Alpha 21064A Microprocessors

Data Sheet

Order Number: EC–QFGKC–TE

This document contains information about the following Alpha microprocessors: 21064A–200, 21064A–233, 21064A–275, 21064A–275–PC, and 21064A–300.

Revision/Update Information: This document supersedes the *Alpha 21064A–233, –275 Microprocessor Data Sheet*, EC–QFGKB–TE.

Digital Equipment Corporation
Maynard, Massachusetts

January 1996

This document was prepared using VAX DOCUMENT Version 2.1.

Contents

Figures

Tables

1 Overview

This document describes the Alpha 21064A microprocessors (21064A). The five versions of the 21064A differ in clock frequency as indicated by their labels (200, 233, 275, 275-PC, and 300).

The 21064A-200, 21064A-233, 21064A-275, and the 21064A-300 are functionally identical. Their memory management operation is very flexible to allow them to enable multiple memory management functions for different operating system environments.

The 21064A-275-PC is functionally identical to the other four except that it differs in its memory management functions. The 21064A-275-PC will only support the memory management functions necessary for the Windows NT operating system and other operating systems using the Windows NT memory management model.

The label 21064A will be used to describe the functions and operations that are identical for the five devices. The label **21064A-275-PC** will be used to identify information that is unique to that one device.

The 21064A has the features listed here:

- 64-bit RISC microprocessor implements the Alpha architecture
- Super-scalar, 200, 233, 275, and 300 MHz
- System clock frequency is the processor clock frequency divided by any value from 2 to 17
- Dual instruction issue that yields a peak instruction execution rate of 400, 466, 550, or 600 MIPS
- Super pipelined
- Two on-chip caches (with data and tag parity protection)
  - 16-Kbyte instruction cache (Icache)
  - 16-Kbyte data cache (Dcache)
- 32 Integer registers and 32 floating-point registers (64-bit)
- 4K x 2-bit branch prediction history table
- External data path selectable for 128 bits or 64 bits
- Byte parity mode available for the external data bus
- Fast lock mode available for use with LDx/L and STx/C instructions
- Backward compatible with the 21064 pin layout and software

- Programmable external cache
  - Set size
  - Set speed
- 43-bit virtual address
- 34-bit physical address
- IEEE and VAX floating-point data types
- Performance monitoring
- 64-bit internal data paths

The 21064A and associated PALcode implement IEEE single- and double-precision, VAX F_floating and G_floating data types, and support longword (32-bit) and quadword (64-bit) integers. Byte (8-bit) and word (16-bit) support is provided by byte manipulation instructions. Limited hardware support is provided for the VAX D_floating data type.

The 21064A consists of four independent functional units:

- Integer execution unit (Ebox)
- Floating-point unit (Fbox)
- Load/store or address unit (Abox)
- Branch unit

Other sections include the central control unit (Ibox), and the Icache and Dcache.

Ebox—Contains a 64-bit fully pipelined integer execution data path including: adder, logic box, barrel shifter, byte extract and mask, and independent integer multiplier. The Ebox also contains a 32-entry, 64-bit integer register file.

Fbox—Contains a fully pipelined floating-point unit and independent divider that supports both IEEE and VAX floating-point data types. IEEE single-precision and double-precision floating-point data types are supported. VAX F_floating and G_floating data types are fully supported, and limited support is provided for the D_floating data type.

Abox—Contains five major sections: address translation data path, load silo, write buffer, data cache interface, and the external bus interface unit (BIU). The Abox supports all integer and floating-point load and store instructions including address calculation and translation, and cache control logic.

Ibox—Performs instruction fetch, resource checks, and dual-instruction issue to the Ebox, Abox, Fbox, or branch unit. In addition, the Ibox controls pipeline stalls, aborts, and restarts.

Pipeline Organization

The 21064A uses a 7-stage pipeline for integer operate and memory reference instructions, and a 10-stage pipleline for floating-point operate instructions. The Ibox maintains state for all pipeline stages to track outstanding register writes.

Cache Organization

The 21064A contains two on-chip caches—data cache (Dcache) and instruction cache (Icache). The 21064A also supports an external cache.

Icache—Contains 16K bytes and is a direct-mapped cache with 32-byte blocks. Virtual address bit 13 and physical address bits [12:5] are the index into the cache.

Dcache—Contains 16K bytes and is a write through, read-allocate cache with 32-byte blocks. It can be used in either of two modes.

- 8K-byte direct-mapped cache (to support 21064 designs). Physical address [12:5] is the index into the cache.
- 16K-byte cache
  - The Dcache appears externally as a 2-way set-associative cache. Physical address [12:5] is the index into the cache.
  - The Dcache appears internally as direct-mapped cache. Virtual address [13] and physical address [12:5] are the index into the cache.

External Cache—The 21064A supports an external cache that is made with readily available static RAMs. The 21064A directly controls RAM operation using its programmable external cache interface, allowing each hardware implementation to make its own external cache speed and configuration trade-offs.

The external cache interface supports cache sizes 0, 256K bytes, 512K bytes, 1M bytes, 2M bytes, 4M bytes, 8M bytes and 16M bytes. The range of operating speeds of the external cache are sub-multiples of the 21064A clock.

Virtual Address Space

The architecture virtual address is a 64-bit unsigned integer that specifies a byte location within the virtual address space. The 21064A implements a 43-bit subset of the virtual address space.

Physical Address Space

The 21064A uses a 34-bit physical address to support 16G bytes of physical address space.

Backward Compatibility

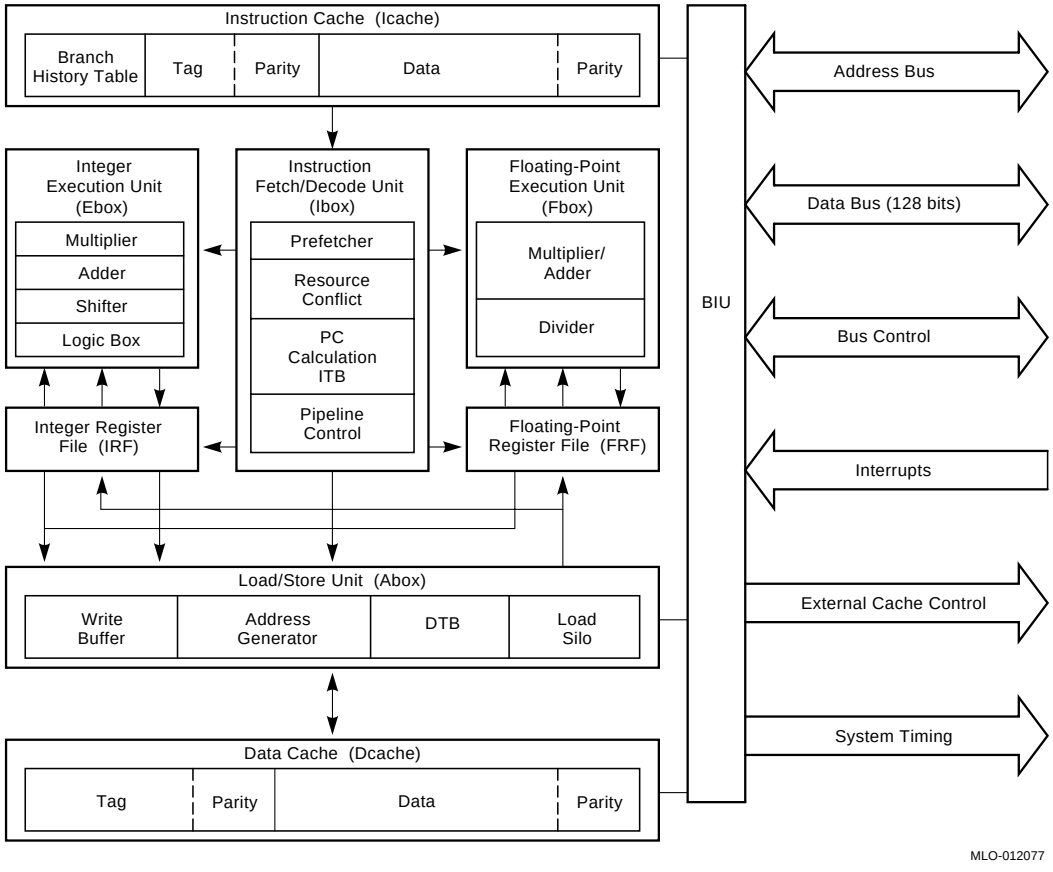
The 21064A is backward compatible with the 21064. The compatibility includes pin layout, PALcode, and application programs.

The following restrictions apply to the compatibility between the 21064A and 21064.

- The 21064A has internal pulldown resistors on inputs which are unused spare pins on the 21064. If these spare pins are unconnected on a module designed for the 21064, then there will be no migration problem with these pins.
- Two pins have been reallocated for other uses. If these pins were not used on a module designed for the 21064, then there will be no migration problem with these pins. On the 21064 the two pins are **tagEq_l** and **tagAdr_h 17**; on the 21064A they are **lockWE_h** and **lockFlag_h** respectively.
- The behavior of the **tagOK** protocol on the 21064A differs from that of the 21064. Designers should investigate the effect of the change if this protocol is used in existing 21064 modules.

Figure 1 shows a block diagram of the 21064A microprocessor.

Figure 1 Block Diagram of the Alpha 21064A Microprocessor



2 Signal Names and Functions

The tables in this section list the various signals grouped by function. The information in the **Type** column identifies a signal as input (I), output (O), or bidirectional (B).

Signals with an **_h** suffix are active (asserted) when high. Those with an **_l** suffix are active (asserted) when low.

Table 1 describes the 21064A data, address, and parity/ECC bus signals.

Table 1 Data, Address, and Parity/ECC Bus Signals

Signal	Type	Count	Function
data_h [127:0]	B	128	Provide the data path between the 21064A and the system.
adr_h [33:5]	B	29	Provide the address path between the 21064A and the system. These address bits provide granularity down to 32-byte internal cache blocks.
check_h [27:0]	B	28	Provide a path for parity or ECC bits between the 21064A and the rest of the system.

Table 2 describes the 21064A primary cache invalidate signals.

Table 2 Primary Cache Invalidate Signals

Signal	Type	Count	Function
iAdr_h [12:5]	I	8	Used to index blocks in the Dcache for Dcache invalidates.
dInvReq_h [1:0][1]	I	2	Used by external logic to invalidate the Dcache entry indexed by **iAdr_h [12:5]**.

[1] **dInvReq_h 1** at PGA location C24 was a spare pin on the 21064. It has an internal pulldown that draws a maximum current of 200 μA at 2.4 V dc.

Table 3 describes the 21064A external cache control signals.

Table 3 External Cache Control Signals

Signal	Type	Count	Function
tagCEOE_h	O	1	Controls tag and tag control RAM chip enable or output enable during the 21064A controlled external cache accesses.
tagCtlWE_h	O	1	Controls tag control RAM write enable during the 21064A controlled transactions.
tagCtlV_h, tagCtlS_h, tagCtlD_h	B	3	Provide read/write path for external cache valid, shared, and dirty bits. The following combinations of the tagCtl RAM bits are allowed. The **tagCtlS_h** can be viewed as a write-protect bit.

tagCtlV_h	tagCtlS_h	tagCtlD_h	Meaning
L	X	X	Invalid
H	L	L	Valid, private
H	L	H	Valid, private, dirty
H	H	L	Valid, shared
H	H	H	Valid, shared, dirty

tagCtlP_h	B	1	Carries parity across **tagCtlV_h**, **tagCtlD_h**, and **tagCtlS_h**.

(continued on next page)

Table 3 (Cont.) External Cache Control Signals

Signal	Type	Count	Function
tagAdr_h [33:18]	I	16	Carries the address contents of the tagAdr RAM to the 21064A address comparator and parity checker.
tagAdrP_h	I	1	Carries the parity contents of the tagAdr RAM to the 21064A address comparator and parity checker.
tagOk_h, tagOk_l	I	2	Bus interface control signals that allow external logic to stall a CPU-controlled access to the external cache RAMs at the last possible moment.
dataCEOE_h [3:0]	O	4	Controls data RAMs output enable or chip enable during the 21064A controlled cache accesses.
dataWE_h [3:0]	O	4	Controls data RAMs write enable during the 21064A controlled cache accesses.
dataA_h [4:3]	O	2	Controls data RAMs **adr_h [4:3]** during the 21064A controlled cache accesses.
holdReq_h	I	1	Asserted by external logic to gain access to the external cache.
holdAck_h	O	1	Asserted by the 21064A to indicate that external logic has access to the external cache.
dMapWE_h [1:0][1]	O	2	Controls the write enable inputs of the (optional) data cache backmap RAM during the 21064A controlled external cache reads.

[1] **dMapWE_h 1** at PGA location M24 was a spare pin on the 21064.

Table 4 describes the signals which allow the 21064A to perform LDxL and STxC transactions to and from an external cache.

Table 4 Fast Lock Mode Signals

Signal	Type	Count	Function
lockWE_h[1]	O	1	The 21064A is able to probe Bcache for an LDxL transaction. If there is a Bcache hit, then the 21064A will assert **lockWE_h** allowing external logic to set a lock flag bit and load a lock address register.
lockFlag_h[2]	I	1	This signal line allows external logic to indicate the state of the lock flag bit (set or clear). When the 21064A performs a STxC transaction, it may probe the Bcache and test this signal. If the signal is asserted, then the 21064A will perform the write to Bcache while asserting **lockWE_h**. This transaction allows the external logic to clear the lock flag bit.

[1] **lockWE_h** at PGA location P24 was used for the signal **tagEq_l** by the 21064.

[2] **lockFlag_h** at PGA location R23 was used for the signal **tagAdr_h 17** by the 21064.

Table 5 describes the 21064A external cycle control signals.

Table 5 External Cycle Control Signals

Signal	Type	Count	Function
dOE_l	I	1	Used by external logic to tell the 21064A to drive the data bus during external write transactions.
dWSel_h [1:0]	I	2	Used by external logic to tell the 21064A which part of the 32-byte block of write data should be driven onto the data bus.
dRAck_h [2:0]	I	3	Informs the 21064A that read data is valid on the data bus, and indicates whether data should be cached and whether ECC or parity checking should be attempted. Read data acknowledge types are:

dRAck_h 2	dRAck_h 1	dRAck_h 0	Type
L	L	L	IDLE
H	L	L	OK_ NCACHE_ NCHK
H	L	H	OK_ NCACHE
H	H	L	OK_ NCHK
H	H	H	OK

(continued on next page)

Table 5 (Cont.) External Cycle Control Signals

Signal	Type	Count	Function
cReq_h [2:0]	O	3	Used by the 21064A to specify a cycle type at the start of an external cycle. The cycle types are:

cReq_h 2	cReq_h 1	cReq_h 0	Type
L	L	L	IDLE
L	L	H	BARRIER
L	H	L	FETCH
L	H	H	FETCH_M
H	L	L	READ_ BLOCK
H	L	H	WRITE_ BLOCK
H	H	L	LDL_L/ LDQ_L
H	H	H	STL_C/ STQ_C

cWMask_h [7:0]	O	8	Supplies longword write masks to external logic during write cycles; contains miss address bits and other miss information during other cycles.
cAck_h [2:0]	I	3	Used by external logic to acknowledge an external cycle. Acknowledge types are:

cAck_h 2	cAck_h 1	cAck_h 0	Type
L	L	L	IDLE
L	L	H	HARD_ ERROR
L	H	L	SOFT_ ERROR
L	H	H	STL_C_FAIL /STQ_C_ FAIL
H	L	L	OK

Table 6 describes the 21064A interrupt signals.

Table 6 Interrupt Signals

Signal	Type	Count	Function
irq_h [5:0]	I	6	These signal lines have two uses: • During normal operation they provide six types of external interrupts to the 21064A. • At reset, they provide initialization information to the 21064A.
sysClkDiv_h[1]	I	1	At reset, this line provides initialization information to the 21064A. When **reset_l** is asserted, **irq_h [4:3]** encodes the delay in CPU clock cycles, from **sysClkOut1** to **sysClkOut2**, as follows:

irq_h 4	irq_h 3	Delay
L	L	0
L	H	1
H	L	2
H	H	3

[1] **sysClkDiv_h** at PGA location AA16 was a spare pin on the 21064. It has an internal pulldown that draws a maximum current of 200 μA at 2.4 V dc.

(continued on next page)

Table 6 (Cont.) Interrupt Signals

Signal	Type	Count	Function
			When **reset_l** is asserted, **sysClkDiv_h** and **irq_h [2:0]** encode the value of the divisor used to generate the system clock from the CPU clock, as follows:

sysClkDiv_h[1]	irq_h [2:0]	Ratio
L	L L L	2
L	L L H	3
L	L H L	4
L	L H H	5
L	H L L	6
L	H L H	7
L	H H L	8
L	H H H	9
H	L L L	10
H	L L H	11
H	L H L	12
H	L H H	13
H	H L L	14
H	H L H	15
H	H H L	16
H	H H H	17

When **reset_l** is asserted, **irq_h 5** is used to select 128-bit or 64-bit mode. If **irq_h 5** is asserted, then 128-bit mode is selected.

[1] **sysClkDiv_h** at PGA location AA16 was a spare pin on the 21064. It has an internal pulldown that draws a maximum current of 200 μA at 2.4 V dc.

Table 7 describes the 21064A instruction cache initialization and serial ROM interface signals.

Table 7 Instruction Cache Initialization and Serial ROM Interface Signals

Signal	Type	Count	Function
icMode_h [2:0][1]	I	3	Determines which Icache initialization mode is used after reset. The 21064A implements several Icache modes used by Digital to support chip and module level testing.

icMode_h [2:0]			Mode
L	L	L	Serial ROM
L	L	H	Disabled
All other combinations			Digital reserved

sRomOE_l	O	1	In serial ROM mode, supplies the output enable to the external serial ROM, serving both as an output enable and as a reset.
sRomD_h	I	1	In serial ROM mode, inputs external serial ROM data to the 21064A.
sRomClk_h	O	1	In serial ROM mode, supplies the clock to the external serial ROM that causes it to advance to the next bit. The signals **sRomOE_l**, **sRomD_h**, and **sRomClk_h** also serve as simple parallel I/O pins to drive a diagnostic terminal. When the serial ROM is not being read, output signal **sRomOE_l** is false. This means that **sRomOE_l** can be wired to the active high enable of an RS422 receiver driving onto **sRomD_h** and to the active high enable of an RS422 driver driving from **sRomClk_h**. The CPU allows **sRomD_h** to be read and **sRomClk_h** to be written by PALcode; this is sufficient hardware support to implement a software-driven serial interface.

[1] **icMode_h 2** at PGA location AD7 was a spare pin on the 21064. It has an internal pulldown that draws a maximum current of 200 μA at 2.4 V dc.

Table 8 describes the initialization signal pins **dcOk_h**, **reset_l** and **reset_SClk_h**.

Table 8 Initialization Signals

Signal	Type	Count	Function
dcOk_h	I	1	Switches clock sources between an on-chip ring oscillator and the external clock oscillator.
reset_l	I	1	Forces the CPU into a known state.
reset_SClk_h[1]	I	1	A test signal. It forces the system clock divider into a known state.

[1] **reset_SClk_h** at PGA location AA11 was a spare pin on the 21064. It has an internal pulldown that draws a maximum current of 200 μA at 2.4 V dc.

Table 9 describes the 21064A clock signals.

Table 9 Clock Signals

Signal	Type	Count	Function
clkIn_h, clkIn_l	I	2	Supplies the 21064A with a differential clock from external logic.
testClkIn_h, testClkIn_l	I	2	Test signals; should be tied to Vdd and Vss, respectively.
cpuClkOut_h	O	1	Supplies the internal chip clock for use by the external interface. The low-to-high transition of **cpuClkOut_h** is the "CPU clock" used in the timing specification for the **tagOk_h** and **tagOk_l** signals.
sysClkOut1_h, sysClkOut1_l	O	2	Provides the system clock for use by the external interface. The low-to-high transition of **sysClkOut1_h** provides the system clock that is used as a timing reference throughout this document.
sysClkOut2_h, sysClkOut2_l	O	2	Provide delayed system clock to the external interface. The delay is between zero and three CPU clock cycles.

Table 10 describes the performance monitoring signals.

Table 10 Performance Monitoring Signals

Signal	Type	Count	Function
perf_cnt_h [1:0]	I	2	Provides the 21064A internal performance monitoring hardware with access to off-chip events.

Table 11 describes some other signals common to the 21064A.

Table 11 Other Signals

Signal	Type	Count	Function
tristate_l	I	1	The assertion of this signal forces all the 21064A signals, with the exception of **cpuClkOut_h**, to the high-impedance state.
cont_l	I	1	The assertion of this signal causes the 21064A to connect all signals to Vss, with the exception of certain clock signals and **vRef**.
vRef	I	1	Supplies a reference voltage of 1.4 V to the input signal sense circuits.
eclOut_h	I	1	Digital reserved; should be tied to Vss.

3 Instruction Set

This section provides information about instructions for the 21064A.

3.1 Instruction Summary

This section contains a summary of all Alpha architecture instructions. All values are in hexadecimal radix. Table 12 describes the contents of the Format and Opcode columns that are in Table 13.

Table 12 Instruction Format and Opcode Notation

Instruction Format	Format Symbol	Opcode Notation	Meaning
Branch	Bra	oo	oo is the 6-bit opcode field.
Floating-point	F-P	oo.fff	oo is the 6-bit opcode field. fff is the 11-bit function code field.
Memory	Mem	oo	oo is the 6-bit opcode field.
Memory/function code	Mfc	oo.ffff	oo is the 6-bit opcode field. ffff is the 16-bit function code in the displacement field.
Memory/branch	Mbr	oo.h	oo is the 6-bit opcode field. h is the high-order two bits of the displacement field.
Operate	Opr	oo.ff	oo is the 6-bit opcode field. ff is the 7-bit function code field.
PALcode	Pcd	oo	oo is the 6-bit opcode field; the particular PALcode instruction is specified in the 26-bit function code field.

Table 13 shows architecture instructions. Table 14 shows qualifiers for IEEE floating-point instructions and Table 15 shows qualifiers for VAX floating-point instructions.

