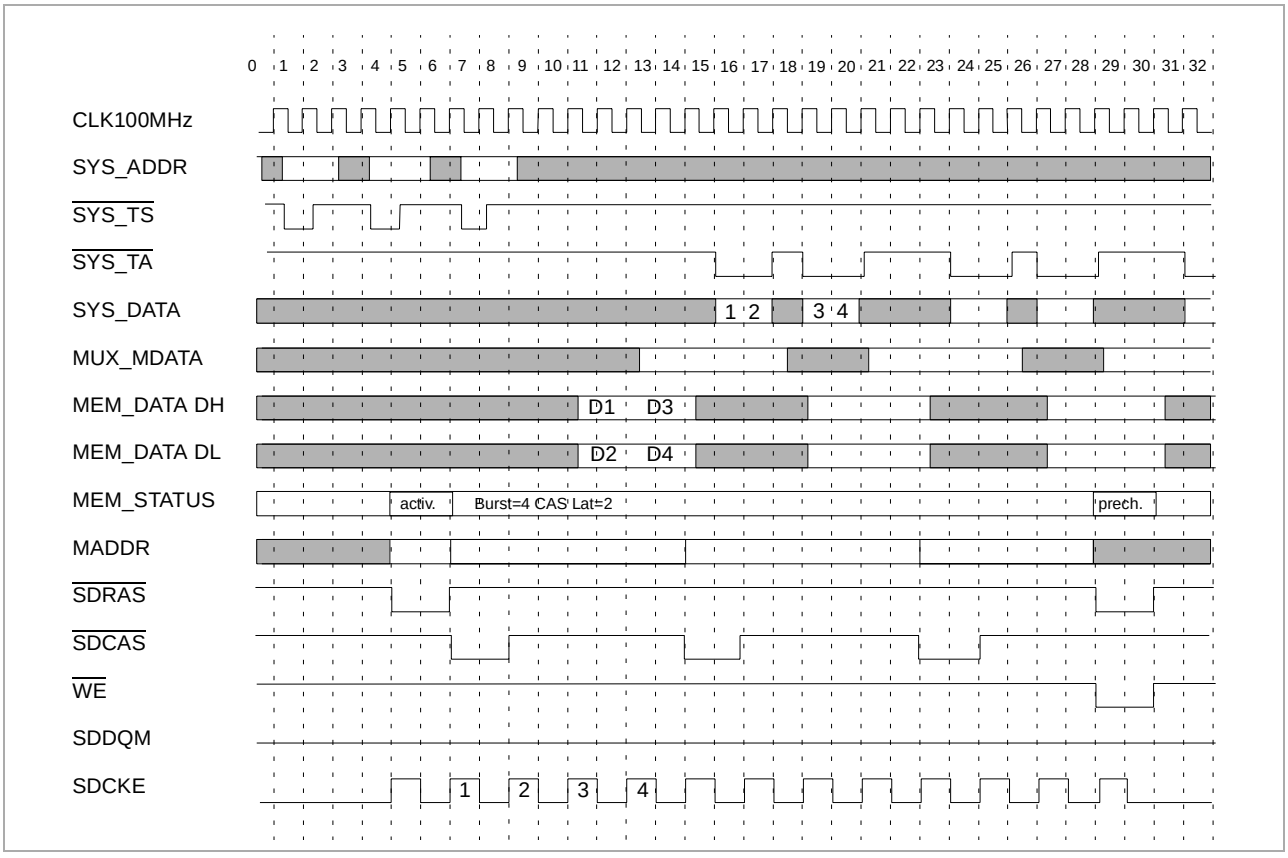
```
/*********************************************************************/
/* Typical CPC710-100 registers setup sequence (from model simulation*/
/*                                                                   */
/*                                  11/17/99          IBM France   */
/*********************************************************************/

/* Begin CPC710-100 registers setup sequence       */
/*================================================*/

/* 60X Interface registers setup                   */
/* ---------------------------------------------- */

  RSTR(0xff000010)  : write 0xf0000000
  UCTL(0xff001000)  : write 0x32f80000
  ABCNTL(0xff001030): write 0xb0000000
  ERRC(0xff001050)  : write 0x00c00000
  SESR(0xff001060)  : write 0x00000000
  SEAR(0xff001070)  : write 0x00000000
  PGCHP(0xff001100) : write 0x00000000

/* Memory Interface registers setup                  */
/* ------------------------------------------------- */

  MESR(0xff001220)  : write 0x00000000
  MEAR(0xff001230)  : write 0x00000000
  MCER0(0xff001300) : write 0x800080c0
  MCER1(0xff001310) : write 0x808080c0
  MCCR(0xff001200)  : write 0x83b06000


/* PCI64 and PCI32 Interfaces Configuration mode setup */
/* ------------------------------------------------- */

  /* Enable configuration mode for PCI64 */
  CNFR(0xff00000c)  : write 0x80000003
  CPU SYNC  /* SYNC OP */
  BAR(0xff200018)   : write 0xff400000
  PCIENB(0xff201000): write 0x80000000

  /* Enable configuration mode for PCI32 */
  CNFR(0xff00000c)  : write 0x80000002
  CPU SYNC  /* SYNC OP */
  BAR(0xff200018)   : write 0xff500000
  PCIENB(0xff201000): write 0x80000000

  /* Disable configuration modes         */
  CNFR(0xff00000c)  : write 0x00000000
  CPU SYNC  /* SYNC OP */


/* PCI64 Interface registers setup                   */
/* ------------------------------------------------- */

  PIBAR(0xff4f7800) : write 0x5c000000
  PMBAR             : write 0x5a000000
```

```
  PR                : write 0x00008000
  ACR               : write 0xff000000
  MSIZE             : write 0xfc000000  /*example with  64 MB*/
  IOSIZE            : write 0xff000000  /*example with  16 MB*/
  SMBAR             : write 0xe0000000
  SIBAR             : write 0x90000000
  CSR               : write 0x00000000
  PLSSR             : write 0x00000000

  /* PCI64 Command register setup */
  CFGA(0xff4f8000)  : write 0x04000080
  CFGD(0xff4f8010)  : write 0x5601


/* PCI32 Interface registers setup                   */
/* ------------------------------------------------- */

  PIBAR(0xff5f7800) : write 0x1c000000
  PMBAR             : write 0x1a000000
  PR                : write 0x0000c000
  ACR               : write 0xfe000000
  MSIZE             : write 0xfe000000  /*example with  32 MB*/
  IOSIZE            : write 0xff800000  /*example with   8 MB*/
  SMBAR             : write 0xc0000000
  SIBAR             : write 0x80000000
  CSR               : write 0x00000000
  PLSSR             : write 0x00000000
  BPMDLK            : write 0xc0400000
  TPMDLK            : write 0xc0800000
  BIODLK            : write 0x80400000
  TIODLK            : write 0x80800000

  /* PCI32 Command register setup */
  CFGA(0xff5f8000)  : write 0x04000080
  CFGD(0xff5f8010)  : write 0x5601

/* ------------------------------------------------- */

/* Wait for SDRAM initialization is complete --> MCCR(2) goes to a 1 */

/* Release external reset to PCI32 bus agents */
  CRR(0xff5f7ef0) : write 0xfc000000

/* Release external reset to PCI64 bus agents */
  CRR(0xff4f7ef0) : write 0xfc000000

/* End of CPC710-100 registers setup sequence */
/*=========================================*/
```

