digital

Digital Semiconductor
Alpha 21164PC Microprocessor

Data Sheet

Order Number: EC–R2W1A–TE

Revision/Update Information: This is a preliminary document.

Preliminary

Digital Equipment Corporation
Maynard, Massachusetts

http://www.digital.com/semiconductor

September 1997

Contents

Figures

Tables

1 About This Data Sheet

This data sheet provides a technical overview of the Digital Semiconductor Alpha 21164PC microprocessor (called the 21164PC), including:

- Functional units
- Signal descriptions
- External interface
- Internal processor register (IPR) summary
- Privileged architecture library code (PALcode) instructions
- Electrical characteristics
- Thermal characteristics
- Mechanical packaging

This data sheet is not intended to provide the reader with everything needed to begin chip implementation. For a more comprehensive description of the 21164PC and the Alpha architecture, refer to documents listed in the Support, Products, and Documentation section located at the end of this document.

Document Conventions

Throughout this data sheet, the following conventions are used:

- INT*n* refers to NATURALLY ALIGNED groups of *n* 8-bit bytes. For example:
  - INT16 — The four least significant address bits are 0.
  - INT8 — The three least significant address bits are 0.
  - INT4 — The two least significant address bits are 0.
- Values of 1, 0, and X are used in some tables. The X signifies a *don't care* (1 or 0) convention, which can be determined by the system designer.

2 Alpha 21164PC Microprocessor Features

The 21164PC has the following features:

- Fully pipelined 64-bit advanced RISC architecture supports multiple operating systems, including:
  - Microsoft Windows NT
  - DIGITAL UNIX
  - OpenVMS
- 400-MHz through 533-MHz operation
- Superscalar 4-way instruction issue
- High-bandwidth (128-bit) interface
- Peak execution rate of 2.1 BIPS
- 0.35-µm CMOS technology
- Two onchip caches:
  - 16KB, direct-mapped, L1 instruction cache
  - 8KB, dual-ported, direct-mapped, write-through L1 data cache
- Supports board-level L2 cache ranging from 512KB to 4MB
- Supports byte and word data types
- 3.3-V external interface and 2.5-V internal interface

The 21164PC implements IEEE S_floating and T_floating, and VAX F_floating and G_floating data types, and supports longword (32-bit) and quadword (64-bit) integers. It also provides byte (8-bit) and word (16-bit) support by byte-manipulation instructions. Limited hardware support is provided for the VAX D_floating data type.

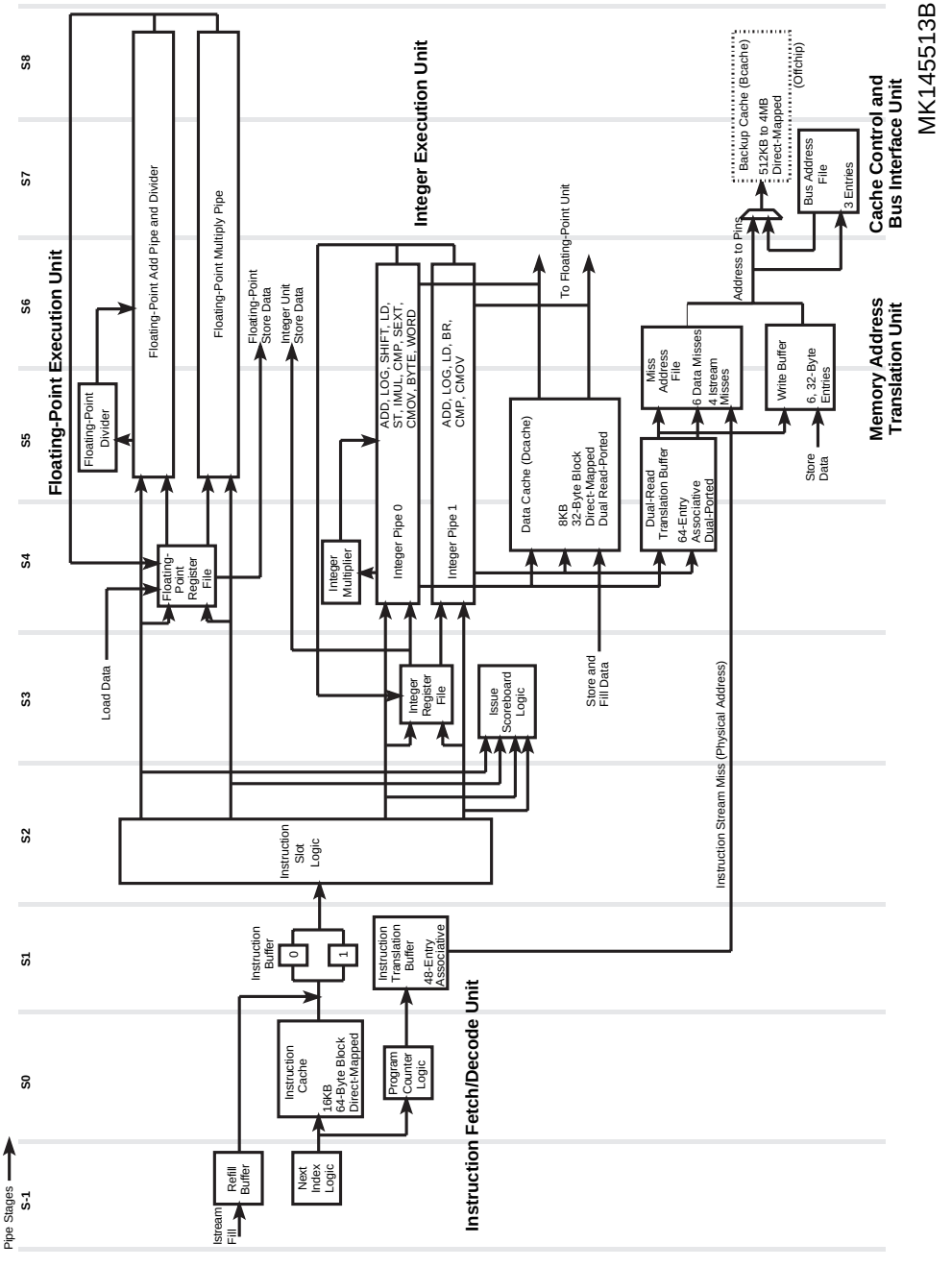
3 Microarchitecture

The 21164PC microprocessor is a high-performance implementation of DIGITAL's Alpha architecture. The following subsections provide an overview of the chip's architecture and major functional units.

Figure 1 shows a block diagram of the 21164PC, which consists of the following:

- Instruction fetch/decode and branch unit (IDU)
- Integer execution unit (IEU)
- Floating-point execution unit (FPU)
- Memory address translation unit (MTU)
- Cache control and bus interface unit (CBU)
- Data cache (Dcache)
- Instruction cache (Icache)
- Serial read-only memory (SROM) interface

Figure 1 21164PC Microprocessor Block/Pipe Flow Diagram



3.1 Instruction Fetch/Decode and Branch Unit

The primary function of the instruction fetch/decode and branch unit (IDU) is to manage and issue instructions to the IEU, MTU, and FPU. It also manages the instruction cache. The IDU contains:

- Prefetcher and instruction buffer
- Instruction slot and issue logic
- Program counter (PC) and branch prediction logic
- 48-entry instruction translation buffers (ITBs)
- Abort logic
- Register conflict logic
- Interrupt and exception logic

3.1.1 Instruction Prefetch and Decode

The IDU handles only NATURALLY ALIGNED groups of four instructions (INT16). The IDU does not advance to a new group of four instructions until all instructions in a group are issued. If a branch to the middle of an INT16 group occurs, then the IDU attempts to issue the instructions from the branch target to the end of the current INT16; the IDU then proceeds to the next INT16 of instructions after all the instructions in the target INT16 are issued. Thus, proper code scheduling is required to achieve optimal performance.

3.1.2 Branch Prediction

The branch unit, or prediction logic, is also part of the IDU. Branch and PC prediction are necessary to predict and begin fetching the target instruction stream before the branch or jump instruction is issued. Each instruction location in the instruction cache (Icache) contains a 2-bit history state to record the outcome of branch instructions.

3.1.3 Instruction Translation Buffer

The IDU includes a 48-entry, fully associative instruction translation buffer (ITB). The buffer stores recently used instruction stream (Istream) address translations and protection information for pages ranging from 8KB to 512KB and uses a not-last-used replacement algorithm.

The 21164PC provides two optional translation extensions called superpages. Access to superpages is allowed only while executing in privileged mode.

- One superpage maps virtual address bits <39:13> to physical address bits <39:13>, on a one-to-one basis, when virtual address bits <42:41> equal 2.
- The other superpage maps virtual address bits <29:13> to physical address bits <29:13>, on a one-to-one basis, and forces physical address bits <39:30> to 0 when virtual address bits <42:30> equal 1FFE(hex).

3.1.4 Interrupts

The IDU exception logic supports three sources of interrupts:

- Hardware interrupts

  There are seven level-sensitive hardware interrupt sources supplied by the following signals:

  irq_h<3:0>
  sys_mch_chk_irq_h
  pwr_fail_irq_h
  mch_halt_irq_h

- Software interrupts

  There are 15 prioritized software interrupts sourced by an onchip internal processor register (IPR).

- Asynchronous system traps

  There are four asynchronous system traps (ASTs) controlled by onchip IPRs.

  Most interrupts can be independently masked in onchip enable registers. In addition, AST interrupts are qualified by the current processor mode. All interrupts are disabled when the processor is executing PALcode.

3.2 Integer Execution Unit

The integer execution unit (IEU) contains two 64-bit integer execution pipelines—E0 and E1, which include the following:

- Two adders
- Two logic boxes
- A barrel shifter

- Byte-manipulation logic
- An integer multiplier

The IEU also includes the 40-entry, 64-bit integer register file (IRF) that contains the 32 integer registers defined by the Alpha architecture and 8 PALshadow registers. The register file has four read ports and two write ports, which provide operands to both integer execution pipelines and accept results from both pipes. The register file also accepts load instruction results (memory data) on the same two write ports.

3.3 Floating-Point Execution Unit

The onchip, pipelined floating-point unit (FPU) can execute both IEEE and VAX floating-point instructions. The 21164PC supports IEEE S_floating and T_floating data types, and all rounding modes. It also supports VAX F_floating and G_floating data types, and provides limited support for the D_floating format. The FPU contains:

- A 32-entry, 64-bit floating-point register file (FRF)
- A user-accessible control register
- A floating-point multiply pipeline
- A floating-point add pipeline — The floating-point divide unit is associated with the floating-point add pipeline but is not pipelined.

The FPU can accept two instructions every cycle, with the exception of floating-point divide instructions. The result latency for nondivide, floating-point instructions is four cycles.

3.4 Memory Address Translation Unit

The memory address translation unit (MTU) contains three major sections:

- Data translation buffer (dual ported)
- Miss address file (MAF)
- Write buffer address file

The MTU receives up to two virtual addresses every cycle from the IEU. The translation buffer generates the corresponding physical addresses and access control information for each virtual address. The 21164PC implements a 43-bit virtual address, a 40-bit noncacheable physical address, and a 33-bit cacheable physical

address. Cacheable addresses consist of bits <32:0> when bit <39> = 0. Physical addresses that set bits <38:33> are not supported by the 21164PC. These addresses are not checked by the 21164PC and could result in erroneous data.

The MTU can perform read and write operations to/from memory on byte, word, longword, and quadword boundaries.

3.4.1 Data Translation Buffer

The 64-entry, fully associative, dual-read-ported data translation buffer (DTB) stores recently used data stream (Dstream) page table entries (PTEs). Each entry supports all four granularity hint-bit combinations, so that a single DTB entry can provide translation for up to 512 contiguously mapped, 8-KB pages.

The DTB also supports the register-enabled superpage extension. The DTB superpage maps provide virtual-to-physical address translation for two regions of the virtual address space.

3.4.2 Miss Address File

The MTU begins the execution of each load instruction by translating the virtual address and by accessing the data cache (Dcache). Translation and Dcache tag read operations occur in parallel. If the addressed location is found in the Dcache (a hit), then the data from the Dcache is formatted and written to either the integer register file (IRF) or floating-point register file (FRF). The formatting required depends on the particular load instruction executed. If the data is not found in the Dcache (a miss), then the address, target register number, and formatting information are entered in the miss address file (MAF).

The MAF performs a load-merging function. When a load miss occurs, each MAF entry is checked to see if it contains a load miss that addresses the same Dcache (32-byte) block. If it does, and certain merging rules are satisfied, then the new load miss is merged with an existing MAF entry. This allows the MTU to service two or more load misses with one data fill from the CBU.

There are six MAF entries for load misses and four more for IDU instruction fetches and prefetches. Load misses are usually the highest MTU priority.

3.4.3 Store Execution

The Dcache follows a write-through protocol. During the execution of a store instruction, the MTU probes the Dcache to determine whether the location to be overwritten is currently cached. If so (a Dcache hit), the Dcache is updated. Regardless of the Dcache state, the MTU forwards the data to the CBU.

A load instruction that is issued one cycle after a store instruction in the pipeline creates a conflict if both the load and store operations access the same memory location. (The store instruction has not yet updated the location when the load instruction reads it.) This conflict is handled by forcing the load instruction to take a replay trap; that is, the IDU flushes the pipeline and restarts execution from the load instruction. By the time the load instruction arrives at the Dcache the second time, the conflicting store instruction has written the Dcache and the load instruction is executed normally.

Replay traps can be avoided by scheduling the load instruction to issue three cycles after the store instruction. If the load instruction is scheduled to issue two cycles after the store instruction, then it will be issue-stalled for one cycle.

3.4.4 Write Buffer

The MTU also contains a write buffer that has six 32-byte entries. The write buffer provides a finite, high-bandwidth resource for receiving store data to minimize the number of CPU stall cycles.

3.5 Cache Control and Bus Interface Unit

The cache control and bus interface unit (CBU) processes all accesses sent by the MTU and implements all memory-related external interface functions, particularly the coherence protocol functions for write-back caching. It controls the board-level L2 cache (Bcache). The CBU handles all instruction and primary Dcache read misses and performs the function of writing data from the write buffer into the shared coherent memory subsystem. The CBU also controls the 128-bit bidirectional data bus, address bus, and I/O control.

3.6 Cache Organization

The 21164PC has two onchip caches — a primary L1 data cache and a primary L1 instruction cache. All memory cells in the onchip caches are fully static, 6-transistor, CMOS structures.

The 21164PC also provides control for a board-level, external L2 cache.

3.6.1 Data Cache

The data cache (Dcache) is a dual-read-ported, single-write-ported, 8-KB cache. It is a write-through, read-allocate, direct-mapped, physical cache with 32-byte blocks.

3.6.2 Instruction Cache

The instruction cache (Icache) is a 16-KB, virtual, direct-mapped cache with 64-byte blocks and 32-byte fills. Each block tag contains:

- A 7-bit address space number (ASN) field as defined by the Alpha architecture
- A 1-bit address space match (ASM) field as defined by the Alpha architecture
- A 1-bit PALcode (physically addressed) indicator

Software, rather than Icache hardware, maintains Icache coherence with memory.

3.6.3 External Cache

The CBU implements control for an external, direct-mapped, physical, write-back, write-allocate cache with 64-byte blocks. The 21164PC supports board-level cache sizes of 512KB, 1MB, 2MB, and 4MB.

3.7 Serial Read-Only Memory Interface

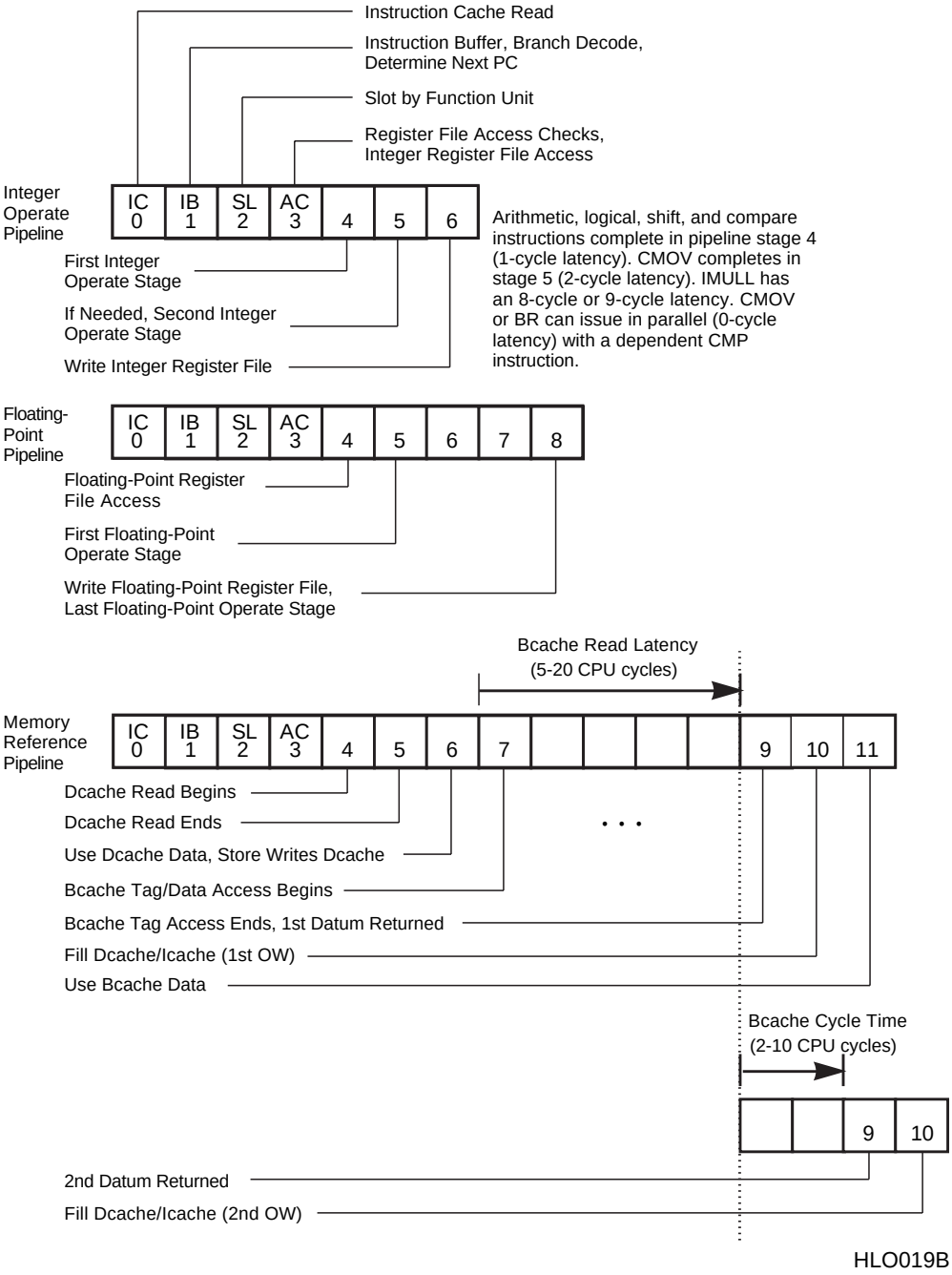
The serial read-only memory (SROM) interface provides the initialization data load path from a system SROM to the instruction cache. Following initialization, this interface can function as a diagnostic port by using privileged architecture library code (PALcode).