Table 13 Architecture Instructions

Mnemonic	Format	Opcode	Description
ADDF	F-P	15.080	Add F_floating
ADDG	F-P	15.0A0	Add G_floating
ADDL	Opr	10.00	Add longword
ADDL/V	Opr	10.40	Add longword
ADDQ	Opr	10.20	Add quadword
ADDQ/V	Opr	10.60	Add quadword
ADDS	F-P	16.080	Add S_floating
ADDT	F-P	16.0A0	Add T_floating
AND	Opr	11.00	Logical product
BEQ	Bra	39	Branch if = zero
BGE	Bra	3E	Branch if ≥ zero
BGT	Bra	3F	Branch if > zero
BIC	Opr	11.0	Bit clear
BIS	Opr	11.20	Logical sum
BLBC	Bra	38	Branch if low bit clear
BLBS	Bra	3C	Branch if low bit set
BLE	Bra	3B	Branch if ≤ zero
BLT	Bra	3A	Branch if < zero
BNE	Bra	3D	Branch if ≠ zero
BR	Bra	30	Unconditional branch
BSR	Mbr	34	Branch to subroutine
CALL_PAL	Pcd	00	Trap to PALcode
CMOVEQ	Opr	11.24	CMOVE if = zero
CMOVGE	Opr	11.46	CMOVE if ≥ zero
CMOVGT	Opr	11.66	CMOVE if > zero
CMOVLBC	Opr	11.16	CMOVE if low bit clear
CMOVLBS	Opr	11.14	CMOVE if low bit set
CMOVLE	Opr	11.64	CMOVE if ≤ zero
CMOVLT	Opr	11.44	CMOVE if < zero
CMOVNE	Opr	11.26	CMOVE if ≠ zero
CMPBGE	Opr	10.0F	Compare byte
CMPEQ	Opr	10.2D	Compare signed quadword equal
CMPGEQ	F-P	15.0A5	Compare G_floating equal

(continued on next page)

Table 13 (Cont.) Architecture Instructions

Mnemonic	Format	Opcode	Description
CMPGLE	F-P	15.0A7	Compare G_floating less than or equal
CMPGLT	F-P	15.0A6	Compare G_floating less than
CMPLE	Opr	10.6D	Compare signed quadword less than or equal
CMPLT	Opr	10.4D	Compare signed quadword less than
CMPTEQ	F-P	16.0A5	Compare T_floating equal
CMPTLE	F-P	16.0A7	Compare T_floating less than or equal
CMPTLT	F-P	16.0A6	Compare T_floating less than
CMPTUN	F-P	16.0A4	Compare T_floating unordered
CMPULE	Opr	10.3D	Compare unsigned quadword less than or equal
CMPULT	Opr	10.1D	Compare unsigned quadword less than
CPYS	F-P	17.020	Copy sign
CPYSE	F-P	17.022	Copy sign and exponent
CPYSN	F-P	17.021	Copy sign negate
CVTDG	F-P	15.09E	Convert D_floating to G_floating
CVTGD	F-P	15.0AD	Convert G_floating to D_floating
CVTGF	F-P	15.0AC	Convert G_floating to F_floating
CVTGQ	F-P	15.0AF	Convert G_floating to quadword
CVTLQ	F-P	17.010	Convert longword to quadword
CVTQF	F-P	15.0BC	Convert quadword to F_floating
CVTQG	F-P	15.0BE	Convert quadword to G_floating
CVTQL	F-P	17.030	Convert quadword to longword
CVTQL/SV	F-P	17.530	Convert quadword to longword
CVTQL/V	F-P	17.130	Convert quadword to longword
CVTQS	F-P	16.0BC	Convert quadword to S_floating
CVTQT	F-P	16.0BE	Convert quadword to T_floating
CVTST	F-P	16.2AC	Convert S_floating to T_floating
CVTTQ	F-P	16.0AF	Convert T_floating to quadword
CVTTS	F-P	16.0AC	Convert T_floating to S_floating
DIVF	F-P	15.083	Divide F_floating
DIVG	F-P	15.0A3	Divide G_floating
DIVS	F-P	16.083	Divide S_floating
DIVT	F-P	16.0A3	Divide T_floating

(continued on next page)

Table 13 (Cont.) Architecture Instructions

Mnemonic	Format	Opcode	Description
EQV	Opr	11.48	Logical equivalence
EXCB	Mfc	18.0400	Exception barrier
EXTBL	Opr	12.06	Extract byte low
EXTLH	Opr	12.6A	Extract longword high
EXTLL	Opr	12.26	Extract longword low
EXTQH	Opr	12.7A	Extract quadword high
EXTQL	Opr	12.36	Extract quadword low
EXTWH	Opr	12.5A	Extract word high
EXTWL	Opr	12.16	Extract word low
FBEQ	Bra	31	Floating branch if = zero
FBGE	Bra	36	Floating branch if $\geq$ zero
FBGT	Bra	37	Floating branch if > zero
FBLE	Bra	33	Floating branch if $\leq$ zero
FBLT	Bra	32	Floating branch if < zero
FBNE	Bra	35	Floating branch if $\neq$ zero
FCMOVEQ	F-P	17.02A	FCMOVE if = zero
FCMOVGE	F-P	17.02D	FCMOVE if $\geq$ zero
FCMOVGT	F-P	17.02F	FCMOVE if > zero
FCMOVLE	F-P	17.02E	FCMOVE if $\leq$ zero
FCMOVLT	F-P	17.02C	FCMOVE if < zero
FCMOVNE	F-P	17.02B	FCMOVE if $\neq$ zero
FETCH	Mfc	18.8000	Prefetch data
FETCH_M	Mfc	18.A000	Prefetch data, modify intent
INSBL	Opr	12.0B	Insert byte low
INSLH	Opr	12.67	Insert longword high
INSLL	Opr	12.2B	Insert longword low
INSQH	Opr	12.77	Insert quadword high
INSQL	Opr	12.3B	Insert quadword low
INSWH	Opr	12.57	Insert word high
INSWL	Opr	12.1B	Insert word low
JMP	Mbr	1A.0	Jump
JSR	Mbr	1A.1	Jump to subroutine
JSR_COROUTINE	Mbr	1A.3	Jump to subroutine return
LDA	Mem	08	Load address
LDAH	Mem	09	Load address high
LDF	Mem	20	Load F_floating
LDG	Mem	21	Load G_floating

(continued on next page)

Table 13 (Cont.) Architecture Instructions

Mnemonic	Format	Opcode	Description
LDL	Mem	28	Load sign-extended longword
LDL_L	Mem	2A	Load sign-extended longword locked
LDQ	Mem	29	Load quadword
LDQ_L	Mem	2B	Load quadword locked
LDQ_U	Mem	0B	Load unaligned quadword
LDS	Mem	22	Load S_floating
LDT	Mem	23	Load T_floating
MB	Mfc	18.4000	Memory barrier
MF_FPCR	F-P	17.025	Move from FPCR
MSKBL	Opr	12.02	Mask byte low
MSKLH	Opr	12.62	Mask longword high
MSKLL	Opr	12.22	Mask longword low
MSKQH	Opr	12.72	Mask quadword high
MSKQL	Opr	12.32	Mask quadword low
MSKWH	Opr	12.52	Mask word high
MSKWL	Opr	12.12	Mask word low
MT_FPCR	F-P	17.024	Move to FPCR
MULF	F-P	15.082	Multiply F_floating
MULG	F-P	15.0A2	Multiply G_floating
MULL	Opr	13.00	Multiply longword
MULL/V	Opr	13.40	Multiply longword
MULQ	Opr	13.20	Multiply quadword
MULQ/V	Opr	13.60	Multiply quadword
MULS	F-P	16.082	Multiply S_floating
MULT	F-P	16.0A2	Multiply T_floating
ORNOT	Opr	11.28	Logical sum with complement
RC	Mfc	18.E000	Read and clear
RET	Mbr	1A.2	Return from subroutine
RPCC	Mfc	18.C000	Read process cycle counter
RS	Mfc	18.F000	Read and set
S4ADDL	Opr	10.02	Scaled add longword by 4
S4ADDQ	Opr	10.22	Scaled add quadword by 4
S4SUBL	Opr	10.0B	Scaled subtract longword by 4
S4SUBQ	Opr	10.2B	Scaled subtract quadword by 4
S8ADDL	Opr	10.12	Scaled add longword by 8

(continued on next page)

Table 13 (Cont.) Architecture Instructions

Mnemonic	Format	Opcode	Description
S8ADDQ	Opr	10.32	Scaled add quadword by 8
S8SUBL	Opr	10.1B	Scaled subtract longword by 8
S8SUBQ	Opr	10.3B	Scaled subtract quadword by 8
SLL	Opr	12.39	Shift left logical
SRA	Opr	12.3C	Shift right arithmetic
SRL	Opr	12.34	Shift right logical
STF	Mem	24	Store F_floating
STG	Mem	25	Store G_floating
STS	Mem	26	Store S_floating
STL	Mem	2C	Store longword
STL_C	Mem	2E	Store longword conditional
STQ	Mem	2D	Store quadword
STQ_C	Mem	2F	Store quadword conditional
STQ_U	Mem	0F	Store unaligned quadword
STT	Mem	27	Store T_floating
SUBF	F-P	15.081	Subtract F_floating
SUBG	F-P	15.0A1	Subtract G_floating
SUBL	Opr	10.09	Subtract longword
SUBL/V	Opr	10.49	Subtract longword
SUBQ	Opr	10.29	Subtract quadword
SUBQ/V	Opr	10.69	Subtract quadword
SUBS	F-P	16.081	Subtract S_floating
SUBT	F-P	16.0A1	Subtract T_floating
TRAPB	Mfc	18.0000	Trap barrier
UMULH	Opr	13.30	Unsigned multiply quadword high
WMB	Mfc	18.44	Write memory barrier
XOR	Opr	11.40	Logical difference
ZAP	Opr	12.30	Zero bytes
ZAPNOT	Opr	12.31	Zero bytes not

3.2 IEEE Floating-Point Instructions

Table 14 lists the hexadecimal value of the 11-bit function code field for the IEEE floating-point instructions, with and without qualifiers. The opcode for these instructions is 16_{16}.

Table 14 IEEE Floating-Point Instruction Function Codes

Mnemonic	None	/C	/M	/D	/U	/UC	/UM	/UD
ADDS	080	000	040	0C0	180	100	140	1C0
ADDT	0A0	020	060	0E0	1A0	120	160	1E0
CMPTEQ	0A5	–	–	–	–	–	–	–
CMPTLT	0A6	–	–	–	–	–	–	–
CMPTLE	0A7	–	–	–	–	–	–	–
CMPTUN	0A4	–	–	–	–	–	–	–
CVTQS	0BC	03C	07C	0FC	–	–	–	–
CVTQT	0BE	03E	07E	0FE	–	–	–	–
CVTTS	0AC	02C	06C	0EC	1AC	12C	16C	1EC
DIVS	083	003	043	0C3	183	103	143	1C3
DIVT	0A3	023	063	0E3	1A3	123	163	1E3
MULS	082	002	042	0C2	182	102	142	1C2
MULT	0A2	022	062	0E2	1A2	122	162	1E2
SUBS	081	001	041	0C1	181	101	141	1C1
SUBT	0A1	021	061	0E1	1A1	121	161	1E1

(continued on next page)

Table 14 (Cont.) IEEE Floating-Point Instruction Function Codes

Mnemonic	/SU	/SUC	/SUM	/SUD	/SUI	/SUIC	/SUIM	/SUID
ADDS	580	500	540	5C0	780	700	740	7C0
ADDT	5A0	520	560	5E0	7A0	720	760	7E0
CMPTEQ	5A5	–	–	–	–	–	–	–
CMPTLT	5A6	–	–	–	–	–	–	–
CMPTLE	5A7	–	–	–	–	–	–	–
CMPTUN	5A4	–	–	–	–	–	–	–
CVTQS	–	–	–	–	7BC	73C	77C	7FC
CVTQT	–	–	–	–	7BE	73E	77E	7FE
CVTTS	5AC	52C	56C	5EC	7AC	72C	76C	7EC
DIVS	583	503	543	5C3	783	703	743	7C3
DIVT	5A3	523	563	5E3	7A3	723	763	7E3
MULS	582	502	542	5C2	782	702	742	7C2
MULT	5A2	522	562	5E2	7A2	722	762	7E2
SUBS	581	501	541	5C1	781	701	741	7C1
SUBT	5A1	521	561	5E1	7A1	721	761	7E1

Mnemonic	None	/S
CVTST	2AC	6AC

Mnemonic	None	/C	/V	/VC	/SV	/SVC	/SVI	/SVIC
CVTTQ	0AF	02F	1AF	12F	5AF	52F	7AF	72F

Mnemonic	D	/VD	/SVD	/SVID	/M	/VM	/SVM	/SVIM
CVTTQ	0EF	1EF	5EF	7EF	06F	16F	56F	76F

3.3 21064A IEEE Floating-Point Conformance

The 21064A supports the IEEE floating-point operations as defined by the Alpha architecture. Support for a complete implementation of the IEEE *Standard for Binary Floating-Point Arithmetic* (ANSI/IEEE Standard 754-1985) is provided by a combination of hardware and software as described in the *Alpha Architecture Reference Manual*.

Additional information about writing code to support precise exception handling (necessary for complete conformance to the standard) is in the *Alpha Architecture Reference Manual*.

The following information is specific to the 21064A:

- Invalid operation (INV)

  The invalid operation trap is always enabled. If the trap occurs, then the destination register is UNPREDICTABLE. This exception is signaled if any VAX architecture operand is non-finite (reserved operand or dirty zero) and the operation can take an exception. (Certain instructions, such as CPYS, never take an exception.) This exception is signaled if any IEEE operand is non-finite (NAN, INF, denorm) and the operation can take an exception. This trap is also signaled for an IEEE format divide of +/– 0 divided by +/– 0. If the exception occurs, then FPCR[INV] is set and the trap is signaled to the Ibox.

- Divide by zero (DZE)

  The divide-by-zero trap is always enabled. If the trap occurs, then the destination register is UNPREDICTABLE. For VAX architecture format, this exception is signaled whenever the numerator is valid and the denominator is zero. For IEEE format, this exception is signaled whenever the numerator is valid and non-zero, with a denominator of +/– 0. If the exception occurs, then FPCR[DZE] is set and the trap is signaled to the Ibox.

  For IEEE format divides, 0/0 signals INV, not DZE.

- Floating overflow (OVF)

  The floating overflow trap is always enabled. If the trap occurs, then the destination register is UNPREDICTABLE. The exception is signaled if the rounded result exceeds in magnitude the largest finite number that can be represented by the destination format. This applies only to operations whose destination is a floating-point data type. If the exception occurs, then FPCR[OVF] is set and the trap is signaled to the Ibox.

- Underflow (UNF)

  The underflow trap can be disabled. If underflow occurs, then the destination register is forced to a true zero, consisting of a full 64 bits of zero. This is done even if the proper IEEE result would have been –0. The exception is signaled if the rounded result is smaller in magnitude than the smallest finite number that can be represented by the destination format. If the exception occurs, then FPCR[UNF] is set. If the trap is enabled, then the trap is signaled to the Ibox.

- Inexact (INE)

  The inexact trap can be disabled. The destination register always contains the properly rounded result, whether the trap is enabled. The exception is signaled if the rounded result is different from what would have been produced if infinite precision (infinitely wide data) were available. For floating-point results, this requires both an infinite precision exponent and fraction. For integer results, this requires an infinite precision integer. If the exception occurs, then FPCR[INE] is set. If the trap is enabled, then the trap is signaled to the Ibox.

  The IEEE-754 specification allows INE to occur concurrently with either OVF or UNF. Whenever OVF is signaled (if the inexact trap is enabled), INE is also signaled. Whenever UNF is signaled (if the inexact trap is enabled), INE is also signaled. The inexact trap also occurs concurrently with integer overflow. All valid opcodes that enable INE also enable both overflow and underflow.

  If a CVTQL results in an integer overflow (IOV), then FPCR[INE] is automatically set. (The INE trap is never signaled to the Ibox because there is no CVTQL opcode that enables the inexact trap.)

  DIVx/I behavior is slightly different. If the DIVx/I instruction does not take an input exception (that is, no INV or DZE), then the Fbox calculates and stores the correct rounded result. The Fbox calculates the inexact flag, setting FPCR [INE] if appropriate, and trapping on DIVx/SI instructions only when the result is really inexact.

- Integer overflow (IOV)

  The integer overflow trap can be disabled. The destination register always contains the low-order bits ([64] or [32]) of the true result (not the truncated bits). Integer overflow can occur with CVTTQ, CVTGQ or CVTQL. In conversions from floating to quadword or longword integer, an integer overflow occurs if the rounded result is outside the range -2^{63} ..$2^{63} - 1$. In conversions from quadword integer to longword integer, an integer overflow occurs if the result is outside the range -2^{31} ..$2^{31} - 1$. If the exception occurs, then the appropriate bit in the FPCR is set. If the trap is enabled, then the trap is signaled to the Ibox.

- Software completion (SWC)

  The software completion signal is not recorded in the FPCR. The state of this signal is always sent to the Ibox. If the Ibox detects the assertion of any of the listed exceptions concurrent with the assertion of the SWC signal, then it sets EXC_SUM[SWC].

Input exceptions always take priority over output exceptions. If both exception types occur, then only the input exception is recorded in the FPCR, and only the input exception is signaled to the Ibox.

Programming Note

Because underflow cannot occur for CMPT*xx*, there is no difference in function or performance between CMPT*xx*/S and CMPT*xx*/SU. It is intended that software generate CMPT*xx*/SU in place of CMPT*xx*/S.

3.4 VAX Floating-Point Instructions

Table 15 lists the hexadecimal value of the 11-bit function code field for the VAX floating-point instructions, with and without qualifiers. The opcode for these instructions is 15_{16}.

Table 15 VAX Floating-Point Instruction Function Codes

Mnemonic	None	/C	/U	/UC	/S	/SC	/SU	/SUC
ADDF	080	000	180	100	480	400	580	500
CVTDG	09E	01E	19E	11E	49E	41E	59E	51E
ADDG	0A0	020	1A0	120	4A0	420	5A0	520
CMPGEQ	0A5	–	–	–	4A5	–	–	–
CMPGLT	0A6	–	–	–	4A6	–	–	–
CMPGLE	0A7	–	–	–	4A7	–	–	–
CVTGF	0AC	02C	1AC	12C	4AC	42C	5AC	52C
CVTGD	0AD	02D	1AD	12D	4AD	42D	5AD	52D
CVTQF	0BC	03C	–	–	–	–	–	–
CVTQG	0BE	03E	–	–	–	–	–	–
DIVF	083	003	183	103	483	403	583	503
DIVG	0A3	023	1A3	123	4A3	423	5A3	523
MULF	082	002	182	102	482	402	582	502
MULG	0A2	022	1A2	122	4A2	422	5A2	522
SUBF	081	001	181	101	481	401	581	501
SUBG	0A1	021	1A1	121	4A1	421	5A1	521

Mnemonic	None	/C	/V	/VC	/S	/SC	/SV	/SVC
CVTGQ	0AF	02F	1AF	12F	4AF	42F	5AF	52F

3.5 Required PALcode Function Codes

The opcodes listed in Table 16 are required for all Alpha implementations. The notation used is oo.ffff, where oo is the hexadecimal 6-bit opcode and ffff is the hexadecimal 26-bit function code.

Table 16 Required PALcode Function Codes

Mnemonic	Type	Function Code
DRAINA	Privileged	00.0002
HALT	Privileged	00.0000
IMB	Unprivileged	00.0086

3.6 Opcodes Reserved for PALcode

The opcodes listed in Table 17 are reserved by the Alpha architecture and used in implementing PALcode for the 21064A. See the *Alpha Architecture Reference Manual* for more information.

Table 17 Opcodes Specific to the 21064A

21064A Mnemonic	Opcode	Architecture Mnemonic	21064A Mnemonic	Opcode	Architecture Mnemonic
HW_MFPR	19	PAL19	HW_LD	1B	PAL1B
HW_MTPR	1D	PAL1D	HW_REI	1E	PAL1E
HW_ST	1F	PAL1F	–	–	–

3.7 Opcodes Reserved for Digital

Table 18 lists the opcodes that are reserved for Digital.

Table 18 Opcodes Reserved for Digital

Mnemonic	Opcode	Mnemonic	Opcode	Mnemonic	Opcode
OPC01	01	OPC02	02	OPC03	03
OPC04	04	OPC05	05	OPC06	06
OPC07	07	OPC0A	0A	OPC0C	0C
OPC0D	0D	OPC0E	0E	OPC14	14

3.8 Instructions Specific to the 21064A

Table 19 lists instructions that are specific to the 21064A.

Table 19 Instructions Specific to the 21064A

Mnemonic	Operation	Type
HW_MTPR	Move data to processor register	PALmode, privileged
HW_MFPR	Move data from processor register	PALmode, privileged
HW_LD	Load data from memory	PALmode, privileged
HW_ST	Store data in memory	PALmode, privileged
HW_REI	Return from PALmode exception	PALmode, privileged

Programming Note

PALcode uses the HW_LD and HW_ST instructions to access memory outside the realm of normal Alpha memory management.

4 Internal Processor Registers

This section describes the internal processor registers of the 21064A microprocessor. The section is organized as follows:

- Ibox and Abox Internal Processor Registers
- PAL_TEMP Registers
- Lock Registers
- Internal Processor Registers Reset State

For the **21064A-275-PC**, the Abox Control Register SPE_1 field, described in Section 4.2.11, functions differently from the other four 21064A microprocessors. This register field controls the memory management operation mode.

4.1 Ibox Internal Processor Registers

This section describes each Ibox internal processor register (IPR).

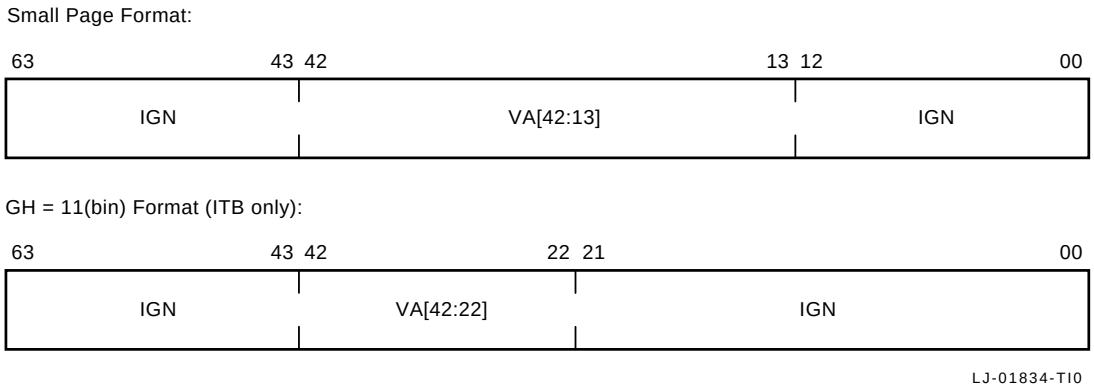
4.1.1 Translation Buffer Tag Register (TB_TAG)

The TB_TAG register is a write-only register that holds the tag for the next translation buffer update operation in the Instruction Translation Buffer (ITB) or the Data Translation Buffer (DTB). The tag is written to a temporary register and not transferred to the ITB or DTB until the Instruction Translation Buffer Page Table Entry (ITB_PTE) or the Data Translation Buffer Page Table Entry (DTB_PTE) register is written. The entry to be written is chosen at the time of the ITB_PTE or DTB_PTE write operation by a not-last-used algorithm, implemented in hardware. Figure 2 shows the TB_TAG register format.

Note

Writing to the Instruction Translation Buffer Tag array (ITB_TAG) is only performed while in PALmode, regardless of the state of the hardware enable (HWE) bit in the ICCSR register.

Figure 2 Translation Buffer Tag Register



4.1.2 Instruction Translation Buffer Page Table Entry Register (ITB_PTE)

The ITB_PTE register is a read/write register, representing twelve page table entries split into two distinct arrays. The first eight page table entries provide small page (8K byte) translations while the remaining four provide large page (4 MB) translations. The entry to be written is chosen by a not-last-used algorithm implemented in hardware for each array independently and the status of the TB_CTL. Writes to the ITB_PTE register use the memory format bit positions as described in the *Alpha Architecture Reference Manual*, with the exception that some fields are ignored.

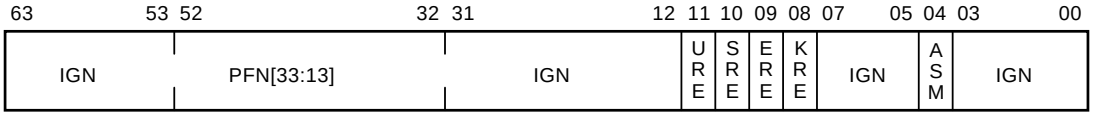
The ITB's tag array is updated simultaneously from the TB_Tag register when the ITB_PTE register is written. Reads of the ITB_PTE register require two instructions. The first instruction sends the PTE data to the Instruction Translation Buffer Page Table Entry Temporary register (ITB_PTE_TEMP) and the second instruction, reading from the ITB_PTE_TEMP register, returns the PTE entry to the register file. Reading or writing the ITB_PTE register increments the TB entry pointer corresponding to the large/small page selection indicated by the TB_CTL, which allows reading the entire set of ITB_PTE register entries. Figure 3 shows the ITB_PTE register format.

Note

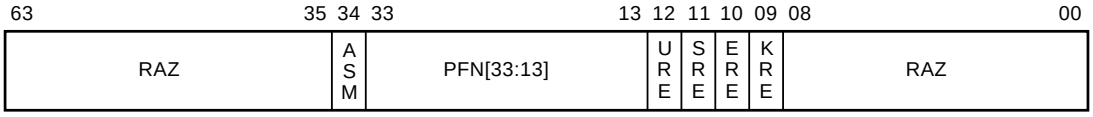
Reading and writing the ITB_PTE register is only performed while in PALmode regardless of the state of the HWE bit in the ICCSR IPR.

Figure 3 Instruction Translation Buffer Page Table Entry Register

Write Format:



Read Format:



LJ-01835-TI0

4.1.3 Instruction Cache Control and Status Register (ICCSR)

The ICCSR register contains various Ibox hardware enables. The only architecturally defined bit in this register is the floating-point enable (FPE), which enables floating-point instructions. When cleared, all floating-point instructions generate exceptions to the FEN entry point in PALcode. Most of this register is cleared by hardware at reset. Fields that are not cleared at reset include ASN, PC0, and PC1.

The hardware enable bit allows the special privileged architecture library code (PALcode) instructions to execute in kernel mode. This bit is intended for diagnostic or operating system alternative PALcode routines only. It does not allow access to the ITB registers if not running in PALmode. Figure 4 shows the ICCSR register format.

Table 20 lists the ICCSR register fields and a brief description. Table 21 lists branch states controlled by the Branch Prediction Enable (BPE) and Branch History Enable (BHE) bits in the ICCSR.