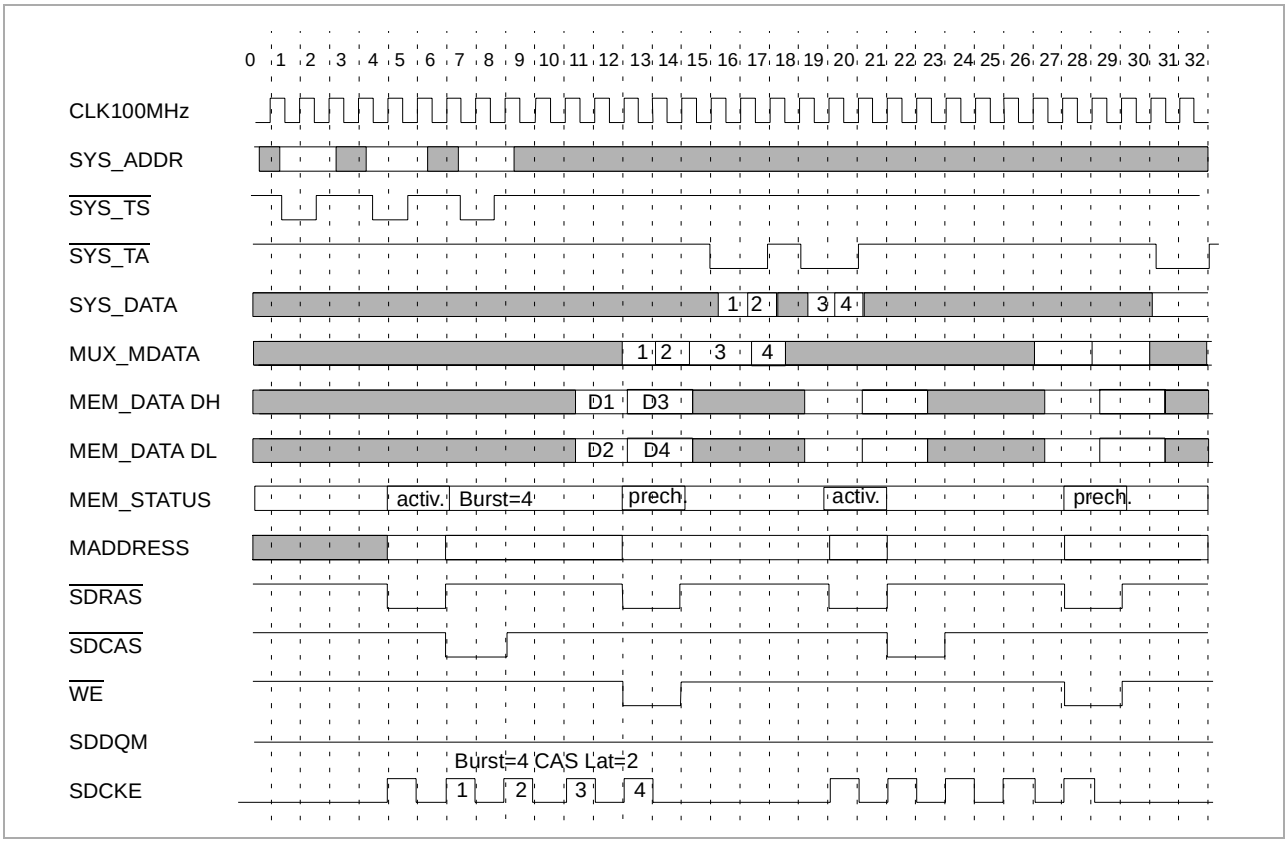
Timing Diagrams

CPU to Memory

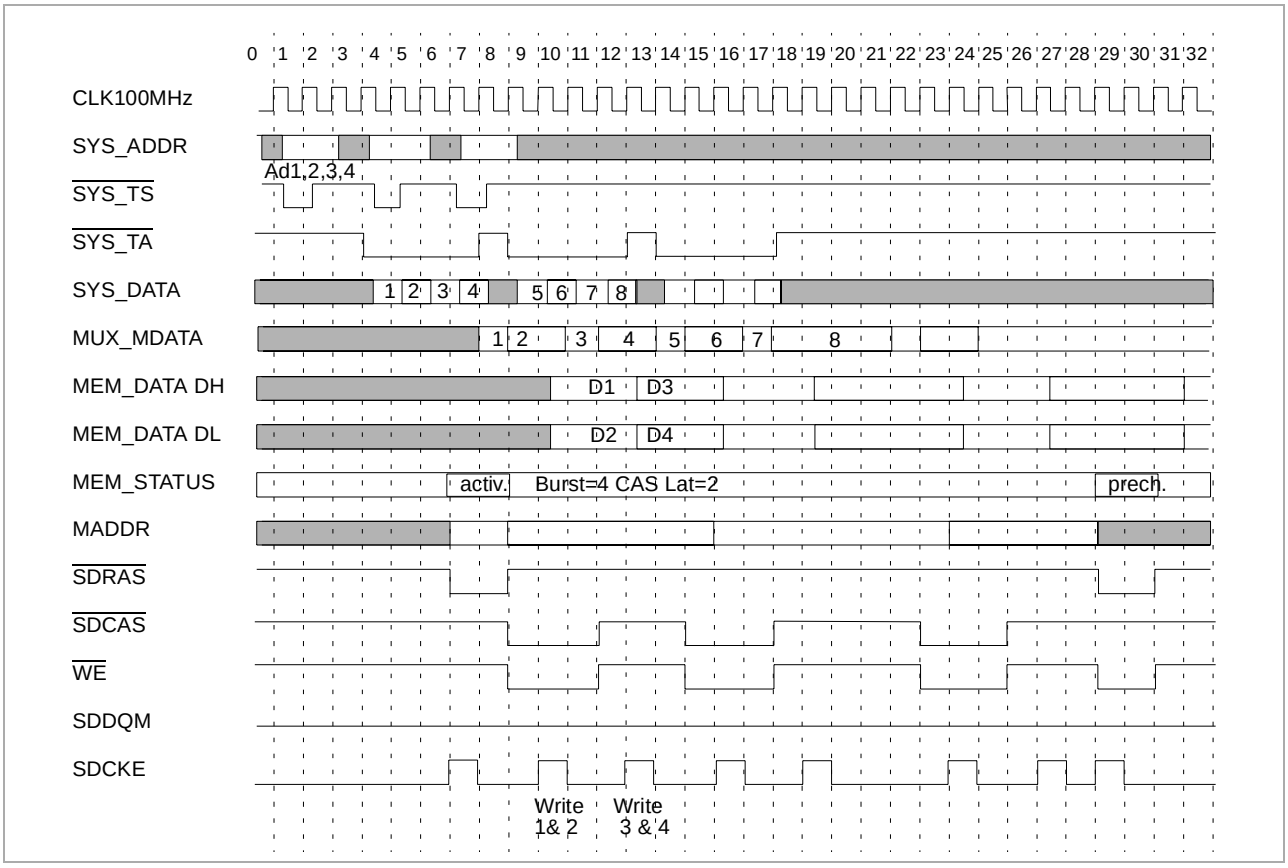
