DECchip 21066
Alpha AXP Microprocessor

Data Sheet

Order Number: EC–N0617–72

Revision/Update Information: This is a new data sheet.

Digital Equipment Corporation
Maynard, Massachusetts

May 1994

Contents

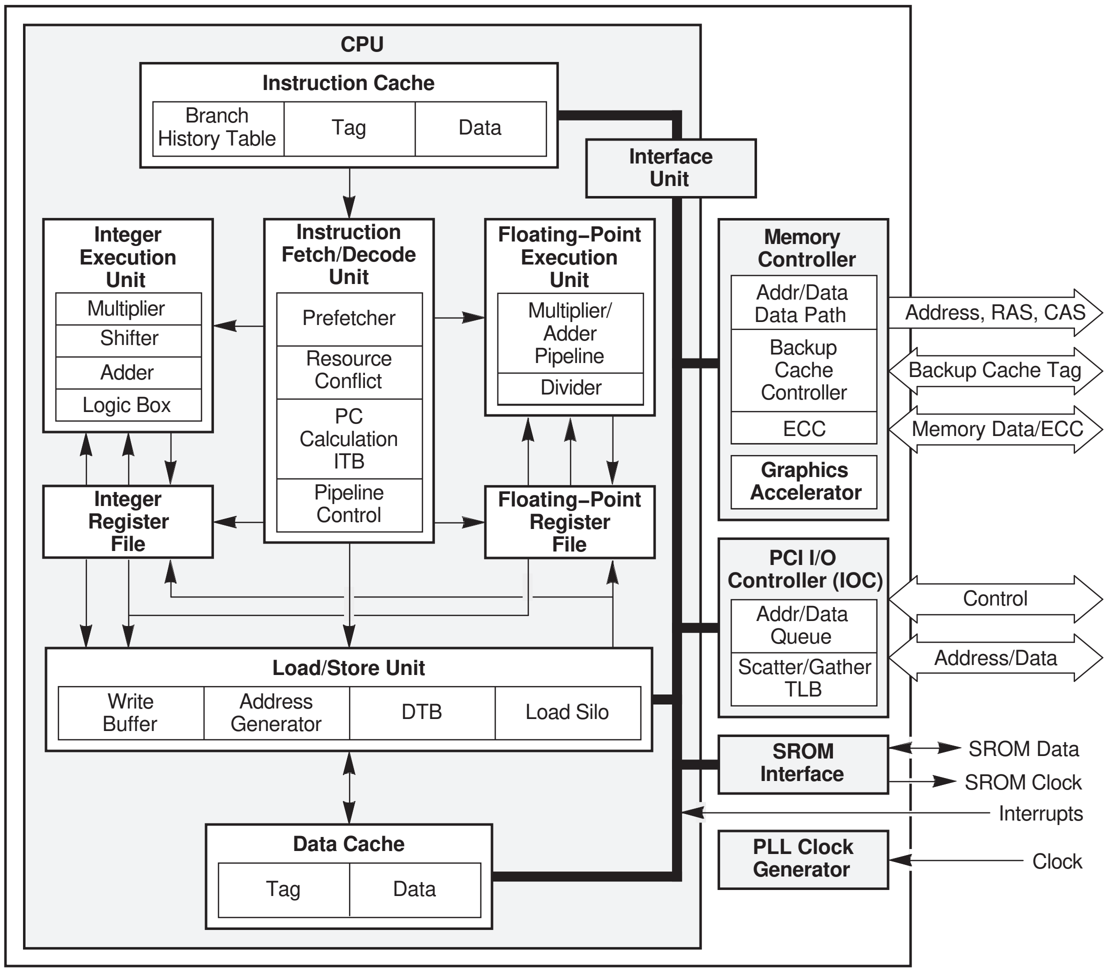
Figures

Tables

1 Microarchitecture

The DECchip 21066 microprocessor implements Digital's Alpha AXP architecture. The following sections provide an overview of the chip's architecture and major functional units. Figure 1 is a block diagram of the DECchip 21066 microprocessor.

Figure 1 DECchip 21066 Block Diagram



1.1 Overview

The DECchip 21066 microprocessor consists of a core central processing unit (CPU), a memory controller, and an I/O controller. The 21066 also contains instruction and data caches and a serial read-only memory (SROM) interface.

