DECchip 21064A–233, –275 Alpha AXP Microprocessor

Data Sheet

Order Number: EC–N0699–72

Revision/Update Information: This is a preliminary document.

Digital Equipment Corporation
Maynard, Massachusetts

January 1994

While Digital believes the information included in this publication is correct as of the date of publication, it is subject to change without notice.

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Printed in U.S.A.

This document was prepared using VAX DOCUMENT Version 2.1.

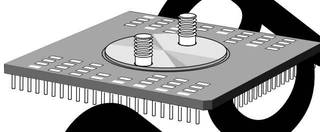
digital

January 1994

DECchip 21064A-233, -275 Alpha AXP Microprocessor Data Sheet

21064A Features:

- 64-Bit RISC microprocessor implements the Alpha AXP architecture
- Super-scalar, 233-MHz, 275-MHz
- Dual instruction issue that yields a peak instruction execution rate of 466 MIPS or 550 MIPS
- Super pipelined
- Two on-chip caches (with parity)
  - 16-Kbyte instruction cache
  - 16-Kbyte data cache
- 32 Integer registers and 32 floating-point registers (64-bit)
- External data path selectable for 128 bits or 64 bits
- Backward compatible with DECchip 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.

Technical Support and Information

North America:	**1-800-332-2717**
TTY:	**1-800-332-2515**
Outside North America:	**+1-508-568-6868**

Chip Ordering Information

North America:	**1-800-344-4825 (1-800-DIGITAL)**
Outside North America:	**Contact your local Digital sales office**

Contents

Figures

Tables

1 Overview

This document describes the DECchip 21064A Alpha AXP Microprocessor (21064A).

The 21064A differs from the 21064 in the following major ways:

- Processor clock has a higher frequency (233 MHz and 275 MHz)
- On-chip instruction cache (Icache) is larger (16K bytes)
  - Icache has data and tag parity protection
- On-chip data cache (Dcache) is larger (16K bytes)
  - Dcache has data and tag parity protection
- Wider choice of system clock frequency
  - System clock frequency is the processor clock frequency divided by any value from **2 to 17**.
- Higher performance floating-point divider
- Branch prediction history table is larger
  - 4K x 2-bit branch prediction history table
- Byte parity mode is now available for the external data bus
- Improved performance for some systems using the fast lock mode with LDx/L and STx/C instructions
- Improved IEEE divide trap handling ability
- Backup cache (Bcache) size of 128K byte is not supported

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 instruction cache (Icache) and data cache (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

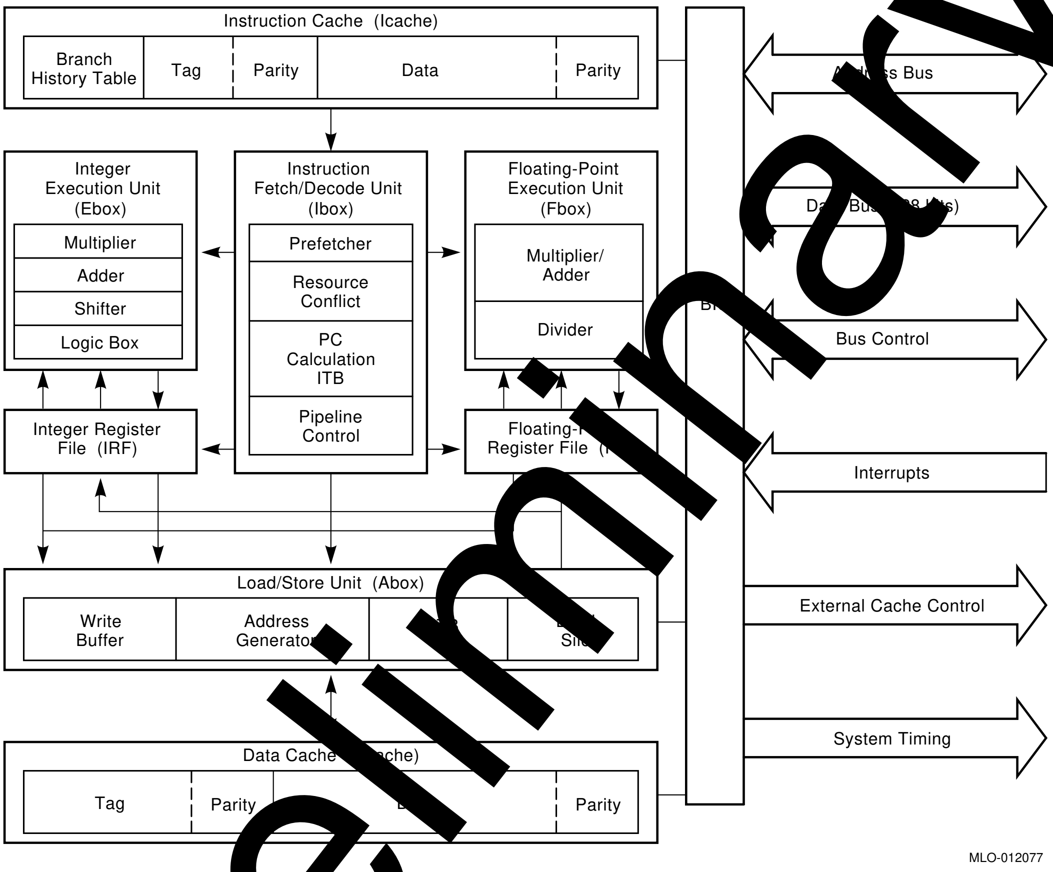
The DECchip 21064A is backward compatible with the DECchip 21064. The compatibility includes pin layout, PALcode, and application programs.