Figure 4 ICCSR Register

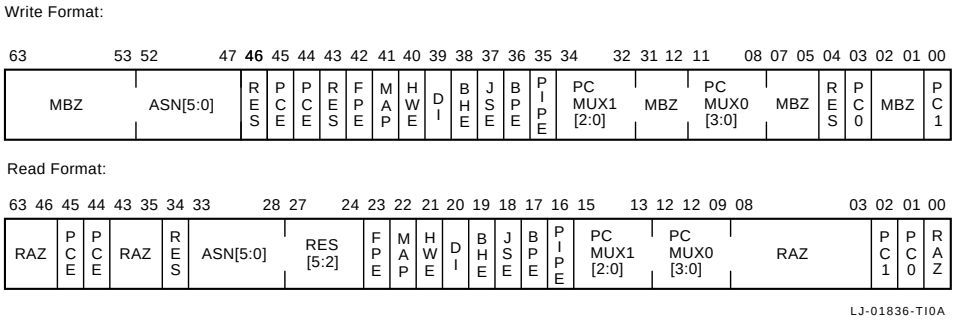


Table 20 ICCSR Fields and Description

Field	Type	Description
ASN	RW	The ASN field is used in conjunction with the Icache to further qualify cache entries and avoid some cache flushes. The ASN is written to the Icache during fill operations and compared with the I-stream data on fetch operations. Mismatches invalidate the fetch without affecting the Icache. (See the *Alpha Architecture Reference Manual*.)
RES	RW,0	The RES state bits are reserved by Digital and should not be used by software.
PCE	RW	If both of these bits are clear, they disable both performance counters. If either bit is set, both performance counters will increment in their usual fashion.
FPE	RW,0	If set, floating-point instructions can be issued. If clear, floating-point instructions cause FEN exceptions.
MAP	RW,0	If set, it allows superpage I-stream memory mapping of virtual PC [33:13] directly to Physical PC [33:13] essentially bypassing ITB for virtual PC addresses containing virtual PC [42:41] = 2. Superpage mapping is allowed in kernel mode only. The Icache ASM bit is always set. If clear, superpage mapping is disabled.

(continued on next page)

Table 20 (Cont.) ICCSR Fields and Description

Field	Type	Description
HWE	RW,0	If set, it allows the five reserved opcodes (PAL19, PAL1B, PAL1D, PAL1E, and PAL1F) instructions to be issued in kernel mode. If cleared, attempts to execute reserved opcodes instructions while not in PALmode result in OPCDEC exceptions.
DI	RW,0	If set, it enables dual issue. If cleared, instructions can only single issue.
BHE	RW,0	Used in conjunction with BPE. See Table 21 for programming information.
JSE	RW,0	If set, it enables the JSR stack to push a return address. If cleared, JSR stack is disabled.
BPE	RW,0	Used in conjunction with BHE. See Table 21 for programming information.
PIPE	RW,0	If clear, it causes all hardware interlocked instructions to drain the machine and waits for the write buffer to empty before issuing the next instruction. Examples of instructions that do not cause the pipe to drain include HW_MTPR, HW_REI, conditional branches, and instructions that have a destination register of R31. If set, pipeline proceeds normally.
PCMUX1	RW,0	See Table 23 for programming information.
PCMUX0	RW,0	See Table 22 for programming information.
PC1	RW	If clear, it enables performance counter 1 interrupt request after 2^{12} events counted. If set, enables performance counter 1 interrupt request after 2^{8} events counted.
PC0	RW	If clear, it enables performance counter 0 interrupt request after 2^{16} events counted. If set, it enables performance counter 0 interrupt request after 2^{12} events counted.

Note

Using the HW_MTPR instruction to update the EXC_ADDR register while in the native mode is restricted to bit [0] being equal to 0. The combination of the native mode and EXC_ADDR bit [0] being equal to one causes UNDEFINED behavior. This combination is only possible through the use of the HWE bit.

Table 21 BHE, BPE Branch Prediction Selection (Conditional Branches Only)

BPE	BHE	Prediction
0	X	Not Taken
1	0	Sign of Displacement
1	1	Branch History Table

4.1.3.1 Performance Counters The performance counters are reset to zero upon powerup. Otherwise, they are never cleared. The counters are intended as a means of counting events over a long period of time, relative to the event frequency. They provide no means of extracting intermediate counter values.

The performance counters may be enabled or disabled using ICCSR [45:44] (PCE [1:0]).

Since the counters continuously accumulate selected events, despite interrupts being enabled, the first interrupt after selecting a new counter input has an error bound as large as the selected overflow range. Some inputs can over count events occurring simultaneously with D-stream errors that abort the actual event very late in the pipeline.

For example, when counting load instructions, attempts to execute a load resulting in a TB miss exception will increment the performance counter after the first aborted execution attempt and again after the TB fill routine when the load instruction reissues and completes.

Performance counter interrupts are reported six cycles after the event that caused the counter to overflow. Additional delay can occur before an interrupt is serviced, if the processor is executing PALcode that always disables interrupts. Events occurring during the interval between counter overflow and interrupt service are counted toward the next interrupt. Only in the case of a complete counter wraparound while interrupts are disabled will an interrupt be missed.

The six cycles before an interrupt is triggered implies that a maximum of 12 instructions may have completed before the start of the interrupt service routine.

When counting Icache misses, no intervening instructions can complete and the exception PC contains the address of the last Icache miss. Branch mispredictions allow a maximum of only two instructions to complete before start of the interrupt service routine.

Table 22 lists performance counter 0 inputs and Table 23 lists performance counter 1 inputs.

Table 22 Performance Counter 0 Input Selection (in ICCSR)

MUX0 [3:0]	Input	Comment
000X	Total Issues/2	Counts total issues divided by 2, dual issue increments count by 1.
001X	Pipeline Dry	Counts cycles where nothing issued due to lack of valid I-stream data. Causes include Icache fill, misprediction, branch delay slots, and pipeline drain for exception.
010X	Load Instructions	Count all Load instructions.
011X	Pipeline Frozen	Counts cycles where nothing issued due to resource conflict.
100X	Branch Instructions	Counts all conditional branches, unconditional branches, JSR, and HW_REI instructions.
1011	PALmode	Counts cycles while executing in PALmode.
1010	Total cycles	Counts total cycles.
110X	Total Non-issues/2	Counts total non-issues divided by 2 ("no issue" increments count by 1).
111X	PERF_CNT_H [0]	Counts external events supplied to a pin at a selected system clock cycle interval.

Table 23 Performance Counter 1 Input Selection (in ICCSR)

MUX1 [2:0]	Input	Comment
000	Dcache miss	Counts total Dcache misses.
001	Icache miss	Counts total Icache misses.
010	Dual issues	Counts cycles of Dual issue.
011	Branch Mispredicts	Counts both conditional branch mispredictions and JSR or HW_REI mispredictions. Conditional branch mispredictions cost 4 cycles and others cost 5 cycles of dry pipeline delay.
100	FP Instructions	Counts total floating-point operate instructions, that is, no FP branch, load, or store.
101	Integer Operate	Counts integer operate instructions including LDA and LDAH with destination other than R31.
110	Store Instructions	Counts total store instructions.
111	PERF_CNT_H [1]	Counts external events supplied to a pin at a selected system clock cycle interval.

4.1.4 Instruction Translation Buffer Page Table Entry Temporary Register (ITB_PTE_TEMP)

The ITB_PTE_TEMP register is a read-only holding register for ITB_PTE read data. Reads of ITB_PTE register require two instructions to return data to the register file. The two instructions are as follows:

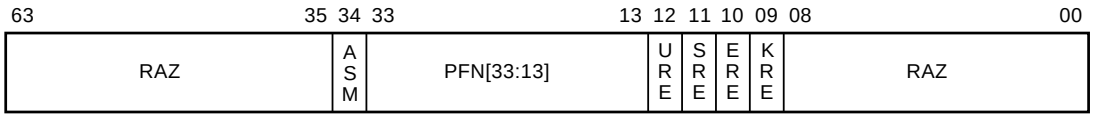
1. Read the ITB_PTE register data to the ITB_PTE_TEMP register.
2. Read the ITB_PTE_TEMP register data to the integer register file.

The ITB_PTE_TEMP register is updated on all ITB accesses, both read and write. A read of the ITB_PTE to the ITB_PTE_TEMP should be followed closely by a read of the ITB_PTE_TEMP to the register file. Figure 5 shows the ITB_PTE_TEMP register format.

Note

Reading the ITB_PTE_TEMP register is only performed while in PALmode regardless of the state of the HWE bit in the ICCSR.

Figure 5 ITB_PTE_TEMP Register



4.1.5 Exceptions Address Register (EXC_ADDR)

The EXC_ADDR register is a read/write register used to restart the system after exceptions or interrupts. The register can be read and written by the software, by way of the HW_MTPR instruction. Also, the EXC_ADDR can be written directly to by the hardware.

The HW_REI instruction executes a jump to the address contained in the EXC_ADDR register. The EXC_ADDR register is written by hardware after an exception to provide a return address for PALcode.

The instruction pointed to by the EXC_ADDR register did not complete its execution. The LSB of the EXC_ADDR register is used to indicate PALmode to the hardware. When the LSB is clear, the HW_REI instruction executes a jump to native (non-PAL) mode, enabling address translation.

CALL_PAL exceptions load the EXC_ADDR with the PC of the instruction following the CALL_PAL. This function allows CALL_PAL service routines to return without needing to increment the value in the EXC_ADDR register.

This feature requires careful treatment in PALcode. Arithmetic traps and machine check exceptions can preempt CALL_PAL exceptions resulting in an incorrect value being saved in the EXC_ADDR register. In the cases of an arithmetic trap or machine check exception (only in these cases), EXC_ADDR [1] takes on special meaning. PALcode servicing these two exceptions must:

- Interpret a 0 in EXC_ADDR [1] as indicating that the PC in EXC_ADDR [63:2] is too large by a value of 4 bytes and subtract 4 before executing a HW_REI from this address.
- Interpret a 1 in EXC_ADDR [1] as indicating that the PC in EXC_ADDR [63:2] is correct and clear the value of EXC_ADDR [1].

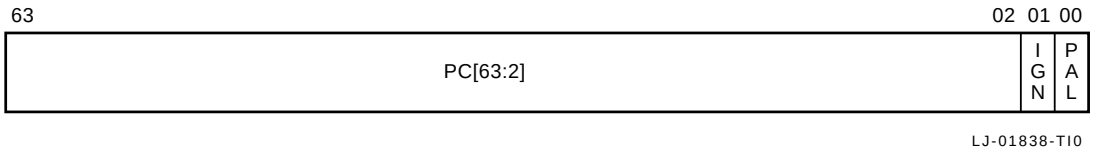
All other PALcode entry points except reset can expect EXC_ADDR [1] to be 0.

The logic allows the following code sequence to conditionally subtract 4 from the address in the EXC_ADDR register without the use of an additional register. This code sequence must be present in arithmetic trap and machine check flows only.

```
HW_MFPR Rx, EXC_ADDR   ; read EXC_ADDR into GPR
SUBQ    Rx, 2,Rx       ; subtract 2 causing borrow if bit [1]=0
BIC     Rx, 2,Rx       ; clear bit [1]
HW_MTPR Rx, EXC_ADDR   ; write back to EXC_ADDR
```

Figure 6 shows the exception address register format.

Figure 6 Exception Address Register



4.1.6 Clear Serial Line Interrupt Register (SL_CLR)

The SL_CLR is a write-only register that clears the:

- Serial line interrupt request
- Performance counter interrupt requests
- CRD interrupt request

The indicated bit must be written with a zero to clear the selected interrupt source. Figure 7 shows the clear serial line interrupt register format. Table 24 lists the register fields and a description.

Figure 7 Clear Serial Line Interrupt Register

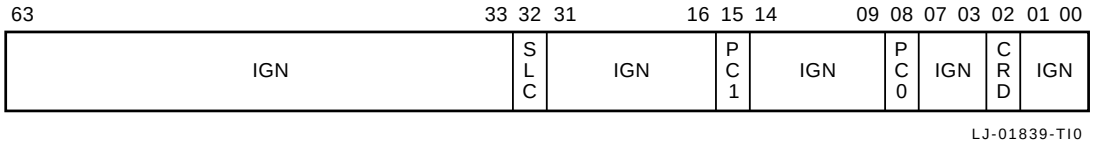


Table 24 Clear Serial Line Interrupt Register Fields

Field	Type	Description
CRD	W0C	Clears the correctable read error interrupt request
PC1	W0C	Clears the performance counter 1 interrupt request
PC0	W0C	Clears the performance counter 0 interrupt request
SLC	W0C	Clears the serial line interrupt request

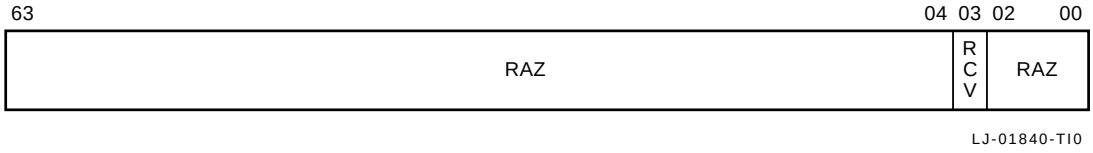
4.1.7 Serial Line Receive Register (SL_RCV)

The SL_RCV register contains a single read-only bit (RCV). This bit is used with the interrupt control registers, the **sRomD_h** pin, and the **sRomClk_h** pin to provide an on-chip serial line function. The RCV bit is functionally connected to the **sRomD_h** pin after the Icache is loaded from the external serial ROM. Using a software timing loop, the RCV bit can be read to receive external data one bit at a time.

A serial line interrupt is requested on detection of any transition on the receive line that sets the SLR bit in the HIRR. The serial line interrupt can be disabled by clearing the HIER register SLE bit.

Figure 8 shows the Serial Line Receive Register format.

Figure 8 Serial Line Receive Register



4.1.8 Instruction Translation Buffer ZAP Register (ITBZAP)

A write to this register invalidates all twelve instruction translation buffer (ITB) entries. It also resets both the NLU pointers to their initial state. The ITBZAP register is only written to in PALmode.

4.1.9 Instruction Translation Buffer ASM Register (ITBASM)

A write to this register invalidates all ITB entries, in which the ITB_PTE ASM bit is equal to zero. The ITBASM register is only written to in PALmode.

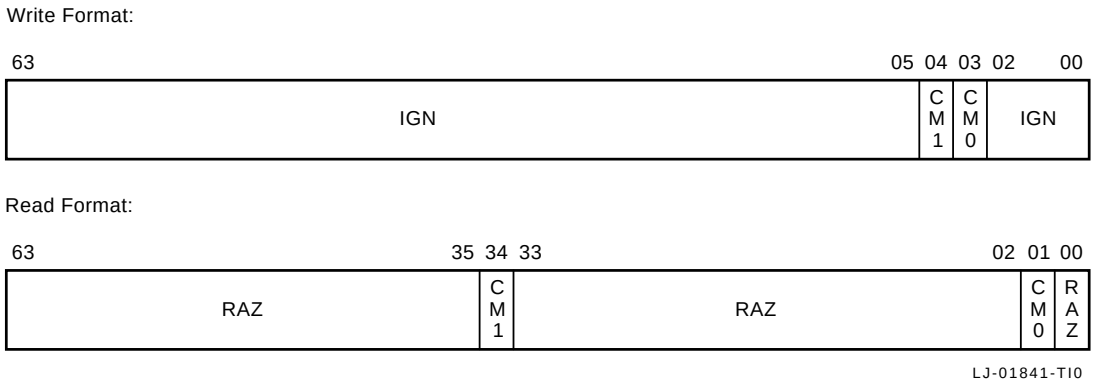
4.1.10 Instruction Translation Buffer IS Register (ITBIS)

A write to the ITBIS register invalidates all twelve ITB entries. It also resets both the NLU pointers to their initial state. The ITBIS register is only written to in PALmode. This register functions the same as the ITBZAP register.

4.1.11 Processor Status Register (PS)

The PS register is a read/write register containing only the current mode bits of the architecturally defined PS. Figure 9 shows the PS register format. See the *Alpha Architecture Reference Manual* for additional information.

Figure 9 Processor Status Register



4.1.12 Exception Summary Register (EXC_SUM)

The EXC_SUM register records the various types of arithmetic traps that occurred since the last time the EXC_SUM was written (cleared). When the result of an arithmetic operation produces an arithmetic trap, the corresponding EXC_SUM bit is set.

The register containing the result of the operation is recorded in the exception register write mask parameter, as a single bit in a 64-bit shift register specifying registers F31-F0 and I31-I0. The EXC_SUM register provides a one-bit window to the exception register write mask parameter. This is visible only through the EXC_SUM register.

Each read to the EXC_SUM shifts one bit in order F31-F0 then I31-I0. The read also clears the corresponding bit. The EXC_SUM must be read 64 times to extract the complete mask and clear the entire register. If no integer traps are present (IOV=0), only the first 32 corresponding floating-point register bits need to be read and cleared.

Any write to EXC_SUM clears bits [8:2] and does not affect the write mask bit.

The Write Mask register bit clears three cycles after a read. Code intended to read the register must allow at least three cycles between reads. This allows the clear and shift operations to complete in order to ensure reading successive bits. Figure 10 shows the exception summary register format. Table 25 lists the register fields and descriptions.

Figure 10 Exception Summary Register

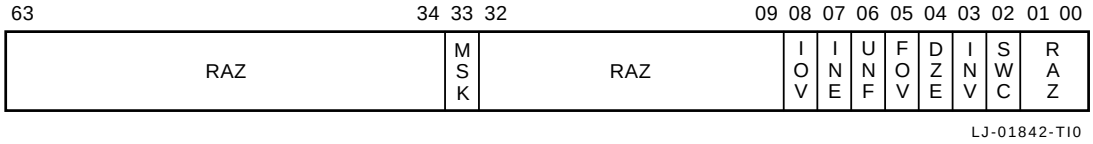


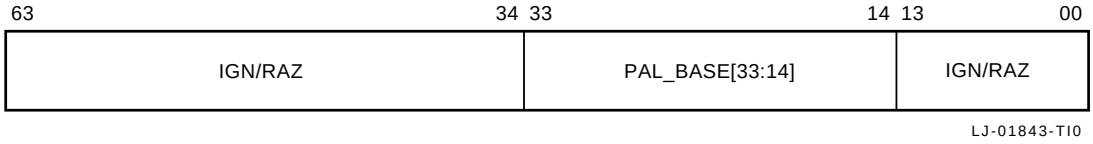
Table 25 Exception Summary Register Fields

Field	Type	Description
SWC	WA	Indicates software completion possible. The bit is set after a floating-point instruction containing the /S modifier completes with an arithmetic trap and all previous floating-point instructions that trapped since the last HW_MTPR EXC_SUM also contained the /S modifier. The SWC bit is cleared whenever a floating-point instruction without the /S modifier completes with an arithmetic trap. The bit remains cleared regardless of additional arithmetic traps until the register is written by way of an HW_MTPR instruction. The bit is always cleared upon any HW_MTPR write to the EXC_SUM register.
INV	WA	Indicates invalid operation.
DZE	WA	Indicates divide by zero.
FOV	WA	Indicates floating-point overflow.
UNF	WA	Indicates floating-point underflow.
INE	WA	Indicates floating inexact error.
IOV	WA	Indicates Fbox convert to integer overflow or integer arithmetic overflow.
MSK	RC	Exception Register Write Mask IPR window.

4.1.13 PAL_BASE Address Register (PAL_BASE)

The PAL_BASE register is a read/write register containing the base address for PALcode. This register is cleared by the hardware at reset. Figure 11 shows the PAL_BASE address register format.

Figure 11 PAL_BASE Address Register



4.1.14 Hardware Interrupt Request Register (HIRR)

The HIRR is a read-only register providing a record of all currently outstanding interrupt requests and summary bits at the time of the read. For each bit of the HIRR [5:0], there is a corresponding bit of the Hardware Interrupt Enable register (HIER) that must be set to request an interrupt.

In addition to returning the status of the hardware interrupt requests, a read of the HIRR returns the state of the software interrupt and AST requests.

Note

A read of the HIRR can return a value of zero if the hardware interrupt was released before the read (passive release).

The register guarantees that the HWR bit reflects the status as shown by the HIRR bits. All interrupt requests are blocked while executing in PALmode. Figure 12 shows the hardware interrupt request register format. Table 26 lists the register fields and gives a description of each.

Figure 12 Hardware Interrupt Request Register

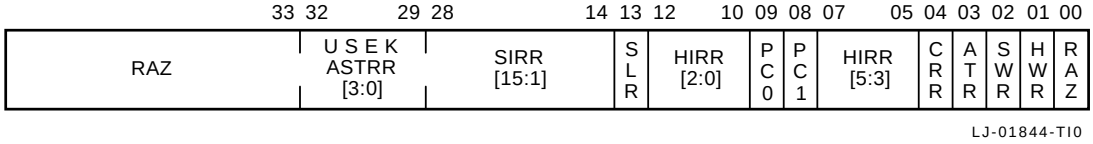


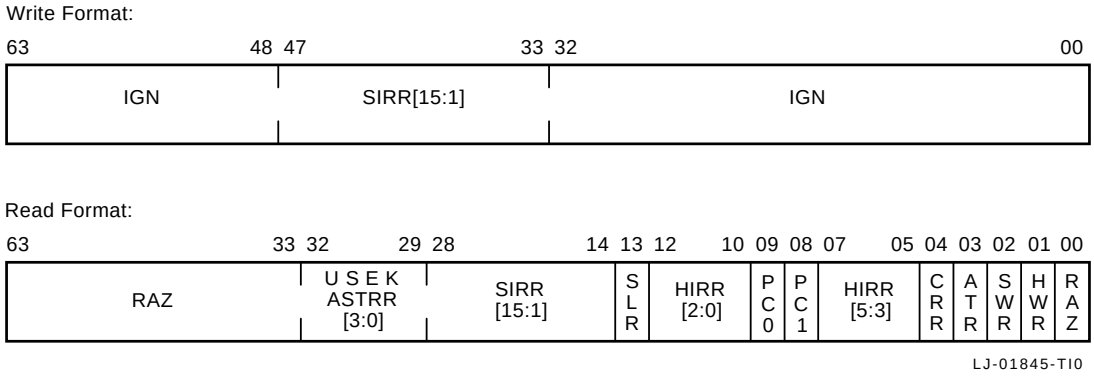
Table 26 Hardware Interrupt Request Register Fields

Field	Type	Description
HWR	RO	Is set if any hardware interrupt request and corresponding enable is set
SWR	RO	Is set if any software interrupt request and corresponding enable is set
ATR	RO	Is set if any AST request and corresponding enable is set. This bit also requires that the processor mode be equal to or higher than the request mode. SIER 2 must be set to allow AST interrupt requests.
CRR	RO	CRD correctable read error interrupt request. This interrupt is cleared by way of the SL_CLR register.
HIRR [5:0]	RO	Contains delayed copies of **Irq_h [5:0]** pins
PC1	RO	Performance counter 1 interrupt request
PC0	RO	Performance counter 0 interrupt request
SLR	RO	Serial line interrupt request. Also see SL_RCV, SL_ XMIT, and SL_CLR
SIRR [15:1]	RO	Corresponds to software interrupt request 15 through 1
ASTRR [3:0]	RO	Corresponds to AST request 3 through 0 (USEK)

4.1.15 Software Interrupt Request Register (SIRR)

The SIRR is a read/write register used to control software interrupt requests. For each bit of the SIRR, there is a corresponding bit of the Software Interrupt Enable register (SIER) that must be set to request an interrupt. Reads of the SIRR return the complete set of interrupt request registers and summary bits (see Table 26 for details). All interrupt requests are blocked while executing in PALmode. Figure 13 shows the SIRR format.

Figure 13 Software Interrupt Request Register

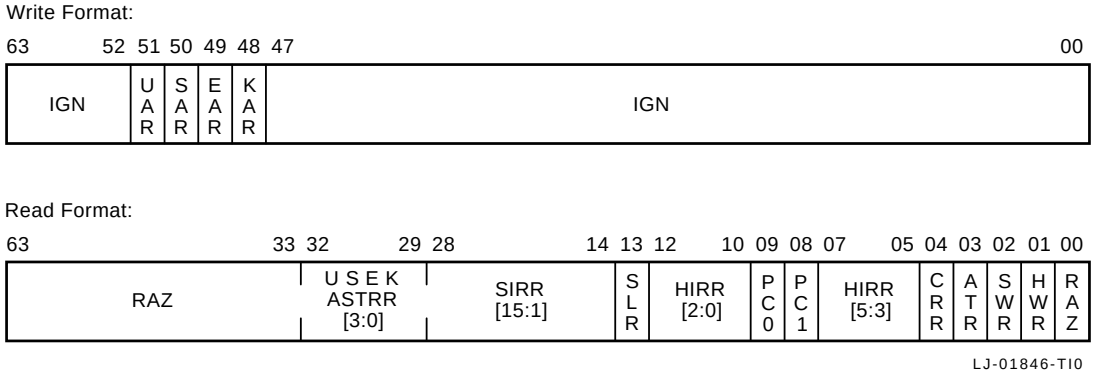


4.1.16 Asynchronous Trap Request Register (ASTRR)

The ASTRR is a read/write register. It contains bits to request AST interrupts in each of the processor modes. To generate an AST interrupt, the corresponding enable bit in the ASTER must be set. Also, the processor must be in the selected processor mode or higher privilege as described by the current value of the PS CM bits. AST interrupts are enabled if the SIER 2 is set. This provides a mechanism to lock out AST requests over certain IPL levels.