Read Page Hit from PowerPC CPU to SDRAM



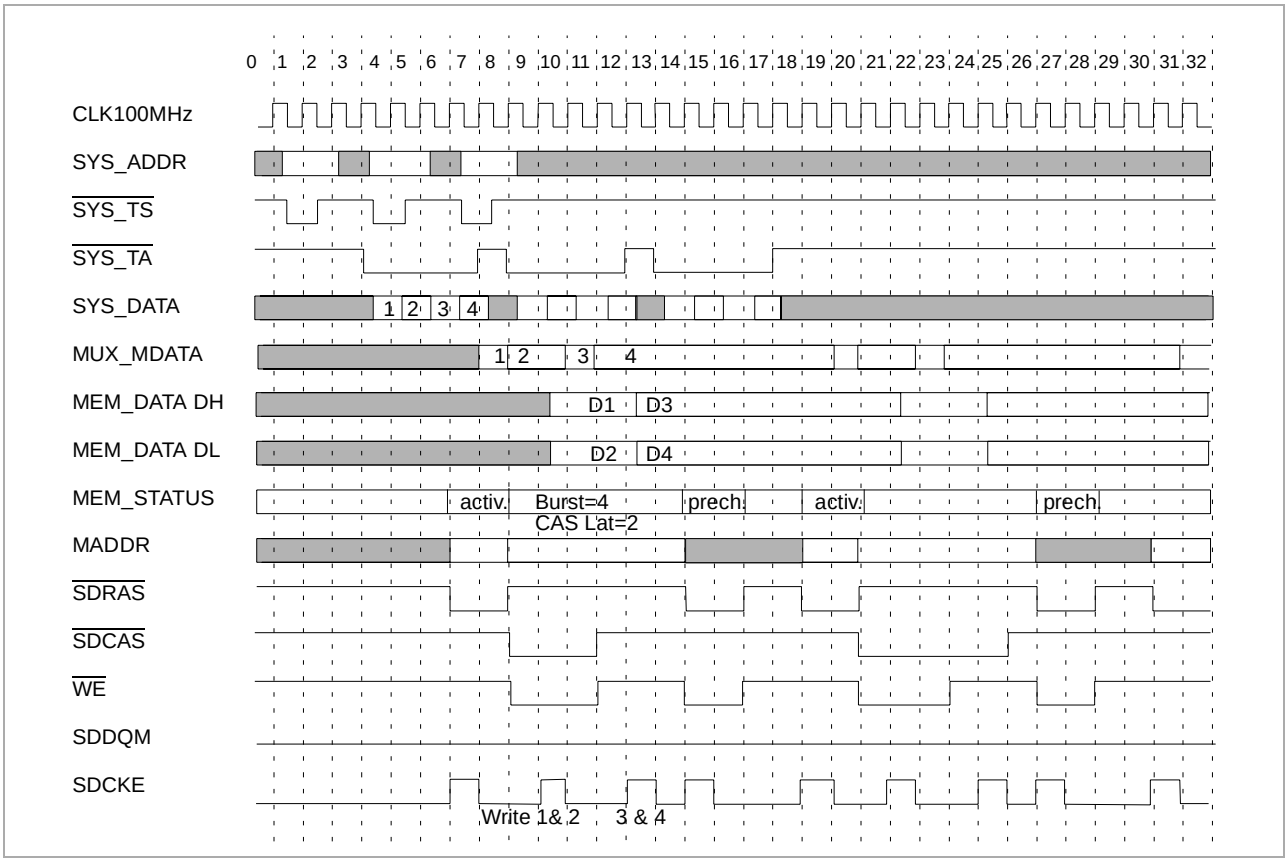
Read Page Miss from PowerPC CPU to SDRAM