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

- The DECchip 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 DECchip 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 DECchip 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 DECchip 21064 modules.

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

Figure 1 Block Diagram of the DECchip 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 DECchip 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

<table>
<tr><th>Signal</th><th>Type</th><th>Count</th><th>Function</th></tr>
<tr><td>tagCEOE_h</td><td>O</td><td>1</td><td>Controls tag and tag control RAM chip enable or output enable during the 21064A controlled external cache accesses.</td></tr>
<tr><td>tagCtlWE_h</td><td>O</td><td>1</td><td>Controls tag control RAM write enable during the 21064A controlled transactions.</td></tr>
<tr><td>tagCtlV_h,
tagCtlS_h,
tagCtlD_h</td><td>B</td><td>3</td><td>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.
<table>
<tr><th>tagCtlV_h</th><th>tagCtlS_h</th><th>tagCtlD_h</th><th>Meaning</th></tr>
<tr><td>L</td><td>X</td><td>X</td><td>Invalid</td></tr>
<tr><td>H</td><td>L</td><td>L</td><td>Valid, private</td></tr>
<tr><td>H</td><td>L</td><td>H</td><td>Valid, private, dirty</td></tr>
<tr><td>H</td><td>H</td><td>L</td><td>Valid, shared</td></tr>
<tr><td>H</td><td>H</td><td>H</td><td>Valid, shared, dirty</td></tr>
</table></td></tr>
<tr><td>tagCtlP_h</td><td>B</td><td>1</td><td>Carries parity across tagCtlV_h, tagCtlD_h, and tagCtlS_h.</td></tr>
<tr><td>tagAdr_h [33:18]</td><td>I</td><td>16</td><td>Carries the address contents of the tagAdr RAM to the 21064A address comparator and parity checker.</td></tr>
<tr><td>tagAdrP_h</td><td>I</td><td>1</td><td>Carries the parity contents of the tagAdr RAM to the 21064A address comparator and parity checker.</td></tr>
<tr><td>tagOk_h,
tagOk_l</td><td>I</td><td>2</td><td>Bus interface control signals that allow external logic to stall a CPU-controlled access to the external cache RAMs at the last possible moment.</td></tr>
</table>

(continued on next page)

Table 3 (Cont.) External Cache Control Signals

Signal	Type	Count	Function
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 DECchip 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

Signal	Type	Count	Function
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 DECchip 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 DECchip 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 DECchip 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 DECchip 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 AXP 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 $\geq$ zero
BGT	Bra	3F	Branch if $>$ zero
BIC	Opr	11.08	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 $\leq$ zero
BLT	Bra	3A	Branch if $<$ zero
BNE	Bra	3D	Branch if $\neq$ 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 $\geq$ 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 $\leq$ zero
CMOVLT	Opr	11.44	CMOVE if $<$ zero
CMOVNE	Opr	11.26	CMOVE if $\neq$ 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 ≥ zero
FBGT	Bra	37	Floating branch if > zero
FBLE	Bra	33	Floating branch if ≤ zero
FBLT	Bra	32	Floating branch if < zero
FBNE	Bra	35	Floating branch if ≠ zero
FCMOVEQ	F-P	17.02A	FCMOVE if = zero
FCMOVGE	F-P	17.02D	FCMOVE if ≥ zero
FCMOVGT	F-P	17.02F	FCMOVE if > zero
FCMOVLE	F-P	17.02E	FCMOVE if ≤ zero
FCMOVLT	F-P	17.02C	FCMOVE if < zero
FCMOVNE	F-P	17.02B	FCMOVE if ≠ 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 AXP 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 AXP 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 AXP 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 AXP memory management.

4 Electrical Characteristics

Table 20 lists the maximum ratings for the 21064A.

Table 20 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
Input or output applied	3.3 V to 5.5 V
Junction temperature	15°C to 90°C (59°F to 194°F)
Case Temperature:	
21064A–233	0°C to 71°C (32°F to 160°F)
21064A–275	0°C to 69°C (32°F to 152°F)
Maximum power @Vdd=3.46 V:	
21064A–233	28.0 W
21064A–275	33 W

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.

4.1 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 21 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 21 lists the dc output characteristics.

The bidirectional pins are ordinary 3.3 V CMOS bidirectional pins.

Table 21 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, cont_l	2.7	–	V	–
Vil	Low-level input voltage (except dcOk_h, cont_l)	–	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%)	600 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	–0.5<Vin<3.465 V

NOTE

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

4.2 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 *DECchip 21064 Microprocessor Hardware Reference Manual* for these restrictions.