All interrupt requests are blocked while executing in PALmode. Reads of the ASTRR return the complete set of interrupt request registers and summary bits. See Table 26 for details. Figure 14 shows the ASTRR format.

Figure 14 Asynchronous Trap Request Register



LJ-01846-TI0

4.1.17 Hardware Interrupt Enable Register (HIER)

The HIER is a read/write register. It is used to enable corresponding bits of the HIRR requesting interrupt. The PC0, PC1, SLE, and CRE bits of this register enable the:

- Performance counters
- Serial line
- Correctable read interrupts

There is a one-to-one correspondence between the interrupt requests and enable bits. As with the reads of the interrupt request registers, reads of the HIER return the complete set of interrupt enable registers. See Table 26 for details. Figure 15 shows the hardware interrupt enable register format. Table 27 lists the register fields and a description of each.

Figure 15 Hardware Interrupt Enable Register

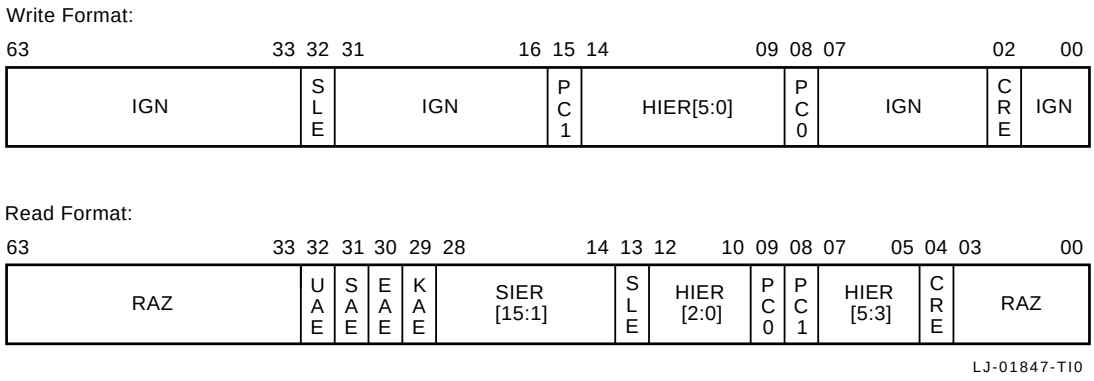


Table 27 Hardware Interrupt Enable Register Fields

Field	Type	Description
HIER [5:0]	RW	Interrupt enables for pins **Irq_h [5:0]**
SIER [15:1]	RW	Corresponds to software interrupt requests 15 through 1
ASTER [3:0]	RW	Corresponds to ASTRR enable 3 through 0 (USEK)
PC1	RW	Performance counter 1 interrupt enable

(continued on next page)

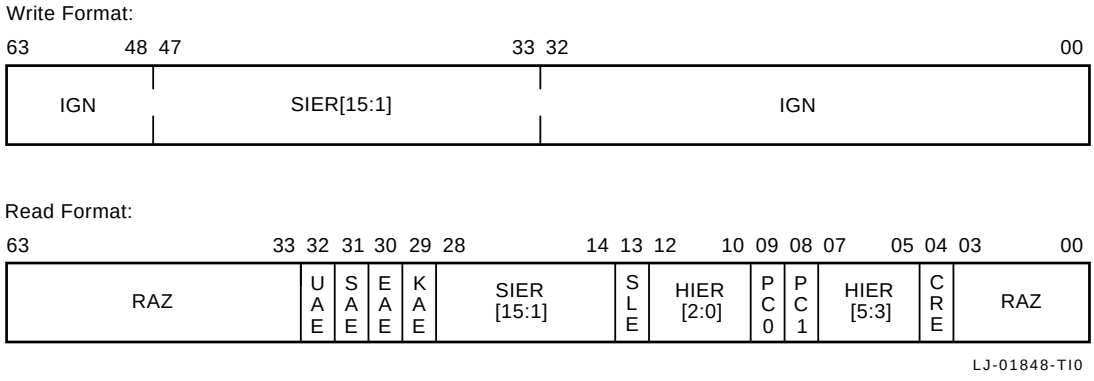
Table 27 (Cont.) Hardware Interrupt Enable Register Fields

Field	Type	Description
PC0	RW	Performance counter 0 interrupt enable
SLE	RW	Serial line interrupt enable Also see SL_RCV, SL_XMIT, and SL_CLR
CRE	RW	CRD correctable read error interrupt enable This interrupt request is cleared by way of the SL_CLR register

4.1.18 Software Interrupt Enable Register (SIER)

The SIER is a read/write register. It is used to enable corresponding bits of the SIRR requesting interrupts. There is a one-to-one correspondence between the interrupt requests and enable bits. As with the reads of the interrupt request registers, reads of the SIER return the complete set of interrupt enable registers. See Table 26 for details. Figure 16 shows the software interrupt enable register format.

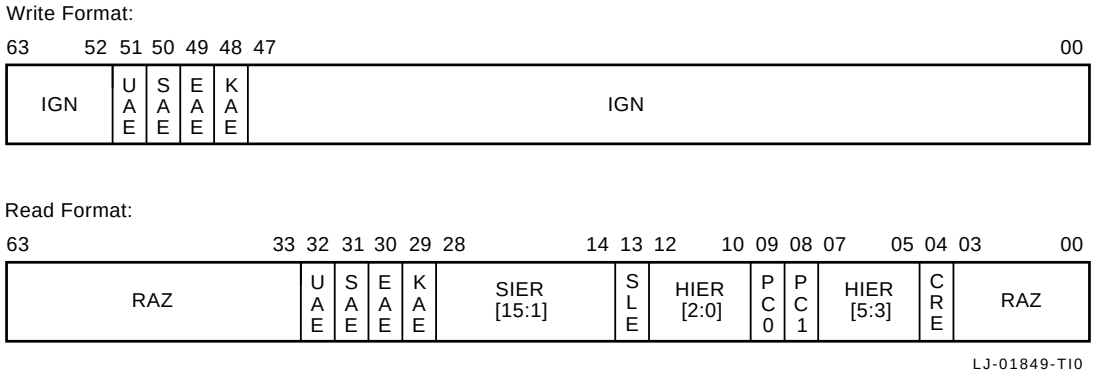
Figure 16 Software Interrupt Enable Register



4.1.19 AST Interrupt Enable Register (ASTER)

The ASTER is a read/write register. It is used to enable corresponding bits of the ASTRR requesting interrupts. There is a one-to-one correspondence between the interrupt requests and enable bits. As with the reads of the interrupt request registers, reads of the ASTER return the complete set of interrupt enable registers. See Table 26 for details. Figure 17 shows the ASTER format.

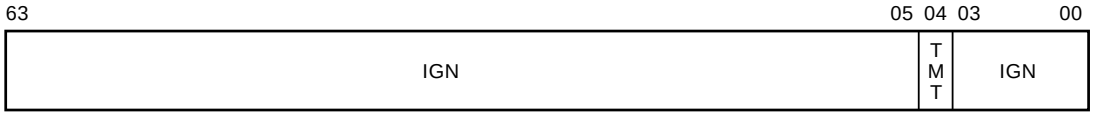
Figure 17 AST Interrupt Enable Register



4.1.20 Serial Line Transmit Register (SL_XMIT)

The SL_XMIT register contains a single write-only bit. This bit is used with the interrupt control registers, the **sRomD_h** pin, and the **sRomClk_h** pin to provide an on-chip serial line function. The TMT bit is functionally connected to the **sRomClk_h** pin after the Icache is loaded from the external serial ROM. Writing the TMT bit can be used to transmit data off chip, one bit at a time under a software timing loop. Figure 18 shows the SL_XMIT register format.

Figure 18 Serial Line Transmit Register



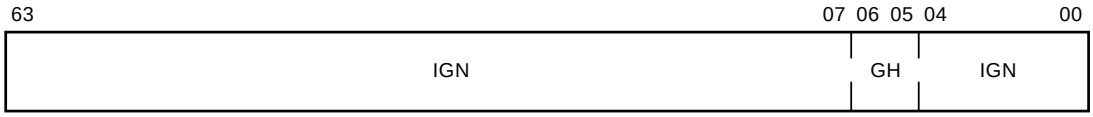
4.2 Abox Internal Processor Registers

The following sections describe the Abox internal processor registers.

4.2.1 Translation Buffer Control Register (TB_CTL)

The granularity hint (GH) field selects between th TB page mapping sizes. There are two sizes in the ITB and four sizes in the DTB. When only two sizes are provided, the large-page-select (GH=11(bin)) field selects the largest mapping size (512 * 8 KB). All other values select the smallest (8 KB) size. The GH field affects both reads and writes to the ITB and DTB. Figure 19 shows the translation buffer control register format. See the *Alpha Architecture Reference Manual* for additional information.

Figure 19 Translation Buffer Control Register

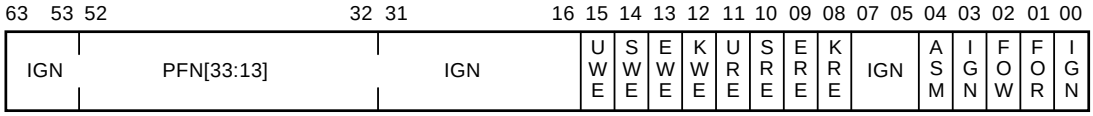


4.2.2 Data Translation Buffer Page Table Entry Register (DTB_PTE)

The DTB_PTE register is a read/write register representing the 32-entry DTB. The entry to be written is chosen by a not-last-used (NLU) algorithm implemented in the hardware. A DTB round robin (DTB_RR) algorithm can be selected by setting ABOX_CTL [9]. Writes to the DTB_PTE use the memory format bit positions as described in the *Alpha Architecture Reference Manual* with the exception that some fields are ignored. The valid bit is not represented in hardware.

The DTB's tag array is updated simultaneously from the TB_Tag register when the DTB_PTE register is written. Reads of the DTB_PTE require two instructions. The first instruction sends the PTE data to the Data Translation Buffer Page Table Entry Temporary register (DTB_PTE_TEMP). The second instruction, reading from the DTB_PTE_TEMP register, returns the PTE entry to the register file. Reading or writing the DTB_PTE register increments the TB entry pointer of the DTB, which allows reading the entire set of DTB_PTE entries. Figure 20 shows the DTB_PTE register format.

Figure 20 Data Translation Buffer Page Table Entry Register



LJ-01852-TI0

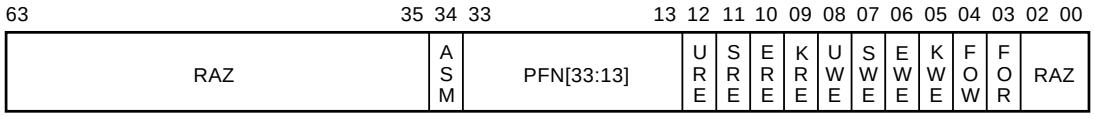
4.2.3 Data Translation Buffer Page Table Entry Temporary Register (DTB_PTE_TEMP)

The DTB_PTE_TEMP register is a read-only holding register for DTB_PTE read data. Reads of the DTB_PTE require two instructions to return the data to the register file. The two instructions are as follows:

- Read the DTB_PTE register data to the DTB_PTE_TEMP register.
- Read the DTB_PTE_TEMP register data to the integer register file.

Figure 21 shows DTB_PTE_TEMP register format.

Figure 21 Data Translation Buffer Page Table Entry Temporary Register



LJ-01853-TI0

4.2.4 Memory Management Control and Status Register (MM_CSR)

When D-stream faults occur the information about the fault is latched and saved in the MM_CSR register. The virtual address register (VA) and MM_CSR registers are locked against further updates until the software reads the Virtual Address register. PALcode must explicitly unlock this register whenever its entry point is higher in priority than a DTB miss. The MM_CSR bits are only modified by the hardware when the register is not locked and a memory management error or a DTB miss occurs. The MM_CSR is unlocked after reset. Figure 22 shows the MM_CSR register format. Table 28 lists the register fields and a brief description.

Figure 22 Memory Management Control and Status Register

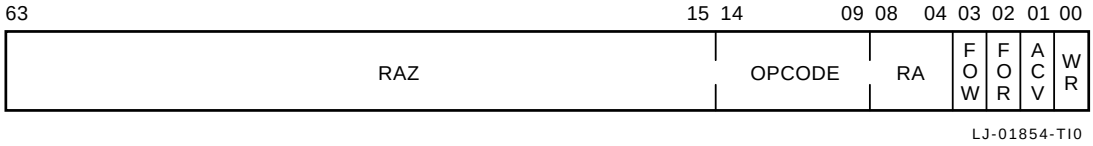


Table 28 Memory Management Control and Status Register

Field	Type	Description
WR	RO	Set if reference that caused error was a write.
ACV	RO	Set if reference caused an access violation.
FOR	RO	Set if reference was a read and the PTE's FOR bit was set.
FOW	RO	Set if reference was a write and the PTE's FOW bit was set.
RA	RO	RA field of the faulting instruction.
OPCODE	RO	Opcode field of the faulting instruction.

4.2.5 Virtual Address Register (VA)

When D-stream faults or DTB misses occur, the effective virtual address associated with the fault or miss is latched in the read-only VA register. The VA and MM_CSR registers are locked against further updates until the software reads the VA register. The VA register is unlocked after reset. PALcode must explicitly unlock this register whenever its entry point is higher in priority than a DTB miss.

4.2.6 Data Translation Buffer ZAP Register (DTBZAP)

The DTBZAP is a pseudo-register. A write to this register invalidates all 32 DTB entries. It also resets the not-last-used (NLU) pointer to its initial state.

4.2.7 Data Translation Buffer ASM Register (DTBASM)

The DTBASM is a pseudo-register. A write to this register invalidates all 32 DTB entries in which the ASM bit is equal to zero.

4.2.8 Data Translation Buffer Invalidate Single Register (DTBIS)

A write to this pseudo-register will invalidate the DTB entry, which maps the virtual address held in the integer register. The integer register is identified by the Rb field of the HW_MTPR instruction, used to perform the write.

4.2.9 Flush Instruction Cache Register (FLUSH_IC)

A write to this pseudo-register flushes the entire instruction cache.

4.2.10 Flush Instruction Cache ASM Register (FLUSH_IC_ASM)

A write to this pseudo-register invalidates all Icache blocks in which the ASM bit is clear.

4.2.11 Abox Control Register (ABOX_CTL)

Figure 23 shows the Abox control register format. Table 29 lists the register fields and descriptions.

Figure 23 Abox Control Register

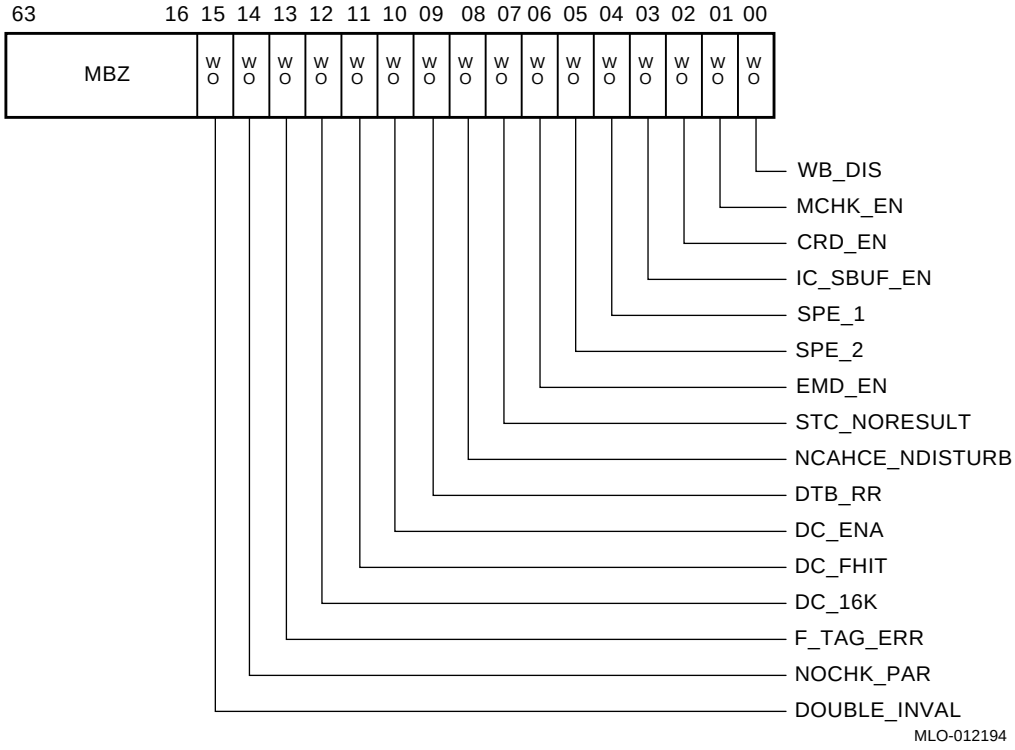


Table 29 Abox Control Register Fields

Field	Type	Description
WB_DIS	WO,0	Write Buffer unload Disable. When set, this bit prevents the write buffer from sending write data to the BIU. It should be set for diagnostics only.

(continued on next page)

Table 29 (Cont.) Abox Control Register Fields

Field	Type	Description
MCHK_EN	WO,0	Machine Check Enable. When this bit is set, the Abox generates a machine check when errors (which are not correctable by the hardware) are encountered. When this bit is cleared, uncorrectable errors do not cause a machine check. However, the BIU_STAT, DC_STAT, BIU_ADDR, and FILL_ADDR registers are updated and locked when the errors occur.
CRD_EN	WO,0	Corrected read data interrupt enable. When this bit is set, the Abox generates an interrupt request whenever a pin bus transaction is terminated with a **cAck_h** code of SOFT_ERROR.
IC_SBUF_EN	WO,0	Icache stream buffer enable. When set, this bit enables operation of a single entry Icache stream buffer.
SPE_1	WO,0	When this bit is set, it enables one-to-one superpage mapping of the D-stream virtual addresses with VA [42:30] = 1FFE (Hex) to the physical addresses with PA [33:30] = 0 (Hex). Access is only allowed in kernel mode. **Note** For the **21064A-275-PC** this bit must always be set when virtual-to-physical mapping is enabled. Operation in native mode (not PALmode) with this bit clear will will cause **21064A-275-PC** operation to be **UNPREDICTABLE**.
SPE_2	WO,0	When this bit is set, it enables one-to-one super page mapping of the D-stream virtual addresses with VA [33:13] directly to physical addresses PA [33:13], if virtual address bits VA [42:41] = 2. Virtual address bits VA [40:34] are ignored in this translation. Access is only allowed in kernel mode.
EMD_EN	WO,0	Limited hardware support is provided for big endian data formats by way of bit [6] of the ABOX_CTL register. When set, this bit inverts the physical address bit [2] for all D-stream references. It is intended that the chip endian mode be selected during initialization of PALcode only.

(continued on next page)

Table 29 (Cont.) Abox Control Register Fields

Field	Type	Description
STC_NORESULT	WO,0	When clear the 21064A implements lock operation in conformance to Alpha Architecture. When set the the 21064A does not conform to Alpha architecture. See the following two items. When STC_NORESULT is set these two items apply. • The result written into the register identified by Ra in STL_C/STQ_C and HW_ST/C instructions is UNPREDICTABLE. This allows the Ibox to restart the memory reference pipeline when the STL_C/STQ_C is transferred from the write buffer to the BIU, and so increases the repetition rate with which STL_C/STQ_C instructions can be processed. • LDL_L/LDQ_L, STL_C/STQ_C and HW_ST/C instructions will invalidate the Dcache line associated with their generated address. These invalidates will not be visible to load or store instructions that issue in the two CPU cycles after the LDL_L/LDQ_L, STL_C/STQ_C or HW_ST/C issues. This bit is cleared by chip reset.
NCACHE_NDISTURB	WO,0	When this bit is set, it enables a mode which make noncacheable only those external reads for which the 21064A does not probe the external cache. This bit is cleared by chip reset.
DTB_RR[2]	WO,0	When this bit is set, it selects the round robin replacement algorithm in the DTB.
DC_ENA	WO,0	Dcache enable. When clear, this bit disables and flushes the Dcache. When set, this bit enables the Dcache.
DC_FHIT	WO,0	Dcache force hit. When set, this bit forces all D-stream references to hit in the Dcache. This bit takes precedence over DC_ENA. That is, when DC_FHIT is set and DC_ENA is clear all D-stream references hit in the Dcache.
DC_16K	WO,0	Set to select 16K byte Dcache. Clear to select 8K byte Dcache.
F_TAG_ERR	WO,0	Set to generate bad Dcache tag parity on fills.
NOCHK_PAR	WO,0	Set to disable checking of Icache and Dcache parity.
DOUBLE_INVAL	WO,0	When set, asserting **dInvReq_h 0** invalidates both Dcache blocks addressed by **iAdr_h [12:5]**.

4.2.12 Alternate Processor Mode Register (ALT_MODE)

The ALT_MODE is a write-only register. The AM field specifies the alternate processor mode used by HW_LD and HW_ST instructions that have their ALT bit (bit [14]) set. Figure 24 shows the alternate processor mode register format and Table 30 lists the register modes.

Figure 24 Alternate Processor Mode Register

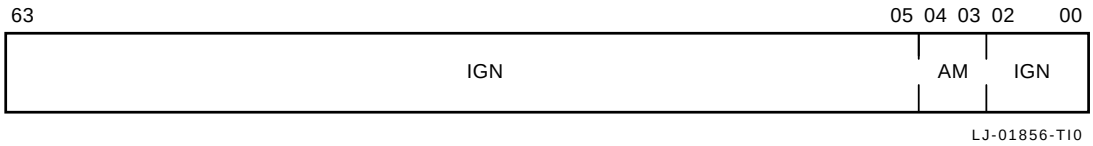


Table 30 Alternate Processor Mode Register

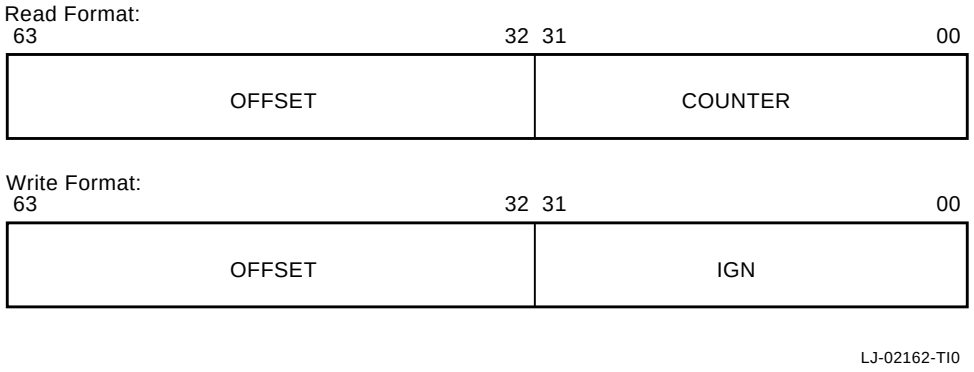
ALT_MODE [4:3]	Mode
0 0	Kernel
0 1	Executive
1 0	Supervisor
1 1	User

4.2.13 Cycle Counter Register (CC)

The 21064A supports a cycle counter, as described in the *Alpha Architecture Reference Manual*. When enabled, the CC increments once each CPU cycle. The HW_MTPR Rn, CC writes the CC [63:32] with the value held in the Rn [63:32]. The CC [31:0] are not changed. This register is read by the RPCC instruction as defined in the *Alpha Architecture Reference Manual*.

Figure 25 shows the register format (top register) when read by the HW_MFPR Rn, CC instruction and when written (bottom register) by the HW_MTPR Rn, CC instruction.

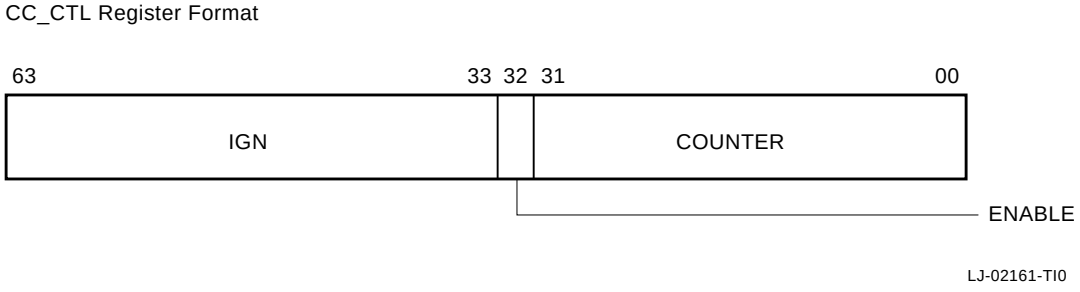
Figure 25 Cycle Counter Register



4.2.14 Cycle Counter Control Register (CC_CTL)

The HW_MTPR Rn, CC_CTL writes the CC [31:0] with the value held in Rn [31:0]. The CC register bits [63:32] are not changed. The CC register bits [3:0] must be written with zero. If Rn bit [32] is set, then the counter is enabled, otherwise the counter is disabled. CC_CTL is a write-only register. Figure 26 shows the register format when written by the HW_MTPR Rn, CC_CTL instruction.