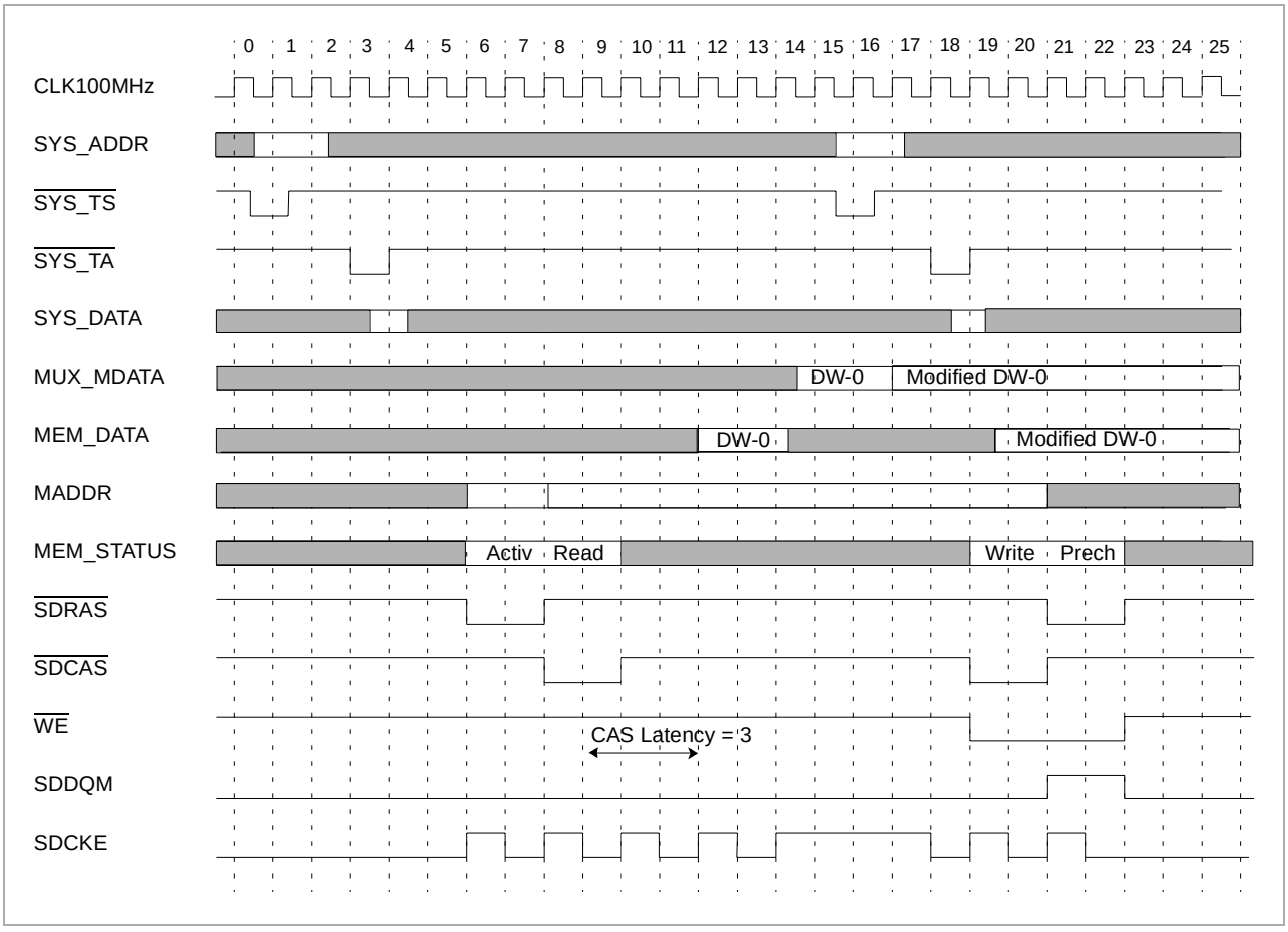
Write Burst Page Hit from PowerPC CPU to SDRAM