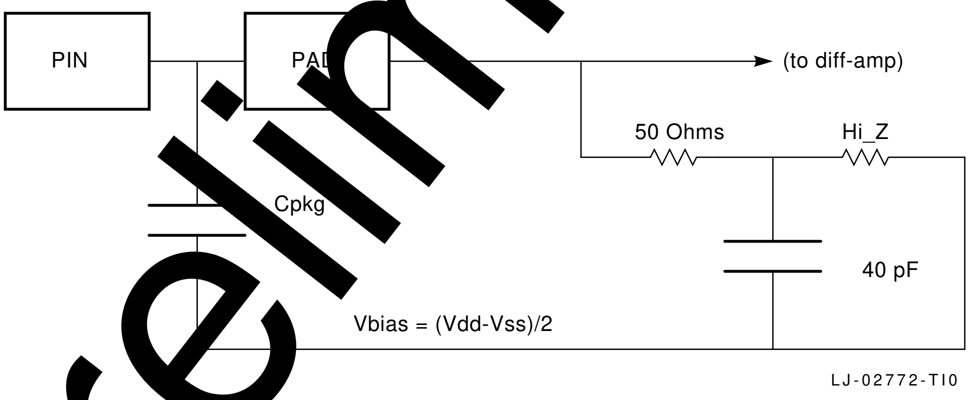
Table 22 lists the possible states of the **testClkIn** pins and the resulting functions.

Table 22 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 2 shows clock termination.

Figure 2 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

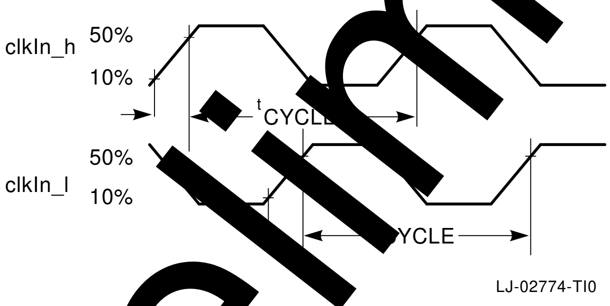
Table 23 lists the input clock cycle times for the two 21064A frequencies. These periods equal one-half the corresponding CPU cycle times.

Table 23 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 3 shows the timing diagram for the input clock.

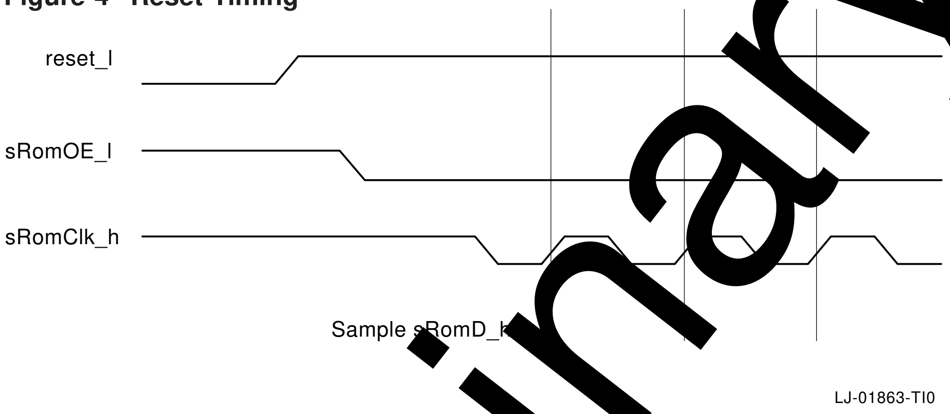
Figure 3 Input Clock Timing Diagram



Reset Timing

Figure 4 shows the SROM timing for the first three bit samples.

Figure 4 Reset Timing

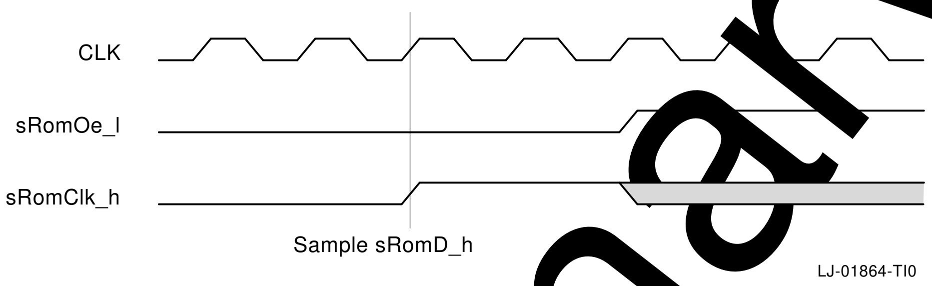


The following list explains the reset timing shown in Figure 4.

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

Figure 5 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 24 lists the times referenced to **sysClkOut1_h**.

Table 24 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 24 (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 24 (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

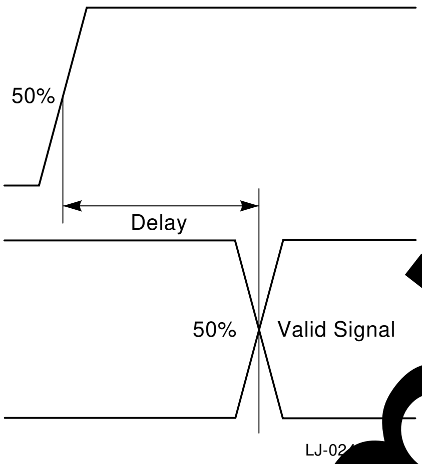
The signals **adr_h [33:5]**, **data_h [127:0]**, and **check_h [27:0]** are only synchronous to **sysClkOut1_h** during an external cycle. During the time that the **cReq_h [2:0]** field is IDLE, the signals can change without regard to the clocks that drive the external system logic. During the time 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 6 shows the 21064A output delay time measurement.