Figure 26 Cycle Counter Control Register



4.2.15 Bus Interface Unit Control Register (BIU_CTL)

Figure 27 shows the bus interface unit control register format. Table 31 lists the register fields and gives a description of each.

Figure 27 21064A Bus Interface Unit Control Register

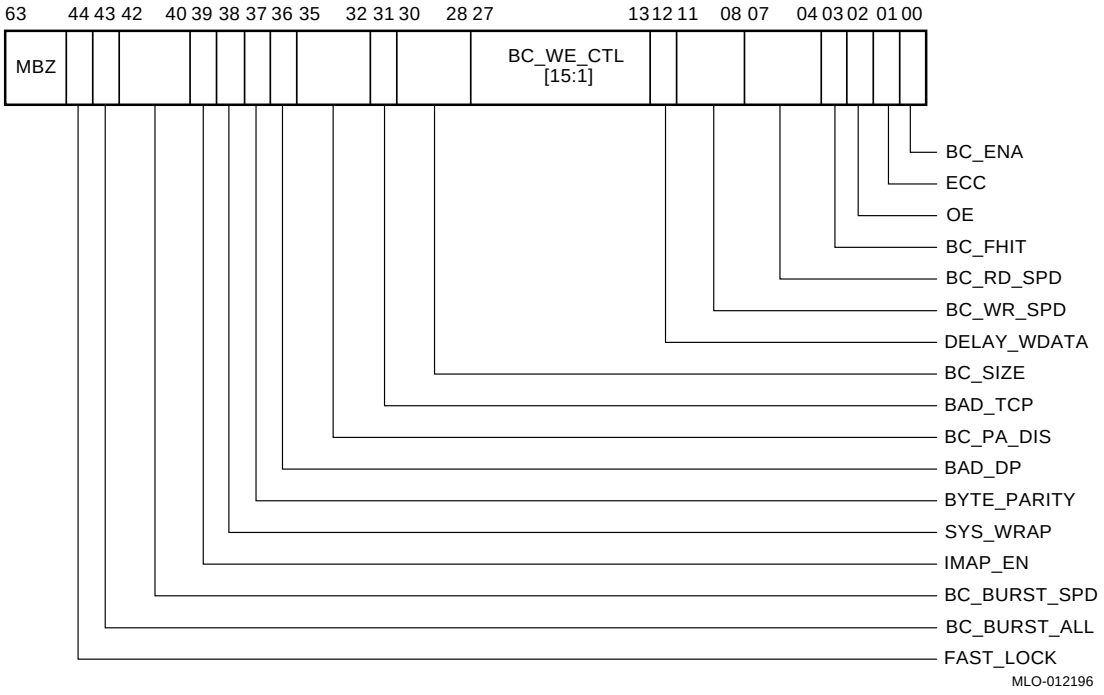


Table 31 Bus Interface Unit Control Register Fields

Field	Type	Description
BC_ENA	WO,0	External cache enable. When this bit is cleared, the bit disables the external cache. When the Bcache is disabled, the BIU does not probe the external cache tag store for read/write references; it launches a request on **cReq_h** immediately.

(continued on next page)

Table 31 (Cont.) Bus Interface Unit Control Register Fields

Field	Type	Description
ECC	WO,0	When this bit is clear, the 21064A generates/expects parity on four of the **check_h** pins. When this bit is set, the 21064A generates/expects ECC on the **check_h** pins.
OE	WO,0	When this bit is set, the 21064A does not assert its chip enable pins during RAM write cycles, thus enabling these pins to be connected to the output enable pins of the cache RAMs. **Caution** The output enable bit in the BIU_CTL register (BIU_CTL [2]) must be set if the system uses SRAMs in the output enable mode (that is, if the **tagCEOE** and/or **dataCEOE** signals are connected to the output enable input of the SRAM and the 21064A enable is always enabled). If this bit is inadvertently cleared, the tag and data SRAMs will be enabled during writes, and damage can result.
BC_FHIT	WO,0	External cache force hit. When this bit is set and the BC_ENA bit is also set, all pin bus READ_BLOCK and WRITE_BLOCK transactions are forced to hit in external cache. Tag and tag control parity are ignored. The BC_ENA takes precedence over BC_FHIT. When BC_ENA is cleared and BC_FHIT is set, no tag probes occur and external requests are directed to the **cReq_h** pins. **Note** The BC_PA_DIS field takes precedence over the BC_FHIT bit.

(continued on next page)

Table 31 (Cont.) Bus Interface Unit Control Register Fields

Field	Type	Description
BC_RD_SPD	WO,0	External cache read speed. This field indicates to the BIU the read access time of the RAMs used to implement the off-chip external cache, measured in CPU cycles. It should be written with a value equal to one less than the read access time of the external cache RAMs. 21064A access times for reads must be in the range [16:3] CPU cycles, which means the values for the BC_RD_SPD field are in the range of [15:2].
BC_WR_SPD	WO,0	External cache write speed. This field indicates to the BIU the write cycle time of the RAMs used to implement the off-chip external cache, measured in CPU cycles. It should be written with a value equal to one less than the write cycle time of the external cache RAMs. The access times for writes must be in the range [16:2] CPU cycles, which means the values for the BC_WR_SPD field are in the range of [15:1].
DELAY_WDATA	WO,0	When this bit is set, it changes the timing of the data bus during external cache writes.
BC_WE_CTL	WO,0	External cache write enable control. This field is used to control the timing of the write enable and chip enable pins during writes into the data and tag control RAMs. It consists of 15 bits, where each bit determines the value placed on the write enable and chip enable pins during a given CPU cycle of the RAM write access. When a given bit of the BC_WE_CTL is set, the write enable and chip enable pins are asserted during the corresponding CPU cycle of the RAM access. The BC_WE_CTL bit [0] (bit [13] in BIU_CTL) corresponds to the second cycle of the write access, BC_WE_CTL [1] (bit [14] in BIU_CTL) to the third CPU cycle, and so on. The write enable pins will never be asserted in the first CPU cycle of a RAM write access. Unused bits in the BC_WE_CTL field must be written with zeros.
BC_SIZE	WO,0	This field is used to indicate the size of the external cache. See Table 32 for the encodings.
BAD_TCP	WO,0	When set, this bit causes the 21064A to write bad parity into the tag control RAM whenever it does a fast external RAM write. (Diagnostic use only.)

(continued on next page)

Table 31 (Cont.) Bus Interface Unit Control Register Fields

Field	Type	Description
BC_PA_DIS	WO,0	This 4-bit field may be used to prevent the CPU chip from using the external cache to service reads and writes based upon the quadrant of physical address space that they reference. The correspondence between this bit field and the physical address space is shown in Table 33. When a read or write reference is presented to the BIU the values of BC_PA_DIS, BC_ENA, and the physical address bits [33:32] determine whether to attempt to use the external cache to satisfy the reference. If the external cache is not to be used for a given reference the BIU does not probe the tag store and makes the appropriate system request immediately. The value of BC_PA_DIS has NO impact on which portions of the physical address space can be cached in the primary caches. System components control this by way of the **dRAck_h** field of the pin bus.
BAD_DP	WO,0	When this bit is set, the BAD_DP causes the 21064A to invert the value placed on bits [0], [7], [14] and [21] of the **check_h [27:0]** field during off-chip writes. This produces bad parity when the 21064A is in parity mode, and bad check bit codes when in ECC mode. (Diagnostic use only.)
SYS_WRAP	WO,0	When this bit is set, it indicates that the system returns read response data wrapped around the requested chunk. This bit is cleared by chip reset.
BC_BURST_SPD	WO,0	When these bits are cleared, the timing of all Bcache reads is controlled by the value of BC_RD_SPD. When these bits are set in 128-bit mode, the second read takes BC_BURST_SPD+1 cycles. When these bits are set in 64-bit mode, the second and fourth reads take BC_BURST_SPD+1 cycles. If BC_BURST_ALL is set, the third read takes BC_BURST_ SPD+1 cycles also.
BC_BURST_ALL	WO,0	In 64-bit mode this bit is set if BC_BURST_SPD should be used to time the third (of four) RAM read cycle.
BYTE_PARITY	WO,0	If set when BIU_CTL ECC is cleared, external byte parity is selected. If set when BIU_CTL ECC is set, this bit is ignored.
IMAP_EN	WO,0	Set to allow **dMapWE_h [1:0]** to assert for I-stream backup cache reads.

(continued on next page)

Table 31 (Cont.) Bus Interface Unit Control Register Fields

Field	Type	Description
FAST_LOCK	WO,0	When set, FAST_LOCK mode operation is selected. FAST_ LOCK mode can only be used when BIU_CTL [2] OE is also set indicating that OE mode Bcache RAMs are used.

Table 32 lists the encoding for BC_SIZE. Table 33 lists the BIU_CTL physical addresses.

Table 32 BC_SIZE

BC_SIZE	Cache Size	BC_SIZE	Cache Size
0 0 0	128 KB	1 0 0	2 MB
0 0 1	256 KB	1 0 1	4 MB
0 1 0	512 KB	1 1 0	8 MB
0 1 1	1 MB	1 1 1	16 MB

Table 33 BC_PA_DIS

BIU_CTL Bits	Physical Address	BIU_CTL Bits	Physical Address
32	PA [33:32] = 0	34	PA [33:32] = 2
33	PA [33:32] = 1	35	PA [33:32] = 3

4.2.16 Cache Status Register (C_STAT)

The C_STAT is a read-only register and is only used by the diagnostics.

Figure 28 shows the 21064A Dcache status register format. Table 34 lists the register fields and gives a description of each.

Figure 28 Cache Status Register

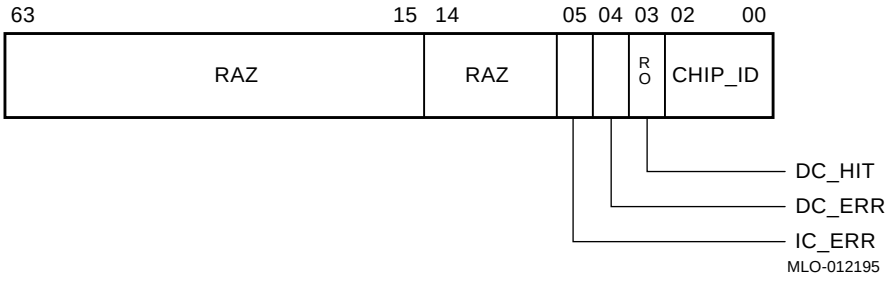


Table 34 Cache Status Register Fields

Field	Type	Description
CHIP_ID	RO	These bits identify the devices as listed here: • 001_2—Early version of **21064A** • 011_2—Production version of **21064A**
DC_HIT	RO	This bit indicates whether the last load or store instruction processed by the Abox hit (DC_HIT set) or missed (DC_HIT clear) the Dcache. Loads that miss the Dcache can be completed without requiring external reads. (Diagnostic use only.)
DC_ERR	RC	Set by Dcache parity error.
IC_ERR	RC	Set by Icache parity error.

4.2.17 Bus Interface Unit Status Register (BIU_STAT)

The BIU_STAT is a read-only register.

Bits [6:0] of the BIU_STAT register are locked against further updates when one of the following bits is set:

- BIU_HERR
- BIU_SERR
- BC_TPERR
- BC_TCPERR

The address associated with the error is latched and locked in the BIU_ADDR register. Bits [6:0] of the BIU_STAT register and BIU_ADDR are also spuriously locked when a parity error or an uncorrectable ECC error occurs during a primary cache fill operation. The BIU_STAT bits [7:0] and BIU_ADDR are unlocked when the BIU_ADDR register is read.

When FILL_ECC or FILL_DPERR is set, BIU_STAT bits [13:8] are locked against further updates. The address associated with the error is latched and locked in the FILL_ADDR register. The BIU_STAT bits [14:8] and FILL_ADDR are unlocked when the FILL_ADDR register is read.

This register is not unlocked or cleared by reset and needs to be explicitly cleared by PALcode.

Figure 29 shows the bus interface unit status register format. Table 35 lists the register fields and gives a description of each.

Figure 29 Bus Interface Unit Status Register

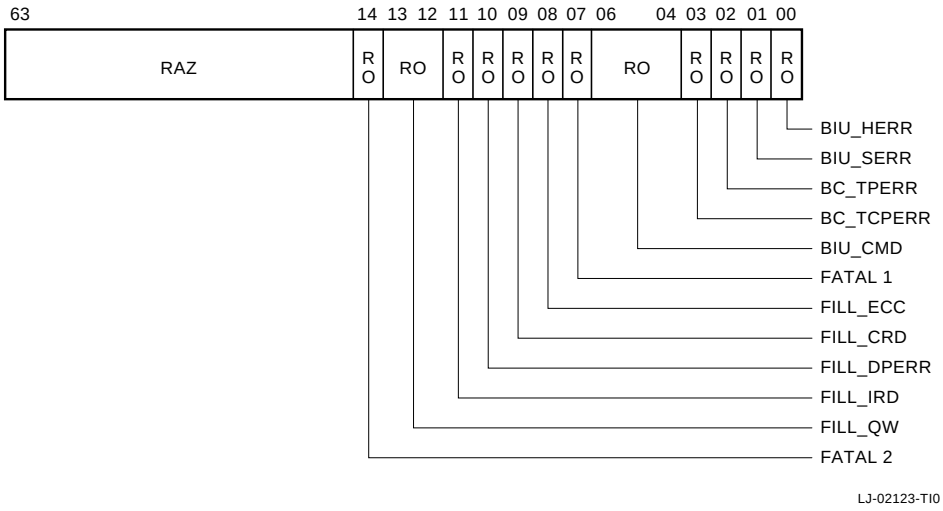


Table 35 Bus Interface Unit Status Register Fields

Field	Type	Description
BIU_HERR	RO	When this bit is set, it indicates that an external cycle was terminated with the **cAck_h** pins indicating HARD_ERROR.
BIU_SERR	RO	When this bit is set, it indicates that an external cycle was terminated with the **cAck_h** pins indicating SOFT_ERROR.
BC_TPERR	RO	When this bit is set, it indicates that an external cache tag probe encountered bad parity in the tag address RAM.
BC_TCPERR	RO	When this bit is set, it indicates that an external cache tag probe encountered bad parity in the tag control RAM.
BIU_CMD	RO	This field latches the cycle type on the cReq_h pins when a BIU_HERR, BIU_SERR, BC_TPERR, or BC_TCPERR error occurs.

(continued on next page)

Table 35 (Cont.) Bus Interface Unit Status Register Fields

Field	Type	Description
FATAL1	RO	When this bit is set, it indicates that an external cycle was terminated with the **cAck_h** pins indicating HARD_ ERROR or that an external cache tag probe encountered bad parity in the tag address RAM or the tag control RAM while one of BIU_HERR, BIU_SERR, BC_TPERR, or BC_TCPERR was already set.
FILL_ECC	RO	ECC error. When this bit is set, it indicates that primary cache fill data received from outside the CPU chip contained an ECC error.
FILL_CRD	RO	Correctable read. This bit only has meaning when FILL_ ECC is set. When this bit is set, it indicates that the information latched in BIU_STAT [13:8], FILL_ADDR, and FILL_SYNDROME relates to an error quadword which does not contain multi-bit errors in either of its component longwords.
FILL_DPERR	RO	Fill Parity Error. When this bit is set, it indicates that the BIU received data with a parity error from outside the CPU chip while performing either a Dcache or Icache fill. FILL_DPERR is only meaningful when the CPU chip is in parity mode, as opposed to ECC mode.
FILL_IRD	RO	This bit is only meaningful when either FILL_ECC or FILL_DPERR is set. The FILL_IRD bit is set to indicate that the error that caused FILL_ECC or FILL_DPERR to set occurred during an Icache fill and clear to indicate that the error occurred during a Dcache fill.
FILL_QW	RO	This field is only meaningful when either FILL_ECC or FILL_DPERR is set. The FILL_QW bit identifies the quadword within the hexaword primary cache fill block which caused the error. It can be used together with FILL_ADDR [33:5] to get the complete physical address of the bad quadword.
FATAL2	RO	When this bit is set, it indicates that a primary cache fill operation resulted in either a multi-bit ECC error or in a parity error while FILL_ECC or FILL_DPERR was already set.

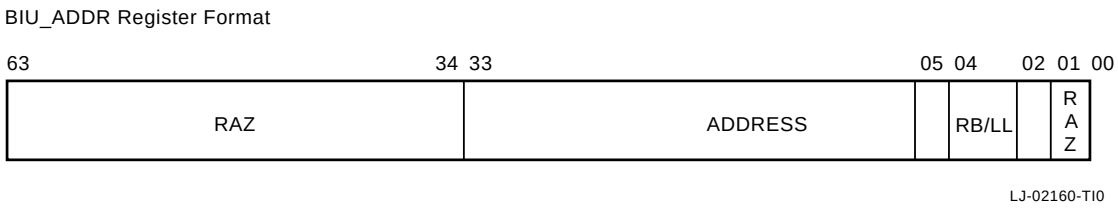
4.2.18 Bus Interface Unit Address Register (BIU_ADDR)

The BIU_ADDR is a read-only register that contains the physical address associated with errors reported by BIU_STAT [7:0]. Its contents are meaningful only when one of BIU_HERR, BIU_SERR, BC_TPERR, or BC_TCPERR are set. Reads of the BIU_ADDR register unlock both BIU_ADDR and BIU_STAT [7:0].

The BIU_ADDR bits [33:5] contain the values of **adr_h** bits [33:5] associated with the pin bus transaction that resulted in the error indicated in BIU_STAT [7:0].

If the BIU_CMD field of the BIU_STAT register indicates that the transaction that received the error was READ_BLOCK or load_locked, then BIU_ADDR [4:2] are UNPREDICTABLE. If the BIU_CMD field of the BIU_STAT register encodes any pin bus command other than READ_BLOCK or load_locked, then BIU_ADDR bits [4:2] will contain zeros. The BIU_ADDR bits [63:34] and BIU_ADDR bits [1:0] always read as zero. Figure 30 shows the bus interface unit address register (BIU_ADDR) format.

Figure 30 Bus Interface Unit Address Register



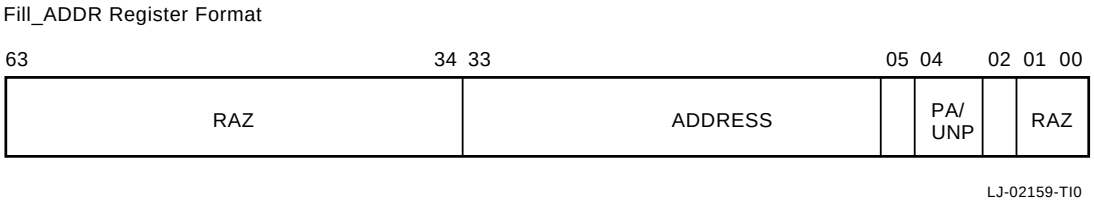
4.2.19 Fill Address Register (FILL_ADDR)

The FILL_ADDR is a read-only register that contains the physical address associated with errors reported by BIU_STAT bits [14:8]. Its contents are meaningful only when FILL_ECC or FILL_DPERR is set. Reads of the FILL_ADDR unlock FILL_ADDR, BIU_STAT bits [14:8] and FILL_SYNDROME.

The FILL_ADDR bits [33:5] identify the 32-byte cache block that the CPU was attempting to read when the error occurred.

If the FILL_IRD bit of the BIU_STAT register is clear, it indicates that the error occurred during a D-stream cache fill. At such times, FILL_ADDR bits [4:2] contain bits [4:2] of the physical address generated by the load instruction that triggered the cache fill. If FILL_IRD is set, then FILL_ADDR bits [4:2] are UNPREDICTABLE. The FILL_ADDR bits [63:34] and FILL_ADDR bits [1:0] will read as zero. Figure 31 shows the fill address register (FILL_ADDR) format.

Figure 31 Fill Address Register



4.2.20 Fill Syndrome Register (FILL_SYNDROME)

The FILL_SYNDROME register is a 14-bit read-only register.

If the chip is in ECC mode and an ECC error is recognized during a primary cache fill operation, the syndrome bits associated with the bad quadword are locked in the FILL_SYNDROME register. The FILL_SYNDROME bits [6:0] contain the syndrome associated with the lower longword of the quadword, and FILL_SYNDROME bits [13:7] contain the syndrome associated with the upper longword of the quadword. A syndrome value of zero means that no errors were found in the associated longword. See Table 36 for a list of syndromes associated with correctable single-bit errors. The FILL_SYNDROME register is unlocked when the FILL_ADDR register is read.

If the chip is in parity mode and a parity error is recognized during a primary cache fill operation, the FILL_SYNDROME register indicates which of the longwords in the quadword got bad parity. The FILL_SYNDROME bit [0] is set to indicate that the lower longword was corrupted, and FILL_SYNDROME bit [7] is set to indicate that the upper longword was corrupted. The FILL_ SYNDROME bits [13:8] and [6:1] are RAZ in parity mode. Figure 32 shows the fill syndrome register format.

Figure 32 FILL_SYNDROME Register

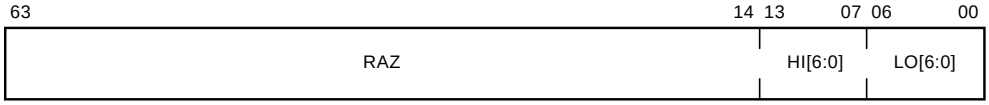


Table 36 Syndromes for Single-Bit Errors

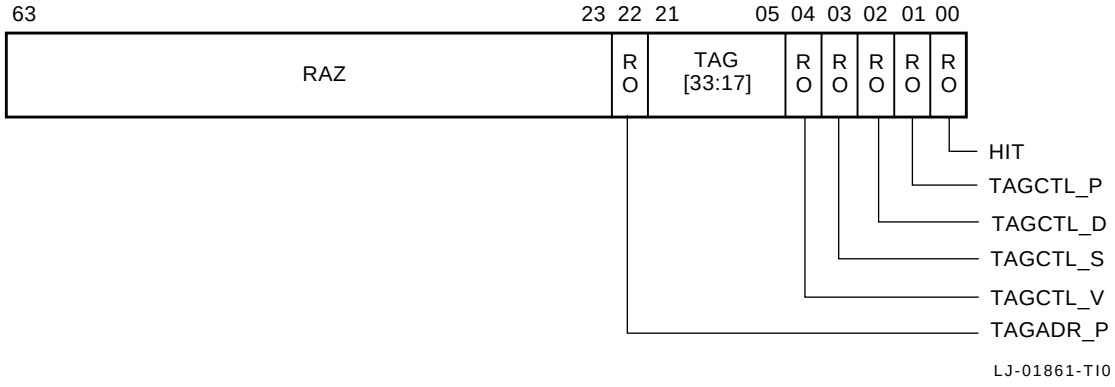
Data Bit	Syndrome (Hex)	Data Bit	Syndrome (Hex)	Check Bit	Syndrome (Hex)
00	4F	16	0E	00	01
01	4A	17	0B	01	02
02	52	18	13	02	04
03	54	19	15	03	08
04	57	20	16	04	10
05	58	21	19	05	20
06	5B	22	1A	06	40
07	5D	23	1C		
08	23	24	62		
09	25	25	64		
10	26	26	67		
11	29	27	68		
12	2A	28	6B		
13	2C	29	6D		
14	31	30	70		
15	34	31	75		

4.2.21 Backup Cache Tag Register (BC_TAG)

The BC_TAG is a read-only register. Unless locked, the BC_TAG register is loaded with the results of every backup cache tag probe. When a tag or tag control parity error or primary fill data error (parity or ECC) occurs, this register is locked against further updates. The software may read the LSB of this register by using the HW_MFPR instruction. Each time an HW_MFPR from BC_TAG completes, the contents of BC_TAG are shifted one bit position to the right, so that the entire register can be read using a sequence of HW_ MFPRs. The software may unlock the BC_TAG register using a HW_MTPR to BC_TAG.

Successive HW_MFPRs from the BC_TAG register must be separated by at least one null cycle. Figure 33 shows the backup cache tag register format. Table 37 lists the register fields and gives a description of each.

Figure 33 Backup Cache Tag Register



Note

Unused tag bits in the TAG field of this register are always clear, based on the size of the external cache as determined by the BC_SIZE field of the BIU_CTL register.