The PCI I/O controller (IOC) is an interface between peripheral devices and the CPU and system memory. It is compatible with the *PCI Local Bus Specification, Revision 2.0*.

The memory controller interfaces to the system memory and an optional, external, backup cache. It also contains the embedded graphics accelerator.

The SROM interface provides the initialization data load path from the SROM to the instruction cache. Following initialization, this interface can be used as a diagnostic port using privileged architecture library code (PALcode).

The interface unit connects the CPU, memory controller, IOC, and SROM interface. It consists of a 64-bit bidirectional data bus, address bus, invalidate address bus, reset logic, and control.

The instruction fetch and decode unit is the CPU's central control unit. It issues instructions, maintains the pipeline, and performs program counter (PC) calculations. The CPU also contains four independent execution units:

- Integer execution unit
- Load and store unit
- Floating-point unit (FPU)
- Branch unit

Each unit accepts no more than one instruction per cycle; however, correctly scheduled code can issue two instructions to two independent units in a single cycle.

1.2 Instruction Fetch and Decode Unit

The primary function of the instruction fetch and decode unit is to issue instructions to the integer execution unit, load and store unit, and floating-point unit. The instruction fetch and decode unit contains:

- Prefetcher
- PC pipeline
- Two instruction translation buffers (ITBs)
- Abort logic
- Register conflict or dirty logic

- Exception logic
- Internal processor registers

Instruction Fetch and Decode

The instruction fetch and decode unit decodes two instructions in parallel and checks that the required resources are available for both instructions, as follows:

- If resources are available, both instructions are issued.
- If resources are available only for the second instruction, neither instruction is issued.
- If the instruction fetch and decode unit issues only the first of a pair of instructions, it does not advance another instruction to attempt another dual issue; dual issue is attempted only on aligned quadword (8-byte) pairs.

Branch Prediction

The branch unit, or prediction logic, is also part of the instruction fetch and decode unit. The microprocessor offers a register-selectable choice of branch prediction strategies. Each instruction location in the instruction cache includes a single history bit to record the outcome of branch instructions. This information can be used to predict the result when the branch instruction is next executed.

Translation Buffers

The instruction fetch and decode unit includes two fully associative instruction translation buffers (ITBs):

- An 8-entry, small-page ITB for 8KB pages.
- A 4-entry, large-page ITB that supports 512 8KB pages.

Both translation buffers store recently used, instruction stream (Istream) page table entries (PTEs) and use a not-last-used replacement algorithm. In addition, both ITBs support a register-enabled extension called the superpage. The ITB superpage mappings provide one-to-one virtual PC <33:13> to physical PC <33:13> translation when virtual address bits <42:41> = 2.

Interrupts

The instruction fetch and decode unit exception logic supports three sources of interrupts:

- Hardware interrupts
  - There are three level-sensitive hardware interrupts sourced by pins **irq<2:0>**.

- There are two internally generated interrupts that respond to external interface error conditions. These are sourced by registers in the memory controller and I/O controller.

- Software interrupts

  There are 15 prioritized software interrupts, sourced by an on-chip register.

- Asynchronous system traps (ASTs)

  There are four ASTs, one for each processor mode: user, supervisor, executive, and kernel. These traps are sourced by an on-chip register.

The interrupt mechanism provides a flexible, software-controlled priority scheme that can be implemented by PALcode or the operating system. All interrupts can be independently masked in on-chip enable registers. In addition, AST interrupts are qualified by the current processor mode.

Performance Monitoring

An on-chip performance recording mechanism counts various hardware events and causes an interrupt upon counter overflow. Two counters are provided to allow accurate comparison of two variables under potentially nonrepeatable, experimental conditions. The events counted include:

- Instruction issues
- Nonissues
- Total cycles
- Pipeline dry
- Pipeline freeze
- Cache misses
- Counts of various instruction classes

In addition, two external interface events, such as direct memory access (DMA) transactions or external cache accesses, can be counted by programming a memory controller register.

1.3 Integer Execution Unit

The integer execution unit contains the 64-bit integer execution data path, which includes the following:

- Adder
- Logic box
- Barrel shifter

- Byte zapper
- Bypassers
- Integer multiplier

The integer execution unit also contains the 32-entry, 64-bit integer register file (IRF). The IRF has four read ports and two write ports to simultaneously read operands to and write operands (results) from the integer execution data path and the load and store unit.