Figure 6 Output Delay Time Measurement



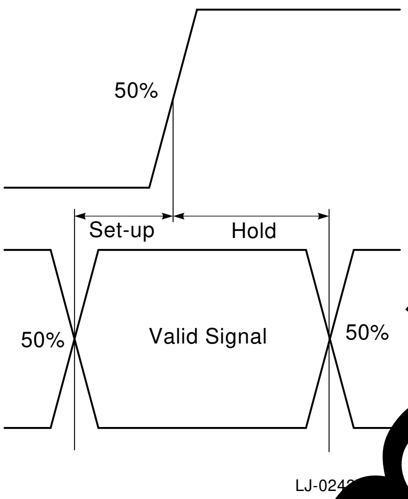
Note

This delay can be positive or negative.

Setup and Hold Time Measurement

Figure 7 shows the 21064A setup and hold time measurement.

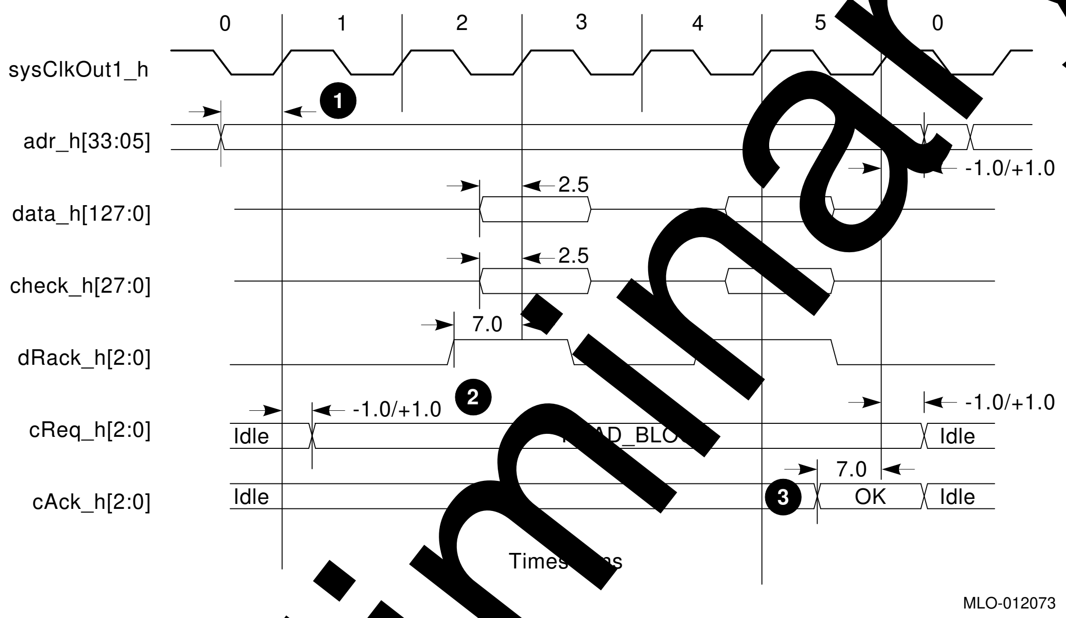
Figure 7 Setup and Hold Time Measurement



READ_BLOCK Timing

Figure 8 shows the 21064A READ_BLOCK timing diagram.

Figure 8 READ_BLOCK Timing Diagram



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

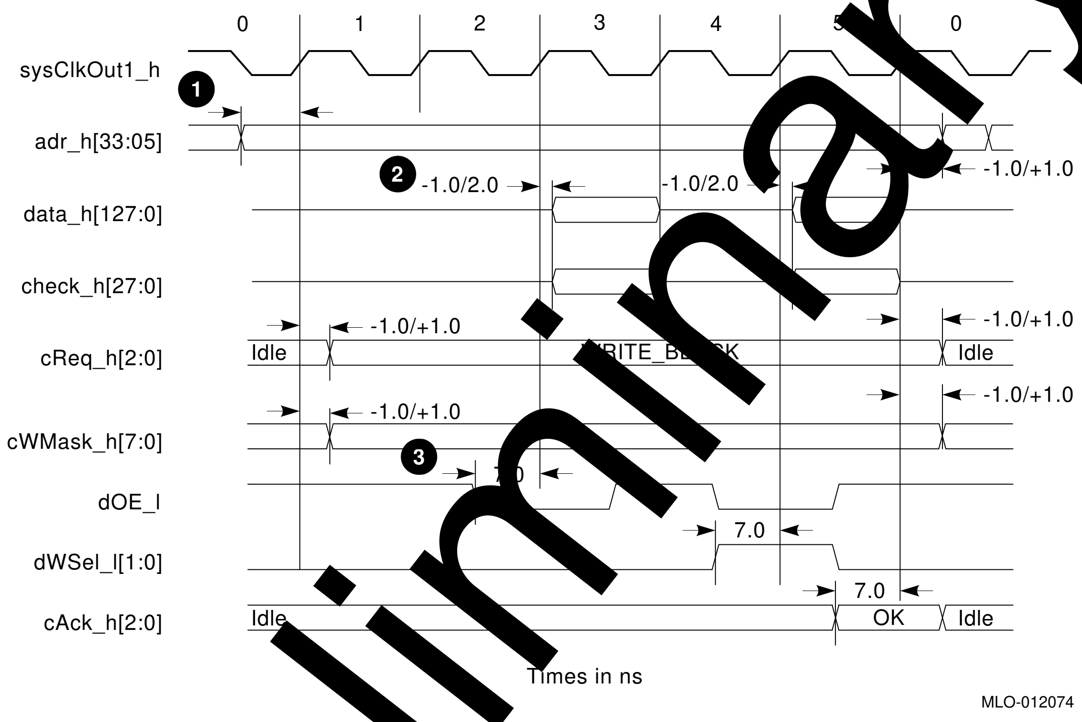
❷ Indicates minimum and maximum.