Table 37 Backup Cache Tag Register Fields

Field	Type	Description
TAGADR_P	RO	Reflects the state of the **tagAdrP_h** signal of the 21064A when a tag, tag control, or data parity error occurs.
TAG	RO	Contains the tag that is being currently probed.
TAGCTL_V	RO	Reflects the state of the **tagCtlV_h** signal of the 21064A when a tag, tag control, or data parity error occurs.
TAGCTL_S	RO	Reflects the state of the **tagCtlS_h** signal of the 21064A when a tag, tag control, or parity error occurs.
TAGCTL_D	RO	Reflects the state of the **tagCtlD_h** signal of the 21064A when a tag, tag control, or data parity error occurs.
TAGCTL_P	RO	Reflects the state of the **tagCtlP_h** signal of the 21064A when a tag, tag control, or data parity error occurs.
HIT		When set, indicates that there was a tag match when a tag, tag control, or data parity error occurred.

4.3 PAL_TEMP Registers

The CPU chip contains 32 (64-bit) registers that are accessible by way of the HW_MxPR instructions. These registers provide temporary storage for PALcode.

4.4 Lock Registers

There are two registers per processor that are associated with the LDQ_L/LDL_L and STQ_C/STL_C instructions: the lock_flag register and the locked_physical_address register. The use of these registers is described in the *Alpha Architecture Reference Manual*. These registers are required by the architecture but are *not* implemented on the 21064A. They must be implemented in the application.

4.5 Internal Processor Registers Reset State

Table 38 lists the state of all the internal processor registers (IPRs) immediately following reset. The table also specifies which registers need to be initialized by power-up PALcode.

Table 38 Internal Process Register Reset State

IPR	Reset State	Comments
TB_TAG	UNDEFINED	
ITB_PTE	UNDEFINED	
ICCSR	cleared except ASN, PC0, PC1	Floating-point disabled, single issue mode, Pipe mode enabled, JSR predictions disabled, branch predictions disabled, branch history table disabled, performance counters reset to zero, Perf Cnt0: Total Issues/2, Perf Cnt1: Dcache Misses, superpage disabled
ITB_PTE_TEMP	UNDEFINED	
EXC_ADDR	UNDEFINED	
SL_RCV	UNDEFINED	
ITBZAP	n/a	PALcode must do a ITBZAP on reset before writing the ITB (must do HW_MTPR to ITBZAP register).
ITBASM	n/a	
ITBIS	n/a	
PS	UNDEFINED	PALcode must set processor status.
EXC_SUM	UNDEFINED	PALcode must clear exception summary and exception register write mask by doing 64 reads.
PAL_BASE	cleared	Cleared on reset.
HIRR	n/a	
SIRR	UNDEFINED	PALcode must initialize.
ASTRR	UNDEFINED	PALcode must initialize.
HIER	UNDEFINED	PALcode must initialize.
SIER	UNDEFINED	PALcode must initialize.

(continued on next page)

Table 38 (Cont.) Internal Process Register Reset State

IPR	Reset State	Comments
ASTER	UNDEFINED	PALcode must initialize.
SL_XMIT	UNDEFINED	PALcode must initialize. Appears on external pin.
TB_CTL	UNDEFINED	PALcode must select between SP/LP DTB prior to any TB fill.
DTB_PTE	UNDEFINED	
DTB_PTE_TEMP	UNDEFINED	
MM_CSR	UNDEFINED	Unlocked on reset.
VA	UNDEFINED	Unlocked on reset.
DTBZAP	n/a	PALcode must do a DTBZAP on reset before writing the DTB (must do HW_MTPR to DTBZAP register).
DTBASM	n/a	
DTBIS	n/a	
BIU_ADDR	UNDEFINED	Potentially locked.
BIU_STAT	UNDEFINED	Potentially locked.
SL_CLR	UNDEFINED	PALcode must initialize.
C_STAT	UNDEFINED	Potentially locked.
FILL_ADDR	UNDEFINED	Potentially locked.
ABOX_CTL	cleared	Write buffer enabled, machine checks disabled, correctable read interrupts disabled, Icache stream buffer disabled, super pages 1 and 2 disabled, endian mode disabled, Dcache disabled, forced hit mode off. (STC_NORESULT disabled, NCACHE_NDISTURB disabled)
ALT_MODE	UNDEFINED	
CC	UNDEFINED	Cycle counter is disabled on reset.

(continued on next page)

Table 38 (Cont.) Internal Process Register Reset State

IPR	Reset State	Comments
CC_CTL	UNDEFINED	
BIU_CTL	cleared	Bcache disabled, parity mode enabled, chip enable asserts during RAM write cycles, Bcache forced-hit mode disabled. BC_PA_DIS field cleared. BAD_TCP cleared. BAD_DP cleared. DELAY_WDATA cleared. SYS_WRAP cleared.
FILL_SYNDROME	UNDEFINED	Potentially locked.
BC_TAG	UNDEFINED	Potentially locked.
PAL_TEMP [31:0]	UNDEFINED	

Note

The Bcache parameters listed here are all undetermined on reset and must be initialized in the BIU_CTL register before enabling the Bcache.

- Bcache RAM read speed (BC_RD_SPD)
- Bcache RAM write speed (BC_WR_SPD)
- Bcache delay write data (DELAY_WDATA)
- Bcache write enable control (BC_WE_CTL)
- Bcache size (BC_SIZE)

5 Electrical Characteristics

Table 39 lists the maximum ratings for the 21064A.

Table 39 21064A Maximum Ratings (PRELIMINARY ESTIMATES)

Characteristics	Ratings
Storage temperature	–55°C to 125°C (–67°F to 257°F)
Supply voltage	Vss –0.5 V, Vdd 3.6 V
Junction temperature	15°C to 90°C (59°F to 194°F)
Voltage applied to pins	
3 V tolerant pins	–0.5 V to Vdd + 0.5 V
5 V tolerant pins	–0.5 V to 5.5 V
Case Temperature:	
21064A–200	0°C to 73°C (32°F to 167.4°F)
21064A–233	0°C to 71°C (32°F to 160°F)
21064A–275, **21064A–275–PC**	0°C to 67°C (32°F to 153°F)
21064A–300	0°C to 65°C (32°F to 149.0°F)
Maximum power @Vdd=3.46 V:	
21064A–200	24.0 W
21064A–233	28.0 W
21064A–275, **21064A–275–PC**	33.0 W
21064A–300	36.0 W

Note

See the *Alpha 21064 and Alpha 21064A Microprocessors Hardware Reference Manual* for formulas to calculate peak power and to calculate maximum power at other values of Vdd and clock frequency.

Caution

Stress beyond the absolute maximum ratings can cause permanent damage to the 21064A. Exposure to absolute maximum rating conditions for extended periods of time can affect the 21064A reliability.

5.1 DC Characteristics

The 21064A uses CMOS/TTL voltage levels.

In CMOS mode, the **Vss** pins are connected to 0.0 V and the **Vdd** pins are connected to 3.3 V nominal +/– 5%.

Caution

To prevent damage to the 21064A, it is important that the **Vdd** power supply be stable before any of its input or bidirectional pins are allowed to rise above 4.0 V.

The **vRef** analog input should be connected to a 1.4 V +/–10% reference supply.

The **clkIn_h** and **clkIn_l** are differential signals generated from an external oscillator circuit. The signals can be ac coupled (if Vcc to the oscillator is greater than **Vdd**), with nominal dc bias of **Vdd**/2 set by a high-impedance (greater than 1K ohm) resistive network on the chip. The signals need not be ac coupled if Vdd is used as the Vcc supply to the oscillator.

The 21064A signal input pins are CMOS inputs that use standard TTL levels, set by **vRef**. Table 40 lists the dc input characteristics.

The following signals are sampled before **vRef** is stable. These signals cannot be driven above the power supply.

dcOk_h
tristate_l (3.3 V)
cont_l (3.3 V)
eclOut_h (GND)

The 21064A output pins are 3.3 V CMOS outputs. These output signals can be driven between **Vdd** and **Vss**. Timing is specified to standard TTL levels. Table 40 lists the dc output characteristics.

The bidirectional pins are ordinary 3.3 V CMOS bidirectional pins.

Table 40 DC Input/Output Characteristics

Symbol	Description	Min	Max	Units	Test Conditions
Vdd	Power supply voltage	3.135	3.465	V	–
Vih	High-level input voltage (except dcOk_h and cont_l)	2.0	–	V	–
Vihs	High-level input voltage (static pins dcOk_h and cont_l)	2.7	–	V	–
Vil	Low-level input voltage	–	0.8	V	–
Voh	High-level output voltage Ioh = 100 μA	2.4	–	V	–
Vol	Low-level output voltage Iol = 3.2 mA	–	0.4	V	–
Vdiffc	Differential clock input swing (duty cycle 45–55%)	300 mV	3.0	V	–
Iil	Input leakage current (except eclOut_h)	–100	100	μA	0<Vin<Vdd V
Iel	Input leakage current (eclOut_h)	–150	150	μA	0<Vin<Vdd V
Ioz	Output leakage current (tristate)	–100	100	μA	–
Icin	Clock input leakage	–4	4	mA	0<Vin<3.465 V

Note

Values in this table are valid only for Vref = 1.4 V.

5.2 AC Characteristics

This section contains the ac characteristics for the 21064A. Timing parameters are given for an internal clock speed of 233 MHz.

Reference Supply

The reference supply (**vRef**) is an analog reference voltage used by the 21064A input buffers of all signals, except for the following:

clkIn_h and clkIn_l
testClkIn_h and testClkIn_l
dcOk_h
eclOut_h
tristate_l
cont_l

Upon power-up, **reset_l** cannot be sampled until **vRef** is stable.

Input Clocks

The **clkIn_h** and **clkIn_l** input clocks have differential inputs. Generally, designers apply the standard 2x input clocks to the pin of **clkIn_h** and **clkIn_l**.

The 21064A input clock circuit also allows for applications that require 1x input clocks (233 MHz input clocks on a 233 MHz current implementation). To use the 21064A with 1x input clocks, the designer needs to drive the 1x clock inputs into the clkIn pins and tie **testClkIn_h** and **testClkIn_l** to 11.

Note

Driving a clock into **testClkIn_h** and **testClkIn_l** results in UNPREDICTABLE behavior.

Electrically, the circuitry attached to the **testClkIn_h** and **testClkIn_l** is identical to the circuitry attached to **tristate_l**. The same restrictions for **tristate_l** apply to **testClkIn_h** and **testClkIn_l**. See the *Alpha 21064 and Alpha 21064A Microprocessors Hardware Reference Manual* for these restrictions.

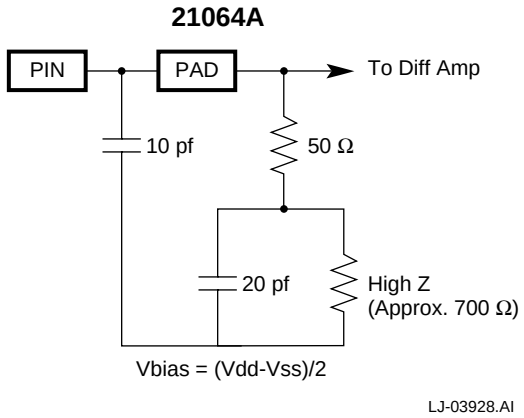
Table 41 lists the possible states of the **testClkIn** pins and the resulting functions.

Table 41 testClkIn Pin States

testClkIn_h	testClkIn_l	Functions
0	0	Reserved for Digital
0	1	Standard 2x input clocks applied to **clkIn** pins
1	0	Standard 2x input clocks applied to **clkIn** pins
1	1	1x input clocks applied to **clkIn** pins

The terminations on these signals (**clkIn** and **testClkIn**) are designed to be compatible with system oscillators of arbitrary dc bias. Figure 34 shows clock termination.

Figure 34 Clock Termination



The timings for all output signals, including clocks (except **cpuClkOut_h**), are specified with respect to their crossings of the midpoint from **Vss** to **Vdd** into a 15 pF lumped capacitive load at the package pin.

Interface Timing

The following sections show the interface timing for the 21064A.

Input Clock Timing

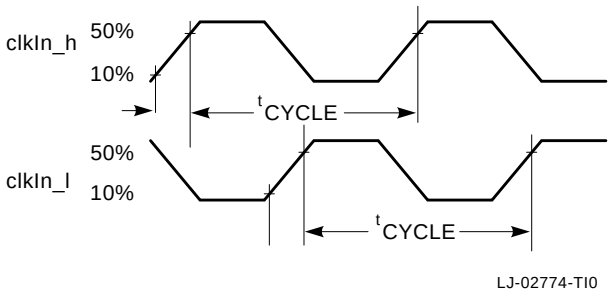
Table 42 lists the input clock cycle times for the two 21064A frequencies. These periods equal one-half the corresponding CPU cycle times.

Table 42 Input Clock Timing

Clock Characteristic	21064A–233	21064A–275
clkIn period minimum	2.15 ns	1.82 ns
clkIn period maximum	15.0 ns	15.0 ns
clkIn symmetry	50% ± 10%	50% ± 10%

Figure 35 shows the timing diagram for the input clock.

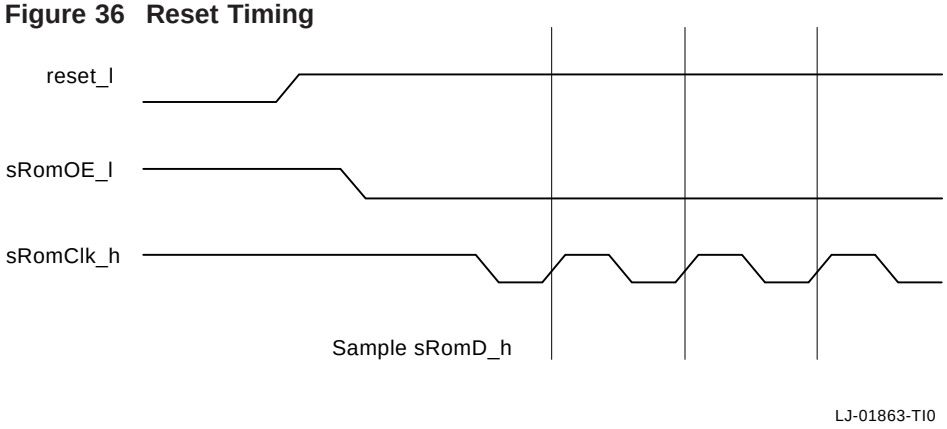
Figure 35 Input Clock Timing Diagram



Reset Timing

Figure 36 shows the SROM timing for the first three bit samples.

Figure 36 Reset Timing

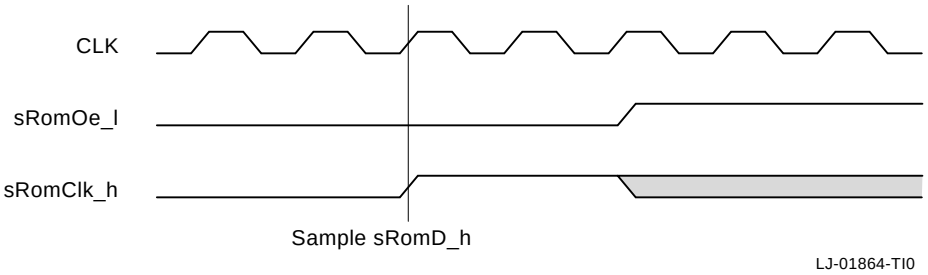


The following list explains the reset timing shown in Figure 4.

1. When **reset_l** is asserted, **sRomOe_l** is deasserted and **sRomClk_h** is asserted.
2. The 21064A internal reset signal remains asserted at least 20 CPU cycles after **reset_l** deasserts, when **sRomOe_l** asserts.
3. The first rising edge of **sRomClk_h** occurs 255 CPU cycles after **sRomOe_l** asserts, and every 254 CPU cycles thereafter.
4. The 21064A samples **sRomD_h** in the last half of each CPU cycle before the rising edge of **sRomClk_h**.

Figure 37 shows the end of the Icache preload sequence. The shaded area indicates UNPREDICTABLE behavior.

Figure 37 Reset Timing—End of Preload Sequence



CLK refers to the 21064A internal CPU clock and is shown as a cycle reference.

1. The 21064A samples the final SROM bit when **sRomClk_h** rises, as shown.
2. Two CPU cycles later, the 21064A deasserts **sRomOe_l** and drives **sRomClk_h** with the value from the TMT bit of the SL_XMIT IPR. Because this bit is not initialized by chip reset, the value driven onto **sRomClk_h** is UNPREDICTABLE.

External Cycle Timing

Table 43 lists the times referenced to **sysClkOut1_h**.

Table 43 External Cycles

Name	Minimum	Maximum	Units
Output enable, sysClkOut1_h to:			
adr_h	–1.0	2.0	ns
data_h (WRITE_BLOCK)	–1.0	2.0	ns
check_h (WRITE_BLOCK)	–1.0	2.0	ns

(continued on next page)

Table 43 (Cont.) External Cycles

Name	Minimum	Maximum	Units
Output delay, sysClkOut1_h to:			
adr_h	–1.0	1.0	ns
data_h (WRITE_BLOCK)	–1.0	1.0	ns
check_h (WRITE_BLOCK)	–1.0	1.0	ns
cReq_h	–1.0	1.0	ns
cWMask_h	–1.0	1.0	ns
holdAck_h	–1.0	1.0	ns
Note: This timing is guaranteed by design.			
Input setup relative to sysClkOut1_h			
cAck_h	7.0	–	ns
dRAck_h	7.0	–	ns
dWSel_h	7.0	–	ns
dOE_l	7.0	–	ns
holdReq_h	3.8	–	ns
dInvReq_h	3.5	–	ns
iAdr_h [12:5]	3.5	–	ns
data_h (READ_BLOCK)	2.5	–	ns
check_h (READ_BLOCK)	2.5	–	ns
perf_cnt_h	3.5	–	ns

(continued on next page)

Table 43 (Cont.) External Cycles

Name	Minimum	Maximum	Units
Input hold relative to sysClkOut1_h			
Input hold relative to sysClkOut1_h			
cAck_h	0	–	ns
dRAck_h	0	–	ns
dWSel_h	0	–	ns
dOE_l	0	–	ns
holdReq_h	0	–	ns
dInvReq_h [1:0]	0	–	ns
iAdr_h	0	–	ns
data_h (READ_BLOCK)	0	–	ns
check_h (READ_BLOCK)	0	–	ns
perf_cnt_h	0	–	ns

Note: This timing is guaranteed by design.

Note

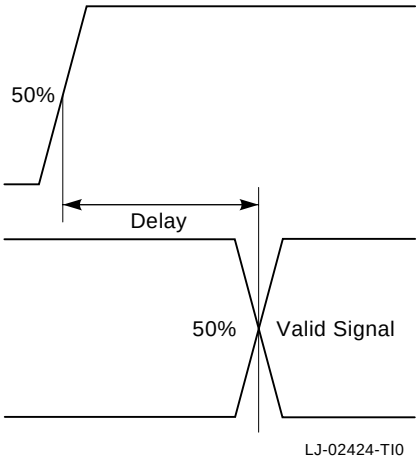
The signals **adr_h [33:5]**, **data_h [127:0]**, and **check_h [27:0]** are only synchronous to **sysClkOut1_h** during an external cycle. During the time that the **cReq_h [2:0]** field is IDLE, the signals can change without regard to the clocks that drive the external system logic. During the time that the field **cReq_h [2:0]** is not IDLE (non-zero), the signals conform to the setup and hold times.

The signals **cReq_h [2:0]**, **holdAck_h**, and **cWMask_h [7:0]** are always synchronous to the external system clocks, even during those times when no external cycle is in progress.

Output Delay Time Measurement

Figure 38 shows the 21064A output delay time measurement.

Figure 38 Output Delay Time Measurement



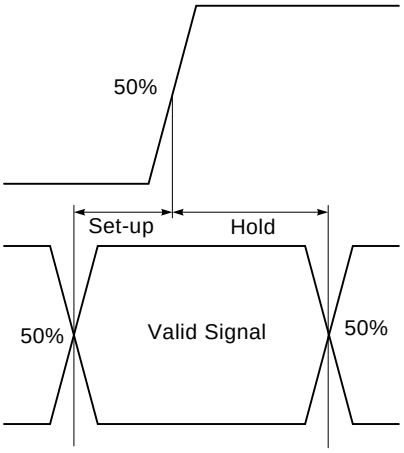
Note

This delay can be positive or negative.

Setup and Hold Time Measurement

Figure 39 shows the 21064A setup and hold time measurement.

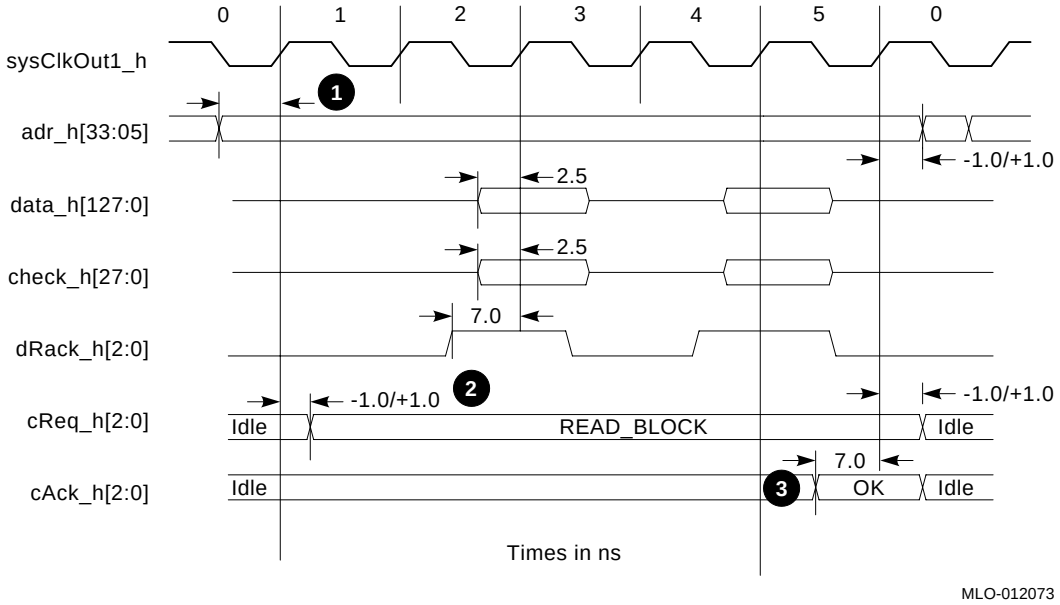
Figure 39 Setup and Hold Time Measurement



READ_BLOCK Timing

Figure 40 shows the 21064A READ_BLOCK timing diagram.

Figure 40 READ_BLOCK Timing Diagram



1 tcycle ± 1.0 ns, where tcycle = period of **cpuClkOut_h**.

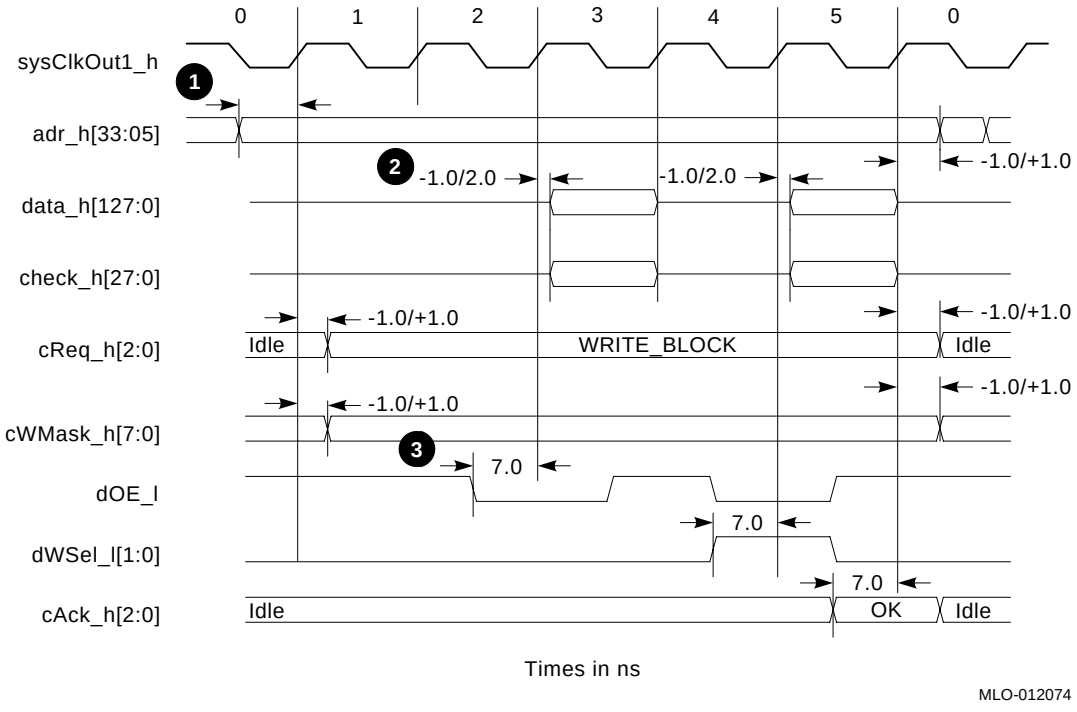
2 Indicates minimum and maximum.