3.8 Pipeline Organization

The 21164PC has a 7-stage (or 7-cycle) pipeline for integer operate and memory reference instructions, and a 9-stage pipeline for floating-point operate instructions. The IDU maintains state for all pipeline stages to track outstanding register write operations.

Figure 2 shows the integer operate, memory reference, and floating-point operate pipelines for the IDU, FPU, IEU, and MTU. The first four stages are executed in the IDU. Remaining stages are executed by the IEU, FPU, MTU, and CBU.

Figure 2 Instruction Pipeline Stages



4 Pinout and Signal Descriptions

Sections 4.1 and 4.2 list and describe the 21164PC microprocessor external signals and their associated pins.

4.1 Pin Assignment

The 21164PC has 413 pins aligned in an interstitial pin grid array (IPGA) design. Table 1 lists the 21164PC signal pins and their corresponding pin grid array (PGA) locations in alphanumeric order; Table 2 lists the voltage reference, power, and ground pins. There are 264 functional signal pins, 2 spare signal pins (unused), 5 voltage reference pins (unused), 46 external power (**Vdd**) pins, 22 internal power (**Vddi**) pins, and 74 ground (**Vss**) pins.

Table 1 Alphanumeric Signal Pin List *(Sheet 1 of 4)*

Signal	PGA Location	Signal	PGA Location	Signal	PGA Location
addr_h<4>	AH12	**addr_h<5>**	AN11	**addr_h<6>**	AJ13
addr_h<7>	AL9	**addr_h<8>**	AK8	**addr_h<9>**	AJ11
addr_h<10>	AN9	**addr_h<11>**	AJ7	**addr_h<12>**	AJ9
addr_h<13>	AL5	**addr_h<14>**	AK6	**addr_h<15>**	AH6
addr_h<16>	AG7	**addr_h<17>**	AK4	**addr_h<18>**	AJ3
addr_h<19>	AH4	**addr_h<20>**	AJ1	**addr_h<21>**	AF6
addr_h<22>	AC31	**addr_h<23>**	AJ35	**addr_h<24>**	AH32
addr_h<25>	AE37	**addr_h<26>**	AK34	**addr_h<27>**	AD32
addr_h<28>	AE33	**addr_h<29>**	AF34	**addr_h<30>**	AL33
addr_h<31>	AK32	**addr_h<32>**	AF32	**addr_h<33>**	AG31
addr_h<34>	AJ31	**addr_h<35>**	AJ37	**addr_h<36>**	AL31
addr_h<37>	AN29	**addr_h<38>**	AL29	**addr_h<39>**	AH34
addr_bus_req_h	A21	**addr_res_h<0>**	D34	**addr_res_h<1>**	E37
cack_h	F18	**clk_mode_h<0>**	AJ19	**clk_mode_h<1>**	AH20
cmd_h<0>	F2	**cmd_h<1>**	E3	**cmd_h<2>**	G3
cmd_h<3>	H6	**cpu_clk_out_h**	AL25	**dack_h**	F20

Table 1 Alphanumeric Signal Pin List *(Sheet 2 of 4)*

Signal	PGA Location	Signal	PGA Location	Signal	PGA Location
data_h<0>	F34	**data_h<1>**	H32	**data_h<2>**	J31
data_h<3>	J33	**data_h<4>**	G37	**data_h<5>**	K32
data_h<6>	H34	**data_h<7>**	J35	**data_h<8>**	K34
data_h<9>	L31	**data_h<10>**	J37	**data_h<11>**	M32
data_h<12>	L35	**data_h<13>**	M36	**data_h<14>**	L37
data_h<15>	M34	**data_h<16>**	N31	**data_h<17>**	P32
data_h<18>	P34	**data_h<19>**	Q31	**data_h<20>**	N35
data_h<21>	Q33	**data_h<22>**	Q35	**data_h<23>**	Q37
data_h<24>	R32	**data_h<25>**	R34	**data_h<26>**	S37
data_h<27>	S31	**data_h<28>**	S35	**data_h<29>**	T32
data_h<30>	T34	**data_h<31>**	T30	**data_h<32>**	T36
data_h<33>	U35	**data_h<34>**	U31	**data_h<35>**	U37
data_h<36>	V36	**data_h<37>**	V34	**data_h<38>**	V30
data_h<39>	W35	**data_h<40>**	V32	**data_h<41>**	X34
data_h<42>	W37	**data_h<43>**	W31	**data_h<44>**	Y37
data_h<45>	Y35	**data_h<46>**	Z34	**data_h<47>**	X32
data_h<48>	Y33	**data_h<49>**	AB36	**data_h<50>**	Y31
data_h<51>	AC35	**data_h<52>**	AA31	**data_h<53>**	Z32
data_h<54>	AD34	**data_h<55>**	AE35	**data_h<56>**	AA37
data_h<57>	AB34	**data_h<58>**	AG37	**data_h<59>**	AB32
data_h<60>	AG35	**data_h<61>**	AH36	**data_h<62>**	AE31
data_h<63>	AC37	**data_h<64>**	J3	**data_h<65>**	J1
data_h<66>	K6	**data_h<67>**	J5	**data_h<68>**	L7
data_h<69>	K4	**data_h<70>**	L3	**data_h<71>**	M2
data_h<72>	L1	**data_h<73>**	M6	**data_h<74>**	N1
data_h<75>	M4	**data_h<76>**	N3	**data_h<77>**	Q7

Table 1 Alphanumeric Signal Pin List *(Sheet 3 of 4)*

Signal	PGA Location	Signal	PGA Location	Signal	PGA Location
data_h<78>	Q1	**data_h<79>**	P6	**data_h<80>**	P4
data_h<81>	R6	**data_h<82>**	S1	**data_h<83>**	Q5
data_h<84>	T2	**data_h<85>**	Q3	**data_h<86>**	R4
data_h<87>	S3	**data_h<88>**	T6	**data_h<89>**	T4
data_h<90>	S7	**data_h<91>**	T8	**data_h<92>**	U3
data_h<93>	U7	**data_h<94>**	U1	**data_h<95>**	V8
data_h<96>	V2	**data_h<97>**	V4	**data_h<98>**	V6
data_h<99>	W7	**data_h<100>**	W1	**data_h<101>**	W3
data_h<102>	X4	**data_h<103>**	X6	**data_h<104>**	Y1
data_h<105>	Y5	**data_h<106>**	Y7	**data_h<107>**	Y3
data_h<108>	Z4	**data_h<109>**	Z6	**data_h<110>**	AA1
data_h<111>	AA3	**data_h<112>**	AC1	**data_h<113>**	AB4
data_h<114>	AB2	**data_h<115>**	AC7	**data_h<116>**	AD6
data_h<117>	AB6	**data_h<118>**	AC3	**data_h<119>**	AE5
data_h<120>	AD4	**data_h<121>**	AE1	**data_h<122>**	AF4
data_h<123>	AG3	**data_h<124>**	AE3	**data_h<125>**	AE7
data_h<126>	AG1	**data_h<127>**	AH2	**data_adsc_l**	D32
data_adv_l	C33	**data_bus_req_h**	E21	**data_ram_oe_l**	D18
data_ram_we_l <0>	B18	**data_ram_we_l<1>**	A19	**data_ram_we_l<2>**	B20
data_ram_we_l <3>	D20	**dc_ok_h**	AL23	**fill_h**	D4
fill_dirty_h	AL11	**fill_error_h**	F6	**fill_id_h**	C5
idle_bc_h	F4	**index_h<4>**	E23	**index_h<5>**	C25
index_h<6>	C21	**index_h<7>**	B26	**index_h<8>**	A23
index_h<9>	C23	**index_h<10>**	A25	**index_h<11>**	A27
index_h<12>	F26	**index_h<13>**	E25	**index_h<14>**	C27

Table 1 Alphanumeric Signal Pin List *(Sheet 4 of 4)*

Signal	PGA Location	Signal	PGA Location	Signal	PGA Location
index_h<15>	C29	**index_h<16>**	E27	**index_h<17>**	E29
index_h<18>	D30	**index_h<19>**	A29	**index_h<20>**	A31
index_h<21>	C31	**int4_valid_h<0>**	E35	**int4_valid_h<1>**	F32
int4_valid_h<2>	H4	**int4_valid_h<3>**	E1	**irq_h<0>**	AJ27
irq_h<1>	AL27	**irq_h<2>**	AH26	**irq_h<3>**	AN27
lw_parity_h<0>	G35	**lw_parity_h<1>**	F36	**lw_parity_h<2>**	J7
lw_parity_h<3>	G1	**mch_hlt_irq_h**	AM26	**osc_clk_in_h**	AN19
osc_clk_in_l	AM20	**port_mode_h<0>**	AH18	**port_mode_h<1>**	AM18
pwr_fail_irq_h	AJ25	**srom_clk_h**	AL17	**srom_data_h**	AK16
srom_oe_l	AJ17	**srom_present_l**	AK18	**st_clk1_h**	E7
st_clk2_h	E31	**st_clk3_h**	G31	**sys_clk_out1_h**	AK22
sys_clk_out2_h	AJ23	**sys_mch_chk_ irq_h**	AN25	**sys_reset_l**	AN23
tag_data_h<19>	A11	**tag_data_h<20>**	D6	**tag_data_h<21>**	E9
tag_data_h<22>	D8	**tag_data_h<23>**	C7	**tag_data_h<24>**	F12
tag_data_h<25>	B6	**tag_data_h<26>**	E11	**tag_data_h<27>**	C9
tag_data_h<28>	A9	**tag_data_h<29>**	C13	**tag_data_h<30>**	C11
tag_data_h<31>	E15	**tag_data_h<32>**	A15	**tag_data_par_h**	B12
tag_dirty_h	A13	**tag_ram_oe_l**	A17	**tag_ram_we_l**	C17
tag_valid_h	E17	**tck_h**	AL13	**tdi_h**	AN17
tdo_h	AL15	**temp_sense**	AN13	**test_status_h<1>**	AJ15
tms_h	AN15	**trst_l**	AM12	**victim_pending_h**	C15
spare1	AJ29	**spare2**	AK30	—	

Table 2 lists the voltage reference, power, and ground pins.

Table 2 Voltage Reference, Power, and Ground Pins

Signal	PGA Location
Vss Metal planes 2, 6	A3, A5, A7, A33, A35, B2, B8, B14, B24, B30, B36, C1, C37, D12, D16, D22, D26, E5, E19, E33, F10, F16, F22, F28, H2, H36, K8, K30, L5, L33, P2, P8, P30, P36, S5, S33, W5, W33, Z2, Z8, Z30, Z36, AC5, AC33, AD8, AD30, AF2, AF36, AH10, AH16, AH22, AH28, AJ5, AJ33, AK12, AK20, AK26, AL1, AL19, AL21, AL37, AM2, AM8, AM14, AM24, AM30, AM36, AN3, AN5, AN7, AN21, AN31, AN33, AN35
Vdd Metal plane 7	B4, B10, B16, B22, B28, B34, C3, C35, D2, D36, F8, F14, F24, F30, K2, K36, M8, M30, R2, R8, R30, R36, X2, X8, X30, X36, AB8, AB30, AD2, AD36, AH8, AH14, AH24, AH30, AK2, AK36, AL3, AL35, AM4, AM6, AM10, AM16, AM22, AM28, AM32, AM34
Vddi Metal plane 4	B32, C19, D10, D14, D24, D28, E13, G5, G33, N5, N7, N33, N37, U5, U33, AA5, AA7, AA33, AA35, AG5, AG33, AJ21, AK10, AK14, AK24, AK28, AL7

4.2 21164PC Packaging

Figure 3 shows the 21164PC pinout from the top view with pins facing down.

Figure 3 21164PC Top View (Pin Down)

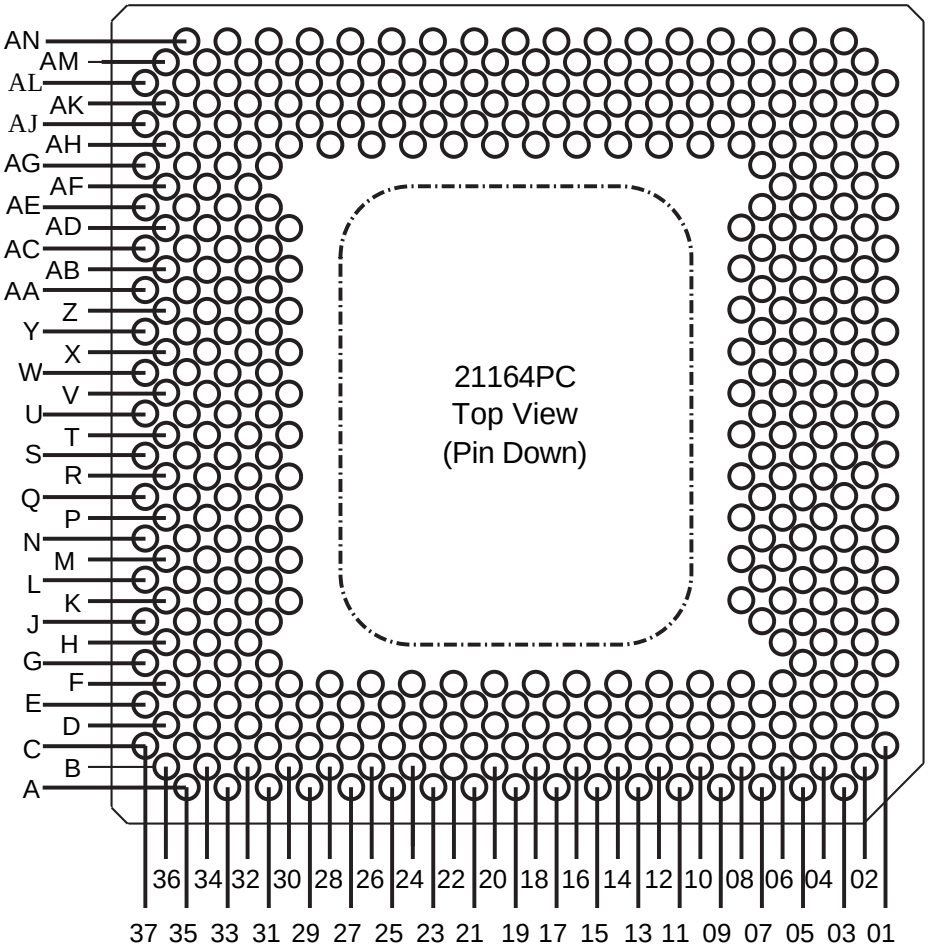
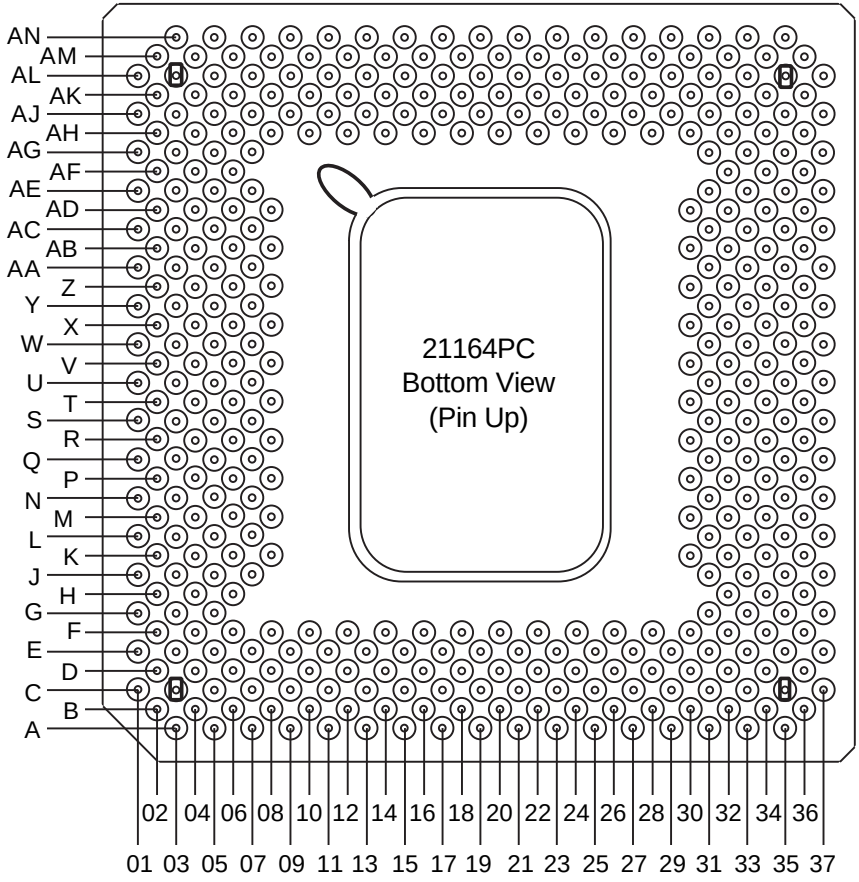


Figure 4 shows the 21164PC pinout from the bottom view with pins facing up.

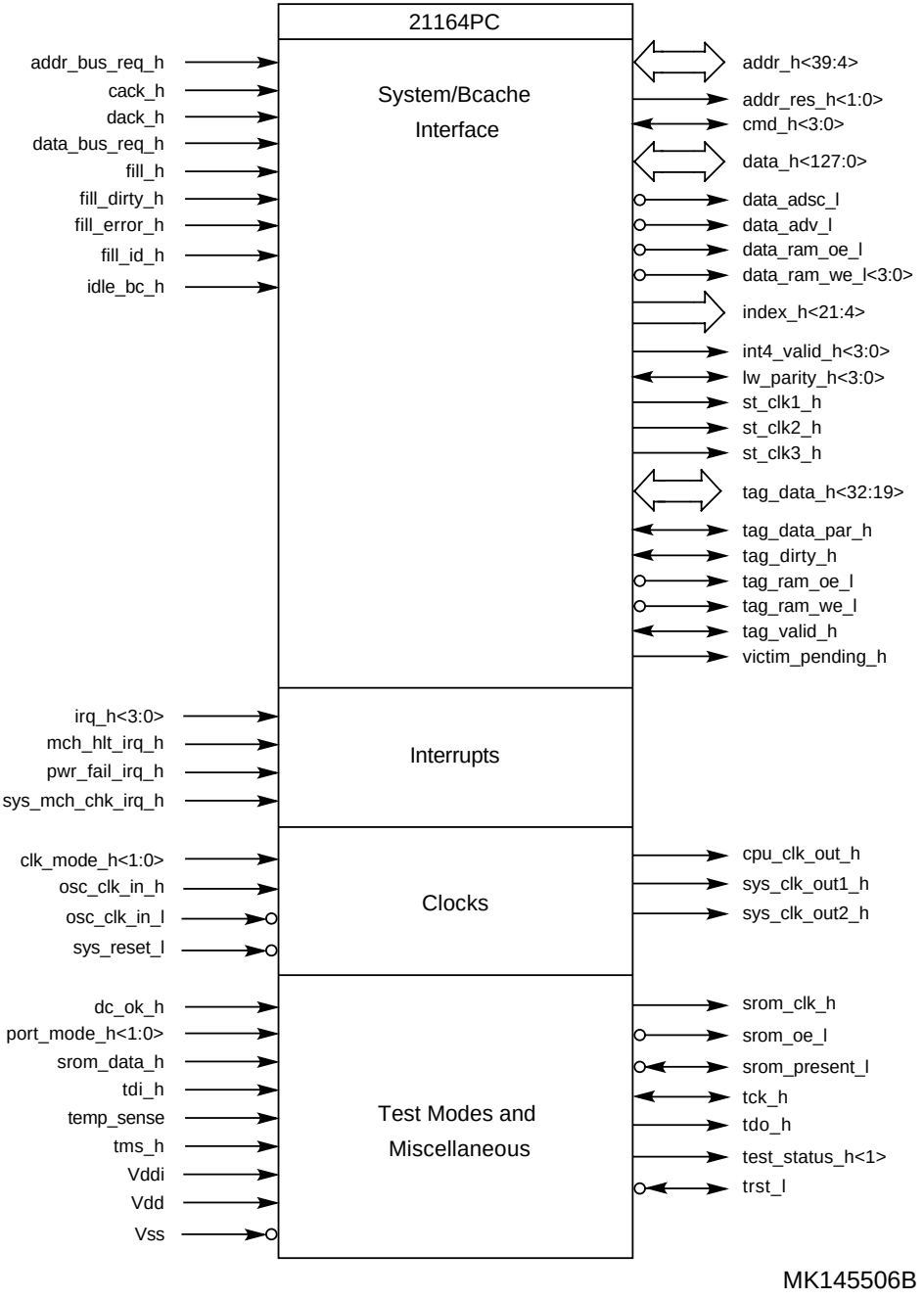
Figure 4 21164PC Bottom View (Pin Up)



4.3 21164PC Microprocessor Logic Symbol

Figure 5 shows the logic symbol for the 21164PC chip.