Write Burst Page Miss from PowerPC CPU to SDRAM

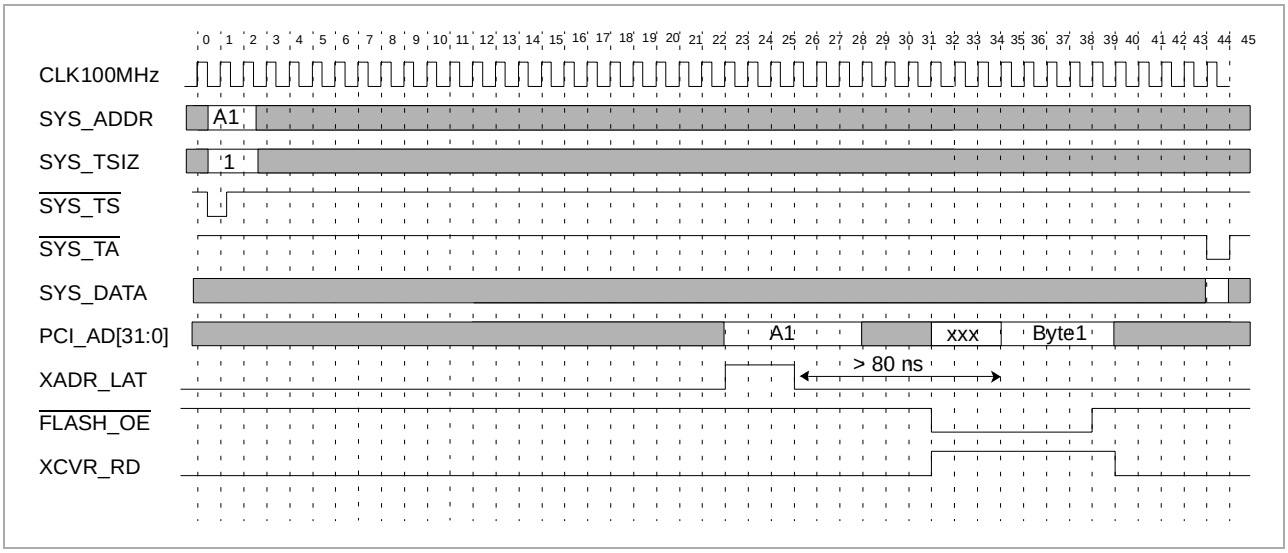


Write One Byte to Memory from CPU: Read Modify Write

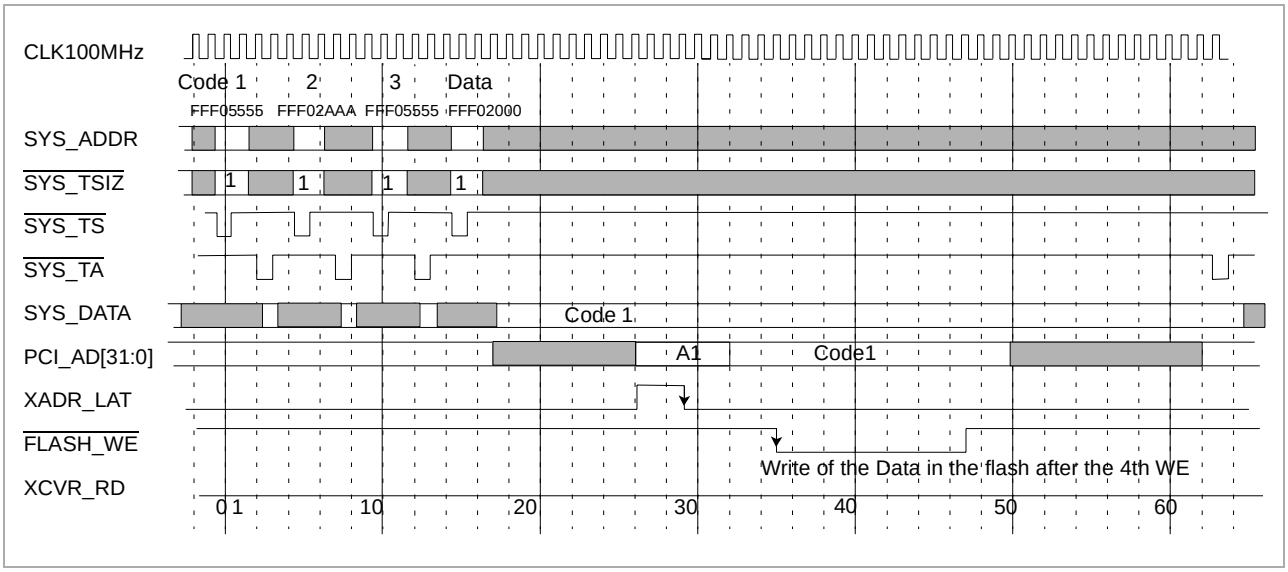


CPU Access to the Boot ROM

Read of One Byte from the Boot ROM

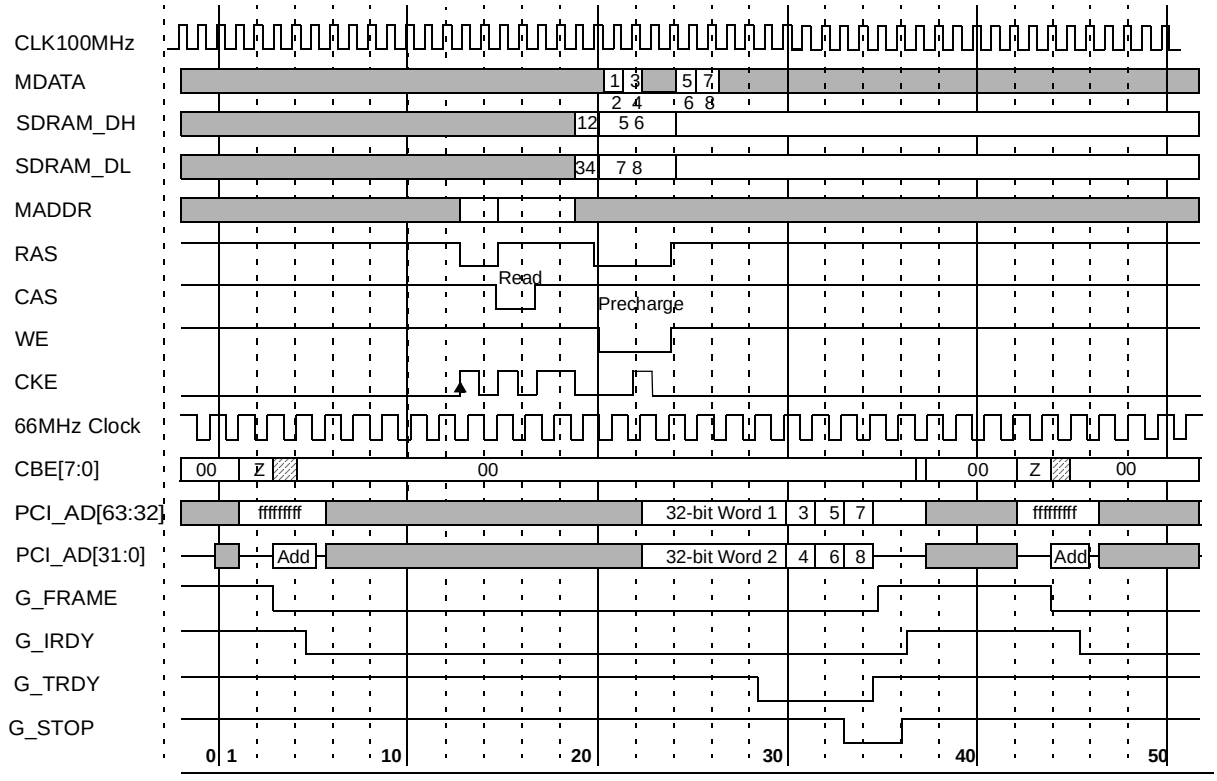


Write of One Byte to the Boot Flash



PCI-64 external Master accessing the SDRAM Memory

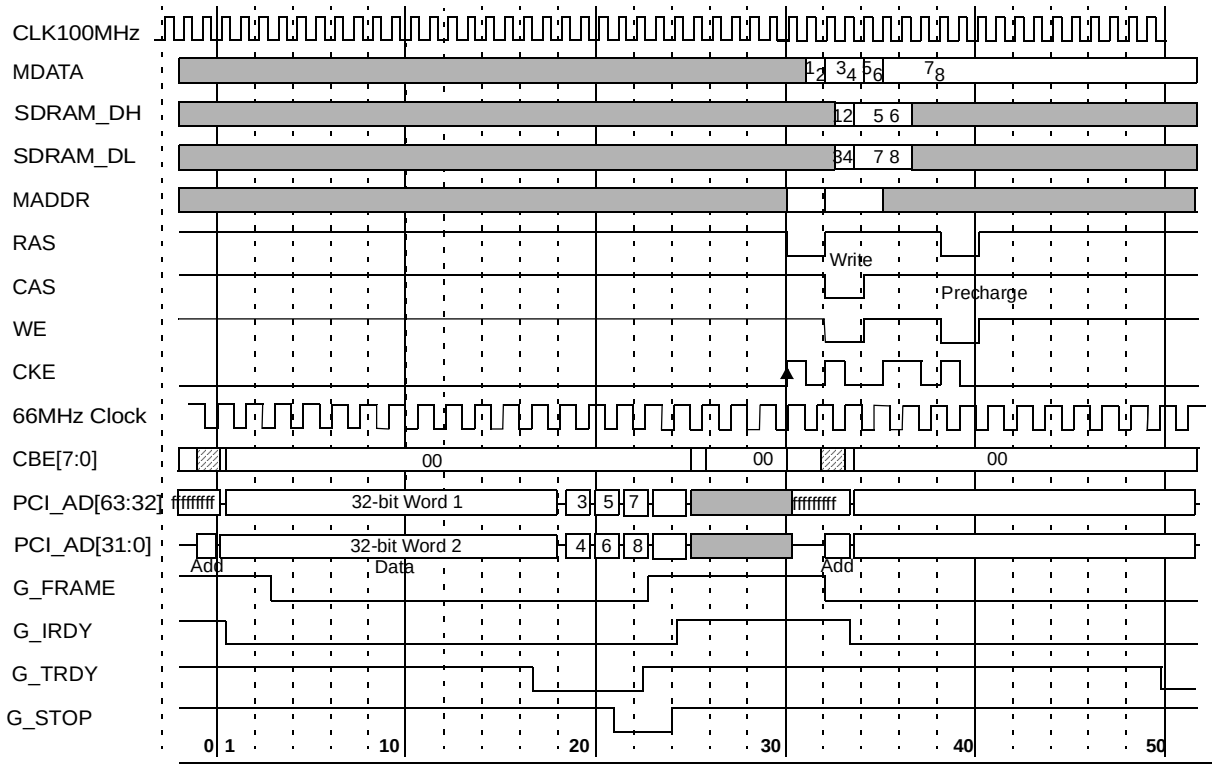
READ 32 Bytes from the SDRAM by a PCI Master on PCI 64- 66MHz bus:



Note: 1,2,3 .. refers to 1st 2nd third 32-bit word.

READ the SDRAM from a PCI Master on PCI 64

Write of 32 Bytes in the SDRAM from a PCI Master on the PCI 64- 66MHz bus



WRITE to the SDRAM from a PCI Master on PCI 64

Electrical Specifications

Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Units
V_{DD}	Supply Voltage	0	3.6	V
V_{IN}	Input Voltage	0	3.6	V
T_{STG}	Storage Temperature Range	-65	150	°C

Recommended DC Operating Conditions

Symbol	Parameter	Min	Typ	Max	Units
V_{DD}	Supply Voltage	3.135	3.3	3.465	V
V_{IH}	Input Logic High (3.3 V receivers)	2.0		V_{DD}	V
V_{IH}	Input Logic High (5.0 V receivers)	2.0		5.0	V
V_{IL}	Input Logic Low	0.0		0.8	V
V_{OH}	Output Logic High	2.4		V_{DD}	V
V_{OL}	Output Logic Low	0.0		0.4	V
I_{IL1}	Input Leakage Current		<1	10	µA
V_{IMAO3}	Input Max Allowable Overshoot (3.3 V receivers)			V_{DD} + 0.6	V
V_{IMAO5}	Input Max Allowable Overshoot (5.0 V receivers)			5.5	V
V_{IMAU3}	Input Max Allowable Undershoot (3.3 V receivers)			-0.6	V
V_{IMAU5}	Input Max Allowable Undershoot (5.0 V receivers)			-0.6	V
V_{OMAO3}	Output Max Allowable Overshoot (3.3 V receivers)			V_{DD} + 0.6	V