1.4 Load and Store Unit

The load and store unit contains four major sections:

- Address translation data path, which includes the data translation buffer
- Load silo
- Write buffer
- Internal processor registers (IPRs)

Address Translation Data Path

The address translation data path has a displacement adder that generates the effective virtual address for load and store instructions, and a data translation buffer (DTB) that generates the corresponding physical address.

Data Translation Buffer

The 32-entry, fully associative DTB stores recently used, data stream (Dstream) page table entries (PTEs). The DTB supports four page-size granularity options (also called granularity hints) that allow an aligned group of 1, 8, 64, or 512 pages to be treated as a single larger page.

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

- The first enables superpage mapping when virtual address bits <42:41> = 2. In this mode, the entire physical address space is mapped multiple times to the quadrant of virtual address space defined by VA <42:41> = 2.
- The second maps a 30-bit region of the total physical address space, defined by PA <33:30> = 0, into a single corresponding region of virtual address space defined by VA <42:30> = 1FFE.

Load Silos

The load and store unit contains a memory reference pipeline that can accept a new load or store instruction every cycle until a data cache fill is required. Instructions are issued in pipeline stage 3, and the result of each data cache lookup is not known until pipeline stage 6. Therefore, there can be two instructions in the load and store unit pipeline behind a load instruction that misses in the data cache. These two instructions are handled as follows:

- Loads that hit in the data cache are allowed to complete (hit-under-miss).
- Loads that miss are placed in a silo and are replayed in sequence after the first load miss completes.
- Store instructions are presented to the data cache at their normal time, with respect to the pipeline. They are placed in a silo and presented to the write buffer in sequence, with respect to loads that miss.

Write Buffer

The load and store unit write buffer has two purposes:

- The 21066 CPU can generate store data faster than the backup cache (Bcache) subsystem can accept the data. This can cause CPU stall cycles. The write buffer provides a finite, high-bandwidth resource for receiving store data to minimize the number of possible CPU stall cycles.
- The write buffer also attempts to aggregate store data into aligned, 32-byte cache blocks to maximize the rate at which the 21066 can write data into the Bcache.

1.5 Floating-Point Unit

The on-chip, pipelined floating-point unit (FPU) can execute both IEEE and VAX floating-point instructions. The 21066 supports IEEE S_floating and T_floating data types, with all rounding modes (except round to ± infinity, which can be provided in software). The 21066 fully supports VAX F_floating and G_floating data types, and provides limited support for the VAX D_floating format.

The FPU contains:

- A 32-entry, 64-bit floating-point register file (FRF)
- A user-accessible control register

The FPU can accept an instruction every cycle, with the exception of floating-point divide instructions. The latency for data-dependent, nondivide instructions is six cycles.

The 21066 supports the IEEE floating-point operations as defined by the Alpha AXP architecture. Support for a complete implementation of the IEEE Standard for Binary Floating-Point Arithmetic (ANSI/IEEE Standard 754-1985) is provided by a combination of hardware and software.

1.6 Pipeline Organization

The 21066 has a 7-stage pipeline for integer operate and memory reference instructions, and a 10-stage pipeline for floating-point operate instructions. The instruction fetch and decode unit maintains state for all pipeline stages, to track outstanding register writes and to determine instruction cache hits and misses.

Figure 2 shows the integer operate, memory reference, and floating-point operate pipelines for the instruction fetch and decode unit, integer execution unit, load and store unit, and FPU. The first four stages of all the pipelines are the same, and are executed in the instruction fetch and decode unit. The last stages are unit specific. All of the units have bypassers that allow the results of one instruction to be the operand of a following instruction, without writing the results of the first instruction to a register file.

1.7 Internal Cache Organization

The 21066 includes two on-chip caches — a data cache and an instruction cache. All memory cells in both caches are fully static, 6-transistor, CMOS structures.

Data Cache

The 8KB data cache (Dcache) is a write-through, direct mapped, read-allocate, physical cache with 32-byte blocks. When a PCI device writes to cacheable memory, the Dcache block corresponding to the memory address is set invalid.