Figure 5 21164PC Microprocessor Logic Symbol



4.4 21164PC Signal Names and Functions

The following table defines the 21164PC signal types referred to in this section:

Signal Type	Definition
B	Bidirectional
I	Input only
O	Output only

The remaining two tables describe the function of each 21164PC external signal. Table 3 lists all signals in alphanumeric order and provides full signal descriptions. Table 4 lists signals by function and provides an abbreviated description.

Table 3 21164PC Signal Descriptions *(Sheet 1 of 10)*

Signal	Type	Count	Description
addr_h<39:4>	B	36	Address bus. These bidirectional signals provide the address of the requested data or operation between the 21164PC and the system. If **addr_h<39>** is asserted, then the reference is to noncached, I/O memory space. When the byte/word instructions are used and **addr_h<39>** is asserted, six additional bits of information are communicated over the pin bus. Two of the new bits are driven over **addr_h<38:37>**, becoming **transfer_size<1:0>**, with the following values: 00 Size = 8 bytes 01 Size = 4 bytes 10 Size = 2 bytes 11 Size = 1 byte
addr_bus_req_h	I	1	Address bus request. The system interface uses this signal to gain control of the **addr_h<39:4>** and **cmd_h<3:0>** pins.

Table 3 21164PC Signal Descriptions *(Sheet 2 of 10)*

Signal	Type	Count	Description
addr_res_h<1:0>	O	2	Address response bits <1> and <0>. For system commands, the 21164PC uses these pins to indicate the state of the block in the Bcache:

Bits	Command	Meaning
00	NOP	Nothing.
01	NOACK	Data not found or clean.
10	—	Reserved.
11	ACK/Bcache	Data from Bcache.

cack_h	I	1	Command acknowledge. The system interface uses this signal to acknowledge any one of the commands driven by the 21164PC.
clk_mode_h<1:0>	I	2	Clock test mode. These signals specify a relationship between **osc_clk_in_h**,**l**, the CPU cycle time, and the duty-cycle equalizer. These signals should be deasserted in normal operation mode.

Bits	Description
00	CPU clock frequency is equal to the input clock frequency.
01	CPU clock frequency is equal to the input clock frequency, with the onchip duty-cycle equalizer enabled.
10	Initialize the CPU clock, allowing the system clock to be synchronized to a stable reference clock.
11	Initialize the CPU clock, allowing the system clock to be synchronized to a stable reference clock, with the onchip duty-cycle equalizer enabled.

Table 3 21164PC Signal Descriptions *(Sheet 3 of 10)*

Signal	Type	Count	Description
cmd_h<3:0>	B	4	Command bus. These signals drive and receive the commands from the command bus. The following tables define the commands that can be driven on the **cmd_h<3:0>** bus by the 21164PC or the system.

21164PC Commands to System:

cmd_h <3:0>	Command	Meaning
0000	NOP	Nothing.
0001	—	Reserved.
0010	—	Reserved.
0011	—	Reserved.
0100	—	Reserved.
0101	—	Reserved.
0110	WRITE BLOCK	Request to write a block.
0111	—	Reserved.
1000	READ MISS0	Request for data.
1001	READ MISS1	Request for data.
1010	—	Reserved.
1011	—	Reserved.
1100	BCACHE VICTIM	Bcache victim should be removed.
1101	—	Reserved.
1110	—	Reserved.
1111	—	Reserved.

Table 3 21164PC Signal Descriptions *(Sheet 4 of 10)*

Signal	Type	Count	Description

System Commands to 21164PC:

cmd_h <3:0>	Command	Meaning
0000	NOP	Nothing.
0001	FLUSH	Removes block from caches; return dirty data.
0010	INVALIDATE	Invalidates the block from caches.
0011	—	Reserved.
0100	READ	Read a block.
0101	—	Reserved.
0111	—	Reserved.
1xxx	—	Reserved.

cpu_clk_out_h	O	1	CPU clock output. This signal is used for test purposes.
dack_h	I	1	Data acknowledge. The system interface uses this signal to control data transfer between the 21164PC and the system.
data_h<127:0>	B	128	Data bus. These signals are used to move data between the 21164PC, the system, and the Bcache.
data_adsc_l	O	1	Loads a new address into the Bcache SSRAM.
data_adv_l	O	1	Advances the Bcache index to the next address.
data_bus_req_h	I	1	Data bus request. If the 21164PC samples this signal asserted on the rising edge of sysclk *n*, then the 21164PC does not drive the data bus on the rising edge of sysclk *n*+1. Before asserting this signal, the system should assert **idle_bc_h** for the correct number of cycles. If the 21164PC samples this signal deasserted on the rising edge of sysclk *n*, then the 21164PC drives the data bus on the rising edge of sysclk *n*+1.
data_ram_oe_l	O	1	Data RAM output enable. This signal is asserted for Bcache read operations.

Table 3 21164PC Signal Descriptions *(Sheet 5 of 10)*

Signal	Type	Count	Description
data_ram_we_l<3:0>	O	4	Data RAM write-enable. These signals are asserted for any Bcache write operation.
dc_ok_h	I	1	dc voltage OK. Must be deasserted until dc voltage reaches proper operating level. After that, **dc_ok_h** is asserted.
fill_h	I	1	Fill warning. If the 21164PC samples this signal asserted on the rising edge of sysclk *n*, then the 21164PC provides the address indicated by **fill_id_h** to the Bcache on the rising edge of sysclk *n*+1. The Bcache begins to write in that sysclk. At the end of sysclk *n*+1, the 21164PC waits for the next sysclk and then begins the write operation again if **dack_h** is not asserted.
fill_dirty_h	I	1	Fill dirty. If the block being filled is dirty, this pin should be asserted.
fill_error_h	I	1	Fill error. If this signal is asserted during a fill from memory, it indicates to the 21164PC that the system has detected an invalid address or hard error. The system still provides an apparently normal read sequence with correct ECC/parity though the data is not valid. The 21164PC traps to the machine check (MCHK) PALcode entry point and indicates a serious hardware error. **fill_error_h** should be asserted when the data is returned. Each assertion produces a MCHK trap.
fill_id_h	I	1	Fill identification. Asserted with **fill_h** to indicate which register is used. The 21164PC supports two outstanding load instructions. If this signal is asserted when the 21164PC samples **fill_h** asserted, then the 21164PC provides the address from miss register 1. If it is deasserted, then the address in miss register 0 is used for the read operation.
idle_bc_h	I	1	Idle Bcache. When asserted, the 21164PC finishes the current Bcache read or write operation but does not start a new read or write operation until the signal is deasserted. The system interface must assert this signal in time to idle the Bcache before fill data arrives.
index_h<21:4>	O	18	Index. These signals index the Bcache.

Table 3 21164PC Signal Descriptions *(Sheet 6 of 10)*

Signal	Type	Count	Description
int4_valid_h<3:0>	O	4	INT4 data valid. During write operations to noncached space, these signals are used to indicate which INT4 bytes of data are valid. This is useful for noncached write operations that have been merged in the write buffer.

int4_valid_h<3:0>	Write Meaning
xxx1	**data_h<31:0>** valid
xx1x	**data_h<63:32>** valid
x1xx	**data_h<95:64>** valid
1xxx	**data_h<127:96>** valid

During read operations to noncached space, these signals indicate which INT8 bytes of a 32-byte block need to be read and returned to the processor. This is useful for read operations to noncached memory.

int4_valid_h<3:0>	Read Meaning
xxx1	**data_h<63:0>** valid
xx1x	**data_h<127:64>** valid
x1xx	**data_h<191:128>** valid
1xxx	**data_h<255:192>** valid

Note: For both read and write operations, multiple **int4_valid_h<3:0>** bits can be set simultaneously.

Table 3 21164PC Signal Descriptions *(Sheet 7 of 10)*

Signal	Type	Count	Description

When **addr_h<39>** is asserted, the **int4_valid_h<3:0>** signals are considered the **addr_h<3:0>** bits required for byte/word transactions. The functionality of these bits is tied to the value stored in **addr_h<38:37>**.

For read transactions:

addr_h <38:37>	int4_valid_h<3:0> Value
00	Valid INT8 mask
01	**addr_h<3:2>** valid on **int4_valid_h<3:2>**; **int4_valid<1:0>** undefined
10	**addr_h<3:1>** valid on **int4_valid_h<3:1>**; **int4_valid<0>** undefined
11	**addr_h<3:0>** valid on **int4_valid_h<3:0>**

For write transactions:

addr_h <38:37>	int4_valid_h<3:0> Value
00	Valid INT4 mask
01	Valid INT4 mask
10	**addr_h<3:1>** valid on **int4_valid_h<3:1>**; **int4_valid<0>** undefined
11	**addr_h<3:0>** valid on **int4_valid_h<3:0>**

Table 3 21164PC Signal Descriptions *(Sheet 8 of 10)*

Signal	Type	Count	Description
irq_h<3:0>	I	4	System interrupt requests. These signals have multiple modes of operation. During normal operation, these level-sensitive signals are used to signal interrupt requests. During initialization, these signals are used to set up the CPU cycle time divisor for **sys_clk_out1_h** as follows:

irq_h<3>	irq_h<2>	irq_h<1>	irq_h<0>	Ratio
Low	High	Low	Low	4
Low	High	Low	High	5
Low	High	High	Low	6
Low	High	High	High	7
High	Low	Low	Low	8
High	Low	Low	High	9
High	Low	High	Low	10
High	Low	High	High	11
High	High	Low	Low	12
High	High	Low	High	13
High	High	High	Low	14
High	High	High	High	15

lw_parity_h<3:0>	B	4	Longword parity. These signals set even INT4 parity for the current data cycle.
mch_hlt_irq_h	I	1	Machine halt interrupt request. This signal has multiple modes of operation. During initialization, this signal is used to set up **sys_clk_out2_h** delay. During normal operation, it is used to signal a halt request.
osc_clk_in_h **osc_clk_in_l**	I I	1 1	Oscillator clock inputs. These signals provide the differential clock input that is the fundamental timing of the 21164PC. These signals are driven at the same frequency as the internal clock frequency (**clk_mode_h<1:0>** = 01).
port_mode_h<1:0>	I	2	Select test port interface modes (normal, manufacturing, and debug). For normal operation, both signals must be deasserted.

Table 3 21164PC Signal Descriptions *(Sheet 9 of 10)*

Signal	Type	Count	Description
pwr_fail_irq_h	I	1	Power failure interrupt request. This signal has multiple modes of operation. During initialization, this signal is used to set up **sys_clk_out2_h** delay. During normal operation, this signal is used to signal a power failure.
srom_clk_h	O	1	Serial ROM clock. Supplies the clock that causes the SROM to advance to the next bit. The cycle time of this clock is 128 times the cycle time of the CPU clock.
srom_data_h	I	1	Serial ROM data. Input for the SROM.
srom_oe_l	O	1	Serial ROM output enable. Supplies the output enable to the SROM.
srom_present_l[1]	B	1	Serial ROM present. Indicates that SROM is present and ready to load the Icache.
st_clk1_h	O	1	STRAM clock. Clock for synchronously timed RAMs (STRAMs). For Bcache, this signal is synchronous with **index_h<21:4>** during private read and write operations, and with **sys_clk_out1_h** during read and fill operations.
st_clk2_h	O	1	This signal is a duplicate of **st_clk1_h**, to increase the fanout capability of the signal.
st_clk3_h	O	1	This signal is another duplicate of **st_clk1_h**, to increase the fanout capability of the signal.
sys_clk_out1_h	O	1	System clock output. Programmable system clock (**cpu_clk_out_h** divided by a value of 3 to 15) is used for board-level cache and system logic.
sys_clk_out2_h	O	1	System clock output. A version of **sys_clk_out1_h** delayed by a programmable amount from 0 to 7 CPU cycles.
sys_mch_chk_irq_h	I	1	System machine check interrupt request. This signal has multiple modes of operation. During initialization, it is used to set up **sys_clk_out2_h** delay. During normal operation, it is used to signal a machine interrupt check request.
sys_reset_l	I	1	System reset. This signal protects the 21164PC from damage during initial power-up. It must be asserted until **dc_ok_h** is asserted. After that, it is deasserted and the 21164PC begins its reset sequence.
tag_data_h<32:19>	B	14	Bcache tag data bits. This bit range supports 0.5MB to 4MB Bcaches.

Table 3 21164PC Signal Descriptions *(Sheet 10 of 10)*

Signal	Type	Count	Description
tag_data_par_h	B	1	Tag data parity bit. This signal indicates odd parity for **tag_data_h<32:19>**.
tag_dirty_h	B	1	Tag dirty state bit. This bit is private to the 21164PC.
tag_ram_oe_l	O	1	Tag RAM output enable. This signal is asserted during any Bcache read operation.
tag_ram_we_l	O	1	Tag RAM write-enable. This signal is asserted during any tag write operation.
tag_valid_h	B	1	Tag valid bit. During fills, this signal is asserted to indicate that the block has valid data.
tck_h	B	1	JTAG boundary-scan clock.
tdi_h	I	1	JTAG serial boundary-scan data-in signal.
tdo_h	O	1	JTAG serial boundary-scan data-out signal.
temp_sense	I	1	Temperature sense. This signal is used to measure the die temperature and is for manufacturing use only. For normal operation, this signal must be left disconnected.
test_status_h<1>	O	1	Icache test status or timeout reset. This signal is used for manufacturing test purposes only to extract Icache test status information from the chip.
tms_h	I	1	JTAG test mode select signal.
trst_l[1]	B	1	JTAG test access port (TAP) reset signal.
victim_pending_h	O	1	Victim pending. When asserted, this signal indicates that the current read miss has generated a victim.

[1] This signal is shown as bidirectional. However, for normal operation, it is input only. The output function is used during manufacturing test and verification only.

Table 4 lists signals by function and provides an abbreviated description.

Table 4 21164PC Signal Descriptions by Function *(Sheet 1 of 3)*

Signal	Type	Count	Description
Clocks			
clk_mode_h<1:0>	I	2	Clock test mode
cpu_clk_out_h	O	1	CPU clock output
osc_clk_in_h,l	I	2	Oscillator clock inputs
st_clk1_h	O	1	Bcache STRAM clock output
st_clk2_h	O	1	Bcache STRAM clock output
st_clk3_h	O	1	Bcache STRAM clock output
sys_clk_out1_h	O	1	System clock output
sys_clk_out2_h	O	1	System clock output
sys_reset_l	I	1	System reset
Bcache			
data_h<127:0>	B	128	Data bus
data_adsc_l	O	1	Data RAM address load enable
data_adv_l	O	1	Data RAM address advance enable
data_ram_oe_l	O	1	Data RAM output enable
data_ram_we_l<3:0>	O	4	Data RAM write-enable bits
index_h<21:4>	O	18	Index
lw_parity_h<3:0>	B	4	Data check
tag_data_h<32:19>	B	14	Bcache tag data bits
tag_data_par_h	B	1	Tag data parity bit
tag_dirty_h	B	1	Tag dirty state bit
tag_ram_oe_l	O	1	Tag RAM output enable
tag_ram_we_l	O	1	Tag RAM write-enable
tag_valid_h	B	1	Tag valid bit

Table 4 21164PC Signal Descriptions by Function *(Sheet 2 of 3)*

Signal	Type	Count	Description
System Interface			
addr_h<39:4>	B	36	Address bus
addr_bus_req_h	I	1	Address bus request
addr_res_h<1:0>	O	2	Address response
cack_h	I	1	Command acknowledge
cmd_h<3:0>	B	4	Command bus
dack_h	I	1	Data acknowledge
data_bus_req_h	I	1	Data bus request
fill_h	I	1	Fill warning
fill_dirty_h	I	1	Fill dirty
fill_error_h	I	1	Fill error
fill_id_h	I	1	Fill identification
idle_bc_h	I	1	Idle Bcache
int4_valid_h<3:0>	O	4	INT4 data valid
victim_pending_h	O	1	Victim pending
Interrupts			
irq_h<3:0>	I	4	System interrupt requests
mch_hlt_irq_h	I	1	Machine halt interrupt request
pwr_fail_irq_h	I	1	Power failure interrupt request
sys_mch_chk_irq_h	I	1	System machine check interrupt request
Test Modes and Miscellaneous			
dc_ok_h	I	1	dc voltage OK
port_mode_h<1:0>	I	2	Selects the test port interface mode (normal, manufacturing, and debug)
srom_clk_h	O	1	Serial ROM clock
srom_data_h	I	1	Serial ROM data

Table 4 21164PC Signal Descriptions by Function *(Sheet 3 of 3)*

Signal	Type	Count	Description
srom_oe_l	O	1	Serial ROM output enable
srom_present_l[1]	B	1	Serial ROM present
tck_h	B	1	JTAG boundary-scan clock
tdi_h	I	1	JTAG serial boundary-scan data in
tdo_h	O	1	JTAG serial boundary-scan data out
temp_sense	I	1	Temperature sense
test_status_h<1>	O	1	Icache test status or timeout reset
tms_h	I	1	JTAG test mode select
trst_l[1]	B	1	JTAG test access port (TAP) reset

[1] This signal is shown as bidirectional. However, for normal operation, it is input only. The output function is used during manufacturing test and verification only.

5 21164PC CPU Functional Overview

This section provides an overview of 21164PC external signals that support the following:

- Clocks
- Bcache interface
- System interface
- Interrupts
- Test modes

See Figure 1 for a block diagram of the 21164PC.