V_{OMAO5}	Output Max Allowable Overshoot (5.0 V receivers)			5.5	V
V_{OMAU3}	Output Max Allowable Undershoot (3.3 V receivers)			-0.6	V
T_J	Die Junction Temperature	-20		105	°C

Driver/Receiver Specifications

DC Voltage Specifications

Driver/Receiver	Function	MAUL (V)	MPUL (V)	LPUL (V)	MPDL (V)	LPDL (V)	MADL (V)	Notes
3.3 V LVTTL Driver	TTL	V_{DD} +0.6 V	V_{DD} (3.0 V to 3.6 V)	2.40	0.40	0.00	-0.60	1
5.0 V- tolerant LVTTL Driver	TTL	5.50	V_{DD}	2.40	0.40	0.00	-0.60	1
3.3 V LVTTL Receiver	TTL	V_{DD} +0.6 V	V_{DD}	2.00	0.80	0.00	-0.60	1
5.0 V- tolerant LVTTL Receiver	TTL	5.50	5.50	2.00	0.80	0.00	-0.60	1

1. Definition of Terms:
   - MAUL — Maximum Allowable Up Level. The maximum voltage that may be applied without affecting the specified reliability. Cell functionality is not implied. Maximum Allowable applies to overshoot only.
   - MPUL — Maximum Positive Up Level. The most positive voltage that maintains cell functionality. The maximum positive logic level.
   - LPUL — Least positive Up Level. The least positive voltage that maintains cell functionality. The minimum positive logic level
   - MPDL — Most Positive Down Level. The most positive voltage that maintains cell functionality. The maximum positive logic level.
   - LPDL — Least positive Down Level. The least positive voltage that maintains cell functionality. The minimum positive logic level
   - MADL — Minimum Allowable Down Level. The minimum voltage that may be applied without affecting the specified reliability. Cell functionality is not implied. Minimum Allowable applies to undershoot only.