3 Minimum setup time is shown. All hold times are a minimum of 0.0 ns.

WRITE_BLOCK Timing

Figure 41 shows the WRITE_BLOCK timing diagram.

Figure 41 WRITE_BLOCK Timing Diagram



1 3x tcycle ± 1.0 ns, where tcycle = period of **cpuClkOut_h**.

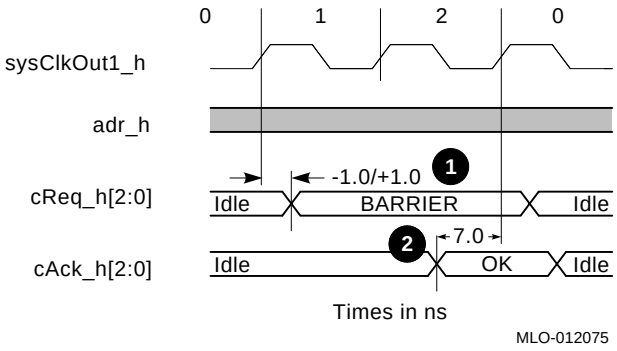
2 Indicates minimum and maximum.

3 Minimum setup time is shown. All hold times are a minimum of 0.0 ns.

BARRIER Timing

Figure 42 shows the BARRIER operation timing.

Figure 42 BARRIER Timing Diagram



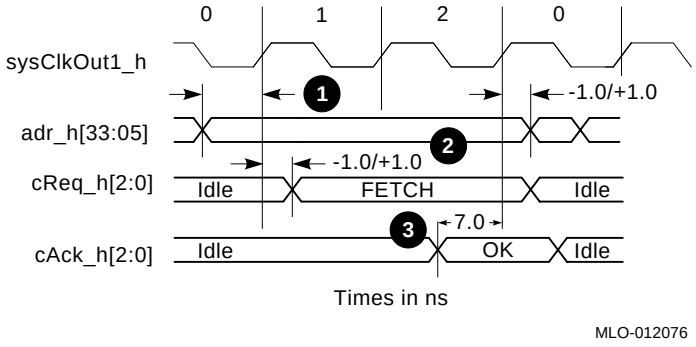
1 Indicates minimum and maximum.

2 Minimum setup time is shown. All hold times are a minimum of 0.0 ns.

FETCH/FETCH_M Timing

Figure 43 shows the FETCH/FETCH_M operation timing.

Figure 43 FETCH/FETCH_M Timing Diagram



1 tcycle ±1.0 ns, where tcycle = period of **cpuClkOut_h**.

2 Indicates minimum and maximum.

3 Minimum setup time is shown. All hold times are a minimum of 0.0 ns.

6 Thermal Considerations

Note

The combination of airflow, heat sink design, and the package thermal characteristics must be considered when calculating the power dissipation, in order not to exceed a maximum junction temperature (Tj) of 90°C (194°F).

The 21064A is packaged in a 431-pin alumina-ceramic (cavity-down) package. This cavity-down design allows the die to be attached to the top surface of the package, which increases the ability of the die to dissipate the heat through the package and attached heat sink surface. A metal slug with two mounting studs is brazed on the ceramic package for the heat sink assembly.

The package has mounting pads for 28 capacitors on the top surface that limits the heat sink contact area. To meet the thermal requirements, the specific dimensions of the heat sink should be determined by the designer.

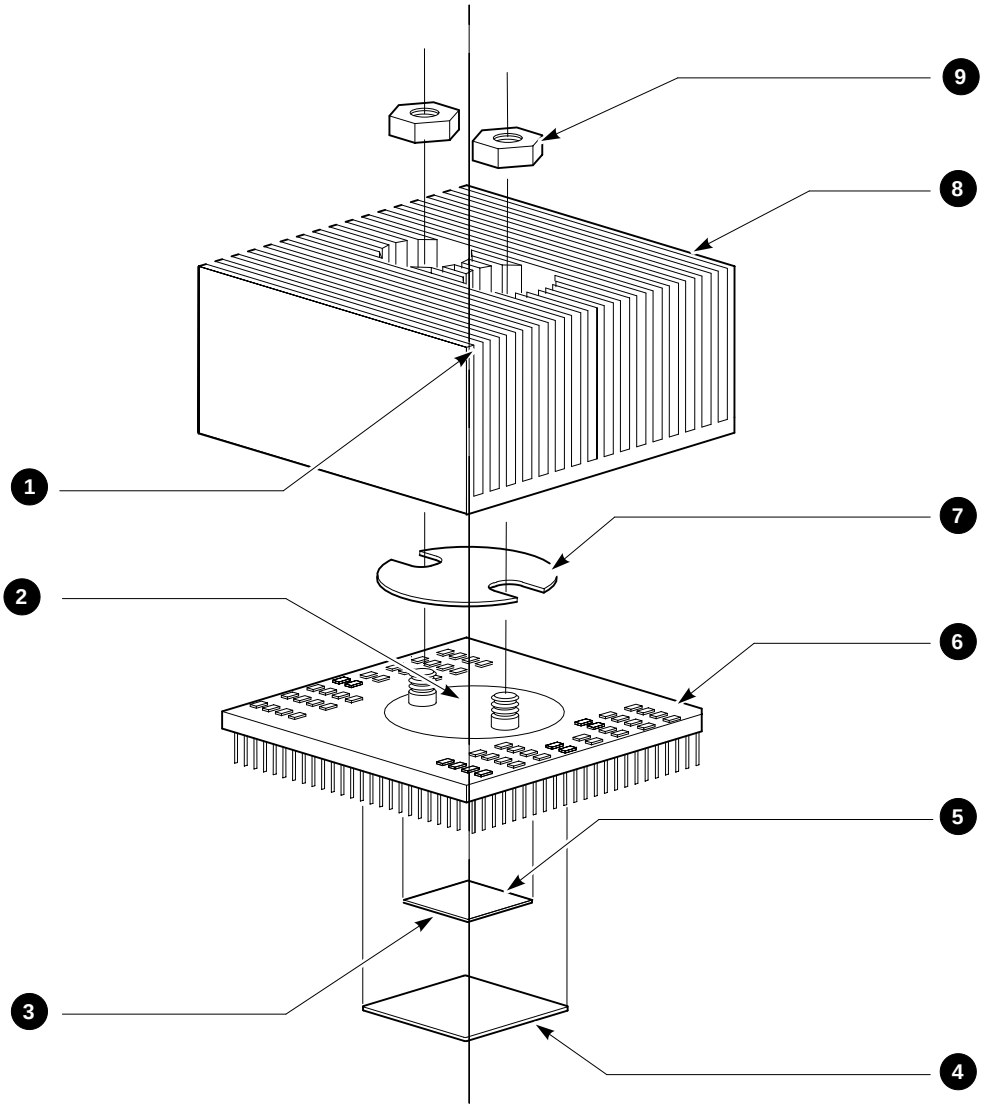
The 21064A has a maximum power rating, which varies depending on the operating frequency. Power dissipation varies directly with the frequency.

Temperature Measurement Locations

The package components and temperature measurement sites are listed here. The locations of the components and sites are indicated in Figure 44.

1. Heat sink temperature (Ths)
2. Case temperature (Tc)
3. Junction temperature (Tj)
4. Package lid
5. Alpha 21064A Microprocessor
6. Package
7. GRAFOIL
8. Heat sink
9. Nut

Figure 44 Package Components and Temperature Measurement Locations



LJ-02416-TI0

6.1 Critical Parameters of Thermal Design

Follow these guidelines for placement of printed-circuit board components:

- Orient the 21064A on the printed-circuit board (PCB) with the heat sink fins aligned with the airflow direction.
- Avoid preheating ambient air. Place the 21064A on the PCB so that inlet air is not preheated by any other PCB components.
- Do not place other high power devices in the vicinity of the 21064A.
- Do not restrict the airflow across the 21064A heat sink. Placement of other devices must allow for maximum system airflow in order to maximize the performance of the heat sink.

Tables 44, 45, 46, and 47 show the thermal characteristics for the 21064A. See Section 7.1 for heat sink information.

Table 44 21064A-200 Thermal Characteristics in a Forced-Air Environment

21064A-200 – Tc= 73.0°C (167.4°F)					
		Heat Sink 1		Heat Sink 2	
Air Velocity	Power	TaMax	θca	TaMax	θca
100 lfpm	24.0 W	43.0°C (109.4°F)	1.25 C/W	33.4°C (92.1°F)	1.65 C/W
200 lfpm	24.0 W	52.6°C (126.7°F)	0.85 C/W	44.2°C (111.6°F)	1.20 C/W
400 lfpm	24.0 W	58.6°C (137.5°F)	0.60 C/W	52.6°C (126.7°F)	0.85 C/W
600 lfpm	24.0 W	60.5°C (140.9°F)	0.52 C/W	57.4°C (135.3°F)	0.65 C/W
1000 lfpm	24.0 W	63.4°C (146.1°F)	0.40 C/W	59.6°C (139.3°F)	0.56 C/W

Table constants and abbreviations

Tj is 90°C (194°F).
θjc is 0.7 C/W.
lfpm is linear feet per minute.

Table 45 21064A-233 Thermal Characteristics in a Forced-Air Environment

21064A-233 – Tc=71.0°C (159.8°F)					
		Heat Sink 1		Heat Sink 2	
Air Velocity	Power	TaMax	θca	TaMax	θca
100 lfpm	28.0 W	36.0°C (96.8°F)	1.25 C/W	24.8°C (76.6°F)	1.65 C/W
200 lfpm	28.0 W	47.2°C (117.0°F)	0.85 C/W	37.4°C (99.3°F)	1.20 C/W
400 lfpm	28.0 W	54.2°C (129.6°F)	0.60 C/W	47.2°C (117.0°F)	0.85 C/W
600 lfpm	28.0 W	56.4°C (133.5°F)	0.52 C/W	52.8°C (127.0°F)	0.65 C/W
1000 lfpm	28.0 W	59.8°C (139.6°F)	0.40 C/W	55.3°C (131.5°F)	0.56 C/W

Table constants and abbreviations

Tj is 90°C (194°F).
θjc is 0.7 C/W.
lfpm is linear feet per minute.

Table 46 21064A-275 and 21064A-275-PC Thermal Characteristics in a Forced-Air Environment

21064A-275 and 21064A-275-PC— Tc=67.0°C (152.6°F)					
		Heat Sink 1		Heat Sink 2	
Air Velocity	Power	TaMax	θc	TaMax	θca
100 lfpm	33.0 W	25.8°C (78.4°F)	1.25 C/W	—	—
200 lfpm	33.0 W	39.0°C (102.2°F)	0.85 C/W	27.4°C (81.3°F)	1.20 C/W
400 lfpm	33.0 W	47.2°C (117.0°F)	0.60 C/W	39.0°C (102.2°F)	0.85 C/W
600 lfpm	33.0 W	49.8°C (121.6°F)	0.52 C/W	45.6°C (114.0°F)	0.65 C/W
1000 lfpm	33.0 W	53.8°C (128.8°F)	0.40 C/W	48.5°C (119.3°F)	0.56 C/W

Table constants and abbreviations

Tj is 90°C (194°F).
θjc is 0.7 C/W.
lfpm is linear feet per minute.

Table 47 21064A-300 Thermal Characteristics in a Forced-Air Environment

21064A-300 — Tc=65.0°C (149.0°F)					
		Heat Sink 1		Heat Sink 2	
Air Velocity	Power	TaMax	θca	TaMax	θca
100 lfpm	36.0 W	20.0°C (68.0°F)	1.25 C/W	—	—
200 lfpm	36.0 W	34.4°C (93.9°F)	0.85 C/W	21.8°C (71.2°F)	1.20 C/W
400 lfpm	36.0 W	43.4°C (110.1°F)	0.60 C/W	34.4°C (93.9°F)	0.85 C/W
600 lfpm	36.0 W	46.3°C (115.3°F)	0.52 C/W	41.6°C (106.9°F)	0.65 C/W
1000 lfpm	36.0 W	50.6°C (123.1°F)	0.40 C/W	44.8°C (112.6°F)	0.56 C/W

Table constants and abbreviations

Tj is 90°C (194°F).
θjc is 0.7 C/W.
lfpm is linear feet per minute.

Note

The values in Tables 44, 45, 46, and 47 are base upon the assumption that maximum power will be 24.0 W, 28.0 W, 33.0 W, 33.0 W, and 36.0 W for the 21064A-200, 21064A-233, 21064A-275, **21064A-275-PC**, and 21064A-300 respectively.

7 Mechanical Specifications

This section provides detailed information about the 21064A package and the complete pinout.

7.1 Package Information

Figure 45 shows two examples of heat sinks which may be used to help cool the 21064A. The primary heat sink (number 2 in Figure 45) is available from Digital.

Figure 45 Heat Sink Dimensions

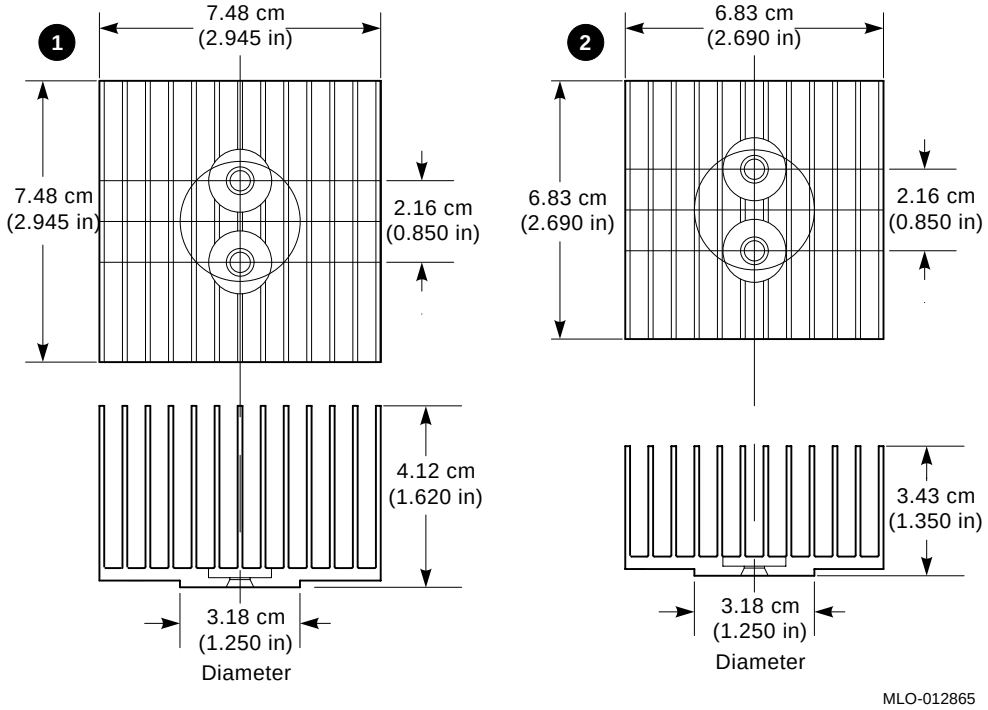
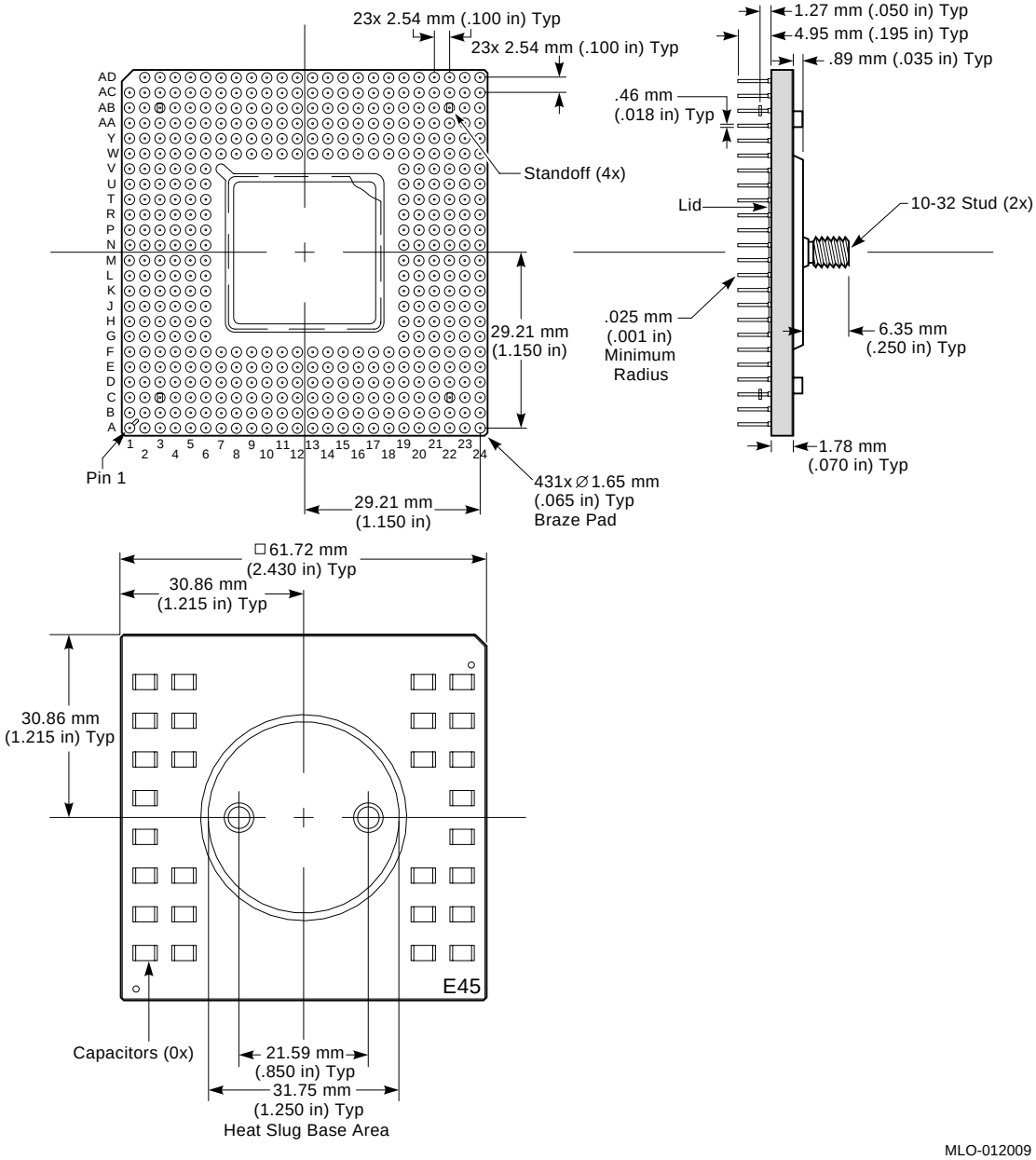


Figure 46 shows the package physical dimensions without a heat sink.

Figure 46 Package Dimensions



MLO-012009

7.2 21064A Pins

Figure 47 shows the 21064A PGA cavity.

Figure 47 PGA Cavity Down View

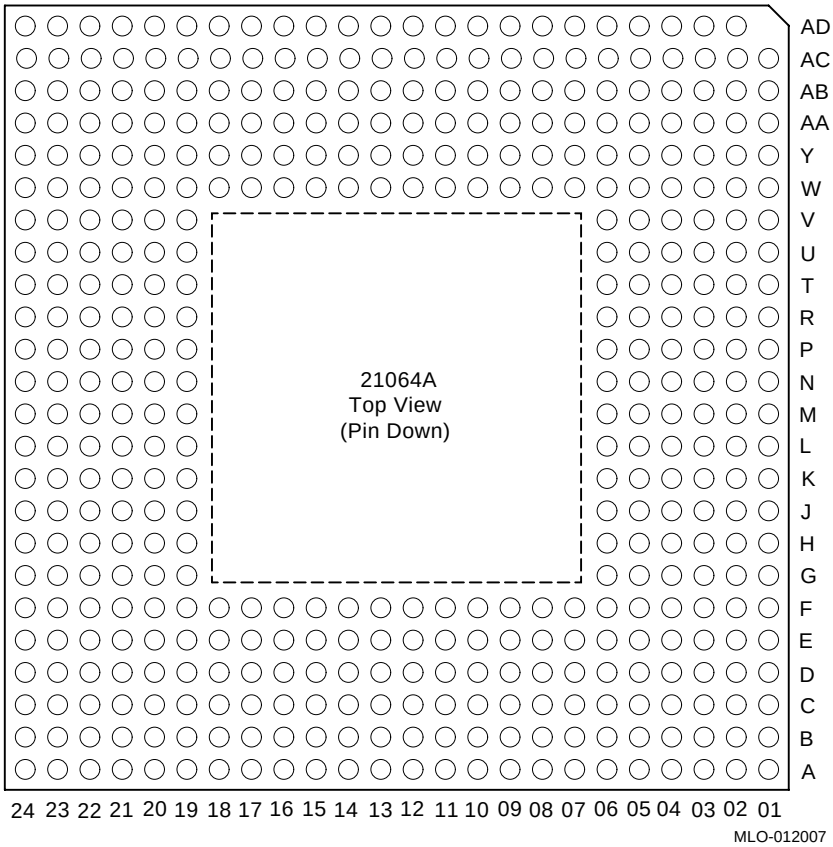


Table 48 lists the order of the 21064A pins by its PGA location.