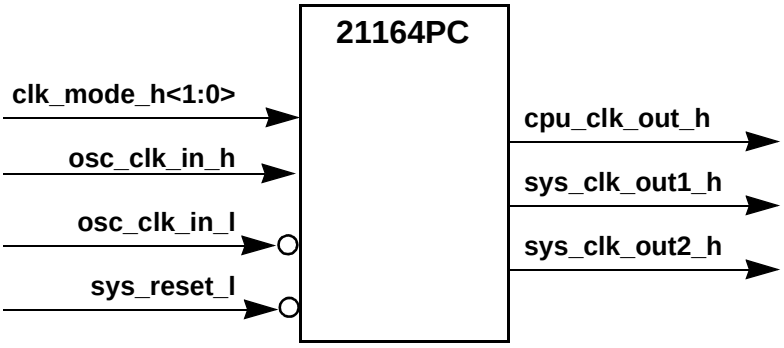
5.1 Clocks

The 21164PC accepts one clock signal input and develops three clock signal outputs:

Signal	Description
Input Clock Signals	
osc_clk_in_h,l	Differential inputs *normally* driven at the desired internal frequency.
Output Clock Signals	
cpu_clk_out_h	A 21164PC internal clock that may or may not drive the system clock.
sys_clk_out1_h	A clock of programmable speed supplied to the external interface.
sys_clk_out2_h	A delayed copy of **sys_clk_out1_h**. The delay is programmable and is an integer number of **cpu_clk_out_h** periods.

Figure 6 shows the 21164PC clock signals.

Figure 6 21164PC Clock Signals



5.1.1 CPU Clock

The 21164PC uses the differential input clock lines **osc_clk_in_h,l** as a source to generate its CPU clock. The input signals **clk_mode_h<1:0>** control generation of the CPU clock.

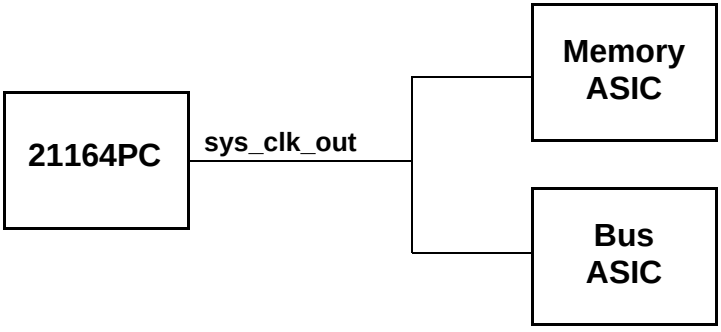
5.1.2 System Clock

The CPU clock is divided by a programmable value of 4 to 15 to generate a system clock. The programmable feature allows the system designer maximum flexibility when choosing external logic to interface with the 21164PC.

The **sys_clk_out1_h** signal is delayed by a programmable number of CPU cycles between 0 and 7 to produce **sys_clk_out2_h**. The output of the programmable divider is symmetric if the divisor is even. The output is asymmetric if the divisor is odd.

Figure 7 shows the 21164PC driving the system clock.

Figure 7 21164PC System Clock



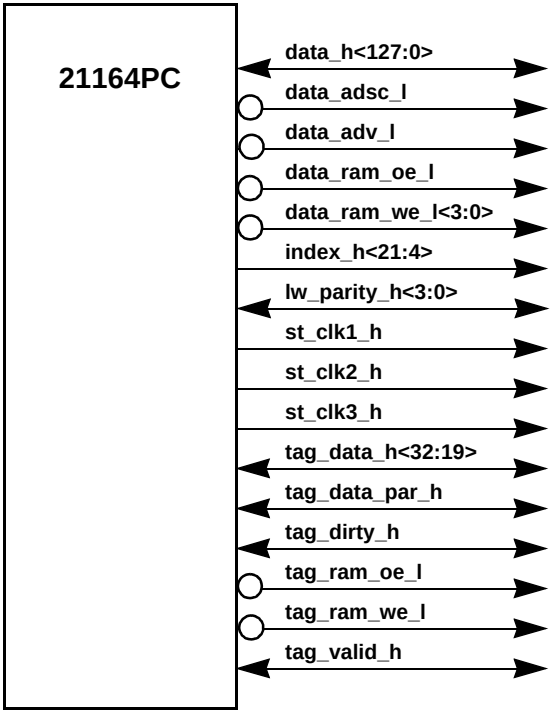
5.2 Board-Level Backup Cache Interface

The 21164PC includes an interface and control for a board-level backup cache (Bcache). This section describes the Bcache interface. The Bcache interface is made up of the following:

- A data bus (which it shares with the system interface)
- Tag and tag control bits for determining hit and coherence
- SRAM output and SRAM write control signals

Figure 8 shows the 21164PC system interface signals.

Figure 8 21164PC Bcache Interface Signals



The Bcache interface is managed by the cache control and bus interface unit (CBU). The Bcache interface is a 128-bit bidirectional data bus. The read and write speed of the Bcache can be programmed independently of each other and independently of the system clock ratio. Optionally, the Bcache can operate in a pseudo-pipeline manner.

Internal processor registers are used to program the Bcache timing and to enable wave pipelining. See the *Digital Semiconductor Alpha 21164PC Microprocessor Hardware Reference Manual* for more information.

The Bcache system supports a block size of 64 bytes.

5.2.1 Bcache Victim Buffers

The 21164PC is designed to support systems with one or more offchip Bcache victim buffers. At least one is required. External victim buffers improve the overall performance of the Bcache. A Bcache victim is generated when the 21164PC deallocates a dirty block from the Bcache. Each time a Bcache victim is produced, the 21164PC stops reading the Bcache until the system takes the current victim, and then the Bcache operations resume.

5.2.2 Cache Coherence Protocol

Cache coherency rules must be followed when designing 21164PC-based uniprocessor systems as there are two levels of caches on a processor module that may be snooped for data by I/O devices.

The system hardware designer need not be concerned about Icache and Dcache coherency. Coherency of the Icache is a software concern — it is flushed with an IMB (PALcode) instruction.

Table 5 describes the Bcache states that determine cache coherence protocol for 21164PC systems.

Table 5 Bcache States for Cache Coherency Protocols

Valid[1]	Dirty[1]	State of Cache Line
0	X	Not valid.
1	0	Valid for read or write operations. This cache line contains the only cached copy of the block, and the copy in memory is identical to this line.
1	1	Valid for read or write operations. This cache line contains the only cached copy of the block. The contents of the block have been modified more recently than the copy in memory.

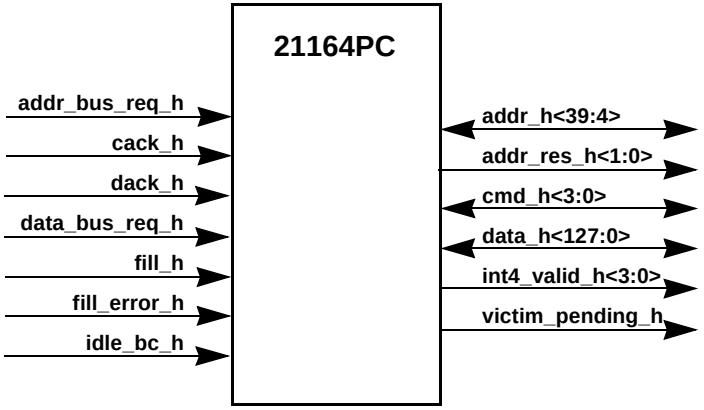
[1] The **tag_valid_h** and **tag_dirty_h** signals are described in Table 3.

5.3 System Interface

The system interface is made up of bidirectional address and command buses, a data bus that it shares with the Bcache interface, and several control signals.

Figure 9 shows the 21164PC system interface signals.

Figure 9 21164PC System Interface Signals



The system interface is under the control of the cache control and bus interface unit (CBU). The system interface is a 128-bit bidirectional data bus. The cycle time of the system interface is programmable to speeds of one-fourth to one-fifteenth the CPU cycle time. All system interface signals are driven or sampled by the 21164PC on the rising edge of **sys_clk_out1_h**.

5.3.1 Commands and Addresses

The 21164PC can take up to two commands from the system at a time. The bus interface buffer can hold one or two misses and one or two Bcache victim addresses at a time. A miss occurs when the 21164PC searches its caches but does not find the addressed block. The 21164PC can queue two misses to the system. A Bcache victim occurs when the 21164PC deallocates a dirty block from the Bcache.

The system requests the misses, and the victims arbitrate for the Bcache.

- The highest priority for the Bcache is data movement for the system, which includes fill, read dirty data, invalidate, and set shared activities.
- If there are no system requests for the Bcache, then a 21164PC command is selected.

Tables 6 and 7 provide a brief description of the commands that the 21164PC and the system can drive on the command bus.

Table 6 21164PC Commands for the System

cmd<3:0>	Command	Meaning
0000	NOP	Nothing.
0001	—	Reserved.
0010	—	Reserved.
0011	—	Reserved.
0100	—	Reserved.
0101	—	Reserved.
0110	WRITE BLOCK	Request to write a block.
0111	—	Reserved.
1000	READ MISS0	Request for data.
1001	READ MISS1	Request for data.
1010	—	Reserved.
1011	—	Reserved.
1100	BCACHE VICTIM	Bcache victim should be removed.
1101	—	Reserved.
1110	—	Reserved.
1111	—	Reserved.

Table 7 System Commands for the 21164PC

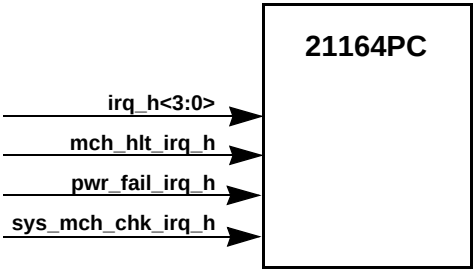
cmd<3:0>	Command	Meaning
0000	NOP	Nothing.
0001	FLUSH	Remove block from caches; return dirty data (flush protocol).
0010	INVALIDATE	Remove the block (write invalidate protocol).
0100	READ	Read a block (flush protocol).

5.4 Interrupts

The 21164PC has seven interrupt signals that have different uses during initialization and normal operation.

Figure 10 shows the 21164PC interrupt signals.

Figure 10 21164PC Interrupt Signals



5.4.1 Interrupt Signals During Initialization

The 21164PC interrupt signals work in tandem with the **sys_reset_l** signal to set the values for many of the user-selectable clocking ratios and interface timing parameters. During initialization, the 21164PC reads system clock configuration parameters from the interrupt pins.

Table 8 shows the system clock divisor settings. The system clock frequency is determined by dividing the ratio into the CPU clock frequency.

Table 8 System Clock Divisor

irq_h<3>	irq_h<2>	irq_h<1>	irq_h<0>	Ratio
Low	High	Low	Low	4
Low	High	Low	High	5
Low	High	High	Low	6
Low	High	High	High	7
High	Low	Low	Low	8
High	Low	Low	High	9
High	Low	High	Low	10
High	Low	High	High	11
High	High	Low	Low	12
High	High	Low	High	13
High	High	High	Low	14
High	High	High	High	15

Table 9 shows how the three remaining interrupt signals are used to determine the length of the **sys_clk_out2** delay. These signals provide flexible timing for system use.

Table 9 System Clock Delay *(Sheet 1 of 2)*

sys_mch_chk_irq_h	pwr_fail_irq_h	mch_hlt_irq_h	Delay Cycles
Low	Low	Low	0
Low	Low	High	1
Low	High	Low	2

Table 9 System Clock Delay *(Sheet 2 of 2)*

sys_mch_chk_irq_h	pwr_fail_irq_h	mch_hlt_irq_h	Delay Cycles
Low	High	High	3
High	Low	Low	4
High	Low	High	5
High	High	Low	6
High	High	High	7

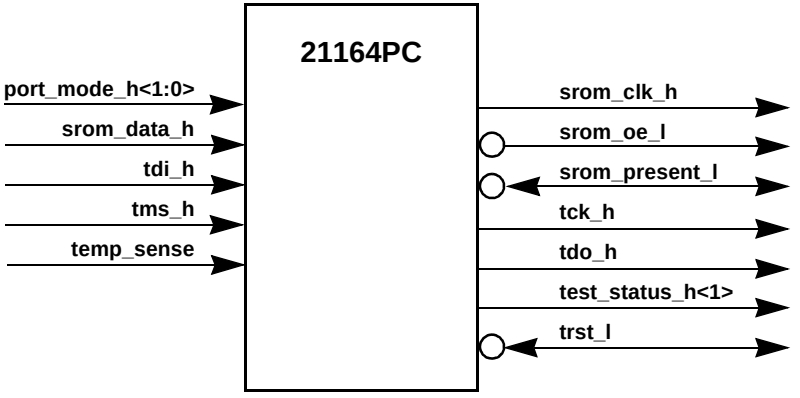
5.4.2 Interrupt Signals During Normal Operation

During normal operation, interrupt signals request various interrupts as described in Table 3.

5.5 Test Modes

Figure 11 shows the 21164PC test signals.

Figure 11 21164PC Test Signals



The 21164PC test interface port consists of 12 dedicated signals. Table 10 summarizes the 21164PC test port signals and their function.

Table 10 21164PC Test Port Pins

Pin Name	Type	Function
port_mode_h<1>	I	Must be false.
port_mode_h<0>	I	Must be false.
srom_present_l	B	Tied low if serial ROMs (SROMs) are present in system.
srom_data_h/Rx	I	Receives SROM or serial terminal data.
srom_clk_h/Tx	O	Supplies clock to SROMs or transmits serial terminal data.
srom_oe_l	O	SROM enable.
tdi_h	I	IEEE 1149.1 TDI port.
tdo_h	O	IEEE 1149.1 TDO port.
tms_h	I	IEEE 1149.1 TMS port.
tck_h	B	IEEE 1149.1 TCK port.
trst_l	B	IEEE 1149.1 optional TRST port.
test_status_h<1>	O	Outputs an IPR-written value and timeout reset.

5.5.1 Normal Test Interface Mode

The test port is in the default or normal test interface mode when the **port_mode_h<1:0>** signals are tied to 00. In this mode, the test port supports the following:

- Serial ROM interface port
- Serial diagnostic terminal interface port
- IEEE 1149.1 test access port

5.5.2 Serial ROM Interface Port

The following signals make up the serial ROM (SROM) interface:

srom_present_l
srom_data_h
srom_oe_l
srom_clk_h

During system reset, the 21164PC samples the **srom_present_l** signal for the presence of SROM. If no SROMs are detected at reset, then **srom_present_l** is deasserted and the SROM load is disabled. The reset sequence clears the Icache valid bits, which causes the first instruction fetch to miss the Icache and seek instructions from offchip memory.

If SROMs are present during setup, then the system performs an SROM load as follows:

1. The **srom_oe_l** signal supplies the output enable to the SROM.
2. The **srom_clk_h** signal supplies the clock to the ROM that causes it to advance to the next bit. The cycle time of this clock is 126± times the system clock ratio.
3. The **srom_data_h** signal reads the SROM data.

5.5.3 Serial Terminal Port

After the serial ROM data is loaded into the Icache, the three SROM load signals become parallel I/O pins that can drive a diagnostic terminal such as an RS422.

5.5.4 IEEE 1149.1 Test Access Port

The test access port complies with all requirements of the IEEE 1149.1 (JTAG) standard. The following signals make up the test access port:

- **tms_h** — Test access port select
- **trst_l** — Test access port reset
- **tck_h** — Test access port clock
- **tdi_h** and **tdo_h** — Input and output for serial boundary-scan, die-ID, bypass, and instruction registers

5.5.5 Test Status Signal

The **test_status_h<1>** signal detects unrepairable Icache by indicating more than two failing Icache rows.

6 Alpha Architecture Basics

This section provides some basic information about the Alpha architecture. For more detailed information about the Alpha architecture, see the *Alpha AXP Architecture Reference Manual*.

6.1 The Architecture

The Alpha architecture is a 64-bit load and store RISC architecture designed with particular emphasis on speed, multiple instruction issue, multiple processors, and software migration from many operating systems.

All registers are 64 bits long and all operations are performed between 64-bit registers. All instructions are 32 bits long. Memory operations are either load or store operations. All data manipulation is done between registers.

The Alpha architecture supports the following data types:

- 8-, 16-, 32-, and 64-bit integers
- IEEE 32-bit and 64-bit floating-point formats
- VAX architecture 32-bit and 64-bit floating-point formats

In the Alpha architecture, instructions interact with each other only by one instruction writing to a register or memory location and another instruction reading from that register or memory location. This use of resources makes it easy to build implementations that issue multiple instructions every CPU cycle.

The 21164PC uses a set of subroutines, called privileged architecture library code (PALcode), that is specific to a particular Alpha operating system implementation and hardware platform. These subroutines provide operating system primitives for context switching, interrupts, exceptions, and memory management. These subroutines can be invoked by hardware or CALL_PAL instructions. CALL_PAL instructions use the function field of the instruction to vector to a specified subroutine. PALcode is written in standard machine code with some implementation-specific extensions to provide direct access to low-level hardware functions. PALcode supports optimizations for multiple operating systems, flexible memory-management implementations, and multi-instruction atomic sequences.

The Alpha architecture performs byte shifting and masking with normal 64-bit, register-to-register instructions; it does not include single-byte load and store instructions.

6.2 Addressing

The basic addressable unit in the Alpha architecture is the 8-bit byte. The 21164PC supports a 43-bit virtual address.

Virtual addresses as seen by the program are translated into physical memory addresses by the memory-management mechanism. The 21164PC supports a 40-bit uncached and a 33-bit cached physical address space.

6.3 Integer Data Types

Alpha architecture supports four integer data types:

Data Type	Description
Byte	A byte is eight contiguous bits that start at an addressable byte boundary. A byte is an 8-bit value. A byte is supported in Alpha architecture by the EXTRACT, INSERT, LDBU, MASK, SEXTB, STB, ZAP, PACK, UNPACK, MIN, MAX, and PERR instructions.
Word	A word is two contiguous bytes that start at an arbitrary byte boundary. A word is a 16-bit value. A word is supported in Alpha architecture by the EXTRACT, INSERT, LDWU, MASK, SEXTW, STW, PACK, UNPACK, MIN, and MAX instructions.
Longword	A longword is four contiguous bytes that start at an arbitrary byte boundary. A longword is a 32-bit value. A longword is supported in Alpha architecture by sign-extended load and store instructions and by longword arithmetic instructions.
Quadword	A quadword is eight contiguous bytes that start at an arbitrary byte boundary. A quadword is supported in Alpha architecture by load and store instructions and quadword integer operate instructions.

Note: Alpha implementations may impose a significant performance penalty when accessing operands that are not NATURALLY ALIGNED. Refer to the *Alpha AXP Architecture Reference Manual* for details.

6.4 Floating-Point Data Types

The 21164PC supports the following floating-point data types:

- Longword integer format in floating-point unit
- Quadword integer format in floating-point unit
- IEEE floating-point formats
  - S_floating
  - T_floating
- VAX floating-point formats
  - F_floating
  - G_floating
  - D_floating (limited support)

7 IEEE Floating-Point Conformance

The 21164PC 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 AXP Architecture Reference Manual*.

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

The following information is specific to the 21164PC:

- 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 nonfinite (reserved operand or dirty zero) and the operation can take an exception (that is, certain instructions, such as CPYS, never take an exception). This exception is signaled if any IEEE operand is nonfinite (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 IDU.

- 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 nonzero, with a denominator of ±0. If the exception occurs, then FPCR<DZE> is set and the trap is signaled to the IDU.

  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, which 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 IDU.

- 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 IDU. The 21164PC never produces a denormal number; underflow occurs instead.