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

WRITE_BLOCK Timing

Figure 9 shows the WRITE_BLOCK timing diagram.

Figure 9 WRITE_BLOCK Timing Diagram



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

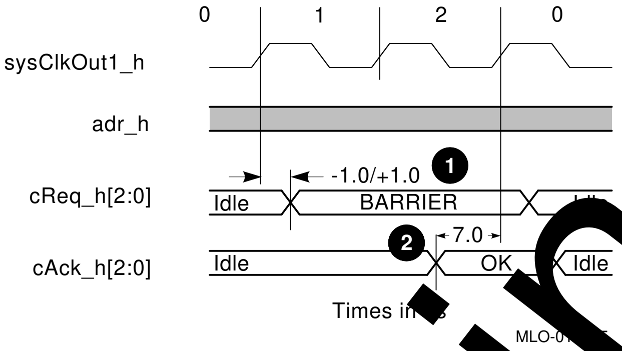
❷ Indicates minimum and maximum.

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

BARRIER Timing

Figure 10 shows the BARRIER operation timing.

Figure 10 BARRIER Timing Diagram



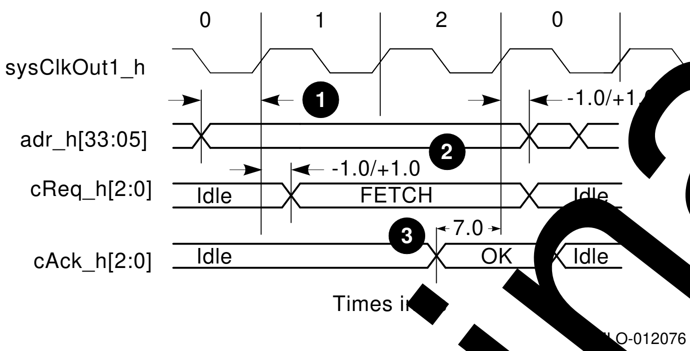
❶ Indicates minimum and maximum.

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

FETCH/FETCH_M Timing

Figure 11 shows the FETCH/FETCH_M operation timing.

Figure 11 FETCH/FETCH_M Timing Diagram



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

❷ Indicates minimum and maximum.

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

5 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.

5.1 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.

5.1.1 Temperature Measurement Locations

The package components and temperature measurement sites are listed here. The locations of the components and sites are indicated in Figure 12.

❶ Heat sink temperature (Ths)

❷ Case temperature (Tc)

❸ Junction temperature (Tj)