LVTTL Driver Minimum DC Currents at Rated Voltage (V_{DD} at 3.0 V, temperature at 100 °C)

Driver Type	V_{HI} (V)	I_{HI} (mA)	V_{LO} (V)	I_{LO} (mA)
50 Ohm Driver Outputs	2.40	11.0	0.40	7.0

Thermal Specifications

Parameter		Typ	Max	Units
Power Dissipation		2.1	2.7	Watts
Thermal Resistance (from junction to air)	No Air Flow	18.1	-	°C/Watt
	100 CFM	16.9	-	°C/Watt
	No Air Flow + Cap	12.6	-	°C/Watt
	No Air Flow + 4.5mm Heat Sink	6.0	-	°C/Watt

AC Timing Specifications

V_{CC} = 3.3 V ±5%, Tj = - 40 °C to +105 °C

60x Bus Timing Specification

60x Bus Input Timings

Signal	IBM25CPC710AB3B100				PowerPC 8p-750 Timing	
	Setup min (ns)	Hold min (ns)	Setup min (ns)	Hold min (ns)	Tp max (ns) 50 pf	Tp min(ns) 0 pf
SYS_BR[0:1]	5.2	0.0			5.00	1.00
SYS_TS, SYS_ARTRY, SYS_TBST	5.2	0.0			4.50	1.00
Others	4.6	0.0			5.00	1.00

60x Bus Output Timings (*)

Signal	IBM25CPC710AB3B100								PowerPC 8p-750 Timing	
	Output Valid		Output Hold		Output Valid		Output Hold			
	Max (ns)	Load (pf)	Min (ns)	Load (pf)	Max (ns)	Load (pf)	Min (ns)	Load (pf)	Setup (ns)	Hold (ns)
SYS_ADDR[0:31]	6.9	30 pf	1.5	10 pf					2.5	+ 0.6
SYS_DATA[0:63]	7.5	30 pf	1.7	10 pf					2.5	+ 0.6
DATAP[0:7]	6.5	30pf	1.1	10pf					2.5	+ 0.6
SYS_ARTRY, SYS_SHD, SYS_AACK , SYS_BG[0:1], DBG[0:1], SYS_TA, SYS_TEA	6.9	30 pf	1.2	10 pf					2.5	+ 0.6
CHKSTOP, SYS_GBL, SYS_HRESET[0:1], SYS_MACHK[0:1], SRESET[0:1], SYS_TBE, SYS_TBST, SYS_TSIZ[0:2], SYS_TT[0:4]	7.4	30 pf	1.3	10 pf					2.5	+ 0.6

PCI-32 bit Bus Timing Specification

33 MHz PCI-32 bit Bus Timings

Description	IBM25CPC710AB3B100				PCI 32-bit 33Mhz Spec	
	Min(ns)	Max(ns)	Min	Max	Min	Max
Output Valid - Bused Signals	2.1	6.9			2	6
Output Valid - Point To Point	6	6			2	6
Input Hold	-0.3				0	
Input Setup - Bused Signals	3.1				3	
Input Setup - GNT#	4.7				5	
Input Setup - REQ#	4.7				5	
Clock Skew						1
System Prop						5

PCI-64 bit Bus Timing Specification

66 MHz PCI-64 bit Bus Timings

Description	IBM25CPC710AB3B100				PCI -64bit 66Mhz Spec	
	Min(ns)	Max (ns)	Min	Max	Min	Max
Output Valid - Bused Signals	2.1	6.9			2	6
Output Valid - Point To Point	6	6			2	6
Input Hold	-0.3				0	
Input Setup - Bused Signals	3.1				3	
Input Setup - GNT#	4.7				5	
Input Setup - REQ#	4.7				5	
Clock Skew						1
System Prop						5

SDRAM Interface Timing Specification

SDRAM Input Timings

Signal	IBM25CPC710AB3B100			
	Input Setup min (ns)	Input Hold min (ns)	Input Setup min (ns)	Input Hold min (ns)
MDATA[0:71]	3.0	0.1		

SDRAM Output Timings

Signal	IBM25CPC710AB3B100							
	Output Valid from SYS_CLK		Output Hold from SYS_CLK		Output Valid from SYS_CLK		Output Hold from SYS_CLK	
	Max (ns)	Load (pf)	Min (ns)	Load (pf)	Max (ns)	Load (pf)	Min (ns)	Load (pf)
SYS_ADDR[0:31]	6.9	30 pf	6.9	30 pf				
SYS_DATA[0:63]	7.5	30 pf	7.5	30 pf				
DATAP[0:7]	6.5	30pf	6.5	30pf				
$\overline{\text{SYS\_ARTRY}}$, $\overline{\text{SYS\_SHD}}$, $\overline{\text{SYS\_AACK}}$, $\overline{\text{SYS\_BG}}$[0:1], $\overline{\text{DBG}}$[0:1], $\overline{\text{SYS\_TA}}$, $\overline{\text{SYS\_TEA}}$	6.9	30 pf	6.9	30 pf				
$\overline{\text{CHKSTOP}}$, $\overline{\text{SYS\_GBL}}$, $\overline{\text{SYS\_HRESET}}$[0:1], $\overline{\text{SYS\_MACHK}}$[0:1], $\overline{\text{SYS\_SRESET}}$[0:1], $\overline{\text{SYS\_TBE}}$, $\overline{\text{SYS\_TBST}}$, $\overline{\text{SYS\_TSIZ}}$[0:2], $\overline{\text{SYS\_TT}}$[0:4]	7.4	30 pf	7.4	30 pf				