- Inexact (INE)

  The inexact trap can be disabled. The destination register always contains the properly rounded result, whether the trap is enabled or not. 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 and an integral result. If the exception occurs, then FPCR<INE> is set. If the trap is enabled, then the trap is signaled to the IDU.

  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 IDU because there is no CVTQL opcode that enables the inexact trap.)

- 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 integer 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 IDU.

- Software completion (SWC)

  The software completion signal is not recorded in the FPCR. The state of this signal is always sent to the IDU. If the IDU 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 IDU.

8 Internal Processor Registers

The tables in this section provide a summary of the 21164PC implementation-specific internal processor registers (IPRs). For detailed register information, see the *Digital Semiconductor Alpha 21164PC Microprocessor Hardware Reference Manual*. For more information about the architecturally specified IPRs, see the *Alpha AXP Architecture Reference Manual.*

8.1 IDU, MTU, Dcache, and PALtemp IPRs

Table 11 lists the IDU, MTU, data cache (Dcache), and PALtemp IPRs. These IPRs are accessible to PALcode by means of the HW_MTPR and HW_MFPR instructions, using the IPR index. The IDU holds a bank of 24 PALtemp registers.

Table 11 IDU, MTU, Dcache, and PALtemp IPRs *(Sheet 1 of 4)*

IPR Mnemonic	Register Name	Access	$Index_{16}$
IDU IPRs			
ISR	Interrupt summary	R	100
ITB_TAG	Istream translation buffer tag	W	101
ITB_PTE	Instruction translation buffer page table entry	R/W	102
ITB_ASN	Instruction translation buffer address space number	R/W	103
ITB_PTE_TEMP	Instruction translation buffer page table entry temporary	R	104
ITB_IA	Instruction translation buffer invalidate all	W	105
ITB_IAP	Instruction translation buffer invalidate all process	W	106
ITB_IS	Instruction translation buffer invalidate single	W	107
SIRR	Software interrupt request	R/W	108
ASTRR	Asynchronous system trap request	R/W	109
ASTER	Asynchronous system trap enable	R/W	10A
EXC_ADDR	Exception address	R/W	10B

Table 11 IDU, MTU, Dcache, and PALtemp IPRs *(Sheet 2 of 4)*

IPR Mnemonic	Register Name	Access	$Index_{16}$
EXC_SUM	Exception summary	R/W0C	10C
EXC_MASK	Exception mask	R	10D
PAL_BASE	Privileged architecture library base address	R/W	10E
ICM	IDU current mode	R/W	10F
IPLR	Interrupt priority level	R/W	110
INTID	Interrupt ID	R	111
IFAULT_VA_FORM	Formatted faulting virtual address	R	112
IVPTBR	Virtual page table base	R/W	113
HWINT_CLR	Hardware interrupt clear	W	115
SL_XMIT	Serial line transmit	W	116
SL_RCV	Serial line receive	R	117
ICSR	IDU control and status	R/W	118
IC_FLUSH_CTL	Icache flush control	W	119
ICPERR_STAT	Icache parity error status	R/W1C	11A
PMCTR	Performance counter	R/W	11C
PALtemp IPRs			
PALtemp0	—	R/W	140
PALtemp1	—	R/W	141
PALtemp2	—	R/W	142
PALtemp3	—	R/W	143
PALtemp4	—	R/W	144
PALtemp5	—	R/W	145
PALtemp6	—	R/W	146
PALtemp7	—	R/W	147
PALtemp8	—	R/W	148

Table 11 IDU, MTU, Dcache, and PALtemp IPRs *(Sheet 3 of 4)*

IPR Mnemonic	Register Name	Access	$Index_{16}$
PALtemp9	—	R/W	149
PALtemp10	—	R/W	14A
PALtemp11	—	R/W	14B
PALtemp12	—	R/W	14C
PALtemp13	—	R/W	14D
PALtemp14	—	R/W	14E
PALtemp15	—	R/W	14F
PALtemp16	—	R/W	150
PALtemp17	—	R/W	151
PALtemp18	—	R/W	152
PALtemp19	—	R/W	153
PALtemp20	—	R/W	154
PALtemp21	—	R/W	155
PALtemp22	—	R/W	156
PALtemp23	—	R/W	157
MTU IPRs			
DTB_ASN	Dstream translation buffer address space number	W	200
DTB_CM	Dstream translation buffer current mode	W	201
DTB_TAG	Dstream translation buffer tag	W	202
DTB_PTE	Dstream translation buffer page table entry	R/W	203
DTB_PTE_TEMP	Dstream translation buffer page table entry temporary	R	204
MM_STAT	Dstream memory-management fault status	R	205
VA	Faulting virtual address	R	206

Table 11 IDU, MTU, Dcache, and PALtemp IPRs *(Sheet 4 of 4)*

IPR Mnemonic	Register Name	Access	$Index_{16}$
VA_FORM	Formatted virtual address	R	207
MVPTBR	MTU virtual page table base	W	208
DTB_IAP	Dstream translation buffer invalidate all process	W	209
DTB_IA	Dstream translation buffer invalidate all	W	20A
DTB_IS	Dstream translation buffer invalidate single	W	20B
ALT_MODE	Alternate mode	W	20C
CC	Cycle counter	W	20D
CC_CTL	Cycle counter control	W	20E
MCSR	MTU control	R/W	20F
DC_FLUSH	Dcache flush	W	210
DC_PERR_STAT	Dcache parity error status	R/W1C	212
DC_TEST_CTL	Dcache test tag control	R/W	213
DC_TEST_TAG	Dcache test tag	R/W	214
DC_TEST_TAG_TEMP	Dcache test tag temporary	R/W	215
DC_MODE	Dcache mode	R/W	216
MAF_MODE	Miss address file mode	R/W	217

8.2 External Interface Control (CBU) IPRs

Table 12 summarizes IPRs for controlling Bcache, system configuration, and logging error information. These IPRs cannot be read or written from the system. They are placed in the 1-MB region of 21164PC-specific I/O address space ranging from FF FFF0 0000 to FF FFFF FFFF. Any read or write operation to an undefined IPR in this address space produces UNDEFINED behavior. The operating system should not map any address in this region as writable in any mode.

Table 12 CBU Internal Processor Register Descriptions

Mnemonic	Register Name	Access	Address
CBOX_CONFIG	CBU configuration	R/W	FF FFF0 0008
CBOX_ADDR	CBU address	R	FF FFF0 0088
CBOX_STATUS	CBU status	R	FF FFF0 0108
CBOX_CONFIG2	CBU configuration 2	R/W	FF FFF0 0188

8.3 PALcode Storage Registers

The 21164PC IEU register file has eight extra registers that are called the PALshadow registers. The PALshadow registers overlay R8 through R14 and R25 when the CPU is in PALmode and ICSR<SDE> is set. Thus, PALcode can consider R8 through R14 and R25 as local scratch. PALshadow registers cannot be written in the last two cycles of a PALcode flow. The normal state of the CPU is ICSR<SDE> = ON. PALcode disables SDE for the unaligned trap and for error flows.

9 PALcode

Privileged architecture library code (PALcode) is macrocode that provides an architecturally defined operating-system-specific programming interface that is common across all Alpha microprocessors. The actual implementation of PALcode differs for each operating system.

PALcode runs with privileges enabled, instruction stream (Istream) mapping disabled, and interrupts disabled. PALcode has privilege to use five special opcodes that allow functions such as physical data stream (Dstream) references and internal processor register (IPR) manipulation.

PALcode can be invoked by the following events:

- Reset
- System hardware exceptions (MCHK, ARITH)
- Memory-management exceptions
- Interrupts
- CALL_PAL instructions

9.1 PALcode Entry Points

PALcode is invoked at specific entry points. The 21164PC has two types of PALcode entry points:

- CALL_PAL entry points are used whenever the IDU encounters a CALL_PAL instruction in the Istream.
  - Privileged CALL_PAL instructions start at offset 2000_{16}.
  - Unprivileged CALL_PAL instructions start at offset 3000_{16}.
- Chip-specific trap entry points start PALcode.

9.1.1 PALcode Trap Entry Points

Table 13 shows the PALcode trap entry points and their offset from the PAL_BASE IPR. Entry points are listed from highest to lowest priority.

Table 13 PALcode Trap Entry Points

Entry Name	$Offset_{16}$	Description
RESET	0000	Reset
IACCVIO	0080	Istream access violation or sign check error on PC
INTERRUPT	0100	Interrupt: hardware, software, and AST
ITBMISS	0180	Istream TBMISS
DTBMISS_SINGLE	0200	Dstream TBMISS
DTBMISS_DOUBLE	0280	Dstream TBMISS during virtual page table entry (PTE) fetch
UNALIGN	0300	Dstream unaligned reference
DFAULT	0380	Dstream fault or sign check error on virtual address
MCHK	0400	Uncorrected hardware error
OPCDEC	0480	Illegal opcode
ARITH	0500	Arithmetic exception
FEN	0580	Floating-point operation attempted with: • Floating-point instructions (LD, ST, and operates) disabled through FPE bit in the ICSR IPR • Floating-point IEEE operation with data type other than S, T, or Q

9.2 Required PALcode Function Codes

Table 14 lists opcodes required for all Alpha implementations. The notation used is *oo.fffffff*, where *oo* is the hexadecimal 6-bit opcode and *fffffff* is the hexadecimal 26-bit function code.

Table 14 Required PALcode Function Codes

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

9.3 Opcodes Reserved for PALcode

Table 15 lists the opcodes reserved by the Alpha architecture for implementation-specific use. These opcodes are privileged and are only available in PALmode. Section 10.1.2 shows the opcodes reserved for PALcode.

Table 15 Opcodes Reserved for PALcode

Opcode	Architecture Mnemonic
1B	PAL1B
1F	PAL1F
1E	PAL1E
19	PAL19
1D	PAL1D

10 Alpha Instruction Summary

This section contains a summary of all Alpha architecture instructions. All values are in hexadecimal radix. Table 16 describes the contents of the Format and Opcode columns that are in Table 17.

Table 16 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 2 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.

Qualifiers for operate instructions are shown in Table 17. Qualifiers for IEEE and VAX floating-point instructions are shown in Tables 20 and 21, respectively.

Table 17 Architecture Instructions *(Sheet 1 of 8)*

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
AMASK	Opr	11.61	Determine byte/word instruction implementation
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

Table 17 Architecture Instructions *(Sheet 2 of 8)*

Mnemonic	Format	Opcode	Description
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
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	15.0AD	Convert G_floating to F_floating

Table 17 Architecture Instructions *(Sheet 3 of 8)*

Mnemonic	Format	Opcode	Description
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
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

Table 17 Architecture Instructions *(Sheet 4 of 8)*

Mnemonic	Format	Opcode	Description
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	18.80	Prefetch data
FETCH_M	Mfc	18.A0	Prefetch data, modify intent
IMPLVER	Opr	11.6C	Determine CPU type
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

Table 17 Architecture Instructions *(Sheet 5 of 8)*

Mnemonic	Format	Opcode	Description
LDBU	Mem	0A	Load zero-extended byte
LDF	Mem	20	Load F_floating
LDG	Mem	21	Load G_floating
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
LDWU	Mem	0C	Load zero-extended word
MAXSB8	Opr	1C.3E	Vector signed byte maximum
MAXSW4	Opr	1C.3F	Vector signed word maximum
MAXUB8	Opr	1C.3C	Vector unsigned byte maximum
MAXUW4	Opr	1C.3D	Vector unsigned word maximum
MB	Mfc	18.4000	Memory barrier
MF_FPCR	F-P	17.025	Move from floating-point control register
MINSB8	Opr	1C.3E	Vector signed byte minimum
MINSW4	Opr	1C.3F	Vector signed word minimum
MINUB8	Opr	1C.3C	Vector unsigned byte minimum
MINUW4	Opr	1C.3D	Vector unsigned word minimum
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

Table 17 Architecture Instructions *(Sheet 6 of 8)*

Mnemonic	Format	Opcode	Description
MSKWH	Opr	12.52	Mask word high
MSKWL	Opr	12.12	Mask word low
MT_FPCR	F-P	17.024	Move to floating-point control register
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
PERR	Opr	1C.31	Pixel error
PKLB	Opr	1C.37	Pack longwords to bytes
PKWB	Opr	1C.36	Pack words to bytes
RC	Mfc	18.E0	Read and clear
RET	Mbr	1A.2	Return from subroutine
RPCC	Mfc	18.C0	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
S8ADDQ	Opr	10.32	Scaled add quadword by 8
S8SUBL	Opr	10.1B	Scaled subtract longword by 8

Table 17 Architecture Instructions *(Sheet 7 of 8)*

Mnemonic	Format	Opcode	Description
S8SUBQ	Opr	10.3B	Scaled subtract quadword by 8
SEXTB	Opr	1C.00	Store byte
SEXTW	Opr	1C.01	Store word
SLL	Opr	12.39	Shift left logical
SRA	Opr	12.3C	Shift right arithmetic
SRL	Opr	12.34	Shift right logical
STB	Mem	0E	Store byte
STF	Mem	24	Store F_floating
STG	Mem	25	Store G_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
STS	Mem	26	Store S_floating
STT	Mem	27	Store T_floating
STW	Mem	0D	Store word
SUBF	F-P	15.081	Subtract F_floating
SUBG	F-P	15.0A1	Subtract G_floating
SUBL	Opr	10.09	Subtract longword
SUBL/V	—	10.49	—
SUBQ	Opr	10.29	Subtract quadword
SUBQ/V	—	10.69	—
SUBS	F-P	16.081	Subtract S_floating
SUBT	F-P	16.0A1	Subtract T_floating
TRAPB	Mfc	18.00	Trap barrier

Table 17 Architecture Instructions *(Sheet 8 of 8)*

Mnemonic	Format	Opcode	Description
UMULH	Opr	13.30	Unsigned multiply quadword high
UNPKBL	Opr	1C.35	Unpack bytes to longwords
UNPKBW	Opr	1C.34	Unpack bytes to words
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

10.1 Reserved Opcodes

This section describes the opcodes that are reserved in the Alpha architecture. They can be reserved for DIGITAL or for PALcode.

10.1.1 Opcodes Reserved for DIGITAL

Table 18 lists opcodes reserved for DIGITAL.

Table 18 Opcodes Reserved for DIGITAL

Mnemonic	Opcode	Mnemonic	Opcode	Mnemonic	Opcode
OPC01	01	OPC05	05	OPC0B	0B
OPC02	02	OPC06	06	OPC0C	0C[1]
OPC03	03	OPC07	07	OPC0D	0D[1]
OPC04	04	OPC0A	0A[1]	OPC0E	0E[1]

[1] Reserved when byte/word instructions are not enabled.

10.1.2 Opcodes Reserved for PALcode

Table 19 lists the 21164PC-specific instructions. For more information, refer to the *Digital Semiconductor Alpha 21164PC Microprocessor Hardware Reference Manual*.

Table 19 Opcodes Reserved for PALcode

21164PC Mnemonic	Opcode	Architecture Mnemonic	Function
HW_LD	1B	PAL1B	Performs Dstream load instructions.
HW_ST	1F	PAL1F	Performs Dstream store instructions.
HW_REI	1E	PAL1E	Returns instruction flow to the program counter (PC) pointed to by EXC_ADDR internal processor register (IPR).
HW_MFPR	19	PAL19	Accesses the IDU, MTU, and Dcache IPRs.
HW_MTPR	1D	PAL1D	Accesses the IDU, MTU, and Dcache IPRs.

10.2 IEEE Floating-Point Instructions

Table 20 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 20 IEEE Floating-Point Instruction Function Codes *(Sheet 1 of 3)*

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

Table 20 IEEE Floating-Point Instruction Function Codes *(Sheet 2 of 3)*

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

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	—	—	—	—	—	—	—

Mnemonic	**/SU**	**/SUC**	**/SUM**	**/SUD**	**/SUI**	**/SUIC**	**/SUIM**	**/SUID**
CVTQS	—	—	—	—	7BC	73C	77C	7FC
CVTQT	—	—	—	—	7BE	73E	77E	7F3
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

Table 20 IEEE Floating-Point Instruction Function Codes *(Sheet 3 of 3)*

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

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.

In the same manner, CVTQS and CVTQT can take an inexact result trap, but not an underflow. Because there is no encoding for a CVTQ*x*/SI instruction, it is intended that software generate CVTQ*x*/SUI in place of CVTQ*x*/SI.

10.3 VAX Floating-Point Instructions

Table 21 lists the hexadecimal value of the 11-bit function code field for the VAX floating-point instructions. The opcode for these instructions is 15_{16}.

Table 21 VAX Floating-Point Instruction Function Codes *(Sheet 1 of 2)*

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

Table 21 VAX Floating-Point Instruction Function Codes *(Sheet 2 of 2)*

Mnemonic	None	/C	/U	/UC	/S	/SC	/SU	/SUC
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

10.4 Opcode Summary

Table 22 lists all Alpha opcodes from 00 (CALL_PAL) through 3F (BGT). In the table, the column headings that appear over the instructions have a granularity of 8_{16}. The rows beneath the Offset column supply the individual hexadecimal number to resolve that granularity.

If an instruction column has a 0 in the right (low) hexadecimal digit, replace that 0 with the number to the left of the slash (/) in the Offset column on the instruction's row. If an instruction column has an 8 in the right (low) hexadecimal digit, replace that 8 with the number to the right of the slash in the Offset column.

For example, the third row (2/A) under the 10_{16} column contains the symbol INTS*, representing the all-integer shift instructions. The opcode for those instructions would then be 12_{16} because the 0 in 10 is replaced by the 2 in the Offset column. Likewise, the third row under the 18_{16} column contains the symbol JSR*, representing all jump instructions. The opcode for those instructions is 1A because the 8 in the heading is replaced by the number to the right of the slash in the Offset column.

The instruction format is listed under the instruction symbol.

Table 22 Opcode Summary

Offset	00	08	10	18	20	28	30	38
0/8	PAL* (pal)	LDA (mem)	INTA* (op)	MISC* (mem)	LDF (mem)	LDL (mem)	BR (br)	BLBC (br)
1/9	Res	LDAH (mem)	INTL* (op)	\PAL\	LDG (mem)	LDQ (mem)	FBEQ (br)	BEQ (br)
2/A	Res	LDBU (mem)	INTS* (op)	JSR* (mem)	LDS (mem)	LDL_L (mem)	FBLT (br)	BLT (br)
3/B	Res	LDQ_U (mem)	INTM* (op)	\PAL\	LDT (mem)	LDQ_L (mem)	FBLE (br)	BLE (br)
4/C	Res	LDWU (mem)	Res	SEXT/ MVI* (op)	STF (mem)	STL (mem)	BSR (br)	BLBS (br)
5/D	Res	STW (mem)	FLTV* (op)	\PAL\	STG (mem)	STQ (mem)	FBNE (br)	BNE (br)
6/E	Res	STB (mem)	FLTI* (op)	\PAL\	STS (mem)	STL_C (mem)	FBGE (br)	BGE (br)
7/F	Res	STQ_U (mem)	FLTL* (op)	\PAL\	STT (mem)	STQ_C (mem)	FBGT (br)	BGT (br)

Symbol	Meaning
FLTI*	IEEE floating-point instruction opcodes
FLTL*	Floating-point operate instruction opcodes
FLTV*	VAX floating-point instruction opcodes
INTA*	Integer arithmetic instruction opcodes
INTL*	Integer logical instruction opcodes
INTM*	Integer multiply instruction opcodes
INTS*	Integer shift instruction opcodes
JSR*	Jump instruction opcodes
MISC*	Miscellaneous instruction opcodes
PAL*	PALcode instruction (CALL_PAL) opcodes
\PAL\	Reserved for PALcode
Res	Reserved for DIGITAL
SEXT/MVI*	Sign extend and motion video instruction set opcodes

10.5 Required PALcode Function Codes

The opcodes listed in Table 23 are required for all Alpha implementations. The notation used is oo.fffffff, where oo is the hexadecimal 6-bit opcode and fffffff is the hexadecimal 26-bit function code.