❹ Package lid

❺ DECchip 21064A Microprocessor

❻ Package

❼ Grafoil

❽ Heat sink

❾ Nut

Figure 12 Package Components and Temperature Measurement Locations

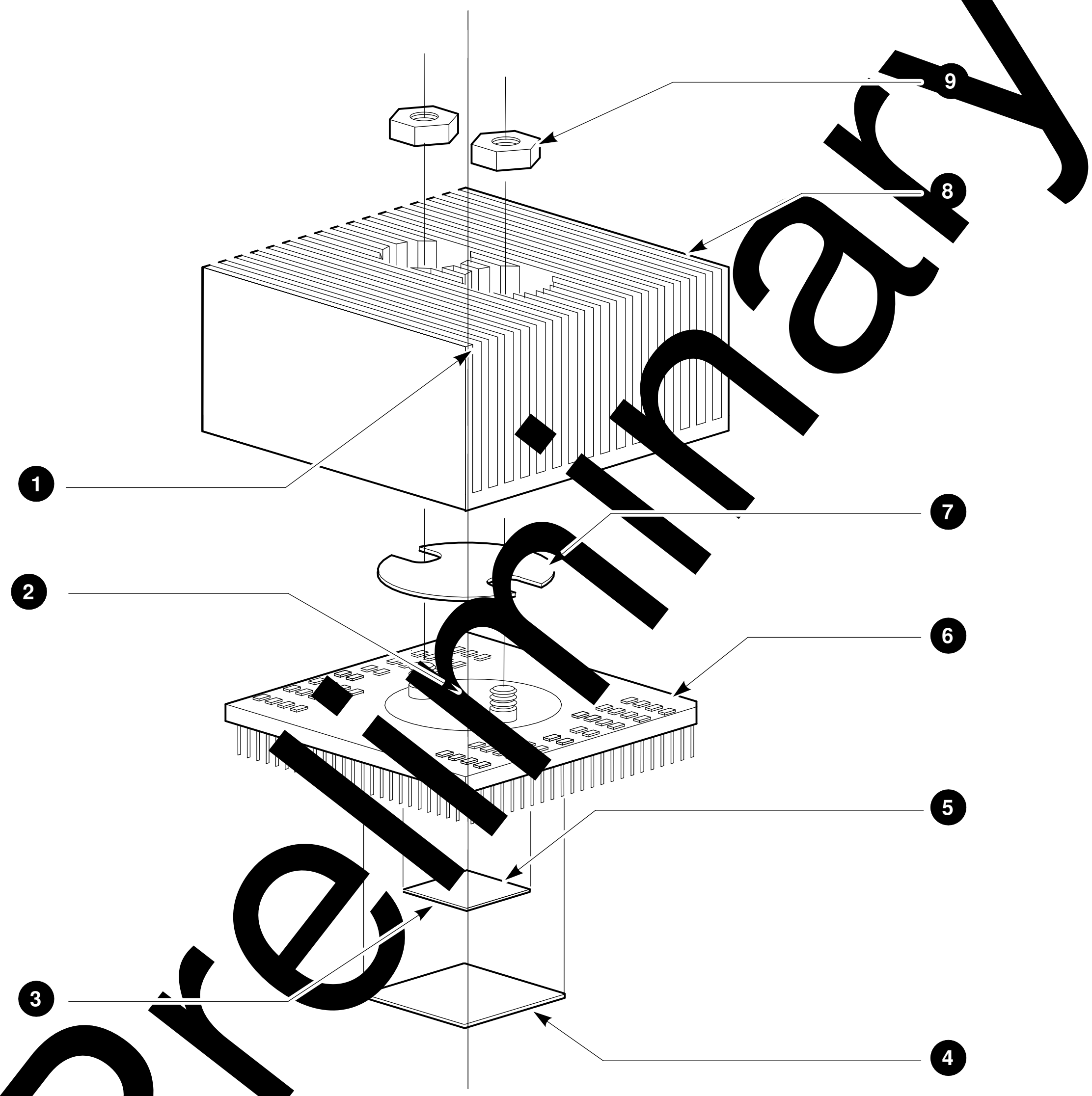


Table 25 lists the thermal characteristics for the 21064A.

Table 25 Thermal Characteristics and Parameters with Heat Sink in a Forced-Air Environment (PRELIMINARY ESTIMATES)

	Defined Parameters						Calculated
Air Velocity (Linear Feet Per Minute)	**21064A**	**Power**	**Tj**	**θ jc**	**Measured θ ja**	**θ ca**	**Maximum Ambient (Ta)**
100 lfpm	–233	28.0 W	90°C (194°F)	0.65 C/W	2.30 C/W	1.65 C/W	25.6°C (78.1°F)
	–275	33.0 W	90°C (194° F)	–	–	–	14.0°C (57.4°F)
200 lfpm	–233	28.0 W	90°C (194°F)	0.65 C/W	1.85 C/W	1.2 C/W	38.2°C (100.8°F)
	–275	33.0 W	90°C (194°F)	–	–	–	29°C (84.1°F)
400 lfpm	–233	28.0 W	90°C (194°F)	0.7 C/W	1.5 C/W	0.8 C/W	48.0°C (118.4°F)
	–275	33.0 W	90°C (194°F)	–	–	–	40.5°C (104.9°F)
600 lfpm	–233	28.0 W	90°C (194°F)	0.7 C/W	1.35 C/W	0.65 C/W	52.2°C (126.0°F)
	–275	33.0 W	90°C (194°F)	–	–	–	45.5°C (113.8°F)
800 lfpm	–233	28.0 W	90°C (194°F)	0.7 C/W	1.28 C/W	0.58 C/W	54.2°C (129.5°F)
	–275	33.0 W	90°C (194°F)	–	–	–	48°C (118°F)
1000 lfpm	–233	28.0 W	90°C (194°F)	0.7 C/W	1.26 C/W	0.56 C/W	54.7°C (130.5°F)
	–275	33.0 W	90°C (194°F)	–	–	–	48.5°C (119.2°F)

6 Mechanical Specifications

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

6.1 Package Information

Figure 13 shows the package physical dimensions of a heat sink (available from Digital).