FLASH Interface Timing Specification

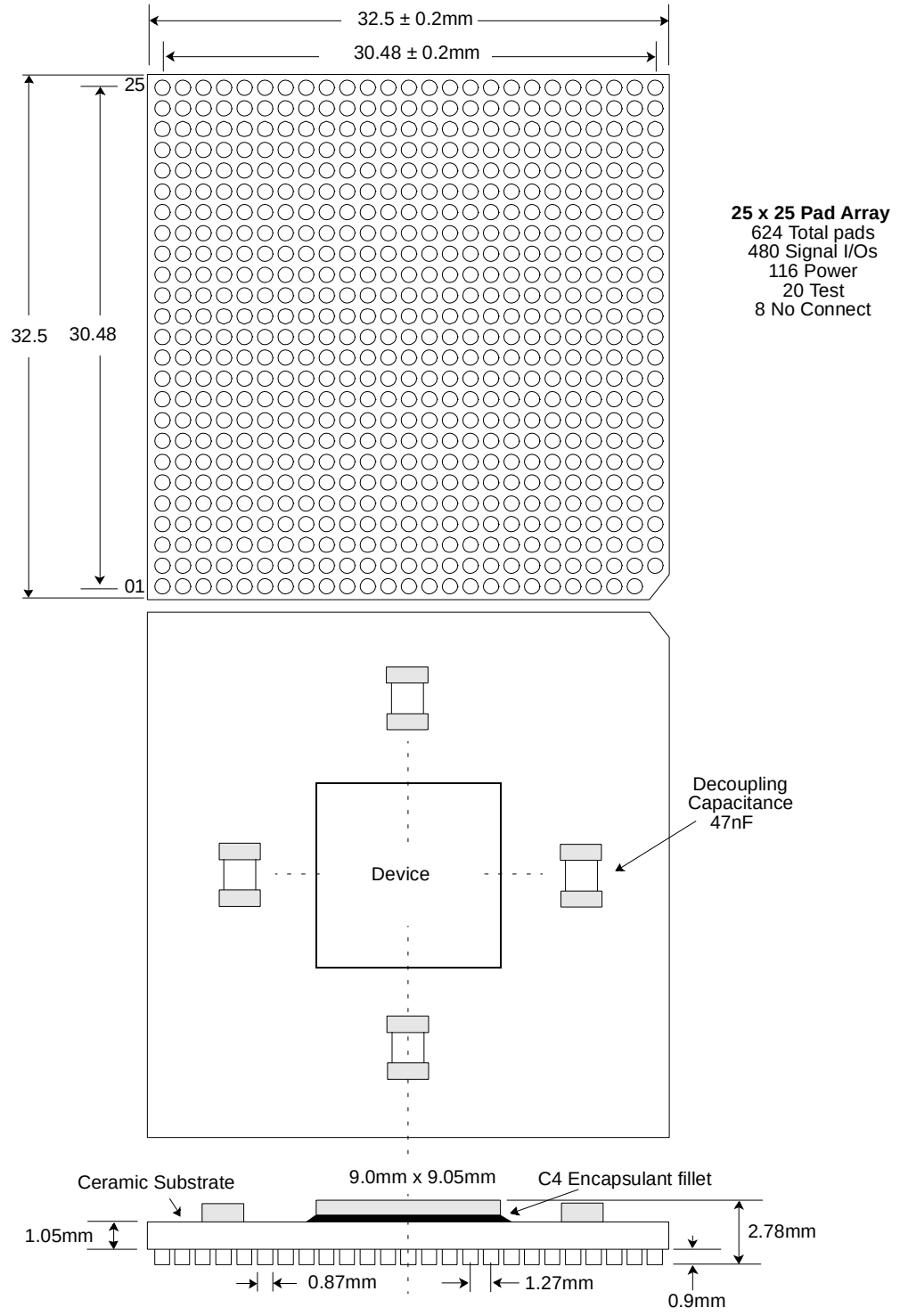
FLASH Output Timings

Signal	IBM25CPC710AB3B100							
	Output Valid from SYS_CLK		Output Hold from SYS_CLK		Output Valid from SYS_CLK		Output Hold from SYS_CLK	
	Max (ns)	Load (pf)	Min (ns)	Load (pf)	Max (ns)	Load (pf)	Min (ns)	Load (pf)
SYS_ADDR[0:31]	6.9	30 pf	6.9	30 pf				
SYS_DATA[0:63]	7.5	30 pf	7.5	30 pf				
DATAP[0:7]	6.5	30pf	6.5	30pf				
$\overline{\text{SYS\_ARTRY}}$, $\overline{\text{SYS\_SHD}}$, $\overline{\text{SYS\_AACK}}$, $\overline{\text{SYS\_BG}}$[0:1], $\overline{\text{DBG}}$[0:1], $\overline{\text{SYS\_TA}}$, $\overline{\text{SYS\_TEA}}$	6.9	30 pf	6.9	30 pf				
$\overline{\text{CHKSTOP}}$, $\overline{\text{SYS\_GBL}}$, $\overline{\text{SYS\_HRESET}}$[0:1], $\overline{\text{SYS\_MACHK}}$[0:1], $\overline{\text{SRESET}}$[0:1], SYS_TBE, $\overline{\text{SYS\_TBST}}$, SYS_TSIZ[0:2], $\overline{\text{SYS\_TT}}$[0:4]	7.4	30 pf	7.4	30 pf				

For FLASH Data Input/Output Timings, refer to signals P_ADL[0:31] into PCI-32 bit bus Timing Specification section.

Packaging Information

Package Dimensions 32.5mm BGA - Standard JEDEC



All measurements are in millimeters unless otherwise noted. Drawing is not to scale.

References

1. PC SDRAM Specification, Revision 1.63. October 1998, Intel
2. PCI Local Bus Specification, Revision 2.1. June 1st 1995, PCI Special Interest Group

Revision Log

Revision Date	Contents of Modification
09/08/99	Initial release (00)
12/17/99	release (1.0) - PCI 64 REQ setting to disable arbiter - XATS removed - P_ISA_MASTER must be tied to GND to be inactive - Initialization: PowerUp at time 0 - SOI1 & SOI2 space access: unpredictable results (See UCTL Register) - ATAS init programing value for PowerPC750 Memory/Cache coherency - Firecoral replaced by PCI-ISA bridge - Boot ROM data on PCI AD[8:15] - Access to SDRAM from PCI or CPU performed in rotating priority - MADDR0 LSB split in MADDR0_ODD & MADDR1_EVEN - G_CBE[0:7] & P_CBE[0:3] active low - CMND Reg reset values reversed - MCER Reg extended code bank for 4 & 8 MB - PCI to PCI transfers: unpredictable results: PPSIZE & BARPP & PPBAR Regs removed - POWERGOOD Active high (Reset when low) - SIOR0 & SIOR1 registers controls PRES_OE0 & PRES_OE1 signals - External Mux control signals renamed - Temperature range: -20 to 105 C - Read and write Waveforms for the PCI 64 -66MHz access to the memory - Typical Register Setup sequence - P_LOCK & G_LOCK are Input only - LOCK on PCI busses not fully supported - MCER0 to 7 address corrected
6/26/00	Release 1.1 - PCI are 3.3V I/OS with 5V compliance - PCI-64 INTA-B-C-D Signals are Outputs - 3 I/Os : GPIO1-2-3 - Inputs G_LOCK & P_LOCK are reserved - JTAG TDO & TMS - Semaphore Register PSEM #60 is reserved - Endian decription enhanced - DMA operation enhanced - SIOA1 & SIOA2 space now reserved