Table 23 Required PALcode Function Codes

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

11 Electrical Data

This section describes the electrical characteristics of the 21164PC component and its interface pins. It is organized as follows:

- Electrical characteristics
- DC characteristics
- Clocking scheme
- AC characteristics
- Power supply considerations

11.1 Electrical Characteristics

Table 24 lists the maximum ratings for the 21164PC and Table 25 lists the operating voltages.

Table 24 21164PC Absolute Maximum Ratings

Characteristics	Ratings
Storage temperature	–55°C to125°C (-67°F to 257°F)
Junction temperature	15°C to 85°C (59°F to 185°F)
Supply voltage	**Vss** = –0.5 V, **Vddi** = 2.5 V, **Vdd** = 3.3 V
Signal input or output applied	–0.5 V to 4.6 V
Typical **Vdd** worst case power @ **Vdd** = 3.3 V	
Frequency = 400 MHz	2.5 W
Frequency = 466 MHz	2.5 W
Frequency = 533 MHz	3.0 W
Typical **Vddi** worst case power @ **Vddi** = 2.5 V	
Frequency = 400 MHz	24 W
Frequency = 466 MHz	28 W
Frequency = 533 MHz	32 W

Caution: Stress beyond the absolute maximum rating can cause permanent damage to the 21164PC. Exposure to absolute maximum rating conditions for extended periods of time can affect the 21164PC reliability.

Table 25 Operating Voltages

	Nominal	Maximum	Minimum
Vdd	3.3 V	3.46 V	3.13 V
Vddi	2.5 V	2.6 V	2.4 V

11.2 DC Characteristics

The 21164PC is designed to run in a 3.3-V CMOS/TTL environment. The 21164PC is tested and characterized in a CMOS environment.

11.2.1 Power Supply

The **Vss** pins are connected to 0.0 V, the **Vddi** pins are connected to 2.5 V ±0.1 V, and the **Vdd** pins are connected to 3.3 V ±5%.

11.2.2 Input Signal Pins

Nearly all input signals are ordinary CMOS inputs with standard TTL levels (see Table 26). (See Section 11.3.1 for a description of an exception — **osc_clk_in_h,l**.)

After power has been applied, input and bidirectional pins can be driven to a maximum dc voltage of **Vclamp** at a maximum current of **Iclamp** without harming the 21164PC. Refer to Table 26 for **Vclamp** and **Iclamp** values. Inputs greater than **Vclamp** will be clamped to **Vclamp** provided that the current does not exceed **Iclamp**. The 21164PC may be damaged if the voltage exceeds **Vclamp** or the current exceeds **Iclamp**.

11.2.3 Output Signal Pins

Output pins are ordinary 3.3-V CMOS outputs. Although output signals are rail-to-rail, timing is specified to **Vdd**/2.

Note: The 21164PC microprocessor chips do not have an onchip resistor for an output driver.

Bidirectional pins are either input or output pins, depending on control timing. When functioning as output pins, they are ordinary 3.3-V CMOS outputs.

Table 26 shows the CMOS dc input and output pins.

Table 26 CMOS DC Input/Output Characteristics *(Sheet 1 of 2)*

Parameter		Requirements			
Symbol	**Description**	**Min.**	**Max.**	**Units**	**Test Conditions**
Vih	High-level input voltage	2.0	—	V	—
Vil	Low-level input voltage	—	0.8	V	—
Voh	High-level output voltage	2.4	—	V	**Ioh** = –6.0 mA
Vol	Low-level output voltage	—	0.4	V	**Iol** = 6.0 mA
Iil_pd	Input with pull-down leakage current	—	±50	µA	**Vin** = 0 V
Iih_pd	Input with pull-down current	—	250	µA	**Vin** = 2.4 V
Iil_pu	Input with pull-up current	—	–800	µA	**Vin** = 0.4 V
Iih_pu	Input with pull-up leakage current	—	±50	µA	**Vin** = **Vdd** V
Iozl_pd	Output with pull-down leakage current (tristate)	—	±100	µA	**Vin** = 0 V
Iozh_pd	Output with pull-down current (tristate)	—	500	µA	**Vin** = 2.4 V
Iozl_pu	Output with pull-up current (tristate)	—	–800	µA	**Vin** = 0.4 V
Iozh_pu	Output with pull-up leakage current (tristate)	—	±100	µA	**Vin** = **Vdd** V
Vclamp	Maximum clamping voltage	—	**Vdd** + 1.0	V	**Iclamp** = 100 mA
Idd	Peak power supply current for **Vdd** power supply	—	1.0[1]	A	**Vdd** = 3.465 V Frequency = 400 MHz
Idd	Peak power supply current for **Vdd** power supply	—	1.0[1]	A	**Vdd** = 3.465 V Frequency = 466 MHz
Idd	Peak power supply current for **Vdd** power supply	—	1.3[1]	A	**Vdd** = 3.465 V Frequency = 533 MHz

Table 26 CMOS DC Input/Output Characteristics *(Sheet 2 of 2)*

Parameter		Requirements			
Symbol	**Description**	**Min.**	**Max.**	**Units**	**Test Conditions**
Iddi	Peak power supply current for **Vddi** power supply	—	11.25	A	**Vddi** = 2.6 V Frequency = 400 MHz
Iddi	Peak power supply current for **Vddi** power supply	—	13.00	A	**Vddi** = 2.6 V Frequency = 466 MHz
Iddi	Peak power supply current for **Vddi** power supply	—	14.75	A	**Vddi** = 2.6 V Frequency = 533 MHz

[1] This assumes a **sysclk** ratio of 4 and worst-case loading of output pins.

Most pins have low current pull-down devices to **Vss**. However, two pins have a pull-up device to **Vdd**. The pull-downs (or pull-ups) are always enabled. This means that some current will flow from the 21164PC (if the pin has a pull-up device) or into the 21164PC (if the pin has a pull-down device) even when the pin is in the high-impedance state. All pins have pull-down devices, except for the pins in the following table:

Signal Name	Notes
tms_h	Has a pull-up device
tdi_h	Has a pull-up device
osc_clk_in_h	50 Ω to **Vterm** (≈ **Vdd**/2) (See Figure 12)
osc_clk_in_l	50 Ω to **Vterm** (≈ **Vdd**/2) (See Figure 12)
temp_sense	150 Ω to **Vss**

11.3 Clocking Scheme

The differential input clock signals **osc_clk_in_h,l** run at the internal frequency of the time base for the 21164PC. The output signal **cpu_clk_out_h** toggles with an unspecified propagation delay relative to the transitions on **osc_clk_in_h,l**.

The 21164PC provides a system clock to run the chip synchronous to the system.

The 21164PC generates and drives out a system clock, **sys_clk_out1_h**. It runs synchronous to the system clock at a selected ratio of the internal clock frequency. There is a small clock skew between the internal clock and **sys_clk_out1_h**.

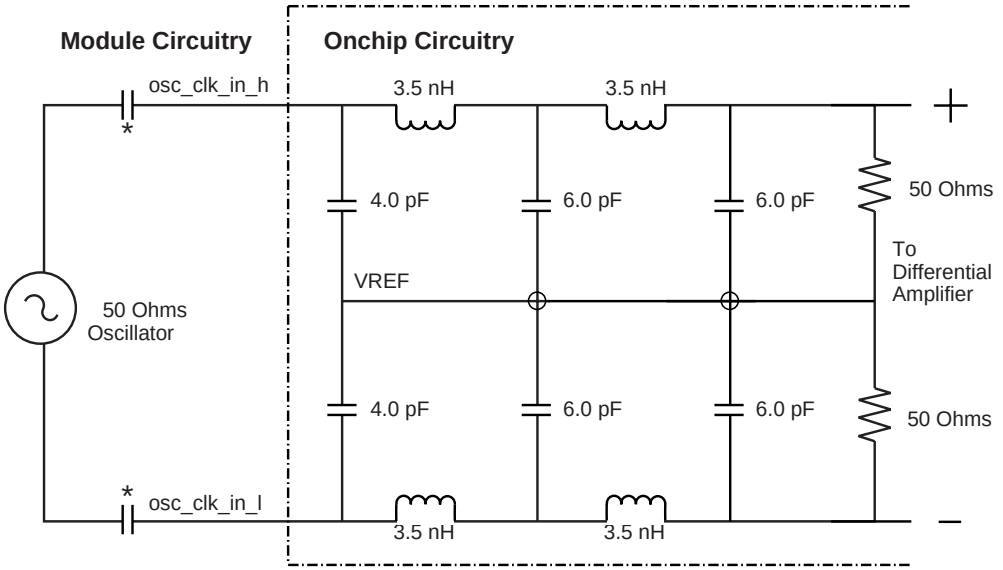
11.3.1 Input Clocks

The differential input clocks **osc_clk_in_h,l** provide the time base for the chip when **dc_ok_h** is asserted. These pins are self-biasing, and must be capacitively coupled to the clock source on the module.

Note: It is not desirable to drive the **osc_clk_in_h,l** pins directly.

The terminations on these signals are designed to be compatible with system oscillators of arbitrary dc bias. The oscillator must have a duty cycle of 60%/40% or tighter. Figure 12 shows the input network and the schematic equivalent of **osc_clk_in_h,l** terminations.

Figure 12 osc_clk_in_h,l Input Network and Terminations



Note:
* Coupling capacitors 47 pF to 220 pF

LJ-05357.AI4

Ring Oscillator

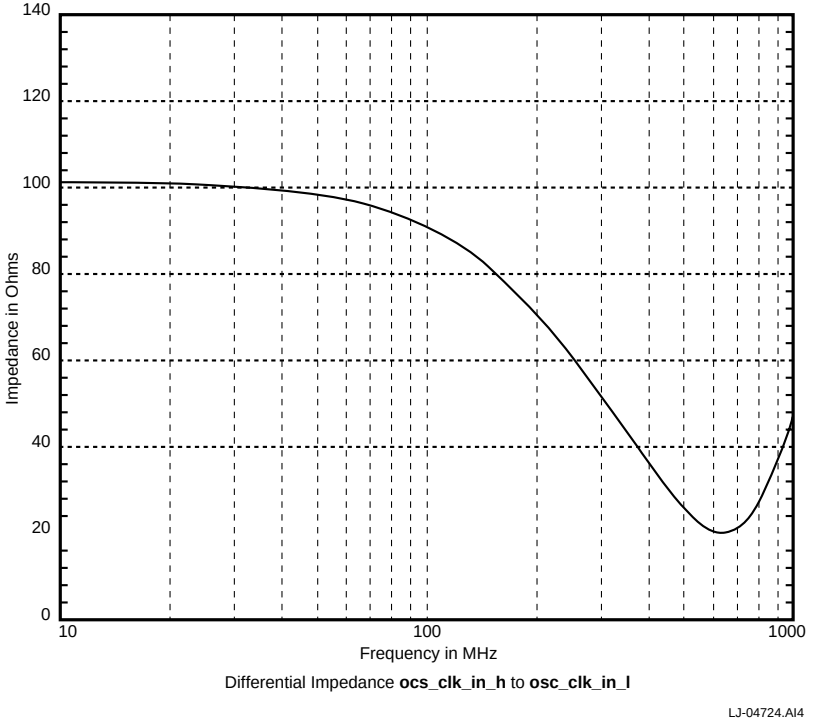
When signal **dc_ok_h** is deasserted, the clock outputs follow the internal ring oscillator. The 21164PC runs off the ring oscillator, just as it would when an external clock is applied. The frequency of the ring oscillator varies from chip to chip within a range of 10 MHz to 100 MHz. This corresponds to an internal CPU clock frequency range of 5 MHz to 50 MHz. The system clock divisor is forced to 8, and the **sys_clk_out2** delay is forced to 3.

11.3.2 Clock Termination and Impedance Levels

In Figure 12, the clock is designed to approximate a 50-Ω termination for the purpose of impedance matching for those systems that drive input clocks across long traces. The clock input pins appear as a 50-Ω series termination resistor connected to a high impedance voltage source. The voltage source produces a nominal voltage value of **Vdd**/2. The source has an impedance of between 130 Ω and 600 Ω. This voltage is called the self-bias voltage and sources current when the applied voltage at the clock input pins is less than the self-bias voltage. It sinks current when the applied voltage exceeds the self-bias voltage. This high impedance bias driver allows a clock source of arbitrary dc bias to be ac coupled to the 21164PC. The peak-to-peak amplitude of the clock source must be between 0.6 V and 3.0 V. Either a square-wave or a sinusoidal source may be used. Full-rail clocks may be driven by testers. In any case, the oscillator should be ac coupled to the **osc_clk_in_h,l** inputs by 47-pF through 220-pF capacitors.

Figure 13 shows a plot of the simulated impedance versus the clock input frequency. Figure 12 is a simplified circuit of the complex model used to create Figure 13.

Figure 13 Impedance Versus Clock Input Frequency



11.3.3 AC Coupling

Using series coupling (blocking) capacitors renders the 21164PC clock input pins insensitive to the oscillator's dc level. When connected this way, oscillators with any dc offset relative to **Vss** can be used provided they can drive a signal into the **osc_clk_in_h,l** pins with a peak-to-peak level of at least 600 mV, but no greater than 3.0 V peak-to-peak.

The value of the coupling capacitor is not overly critical. However, it should be sufficiently low impedance at the clock frequency so that the oscillator's output signal (when measured at the **osc_clk_in_h,l** pins) is not attenuated below the 600-mV, peak-to-peak lower limit. For sine waves or oscillators producing nearly sinusoidal (pseudo square wave) outputs, 220 pF is recommended at 533 MHz. A high-quality dielectric such as NPO is required to avoid dielectric losses.

Table 27 shows the input clock specification.

Table 27 Input Clock Specification

Signal Parameter	Nominal Bin[1]	Unit
osc_clk_in_h,l symmetry	50 ± 10	%
osc_clk_in_h,l minimum voltage	0.6	V (peak-to-peak)
osc_clk_in_h,l Z input	50	Ω

[1] Minimum clock frequency = 50.0 MHz; Maximum clock frequency = 533 MHz = 1/Tcycle

11.4 AC Characteristics

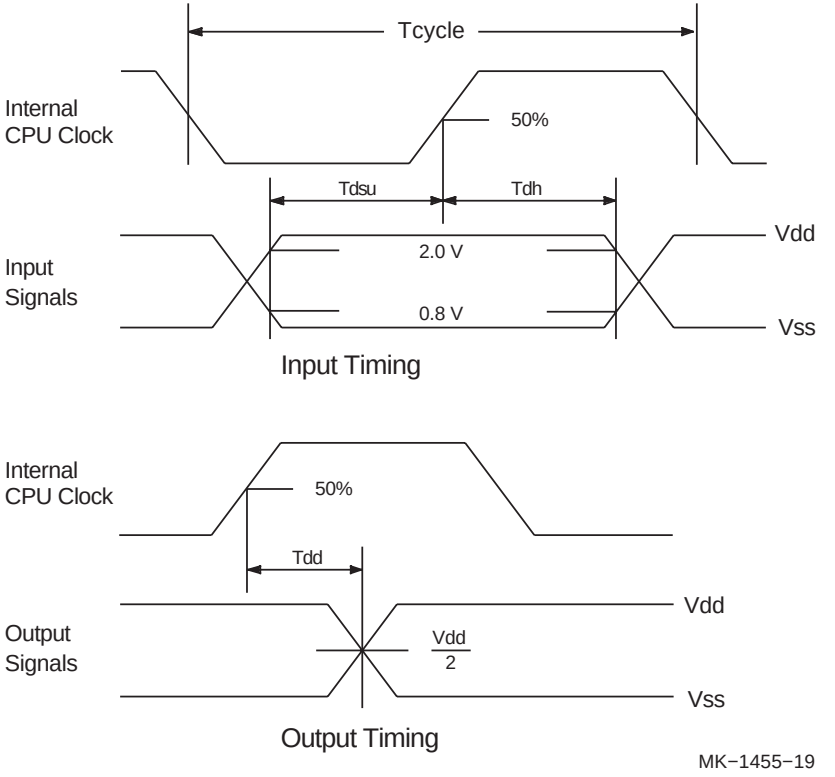
This section describes the ac timing specifications for the 21164PC.

11.4.1 Test Configuration

All input timing is specified relative to the crossing of standard TTL input levels of 0.8 V and 2.0 V. Output timing is to the nominal CMOS switch point of **Vdd**/2 (see Figure 14).

Because the speed and complexity of microprocessors has increased substantially over the years, it is necessary to change the way they are tested. Traditional assumptions that all loads can be lumped into some accumulation of capacitance cannot be employed any more. Rather, the model of a transmission line with discrete loads is a much more realistic approach for current test technology.

Figure 14 Input/Output Pin Timing



Typically, printed circuit board (PCB) etch has a characteristic impedance of approximately 75 Ω. This may vary from 60 Ω to 90 Ω with tolerances. If the line is driven in the electrical center, the load could be as low as 30 Ω. Therefore, a characteristic impedance range of 30 Ω to 90 Ω could be experienced.

The 21164PC output drivers are designed with typical printed circuit board applications in mind rather than trying to accommodate a 40-pF test load specification. As such, it “launches” a voltage step into a characteristic impedance, ranging from 30 Ω to 90 Ω.

There is no source termination resistor in the 21164PC fabricated in 0.35-µm CMOS process technology. The source impedance of the driver is approximately 32 Ω ±17. The circuit is designed to deliver a TTL signal under worst-case conditions. Under light load, high drive voltages, and fast process conditions there may be considerable overdrive. It may be necessary to install termination or clamping elements to the signal etches or loads.

11.4.2 Pin Timing

The following sections describe Bcache loop timing and sys_clk-based system timing.

11.4.2.1 Backup Cache Loop Timing

The 21164PC must be configured to support an offchip backup cache (Bcache). Private Bcache read or write transactions initiated by the 21164PC are independent of the system clocking scheme. Bcache loop timing must be an integer multiple of the 21164PC cycle time.

Table 28 lists the Bcache loop timing.

Table 28 Bcache Loop Timing

Signal	Specification	Value	Name
data_h<127:0>	Input setup	1.1 ns	**Tdsu**
data_h<127:0>	Input hold	0.0 ns	**Tdh**
data_h<127:0>	Output delay	**Tdd** + 0.2 ns[1]	**Tdbd**[3]
data_h<127:0>	Output hold	**Tmdd**	**Tdbh**[3]
index_h<21:4>, **st_clk1_h**, **st_clk2_h**, **st_clk3_h**	Output delay	**Tbedd** + 0.2 ns, or **Tbddd** + 0.2 ns[1,2]	**Tiod**
index_h<21:4>, **st_clk1_h**, **st_clk2_h**, **st_clk3_h**	Output hold time	**Tmdd**	**Tioh**

[1] The value 0.2 ns accounts for onchip driver and clock skew.
[2] For big drive enabled or big drive disabled, respectively. See Table 30.
[3] For private Bcache write operations, 21164PC drives **data_h<127:0>** coincident with driving **index_h<21:4>**.