Figure 13 Heat Sink Dimensions

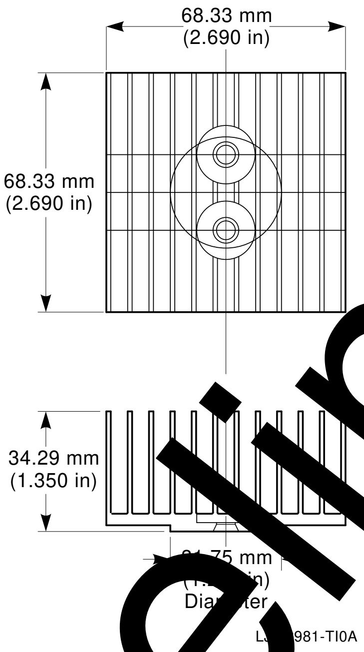
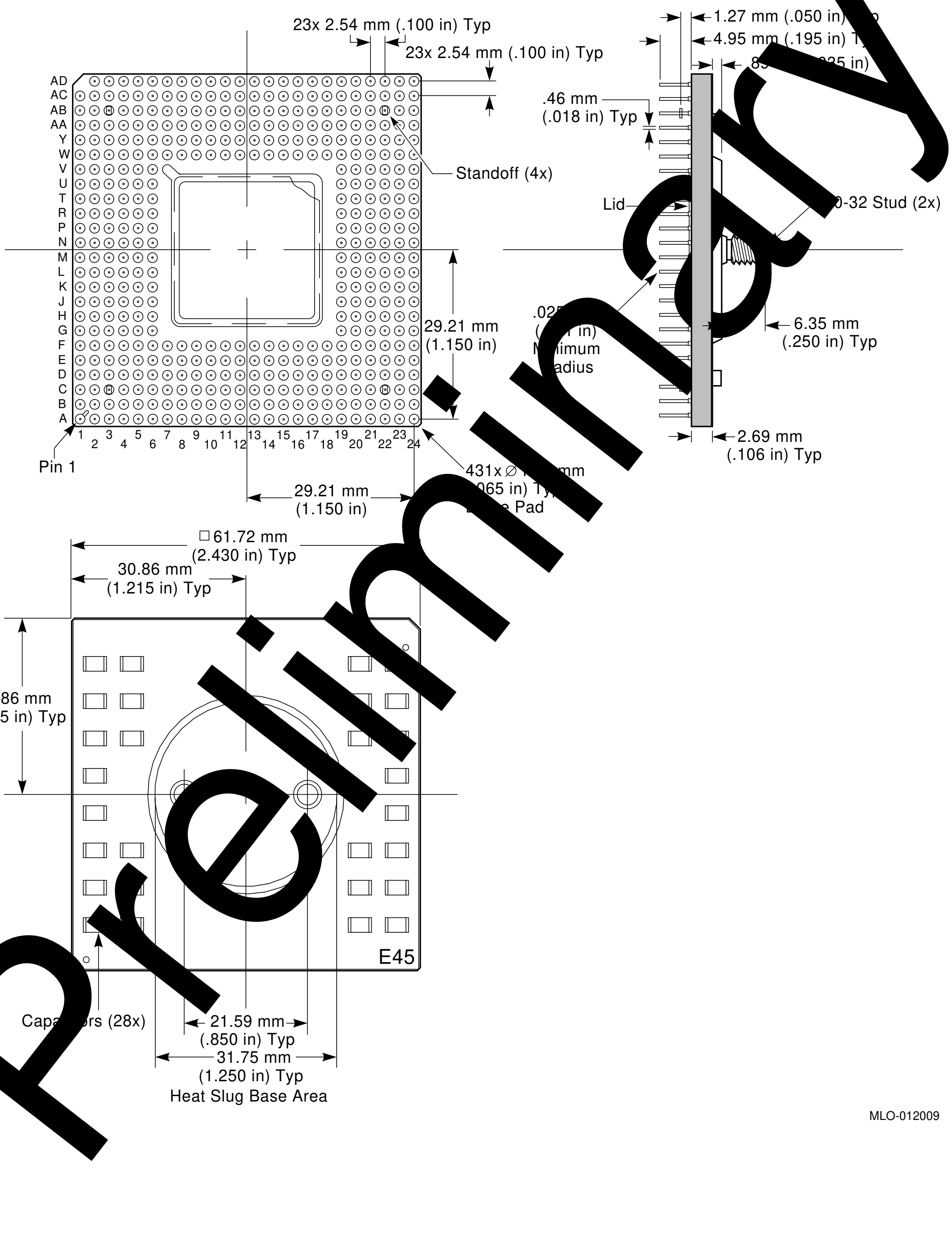


Figure 14 shows the package physical dimensions without a heat sink.

Figure 14 Package Dimensions



6.2 21064A Pins

Figure 15 shows the 21064A PGA cavity.

Figure 15 PGA Cavity Down View

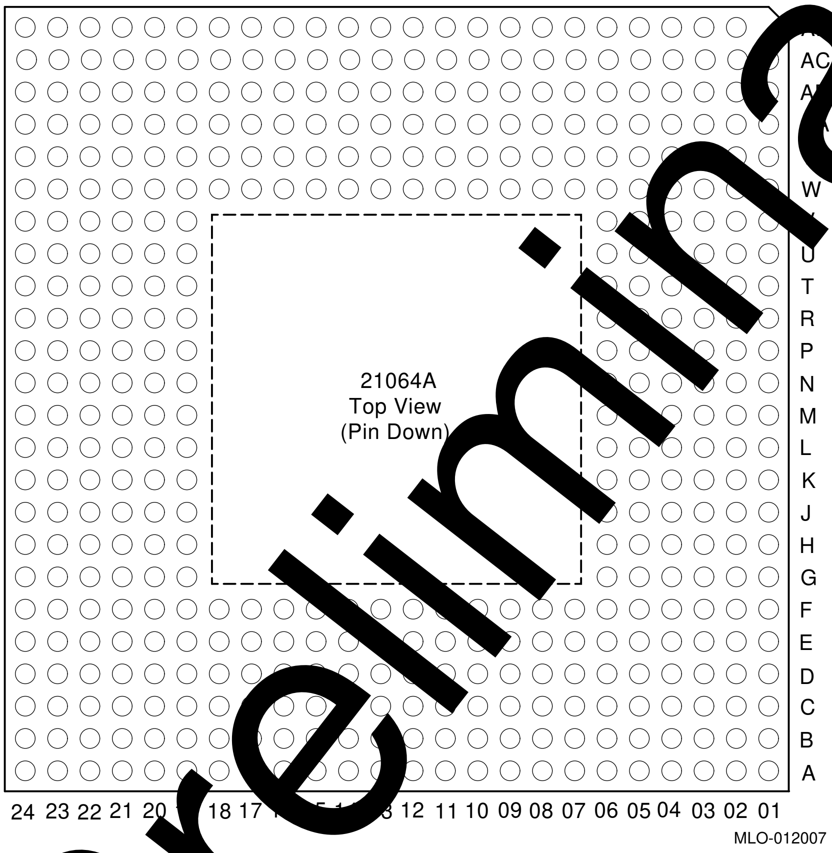


Table 26 lists the order of the 21064A pins by its PGA location.