Table 48 Pin List

PGA Location	Name	PGA Location	Name	PGA Location	Name
A1	data_h 33	B1	check_h 15	C1	check_h 16
A2	data_h 97	B2	VDD plane	C2	VSS plane
A3	data_h 98	B3	data_h 35	C3	data_h 96
A4	data_h 100	B4	VSS plane	C4	data_h 99
A5	data_h 38	B5	data_h 101	C5	data_h 37
A6	check_h 27	B6	VDD plane	C6	check_h 13
A7	data_h 104	B7	data_h 40	C7	data_h 103
A8	data_h 42	B8	VSS plane	C8	data_h 105
A9	data_h 44	B9	data_h 107	C9	data_h 43
A10	data_h 109	B10	VDD plane	C10	data_h 45
A11	data_h 47	B11	data_h 110	C11	data_h 46
A12	data_h 49	B12	VSS plane	C12	data_h 112
A13	data_h 113	B13	data_h 50	C13	data_h 114
A14	data_h 52	B14	VDD plane	C14	data_h 116
A15	check_h 12	B15	check_h 26	C15	data_h 54
A16	data_h 55	B16	VSS plane	C16	data_h 119
A17	data_h 120	B17	data_h 57	C17	data_h 121
A18	data_h 122	B18	VDD plane	C18	check_h 11
A19	check_h 7	B19	check_h 21	C19	data_h 59
A20	data_h 60	B20	VSS plane	C20	data_h 124
A21	data_h 61	B21	data_h 125	C21	data_h 126
A22	data_h 62	B22	VDD plane	C22	check_h 23
A23	data_h 127	B23	VSS plane	C23	dRAck_h 0
A24	check_h 9	B24	check_h 8	C24	dInvReq_h 1

(continued on next page)

Table 48 (Cont.) Pin List

PGA Location	Name	PGA Location	Name	PGA Location	Name
D1	data_h 94	E1	data_h 30	F1	data_h 92
D2	check_h 2	E2	VDD plane	F2	data_h 29
D3	check_h 1	E3	data_h 31	F3	data_h 93
D4	data_h 34	E4	data_h 32	F4	data_h 95
D5	data_h 36	E5	VDD plane	F5	VSS plane
D6	data_h 102	E6	VSS plane	F6	VDD plane
D7	data_h 39	E7	VDD plane	F7	VSS plane
D8	data_h 41	E8	VSS plane	F8	VDD plane
D9	data_h 106	E9	VDD plane	F9	VSS plane
D10	data_h 108	E10	VSS plane	F10	VDD plane
D11	check_h 24	E11	check_h 10	F11	VSS plane
D12	data_h 48	E12	data_h 111	F12	VDD plane
D13	data_h 51	E13	data_h 115	F13	VSS plane
D14	data_h 53	E14	data_h 117	F14	VDD plane
D15	data_h 118	E15	VDD plane	F15	VSS plane
D16	data_h 56	E16	VSS plane	F16	VDD plane
D17	data_h 58	E17	VDD plane	F17	VSS plane
D18	check_h 25	E18	VSS plane	F18	VDD plane
D19	data_h 123	E19	VDD plane	F19	VSS plane
D20	data_h 63	E20	VSS plane	F20	VDD plane
D21	check_h 22	E21	dRAck_h 1	F21	cAck_h 1
D22	dRAck_h 2	E22	dWSel_h 0	F22	cAck_h 2
D23	VDD plane	E23	dWSel_h 1	F23	VSS plane
D24	dOE_l	E24	cAck_h 0	F24	holdReq_h

(continued on next page)

Table 48 (Cont.) Pin List

PGA Location	Name	PGA Location	Name	PGA Location	Name
G1	data_h 27	J1	data_h 89	L1	check_h 19
G2	VSS plane	J2	VDD plane	L2	VSS plane
G3	data_h 91	J3	data_h 26	L3	data_h 22
G4	data_h 28	J4	data_h 90	L4	data_h 86
G5	VDD plane	J5	VDD plane	L5	data_h 23
G6	VSS plane	J6	VSS plane	L6	VSS plane
G19	VDD plane	J19	VDD plane	L19	VDD plane
G20	VSS plane	J20	VSS plane	L20	dataWE_h 0
G21	holdAck_h	J21	cWMask_h 1	L21	dataWE_h 1
G22	dataCEOE_h 0	J22	cWMask_h 2	L22	dataWE_h 2
G23	dataCEOE_h 1	J23	cWMask_h 3	L23	dataWE_h 3
G24	dataCEOE_h 2	J24	cWMask_h 4	L24	dMapWE_h 0
H1	check_h 4	K1	data_h 87	M1	data_h 20
H2	check_h 18	K2	data_h 24	M2	data_h 84
H3	check_h 0	K3	data_h 88	M3	data_h 21
H4	check_h 14	K4	data_h 25	M4	data_h 85
H5	VSS plane	K5	VSS plane	M5	check_h 5
H6	VDD plane	K6	VDD plane	M6	VDD plane
H19	VSS plane	K19	VSS plane	M19	VSS plane
H20	VDD plane	K20	VDD plane	M20	cReq_h 0
H21	dataCEOE_h 3	K21	cWMask_h 5	M21	cReq_h 1
H22	tagCtlWE_h	K22	cWMask_h 6	M22	cReq_h 2
H23	VDD plane	K23	VSS plane	M23	VDD plane
H24	cWMask_h 0	K24	cWMask_h 7	M24	dMapWe_h 1

(continued on next page)

Table 48 (Cont.) Pin List

PGA Location	Name	PGA Location	Name	PGA Location	Name
N1	data_h 83	R1	data_h 15	U1	data_h 76
N2	VDD plane	R2	VSS plane	U2	VDD plane
N3	data_h 19	R3	data_h 78	U3	data_h 12
N4	data_h 82	R4	data_h 14	U4	data_h 75
N5	data_h 18	R5	VDD plane	U5	VDD plane
N6	VSS plane	R6	VSS plane	U6	VSS plane
N19	VDD plane	R19	VDD plane	U19	VDD plane
N20	tagOk_l	R20	VSS plane	U20	VSS plane
N21	tagOk_h	R21	tagadr_h 19	U21	tagadr_h 26
N22	dataA_h 4	R22	tagadr_h 18	U22	tagadr_h 25
N23	dataA_h 3	R23	lockFlag_h	U23	tagadr_h 24
N24	tagCEOE_h	R24	tagCtlV_h	U24	tagadr_h 23
P1	data_h 81	T1	check_h 17	V1	data_h 11
P2	data_h 17	T2	check_h 3	V2	data_h 74
P3	data_h 80	T3	data_h 77	V3	data_h 10
P4	data_h 16	T4	data_h 13	V4	data_h 73
P5	data_h 79	T5	VSS plane	V5	VSS plane
P6	VDD plane	T6	VDD plane	V6	VDD plane
P19	VSS plane	T19	VSS plane	V19	VSS plane
P20	tagCtlS_h	T20	VDD plane	V20	VDD plane
P21	tagCtlD_h	T21	tagadr_h 22	V21	tagadr_h 29
P22	tagCtlP_h	T22	tagadr_h 21	V22	tagadr_h 28
P23	VSS plane	T23	VDD plane	V23	VSS plane
P24	lockWE_h	T24	tagadr_h 20	V24	tagadr_h 27

(continued on next page)

Table 48 (Cont.) Pin List

PGA Location	Name	PGA Location	Name	PGA Location	Name
W1	data_h 9	Y1	data_h 8	AA1	check_h 20
W2	VSS plane	Y2	data_h 71	AA2	VDD plane
W3	data_h 72	Y3	data_h 7	AA3	data_h 5
W4	check_h 6	Y4	data_h 68	AA4	data_h 66
W5	VDD plane	Y5	VSS plane	AA5	data_h 0
W6	VSS plane	Y6	VDD plane	AA6	iAdr_h 6
W7	VDD plane	Y7	VSS plane	AA7	iAdr_h 10
W8	VSS plane	Y8	VDD plane	AA8	vRef
W9	testClkIn_h	Y9	VSS plane	AA9	sysClkOut2_h
W10	testClkIn_l	Y10	VDD plane	AA10	sysClkOut2_l
W11	VDD plane	Y11	VSS plane	AA11	resetSClk_h
W12	clkIn_h	Y12	VDD plane	AA12	sysClkOut1_h
W13	clkIn_l	Y13	VSS plane	AA13	sysClkOut1_l
W14	VSS plane	Y14	VDD plane	AA14	cont_l
W15	VDD plane	Y15	VSS plane	AA15	irq_h 5
W16	VSS plane	Y16	VDD plane	AA16	sysClkDiv_h
W17	VDD plane	Y17	VSS plane	AA17	adr_h 31
W18	VSS plane	Y18	VDD plane	AA18	adr_h 27
W19	VDD plane	Y19	VSS plane	AA19	adr_h 24
W20	VSS plane	Y20	VDD plane	AA20	adr_h 17
W21	tagadrP_h	Y21	adr_h 8	AA21	adr_h 15
W22	tagadr_h 32	Y22	adr_h 5	AA22	adr_h 11
W23	tagadr_h 31	Y23	VDD plane	AA23	adr_h 7
W24	tagadr_h 30	Y24	tagadr_h 33	AA24	adr_h 6

(continued on next page)

Table 48 (Cont.) Pin List

PGA Location	Name	PGA Location	Name	PGA Location	Name
AB1	data_h 70	AC1	data_h 6	AD2	data_h 4
AB2	data_h 69	AC2	VSS plane	AD3	data_h 3
AB3	data_h 67	AC3	VDD plane	AD4	data_h 1
AB4	data_h 2	AC4	data_h 65	AD5	iAdr_h 5
AB5	data_h 64	AC5	VSS plane	AD6	iAdr_h 9
AB6	iAdr_h 7	AC6	iAdr_h 8	AD7	icMode_h 2
AB7	iAdr_h 12	AC7	VDD plane	AD8	eclOut_h
AB8	reset_l	AC8	iAdr_h 11	AD9	dInvReq_h 0
AB9	sRomD_h	AC9	VSS plane	AD10	spare 2
AB10	sRomOE_l	AC10	sRomClk_h	AD11	spare 4
AB11	cpuClkOut_h	AC11	VDD plane	AD12	icMode_h 1
AB12	dcOk_h	AC12	spare 5	AD13	irq_h 0
AB13	triState_l	AC13	VSS plane	AD14	irq_h 1
AB14	icMode_h 0	AC14	irq_h 2	AD15	irq_h 3
AB15	irq_h 4	AC15	VDD plane	AD16	Digital Reserved
AB16	perf_cnt_h 0	AC16	perf_cnt_h 1	AD17	adr_h 33
AB17	adr_h 32	AC17	VSS plane	AD18	adr_h 30
AB18	adr_h 28	AC18	adr_h 29	AD19	adr_h 26
AB19	adr_h 25	AC19	VDD plane	AD20	adr_h 23
AB20	adr_h 21	AC20	adr_h 22	AD21	adr_h 20
AB21	adr_h 18	AC21	VSS plane	AD22	adr_h 19
AB22	adr_h 14	AC22	adr_h 16	AD23	adr_h 13
AB23	VSS plane	AC23	VDD plane	AD24	adr_h 12
AB24	adr_h 9	AC24	adr_h 10	—	—

7.3 Signal Pin Lists

Tables 49 through 64 list the 21064A pinout. The order of the pinout is by signal name. The following list describes the abbreviations used in the Type column of the pin lists.

- B = Bidirectional
- I = Input
- N = Not connected
- P = Power or ground
- O = Output

Table 49 Data Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
data_h 127	A23	B	data_h 63	D20	B
data_h 126	C21	B	data_h 62	A22	B
data_h 125	B21	B	data_h 61	A21	B
data_h 124	C20	B	data_h 60	A20	B
data_h 123	D19	B	data_h 59	C19	B
data_h 122	A18	B	data_h 58	D17	B
data_h 121	C17	B	data_h 57	B17	B
data_h 120	A17	B	data_h 56	D16	B
data_h 119	C16	B	data_h 55	A16	B
data_h 118	D15	B	data_h 54	C15	B
data_h 117	E14	B	data_h 53	D14	B
data_h 116	C14	B	data_h 52	A14	B
data_h 115	E13	B	data_h 51	D13	B
data_h 114	C13	B	data_h 50	B13	B
data_h 113	A13	B	data_h 49	A12	B
data_h 112	C12	B	data_h 48	D12	B
data_h 111	E12	B	data_h 47	A11	B
data_h 110	B11	B	data_h 46	C11	B
data_h 109	A10	B	data_h 45	C10	B
data_h 108	D10	B	data_h 44	A9	B
data_h 107	B9	B	data_h 43	C9	B
data_h 106	D9	B	data_h 42	A8	B
data_h 105	C8	B	data_h 41	D8	B
data_h 104	A7	B	data_h 40	B7	B
data_h 103	C7	B	data_h 39	D7	B
data_h 102	D6	B	data_h 38	A5	B
data_h 101	B5	B	data_h 37	C5	B
data_h 100	A4	B	data_h 36	D5	B

(continued on next page)

Table 49 (Cont.) Data Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
data_h 99	C4	B	data_h 35	B3	B
data_h 98	A3	B	data_h 34	D4	B
data_h 97	A2	B	data_h 33	A1	B
data_h 96	C3	B	data_h 32	E4	B
data_h 95	F4	B	data_h 31	E3	B
data_h 94	D1	B	data_h 30	E1	B
data_h 93	F3	B	data_h 29	F2	B
data_h 92	F1	B	data_h 28	G4	B
data_h 91	G3	B	data_h 27	G1	B
data_h 90	J4	B	data_h 26	J3	B
data_h 89	J1	B	data_h 25	K4	B
data_h 88	K3	B	data_h 24	K2	B
data_h 87	K1	B	data_h 23	L5	B
data_h 86	L4	B	data_h 22	L3	B
data_h 85	M4	B	data_h 21	M3	B
data_h 84	M2	B	data_h 20	M1	B
data_h 83	N1	B	data_h 19	N3	B
data_h 82	N4	B	data_h 18	N5	B
data_h 81	P1	B	data_h 17	P2	B
data_h 80	P3	B	data_h 16	P4	B
data_h 79	P5	B	data_h 15	R1	B
data_h 78	R3	B	data_h 14	R4	B
data_h 77	T3	B	data_h 13	T4	B
data_h 76	U1	B	data_h 12	U3	B
data_h 75	U4	B	data_h 11	V1	B
data_h 74	V2	B	data_h 10	V3	B
data_h 73	V4	B	data_h 9	W1	B
data_h 72	W3	B	data_h 8	Y1	B

(continued on next page)

Table 49 (Cont.) Data Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
data_h 71	Y2	B	data_h 7	Y3	B
data_h 70	AB1	B	data_h 6	AC1	B
data_h 69	AB2	B	data_h 5	AA3	B
data_h 68	Y4	B	data_h 4	AD2	B
data_h 67	AB3	B	data_h 3	AD3	B
data_h 66	AA4	B	data_h 2	AB4	B
data_h 65	AC4	B	data_h 1	AD4	B
data_h 64	AB5	B	data_h 0	AA5	B

Table 50 Address Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
adr_h 33	AD17	B	adr_h 18	AB21	B
adr_h 32	AB17	B	adr_h 17	AA20	B
adr_h 31	AA17	B	adr_h 16	AC22	B
adr_h 30	AD18	B	adr_h 15	AA21	B
adr_h 29	AC18	B	adr_h 14	AB22	B
adr_h 28	AB18	B	adr_h 13	AD23	B
adr_h 27	AA18	B	adr_h 12	AD24	B
adr_h 26	AD19	B	adr_h 11	AA22	B
adr_h 25	AB19	B	adr_h 10	AC24	B
adr_h 24	AA19	B	adr_h 9	AB24	B
adr_h 23	AD20	B	adr_h 8	Y21	B
adr_h 22	AC20	B	adr_h 7	AA23	B
adr_h 21	AB20	B	adr_h 6	AA24	B
adr_h 20	AD21	B	adr_h 5	Y22	B
adr_h 19	AD22	B	–	–	–

Table 51 Parity/ECC Bus Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
check_h 27	A6	B	check_h 13	C6	B
check_h 26	B15	B	check_h 12	A15	B
check_h 25	D18	B	check_h 11	C18	B
check_h 24	D11	B	check_h 10	E11	B
check_h 23	C22	B	check_h 9	A24	B
check_h 22	D21	B	check_h 8	B24	B
check_h 21	B19	B	check_h 7	A19	B
check_h 20	AA1	B	check_h 6	W4	B
check_h 19	L1	B	check_h 5	M5	B
check_h 18	H2	B	check_h 4	H1	B
check_h 17	T1	B	check_h 3	T2	B
check_h 16	C1	B	check_h 2	D2	B
check_h 15	B1	B	check_h 1	D3	B
check_h 14	H4	B	check_h 0	H3	B

Table 52 Primary Cache Invalidate Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
iAdr_h 12	AB7	I	iAdr_h 8	AC6	I
iAdr_h 11	AC8	I	iAdr_h 7	AB6	I
iAdr_h 10	AA7	I	iAdr_h 6	AA6	I
iAdr_h 9	AD6	I	iAdr_h 5	AD5	I
dInvReq_h 0	AD9	I	dInvReq_h 1	C24	I

Table 53 External Cache Control Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
tagCEOE_h	N24	O	tagCtlWE_h	H22	O
tagCtlV_h	R24	B	tagCtlD_h	P21	B
tagCtlS_h	P20	B	tagCtlP_h	P22	B
tagadr_h 33	Y24	I	tagadr_h 25	U22	I
tagadr_h 32	W22	I	tagadr_h 24	U23	I
tagadr_h 31	W23	I	tagadr_h 23	U24	I
tagadr_h 30	W24	I	tagadr_h 22	T21	I
tagadr_h 29	V21	I	tagadr_h 21	T22	I
tagadr_h 28	V22	I	tagadr_h 20	T24	I
tagadr_h 27	V24	I	tagadr_h 19	R21	I
tagadr_h 26	U21	I	tagadr_h 18	R22	I
tagadrP_h	W21	I	–	–	–
tagOk_h	N21	I	tagOk_l	N20	I
dataCEOE_h 3	H21	O	dataCEOE_h 1	G23	O
dataCEOE_h 2	G24	O	dataCEOE_h 0	G22	O
dataWE_h 3	L23	O	dataWE_h 1	L21	O
dataWE_h 2	L22	O	dataWE_h 0	L20	O
dataA_h 4	N22	O	dataA_h 3	N23	O
holdReq_h	F24	I	holdAck_h	G21	O
dMapWE_h 0	L24	O	dOE_l	D24	I
dMapWE_h 1	M24	O	–	–	–
dWSel_h 1	E23	I	dWSel_h 0	E22	I
dRAck_h 2	D22	I	dRAck_h 0	C23	I
dRAck_h 1	E21	I	–	–	–
cReq_h 2	M22	O	cReq_h 0	M20	O

(continued on next page)

Table 53 (Cont.) External Cache Control Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
cReq_h 1	M21	O	–	–	–
cWMask_h 7	K24	O	cWMask_h 3	J23	O
cWMask_h 6	K22	O	cWMask_h 2	J22	O
cWMask_h 5	K21	O	cWMask_h 1	J21	O
cWMask_h 4	J24	O	cWMask_h 0	H24	O
cAck_h 2	F22	I	cAck_h 0	E24	I
cAck_h 1	F21	I	–	–	–

Table 54 Interrupt Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
irq_h 5	AA15	I	irq_h 2	AC14	I
irq_h 4	AB15	I	irq_h 1	AD14	I
irq_h 3	AD15	I	irq_h 0	AD13	I

Table 55 Instruction Cache Initialization Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
icMode_h 1	AD12	I	icMode_h 0	AB14	I
icMode_h 2	AD7	I	–	–	–

Table 56 Serial ROM Interface Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
sRomOE_l	AB10	O	sRomClk_h	AC10	O
sRomD_h	AB9	I	–	–	–

Table 57 Initialization Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
dcOk_h	AB12	I	reset_l	AB8	I
resetSClk_h	AA11	I	–	–	–

Table 58 Load/Lock and Store/Conditional Fast Lock Mode Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
lockFlag_h	R23	O	lockWE_h	P24	O

Table 59 Clock Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
clkIn_h	W12	I	clkIn_l	W13	I
testClkIn_h	W9	I	testClkIn_l	W10	I
cpuClkOut_h	AB11	O	sysClkDiv_h	AA16	I
sysClkOut1_h	AA12	O	sysClkOut1_l	AA13	O
sysClkOut2_h	AA9	O	sysClkOut2_l	AA10	O

Table 60 Performance Monitoring Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
perf_cnt_h 1	AC16	I	perf_cnt_h 0	AB16	I

Table 61 Other Signals Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
triState_l	AB13	I	vRef	AA8	I
cont_l	AA14	I	eclOut_h	AD8	I

Table 62 Power Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
Vdd plane	B2	P	Vdd plane	N2	P
Vdd plane	B6	P	Vdd plane	N19	P
Vdd plane	B10	P	Vdd plane	P6	P
Vdd plane	B14	P	Vdd plane	R5	P
Vdd plane	B18	P	Vdd plane	R19	P
Vdd plane	B22	P	Vdd plane	T6	P
Vdd plane	D23	P	Vdd plane	T20	P
Vdd plane	E2	P	Vdd plane	T23	P
Vdd plane	E5	P	Vdd plane	U2	P
Vdd plane	E7	P	Vdd plane	U5	P
Vdd plane	E9	P	Vdd plane	U19	P
Vdd plane	E15	P	Vdd plane	V6	P
Vdd plane	E17	P	Vdd plane	V20	P
Vdd plane	E19	P	Vdd plane	W5	P
Vdd plane	F6	P	Vdd plane	W7	P
Vdd plane	F8	P	Vdd plane	W11	P
Vdd plane	F10	P	Vdd plane	W15	P
Vdd plane	F12	P	Vdd plane	W17	P
Vdd plane	F14	P	Vdd plane	W19	P
Vdd plane	F16	P	Vdd plane	Y6	P
Vdd plane	F18	P	Vdd plane	Y8	P
Vdd plane	F20	P	Vdd plane	Y10	P
Vdd plane	G5	P	Vdd plane	Y12	P
Vdd plane	G19	P	Vdd plane	Y14	P
Vdd plane	H6	P	Vdd plane	Y16	P
Vdd plane	H20	P	Vdd plane	Y18	P
Vdd plane	H23	P	Vdd plane	Y20	P

(continued on next page)

Table 62 (Cont.) Power Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
Vdd plane	J2	P	Vdd plane	Y23	P
Vdd plane	J5	P	Vdd plane	AA2	P
Vdd plane	J19	P	Vdd plane	AC3	P
Vdd plane	K6	P	Vdd plane	AC7	P
Vdd plane	K20	P	Vdd plane	AC11	P
Vdd plane	L19	P	Vdd plane	AC15	P
Vdd plane	M6	P	Vdd plane	AC19	P
Vdd plane	M23	P	Vdd plane	AC23	P

Table 63 Ground Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
Vss plane	B4	P	Vss plane	N6	P
Vss plane	B8	P	Vss plane	P19	P
Vss plane	B12	P	Vss plane	P23	P
Vss plane	B16	P	Vss plane	R2	P
Vss plane	B20	P	Vss plane	R6	P
Vss plane	B23	P	Vss plane	R20	P
Vss plane	C2	P	Vss plane	T5	P
Vss plane	E6	P	Vss plane	T19	P
Vss plane	E8	P	Vss plane	U6	P
Vss plane	E10	P	Vss plane	U20	P
Vss plane	E16	P	Vss plane	V5	P
Vss plane	E18	P	Vss plane	V19	P
Vss plane	E20	P	Vss plane	V23	P
Vss plane	F5	P	Vss plane	W2	P
Vss plane	F7	P	Vss plane	W6	P
Vss plane	F9	P	Vss plane	W8	P

(continued on next page)

Table 63 (Cont.) Ground Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
Vss plane	F11	P	Vss plane	W14	P
Vss plane	F13	P	Vss plane	W16	P
Vss plane	F15	P	Vss plane	W18	P
Vss plane	F17	P	Vss plane	W20	p
Vss plane	F19	P	Vss plane	Y5	P
Vss plane	F23	P	Vss plane	Y7	P
Vss plane	G2	P	Vss plane	Y9	P
Vss plane	G6	P	Vss plane	Y11	P
Vss plane	G20	P	Vss plane	Y13	P
Vss plane	H5	P	Vss plane	Y15	P
Vss plane	H19	P	Vss plane	Y17	P
Vss plane	J6	P	Vss plane	Y19	P
Vss plane	J20	P	Vss plane	AB23	P
Vss plane	K5	P	Vss plane	AC2	P
Vss plane	K19	P	Vss plane	AC5	P
Vss plane	K23	P	Vss plane	AC9	P
Vss plane	L2	P	Vss plane	AC13	P
Vss plane	L6	P	Vss plane	AC17	P
Vss plane	M19	P	Vss plane	AC21	P

Table 64 Spare Pin List

Signal	PGA Location	Type	Signal	PGA Location	Type
spare	AD16	N	spare	AD10	N
spare	AC12	N	spare	AD11	N

Technical Support and Ordering Information

Technical Support

If you need technical support or help deciding which literature best meets your needs, call the Digital Semiconductor Information Line:

United States and Canada	**1–800–332–2717**
Outside North America	**+1–508–628–4760**

Ordering Digital Semiconductor Products

To order the Alpha 21064A microprocessors and related products, contact your local distributor.

You can order the following semiconductor products from Digital:

Product	Order Number
Alpha 21064A–200 Microprocessor	21064–AB
Alpha 21064A–233 Microprocessor	21064–BB
Alpha 21064A–275 Microprocessor	21064–DB
Alpha 21064A–275-PC Microprocessor	21064–P1
Alpha 21064A–300 Microprocessor	21064–EB
AlphaPC64 Evaluation Board 275-MHz Kit	21A02–03
AlphaPC64 Evaluation Board Design Kit	21A02–13
Alpha Evaluation Board Software Developer's Kit	21B02–02
DECchip 21064 Evaluation Board Design Package	21A01–13
Heat Sink Assembly	2106H–AA

Ordering AlphaPC64 Boards

To order an AlphaPC64 board, contact your local distributor.

Board Product	Order Number
AlphaPC64 Board[1]	21A02–A3
AlphaPC64 Board[2](2MB Level 2 Cache)	21A02–A4
AlphaPC64 Board[2](512KB Level 2 Cache)	21A02–A5

[1]Alpha 21064A microprocessors, main memory, and level 2 cache must be purchased separately.

[2]Alpha 21064A microprocessors and main memory must be purchased separately.

Ordering Associated Literature

The following table lists some of the available Digital Semiconductor literature. For a complete list, contact the Digital Semiconductor Information Line.

Title	Order Number
Alpha 21064 and Alpha 21064A Microprocessors Hardware Reference Manual	EC–Q9ZUA–TE
PALcode for Alpha Microprocessors System Design Guide	EC–QFGLB–TE
Designing a Memory/Cache Subsystem for the DECchip 21064 Microprocessor: An Application Note	EC–N0301–72
Designing a System with the DECchip 21064 Microprocessor: An Application Note	EC–N0107–72
Calculating a System I/O Address for the DECchip 21064 Evaluation Board: An Application Note	EC–N0567–72
DECchip 21064 Bus Transactor User's Guide	EC–N0448–72
Alpha Microprocessors Evaluation Board Debug Monitor User's Guide	EC–QHUVB–TE
AlphaPC64 Evaluation Board User's Guide	EC–QGY2C–TE
AlphaPC64 Evaluation Board Read Me First	EC–QGY3C–TE
DECchip 21064 Evaluation Board Product Brief	EC–N0353–72
Alpha Microprocessors SROM Mini-Debugger User's Guide	EC–QHUXA–TE
Alpha Microprocessors Evaluation Board Software Design Tools User's Guide	EC–QHUWA–TE