Outgoing Bcache index and data signals are driven off the internal clock edge, and the incoming Bcache tag and data signals are latched on the same internal clock edge. Table 29 and Table 30 show the output driver characteristics for the normal driver and big driver, respectively.

Additional drive for the following pins can be enabled by setting bit <27> (BC_BIG_DRV) of the CBOX_CONFIG register:

- **index_h<21:4>**
- **tag_ram_oe_l**, **tag_ram_we_l**
- **data_ram_oe_l**, **data_ram_we_l<3:0>**
- **st_clk1_h**, **st_clk2_h, st_clk3_h**
- **data_adsc_l, data_adv_l**

If any of the previous pins are connected to lightly loaded lines (less than 40 pF), additional drive should not be enabled or the lines should be properly terminated to avoid transmission line ringing.

Table 29 Normal Output Driver Characteristics

Specification	40-pF Load	10-pF Load	Name
Maximum driver delay	2.7 ns	1.4 ns	**Tdd**
Minimum driver delay	0.8 ns	0.8 ns	**Tmdd**

Table 30 Big Output Driver Characteristics

Specification	60-pF Load	40-pF Load	10-pF Load	Name
Extra Drive Disabled				
Maximum driver delay	NA[1]	2.6 ns	1.5 ns	**Tbddd**
Minimum driver delay	NA[1]	0.8 ns	0.8 ns	**Tmdd**
Extra Drive Enabled				
Maximum driver delay	2.7 ns	2.0 ns	1.5 ns	**Tbedd**
Minimum driver delay	1.0 ns	0.8 ns	0.8 ns	**Tmdd**

[1] NA = Not applicable.

Output pin timing is specified for lumped 40-pF and 10-pF loads for the normal driver and lumped 60-pF, 40-pF, and 10-pF loads for the big driver. In some cases, the circuit may have loads higher than 40 pF (60 pF for big driver). The 21164PC can safely drive higher loads provided the average charging or discharging current

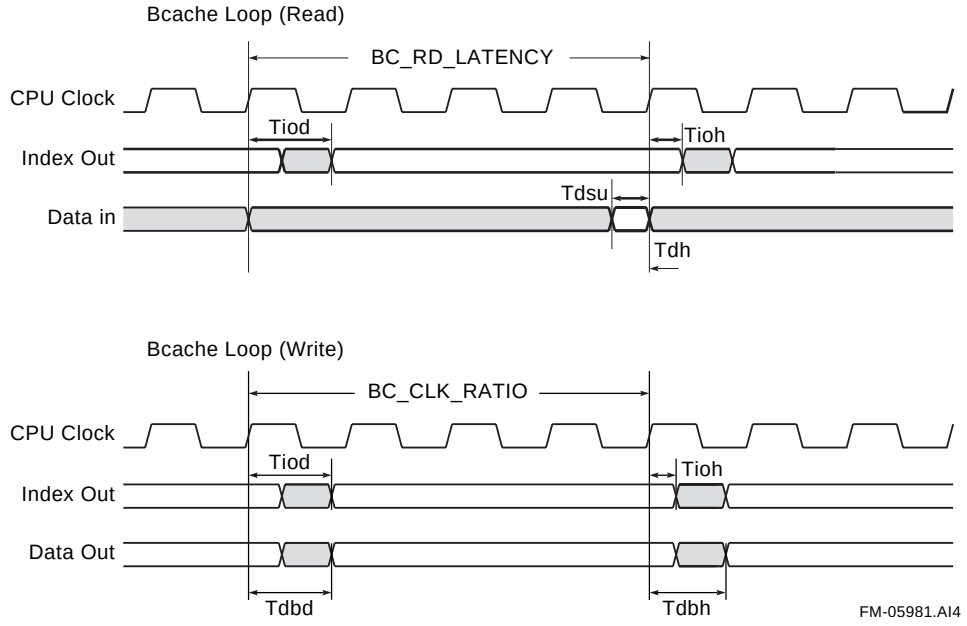
from each pin is 11 mA or less for normal output drivers or 25 mA or less for big output drivers. The following equation can be used to determine the maximum capacitance that can be safely driven by each pin:

- For normal output drivers: C_{max} (in pF) = 5t, where t is the waveform period (measured from rising to rising or falling to falling edge), in nanoseconds.
- For big output drivers: C_{max} (in pF) = 7t, where t is the waveform period (measured from rising to rising or falling to falling edge), in nanoseconds.

For example, if the waveform appearing on a given normal I/O pin has a 15.0-ns period, it can safely drive up to and including 75 pF.

Figure 15 shows the Bcache read and write timing.

Figure 15 Bcache Timing



11.4.2.2 sys_clk-Based Systems

All timing is specified relative to the rising edge of the internal CPU clock.

Table 31 shows 21164PC system clock **sys_clk_out1_h** output timing. Setup and hold times are specified independent of the relative capacitive loading of **sys_clk_out1_h,l**, **addr_h<39:4>**, **data_h<127:0>**, and **cmd_h<3:0>** signals.

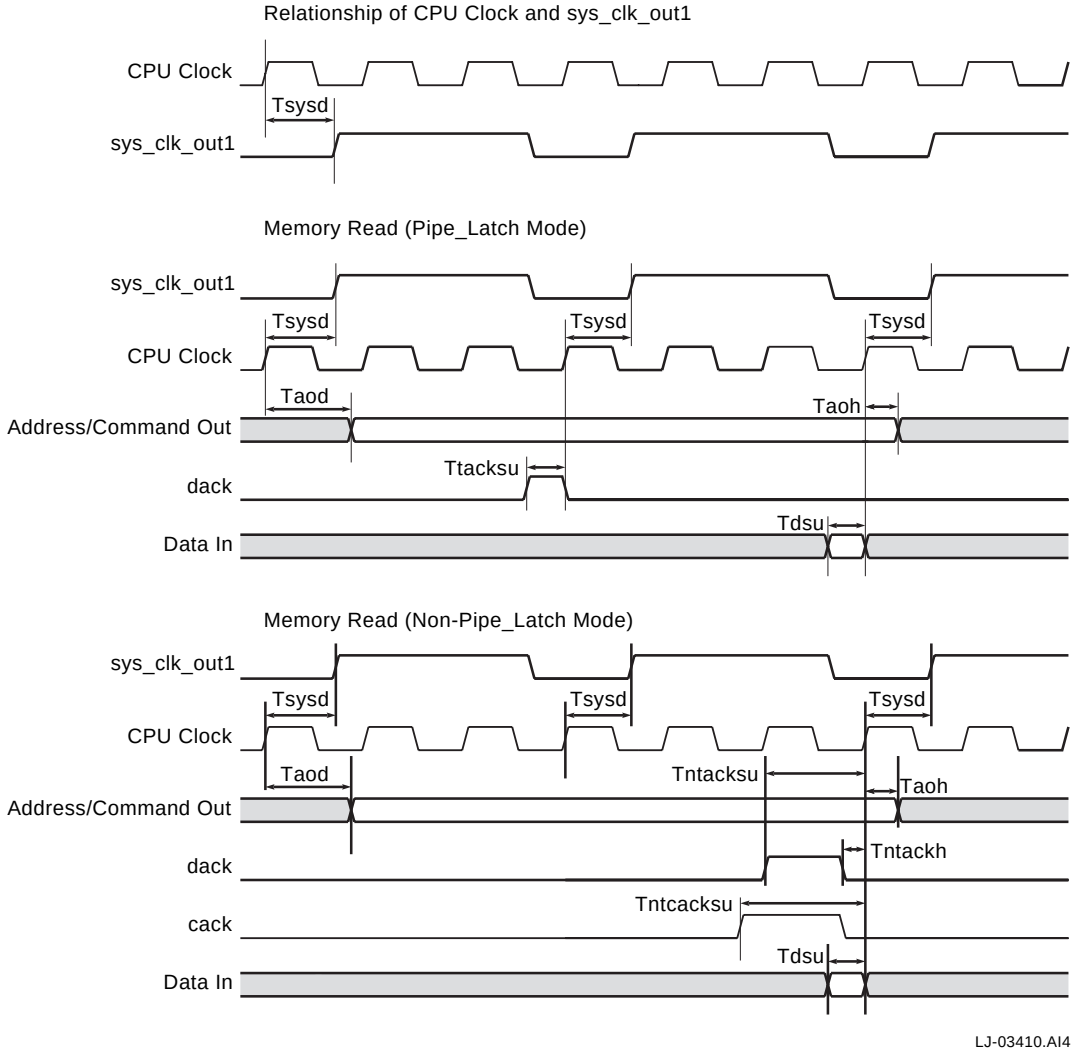
Table 31 21164PC System Clock Output Timing (sysclk=$T_{ø}$)

Signal	Specification	Value	Name
sys_clk_out1_h	Output delay	**Tdd**	**Tsysd**
sys_clk_out1_h	Minimum output delay	**Tmdd**	**Tsysdm**
data_bus_ req_h, data_h<127:0>, addr_h<39:4>	Input setup	1.1 ns	**Tdsu**
data_bus_ req_h, data_h<127:0>, addr_h<39:4>	Input hold	0 ns	**Tdh**
addr_h<39:4>	Output delay	**Tdd** + 0.2 ns[1]	**Taod**
addr_h<39:4>	Output hold time	**Tmdd**	**Taoh**
data_h<127:0>	Output delay	**Tdd** [+ **Tcycle**][2] + 0.2 ns[1]	**Tdod**[2]
data_h<127:0>	Output hold time	**Tmdd** [+ **Tcycle**][2]	**Tdoh**[2]
addr_bus_req_h	Input setup	3.4 ns	**Tabrsu**
addr_bus_req_h	Input hold	–1.0 ns	**Tabrh**
dack_h	Input setup	3.2 ns	**Tntacksu**
cack_h	Input setup	3.4 ns	**Tntcacksu**
cack, dack	Input hold	–1.0 ns	**Tntackh**

[1] The value 0.2 ns accounts for onchip driver and clock skew.
[2] For all system write transactions initiated by the 21164PC, data is driven **Tcycle (=** 1 cpu_clk) after the **sys_clk_out1_h** pin. For all private write transactions, data is driven coincident with (**Tcycle** = 0 cpu_clk) the driving of **index_h<21:4>.**

Figure 16 shows sys_clk system timing.

Figure 16 sys_clk System Timing



11.4.3 Timing — Additional Signals

This section lists timing for all other signals.

11.4.3.1 Asynchronous Input Signals

The following is a list of the asynchronous input signals:

clk_mode_h<1:0>
dc_ok_h
sys_reset_l
irq_h<3:0>[1]
mch_hlt_irq_h[1]
pwr_fail_irq_h[1]
sys_mch_chk_irq_h[1]

[1] These signals can also be used synchronously.

11.4.3.2 Miscellaneous Signals

Table 32 and Table 33 list the timing for miscellaneous input-only and output-only signals. All timing is expressed in nanoseconds.

Table 32 Input Timing for sys_clk_out-Based Systems

Signal	Specification	Value	Name
fill_h, fill_error_h, fill_id_h, idle_bc_h **irq_h<3:0>, mch_hlt_irq_h, pwr_fail_irq_h, sys_mch_chk_irq_h** Testability pins: **port_mode_h, srom_data_h, srom_present_l**	Input setup	1.1 ns	**Tdsu**
fill_h, fill_error_h, fill_id_h, idle_bc_h **irq_h<3:0>, mch_hlt_irq_h, pwr_fail_irq_h, sys_mch_chk_irq_h** **sys_reset_l** Testability pins: **port_mode_h, srom_data_h, srom_present_l**	Input hold	0 ns	**Tdh**

Table 33 Output Timing for sys_clk_out-Based Systems

Signal	Specification	Value	Name
Unidirectional Signals			
addr_res_h, **int4_valid_h**,[1] **srom_clk_h**, **srom_oe_l**, **victim_pending_h**	Output delay	**Tdd** + 0.2 ns	**Taod**
addr_res_h, **int4_valid_h**,[1] **srom_clk_h**, **srom_oe_l**, **victim_pending_h**	Output hold	**Tmdd**	**Taoh**
int4_valid_h[2]	Output delay	**Tdd** + **Tcycle** + 0.2 ns	**Tdod**
int4_valid_h[2]	Output hold	**Tmdd** + **Tcycle**	**Tdoh**
Bidirectional Signals			
Input mode:			
cmd_h, **lw_parity_h**,[1] **tag_dirty_h**[3]	Input setup	1.1 ns	**Tdsu**
cmd_h, **lw_parity_h**,[1] **tag_dirty_h**[3]	Input hold	0 ns	**Tdh**
Output mode:			
cmd_h, **tag_dirty_h**,[4] **tag_valid_h**[4]	Output delay	**Tdd** + 0.2 ns	**Taod**
lw_parity_h[2]	Output delay	**Tdd** + **Tcycle** + 0.2 ns	**Tdod**
cmd_h, **tag_dirty_h**,[4] **tag_valid_h**[4]	Output hold	**Tmdd**	**Taoh**
lw_parity_h[2]	Output hold	**Tmdd** + **Tcycle**	**Tdoh**

[1] Read transaction
[2] Write transaction
[3] Fills from memory
[4] Only for write broadcasts and system transactions

Signals in Table 34 are used to control Bcache data transfers. These signals are driven off the CPU clock. The timing of these signals does not change when switching over to the **sys_clk_out** timing domain.

Table 34 Bcache Control Signal Timing

Signal	Specification	Value	Name
Input mode:			
tag_data_h, **tag_data_par_h**, **tag_valid_h**	Input setup	1.1 ns	**Tdsu**
tag_data_h, **tag_data_par_h**, **tag_valid_h**	Input hold	0 ns	**Tdh**
Output mode:			
data_ram_oe_l, **data_ram_we_l<3:0>**, **tag_ram_oe_l**, **tag_ram_we_l**	Output delay	**Tbedd** + 0.2 ns or **Tbddd** + 0.2 ns[1,2]	**Taod**
tag_data_h, **tag_data_par_h**, **tag_valid_h**	Output delay	**Tdd** + 0.2 ns[1]	**Taod**
data_ram_oe_l, **data_ram_we_l<3:0>**, **tag_ram_oe_l**, **tag_ram_we_l**	Output hold	**Tmdd**	**Taoh**
tag_data_h, **tag_data_par_h**, **tag_valid_h**	Output hold	**Tmdd**	**Taoh**

[1] The value 0.2 ns accounts for onchip driver and clock skew.
[2] For big drive enabled or big drive disabled, respectively. See Table 30.

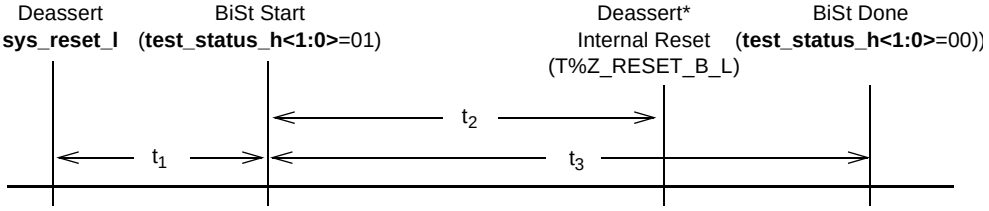
11.4.4 Timing of Test Features

Timing of 21164PC testability features depends on the system clock rate and the test port's operating mode. The following sections provide timing information that may be needed for most common operations.

11.4.4.1 Icache BiSt Operation Timing

The Icache BiSt is invoked by deasserting the external reset signal **sys_reset_l**. Figure 17 shows the timing between various events relevant to BiSt operations.

Figure 17 BiSt Timing Event — Timeline



The timing for deassertion of internal reset (time t_2, see asterisk) is valid only if an SROM is not present (indicated by keeping signal **srom_present_l** deasserted). If an SROM is present, the SROM load is performed once the BiSt completes. The internal reset signal T%Z_RESET_B_L is extended until the end of the SROM load (Section 11.4.4.2). In this case, the end of the timeline shown in Figure 17 connects to the beginning of the timeline shown in Figure 18.

Table 35 and Table 36 list timing shown in Figure 17 for some of the system clock ratios. Time t_1 is measured starting from the rising edge of sys_clk following the deassertion of the **sys_reset_l** signal.

Table 35 BiSt Timing for Some System Clock Ratios, Port Mode=Normal (System Cycles)

	System Cycles		
Sys_clk Ratio	t_1	t_2	t_3
4	7	28569 + 3½	28570
15	7	15749 + 14½	15750

Table 36 BiSt Timing for Some System Clock Ratios, Port Mode=Normal (CPU Cycles)

	CPU Cycles		
Sys_clk Ratio	t_1	t_2	t_3
4	28	114279½	114280
15	105	236249½	236250

11.4.4.2 Automatic SROM Load Timing

The SROM load is triggered by the conclusion of BiSt if **srom_present_l** is asserted. The SROM load occurs at the internal cycle time of approximately 126 CPU cycles for **srom_clk_h**, but the behavior at the pins may shift slightly.

Timing events are shown in Figure 18 and are listed in Table 37 and Table 38.

Figure 18 SROM Load Timing Event — Timeline

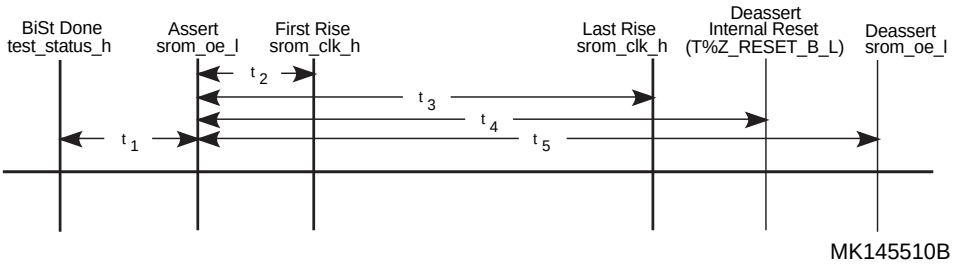


Table 37 SROM Load Timing for Some System Clock Ratios (System Cycles)

	System Cycles[1]				
Sys_clk Ratio	t_1	t_2	t_3	t_4	t_5
4	3	48	4209267	4209361 + 3½	4209362
15	3	13	1122472	1122496 + 14½	1122497

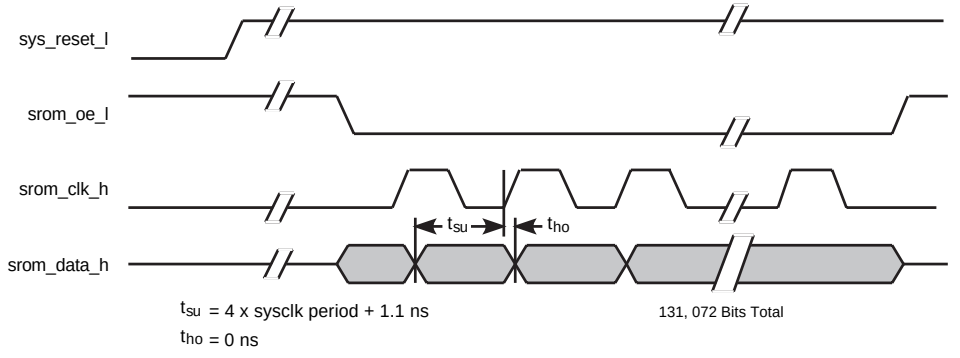
[1] Measured in **sysclk** cycles, where "+ *n*" refers to an additional *n* CPU cycles

Table 38 SROM Load Timing for Some System Clock Ratios (CPU Cycles)

	CPU Cycles				
Sys_clk Ratio	t_1	t_2	t_3	t_4	t_5
4	12	192	16837068	16837447½	16837448
15	45	195	16837080	16837454½	16837455

Figure 19 is a timing diagram of an SROM load sequence.