Table 26 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 26 (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 26 (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 26 (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 26 (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 26 (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	—	—

6.3 Signal Pin Lists

Tables 27 through 42 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 27 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 27 (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 27 (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 28   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 29 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 30 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 31 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 31 (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 32 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 33 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 34 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 35 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 36 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	R24	O

Table 37 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 38 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 39 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 40 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 40 (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 41 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 41 (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 42 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, Ordering, and Associated Literature

This section explains how to:

- Call for DECchip information and technical support.
- Order DECchip products.
- Order associated literature for the DECchip 21064A Microprocessor.

Information and Technical Support

Call the following phone numbers for information and technical support:

United States and Canada	**1–800–332–2717 (1–800–DEC–2717)**
TTY (United States only)	**1–800–332–2515 (1–800–DEC–2515)**
Outside North America	**+1–508–568–6868**

Ordering DECchip Products

To order the DECchip 21064/21064A microprocessors and evaluation boards, contact your local Digital sales office. Working with your sales representative, you may be able to take advantage of discounts and volume pricing.

You can order the following DECchip products from Digital.

Product	Order Number
DECchip 21064A–233 Microprocessor	21064–BB
DECchip 21064A–275 Microprocessor	21064–DB
DECchip 21064–150 Microprocessor	21064–AA
DECchip 21064–166 Microprocessor	21064–CA
DECchip 21064–200 Microprocessor	21064–BA
DECchip 21064 Evaluation Board 150 MHz Kit (Supports OSF/1 and Windows NT operating systems.)	21A01–03
DECchip 21064 Evaluation Board Design Package	21A01–13
Heat Sink Assembly	2106H–AA

Ordering DECchip Sample Kits

To order a DECchip 21064 Sample Kit, which contains one DECchip 21064, one heat sink, and supporting documentation, call **1–800–DIGITAL**. You will need a purchase order number or credit card to order the following DECchip products.

Product	Order Number
DECchip 21064–150 Sample Kit	21064–SA
DECchip 21064–166 Sample Kit	21064–SC
DECchip 21064–200 Sample Kit	21064–SB

Associated DECchip Literature

The following table lists DECchip literature in these categories:

- DECchip 21064/21064A Literature
- DECchip 21064 Evaluation Board Literature
- DECchip Marketing Literature

Title	Order Number
DECchip 21064/21064A Literature	
DECchip 21064A–233, –275 Alpha AXP Microprocessor Data Sheet	EC–N0699–72
DECchip 21064–150, –166, –200 Alpha AXP Microprocessor Data Sheet	EB–N0136–72
DECchip 21064 Microprocessor Hardware Reference Manual	EC–N0079–72
DECchip 21064 PALcode System Design Guide	EC–N0543–72
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

Title	Order Number
DECchip 21064 Evaluation Board Literature	
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
DECchip 21064 Evaluation Board Debug Monitor User’s Guide	EC–N0392–72
DECchip 21064 Evaluation Board Design Package Read Me First	EC–N0352–72
DECchip 21064 Evaluation Board Product Brief	EC–N0353–72
DECchip 21064 SROM Mini-Debugger User’s Guide	EC–N0357–72
DECchip 21064 Evaluation Board User’s Guide	EC–N0351–72
DECchip 21064 Software Design Tools User’s Guide	EC–N0441–72
DECchip Marketing Literature	
AMP PGA Burn-In and Product Sockets for DECchip 21064 with Alpha AXP Architecture AXP	EC–X3013–72
DECchip Information Line Brochure	EB–N0109–72
DECchip Preprocessor for Hewlett-Packard Logic Analyzer	EC–X2454–72
DECchip Sample Kit Brochure	EB–N0108–72

Ordering Associated DECchip Literature

To order any of the previously listed DECchip literature, send the following information by **FAX** or **mail**:

Name
Company name
Street address
City, state or county, and country
Document order number
Quantity

To order by . . .	Do this . . .
FAX (available 24 hours)	Call 1–508–351–4467.
Mail	Write to: Digital Equipment Corporation Order Administration NRO2–2/J6 444 Whitney Street Northboro, MA 01532 USA

Associated Digital Literature

You can order and purchase the following literature from Digital.

Title	Order Number
Alpha Architecture Reference Manual	EY–L520E–DP–YCH

The following table explains how to order the previously listed Digital literature.

If you are from . . .	Then use this method to order . . .
The United States or Canada	Call **1–800–DIGITAL**.
Outside the United States	Contact your local Digital office, or technical or reference bookstore where Digital Press books are distributed by Prentice Hall.