Figure 19 Serial ROM Load Timing



The minimum **srom_clk_h** cycle = (126 – sysclk ratio) × (CPU cycle time).

The maximum **srom_clk_h** to **srom_data_h** delay allowable (in order to meet the required setup time) = [126 – (5 × sysclk ratio)] × (CPU cycle time).

11.4.5 Clock Test Modes

This section describes the 21164PC clock test modes.

11.4.5.1 Normal (1× Clock) Mode

When **clk_mode_h<1>** is not asserted, the **osc_clk_in_h,l** frequency is used to drive the input clock frequency. The **clk_mode_h<0>** signal is used to enable/disable a clock equalizing circuit, called a **symmetrator**. The symmetrator equalizes the duty-cycle of the input clock for use onchip. The **osc_clk_ in_h,l** signals must have a duty cycle of at least 60/40 for the symmetrator to work properly. Normal clock mode with the symmetrator enabled is the preferred clocking mode of the 21164PC.

11.4.5.2 Clock Test Reset Mode

When **clk_mode_h<1>** is asserted, the **sys_clk_out** generator circuit is forced to reset to a known state. This allows the chip manufacturing tester to synchronize the chip to the tester cycle. This mode can be used with the symmetrator either enabled or disabled.

Table 39 lists the clock test modes.

Table 39 Clock Test Modes

	clk_mode_h		
Mode	**<1>**	**<0>**	**Notes**
Normal (1×) clock mode	0	0	
Normal (1×) clock mode	0	1	Symmetrator is enabled.
Clock reset	1	0	
Clock reset	1	1	Symmetrator is enabled.

11.4.6 IEEE 1149.1 (JTAG) Performance

Table 40 lists the standard mandated performance specifications for the IEEE 1149.1 circuits.

Table 40 IEEE 1149.1 Circuit Performance Specifications

Item	Specification
trst_l is asynchronous. Minimum pulse width.	4 ns
trst_l setup time for deassertion before a transition on **tck_h**.	4 ns
Maximum acceptable **tck_h** clock frequency.	16.6 MHz
tdi_h/**tms_h** setup time (referenced to **tck_h** rising edge).	4 ns
tdi_h/**tms_h** hold time (referenced to **tck_h** rising edge).	4 ns
Maximum propagation delay at pin **tdo_h** (referenced to **tck_h** falling edge).	14 ns
Maximum propagation delay at system output pins (referenced to **tck_h** falling edge).	20 ns

11.5 Power Supply Considerations

For correct operation of the 21164PC, all of the **Vss** pins must be connected to ground, all of the **Vdd** pins must be connected to a 3.3-V ±5% power source, and all of the **Vddi** pins must be connected to a 2.5-V ±0.1 V power source. This source voltage should be guaranteed (even under transient conditions) at the 21164PC pins, and not just at the PCB edge.

Plus 5 V is not used in the 21164PC. The voltage difference between the **Vdd** pins and **Vss** pins must never be greater than 3.46 V, and the voltage difference between the **Vddi** pins and **Vss** pins must never be greater than 2.6 V. If the differentials exceed these limits, the 21164PC chip will be damaged.

11.5.1 Decoupling

The effectiveness of decoupling capacitors depends on the amount of inductance placed in series with them. The inductance depends both on the capacitor style (construction) and on the module design. In general, the use of small, high-frequency capacitors placed close to the chip package's power and ground pins with very short module etch will give best results. Depending on the user's power supply and power supply distribution system, bulk decoupling may also be required on the module.

The 21164PC requires two sets of decoupling capacitors: one for **Vdd** and one for **Vddi**.

11.5.1.1 Vdd Decoupling

The amount of decoupling capacitance connected between **Vdd** and **Vss** should be roughly equal to 10 times the amount of capacitive load that 21164PC is required to drive at any one time. This should guarantee a voltage drop of no more than 10% on **Vdd** during heavy drive conditions.

Use capacitors that are as physically small as possible. Connect the capacitors directly to the 21164PC **Vdd** and **Vss** pins by short surface etch (0.64 cm [0.25 in] or less). The small capacitors generally have better electrical characteristics than the larger units and will more readily fit close to the IPGA pin field.

When designing the placement of decoupling capacitors, **Vdd** decoupling capacitors should be favored over **Vddi** decoupling capacitors (that is, **Vdd** capacitors should be placed closer to the 21164PC than the **Vddi** capacitors).

11.5.1.2 Vddi Decoupling

Each individual case must be separately analyzed, but generally designers should plan to use at least 4 µF of capacitance connected between **Vddi** and **Vss**. Typically, 30 to 40 small, high-frequency 0.1-µF capacitors are placed near the chip's **Vddi** and **Vss** pins. Actually placing the capacitors in the pin field is the best approach. Several tens of µF of bulk decoupling (comprised of tantalum and ceramic capacitors) should be positioned near the 21164PC chip.

Use capacitors that are as physically small as possible. Connect the capacitors directly to the 21164PC **Vddi** and **Vss** pins by short surface etch (0.64 cm [0.25 in] or less). The small capacitors generally have better electrical characteristics than the larger units, and will more readily fit close to the IPGA pin field.

11.5.2 Power Supply Sequencing

When applying or removing power to the 21164PC, **Vdd** (the 3.3-V supply voltage) must be no less than **Vddi** (the 2.5-V supply voltage).

The following rules must be followed when either applying or removing the supply voltages:

1. **Vdd** must always be at the same or a higher voltage than **Vddi** during normal operation.
2. The *signal voltage* must not exceed **Vclamp.**
3. The *signal voltage* must not be more than 2.4 V higher than **Vddi.**

Rule 1 means that either **Vdd** and **Vddi** can be brought up and down in unison or **Vddi** can be applied after and removed before **Vdd**.

Rule 2 means that the signal voltage must not be allowed to exceed **Vclamp** during the application or removal of power. Refer to Table 26 for the value of **Vclamp**. Note that it is acceptable for the signal voltage either to be held at zero or to follow **Vdd** during the application or removal of power.

Rule 3 means that, if the signal voltage follows **Vdd**, the signal voltage must never be greater than 2.4 V above the value of **Vddi**. This applies equally during the application or the removal of power.

Note that if the signal voltage is held at 0 V during power-up reset (that is, the ASICs and SRAMs are set to drive 0 V during reset), **Vdd** and **Vddi** can be brought up together. In a similar manner, the power-down situation can be managed if the signal voltages are forced to 0 V when the loss of **Vddi** is detected.

During power-up, **Vddi** can momentarily exceed the maximum steady-state value under the following conditions:

- The transient voltage is 200 mV or less.
- The transient period lasts for 200 µs or less.

The transient voltage is defined as the voltage that rises above the maximum-allowed steady-state value. The transient period is defined as the time beginning when the transient voltage exceeds the steady-state value and ending when it falls back to it.

There is no derating for shorter transient periods or lower transient voltages (for example, a 400-mV transient voltage lasting for 100 µs is not acceptable).

All input and bidirectional signals are diode-clamped to **Vdd** and **Vss**. A current greater than **Iclamp** on an individual pin could damage the 21164PC. Designers must take care that currents greater than **Iclamp** will not be achieved during power-supply sequencing. While currents less than **Iclamp** will not damage the 21164PC, other source drivers connected to the 21164PC could be damaged by the clamp. Designers must verify that the source drivers will not be damaged by currents up to **Iclamp**.

12 Thermal Management

This section describes the 21164PC thermal management and thermal design considerations.

12.1 Operating Temperature

The 21164PC is specified to operate when the temperature at the center of the heat sink (T_c) is 71.8°C for 400 MHz, 69.8°C for 466 MHz, or 67.5°C for 533 MHz. Temperature (T_c) should be measured at the center of the heat sink (between the two package studs). The GRAFOIL pad is the interface material between the package and the heat sink.

Table 41 lists the values for the center of heat-sink-to-ambient ($\Theta_c a$) for the 413-pin grid array. Table 42 shows the allowable T_a (without exceeding T_c) at various airflows.

Note: DIGITAL recommends using the heat sink because it greatly improves the ambient temperature requirement.

Table 41 $\Theta_c a$ at Various Airflows

Frequency: 400 MHz, 466 MHz, and 533 MHz						
	Airflow (linear ft/min)					
	100	200	400	600	800	1000
$\Theta_c a$ with heat sink 1 (°C/W)	3.2	1.7	0.95	0.75	0.65	0.55
$\Theta_c a$ with heat sink 2 (°C/W) (includes 52×10 mm fan)	0.75[1]					

[1] With the heat-sink fan, performance does not depend on system airflow.

Table 42 Maximum T_a at Various Airflows

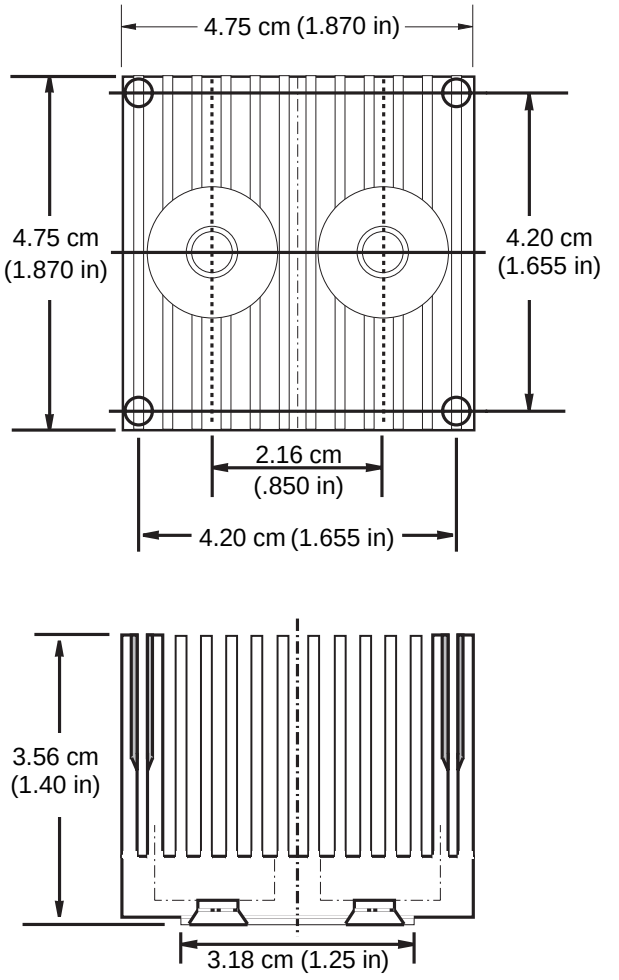
	Airflow (linear ft/min)					
	100	**200**	**400**	**600**	**800**	**1000**
Frequency: 400 MHz, Power: 26.5 W @Vdd = 3.3 V and @Vddi = 2.5 V						
T_a with heat sink 1 (°C)	—	26.8	46.6	51.9	54.6	57.2
T_a with heat sink 2 (°C) (includes 52×10 mm fan)	←			51.9[1]		→
Frequency: 466 MHz, Power: 30.5 W @Vdd = 3.3 V and @Vddi = 2.5 V						
T_a with heat sink 1 (°C)	—	18.0	40.8	46.9	50.0	53.0
T_a with heat sink 2 (°C) (includes 52×10 mm fan)	←			46.9[1]		→
Frequency: 533 MHz, Power: 35 W @Vdd = 3.3 V and @Vddi = 2.5 V						
T_a with heat sink 1 (°C)	—	—	34.3	41.3	44.8	48.3
T_a with heat sink 2 (°C) (includes 52×10 mm fan)	←			41.3[1]		→

[1] With the heat-sink fan, performance does not depend on system airflow.

12.2 Heat-Sink Specifications

Figure 20 describes the specifications of heat sink 1. Heat sink 2 has the exact same specifications, plus an added 52×10 mm fan.

Figure 20 Heat Sink 1



PCA030

12.3 Thermal Design Considerations

Follow these guidelines for printed circuit board (PCB) component placement:

- Orient the 21164PC on the PCB with the heat-sink fins aligned with the airflow direction.
- Avoid preheating ambient air. Place the 21164PC 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 21164PC.
- Do not restrict the airflow across the 21164PC heat sink. Placement of other devices must allow for maximum system airflow in order to maximize the performance of the heat sink.

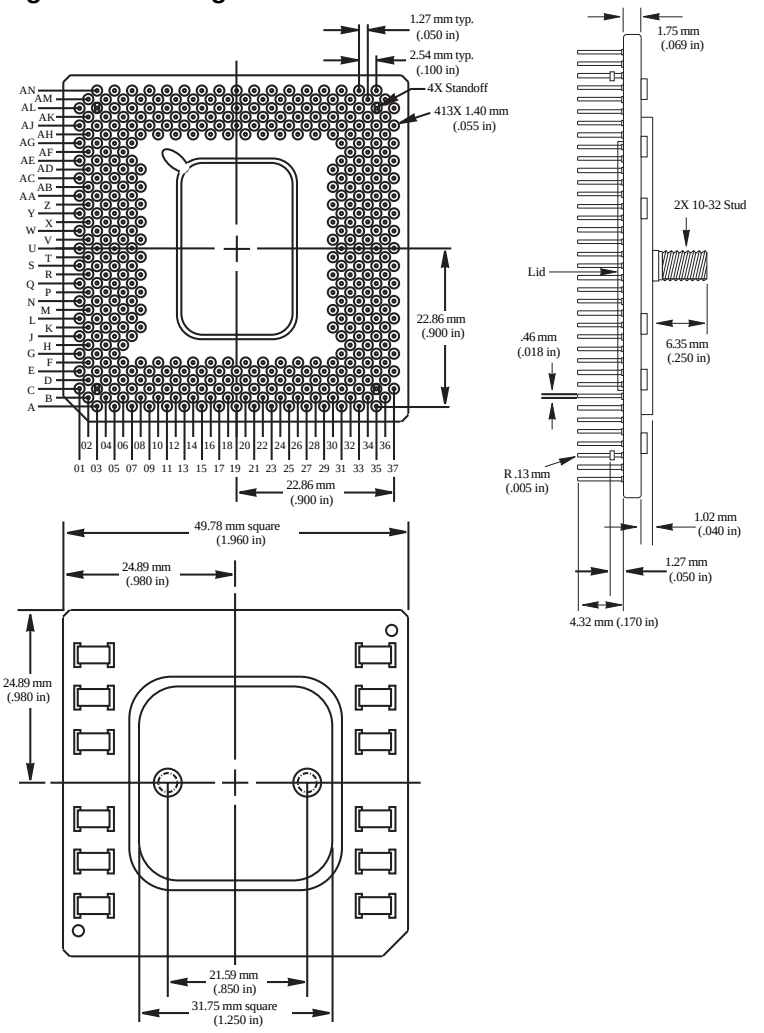
13 Mechanical Specifications

This section shows the 21164PC mechanical packaging dimensions without a heat sink. For heat sink dimensions, refer to Section 12.

Package Dimensions

Figure 21 shows the package physical dimensions without a heat sink.

Figure 21 Package Dimensions



PCA027

A Support, Products, and Documentation

If you need technical support, a *Digital Semiconductor Product Catalog*, or help deciding which documentation best meets your needs, visit the Digital Semiconductor World Wide Web Internet site:

http://www.digital.com/semiconductor

You can also call the Digital Semiconductor Information Line or the Digital Semiconductor Customer Technology Center. Please use the following information lines for support.

For documentation and general information:	
Digital Semiconductor Information Line	
United States and Canada:	1–800–332–2717
Outside North America:	1–510–490–4753
Electronic mail address:	semiconductor@digital.com

For technical support:	
Digital Semiconductor Customer Technology Center	
Phone (U.S. and international):	1–978–568–7474
Fax:	1–978–568–6698
Electronic mail address:	ctc@hlo.mts.dec.com

Digital Semiconductor Products

To order the Digital Semiconductor Alpha 21164PC microprocessor, contact your local distributor. The following table lists some of the semiconductor products available from Digital Semiconductor.

Note: The following products and order numbers might have been revised. For the latest versions, contact your local distributor.

Chips	Order Number
Digital Semiconductor Alpha 21164PC 400-MHz microprocessor	211PC–01
Digital Semiconductor Alpha 21164PC 466-MHz microprocessor	211PC–02
Digital Semiconductor Alpha 21164PC 533-MHz microprocessor	211PC–03

For information about other Alpha microprocessors, visit the Alpha World Wide Web Internet site:

http://www.alpha.digital.com

Digital Semiconductor Documentation

The following table lists some of the available Digital Semiconductor documentation.

Title	Order Number
Alpha AXP Architecture Reference Manual[1]	EY–T132E–DP
Alpha Architecture Handbook[2]	EC–QD2KB–TE
Digital Semiconductor Alpha 21164PC Microprocessor Hardware Reference Manual	EC–R2W0A–TE
Digital Semiconductor Alpha 21164PC Microprocessor Product Brief	EC–R2W2B–TE
Digital Semiconductor 21172 Core Logic Chipset Product Brief	EC–QUQHA–TE
Digital Semiconductor 21172 Core Logic Chipset Technical Reference Manual	EC–QUQJA–TE
Answers to Common Questions about PALcode for Alpha AXP Systems	EC–N0647–72
PALcode for Alpha Microprocessors System Design Guide	EC–QFGLC–TE

Title	Order Number
Alpha Microprocessors Evaluation Board Windows NT 3.51 and 4.0 Installation Guide	EC–QLUAG–TE
SPICE Models for Alpha Microprocessors and Peripheral Chips: An Application Note	EC–QA4XG–TE
Alpha Microprocessors SROM Mini-Debugger User's Guide	EC–QHUXC–TE
Alpha Microprocessors Evaluation Board Debug Monitor User's Guide	EC–QHUVE–TE
Alpha Microprocessors Evaluation Board Software Design Tools User's Guide	EC–QHUWC–TE

[1] To purchase the *Alpha AXP Architecture Reference Manual*, contact your local distributor or call Butterworth-Heinemann (Digital Press) at 1-800-366-2665.
[2] This handbook provides information subsequent to the *Alpha AXP Architecture Reference Manual*.

Third–Party Documentation

You can order the following third-party documentation directly from the vendor.

Title	Vendor
PCI Local Bus Specification, Revision 2.1 PCI System Design Guide	PCI Special Interest Group U.S. 1–800–433–5177 International 1–503–797–4207 Fax 1–503–234–6762
IEEE Standard 754, Standard for Binary Floating-Point Arithmetic IEEE Standard 1149.1, A Test Access Port and Boundary Scan Architecture	The Institute of Electrical and Electronics Engineers, Inc. U.S. 1–800–701–4333 International 1–908–981–0060 Fax 1–908–981–9667