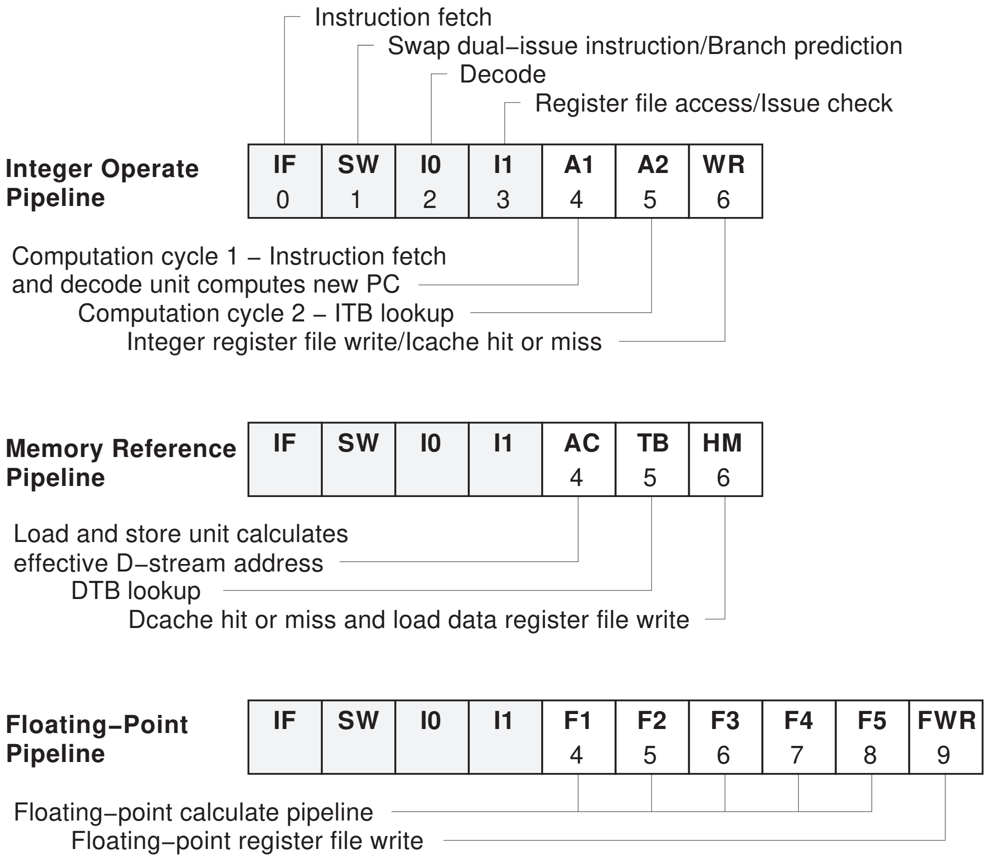
Instruction Cache

The 8KB instruction cache (Icache) is a physical direct-mapped cache. Each Icache block (line) contains:

- 32 bytes of Istream data
- 21-bit associated tag
- 6-bit address space number (ASN) field
- 1-bit address space match (ASM) field
- 8-bit branch history (BHT) field

The Icache does not contain hardware for maintaining coherency with memory, and it is unaffected by PCI writes to memory.

Figure 2 Instruction Pipelines



1.8 Memory Controller

The on-chip memory controller interfaces the CPU to the system memory and the optional Bcache. It has several memory-mapped control and status registers (CSRs) to program organization, timing, and size of DRAM, VRAM, and Bcache SRAM. It controls CPU requests and DMA requests (from the I/O controller) to and from memory and the Bcache. It also controls VRAM shift register loads and memory refreshes.

The memory controller decodes the address of a CPU request to determine whether the request is for memory or the I/O controller (IOC). It handles the access to the memory controller CSRs, the memory, and Bcache. If the request is directed at the IOC, the memory controller passes control to the IOC.

The memory controller can also perform the following graphics operations:

- Dumb frame buffer operation
- Transparent stipple operation
- Write-per-bit plane masking
- Byte writes (with external gating)
- Full and split VRAM shift-register loads

1.9 I/O Controller

The on-chip I/O controller (IOC) is an interface bridge between peripheral devices and the CPU and system memory. The IOC interface protocol complies with the *PCI Local Bus Specification, Revision 2.0*. All peripheral devices in a 21066-based system can communicate with the CPU and system memory through the IOC. Peripheral chips that are PCI-compliant can be connected directly to the 21066 without any glue logic. The IOC runs asynchronously to the CPU using the PCI clock input.

The IOC incorporates scatter-gather mapping logic to translate 32-bit addresses generated by PCI bus masters to the 34-bit CPU physical address space. The IOC implements an 8-entry translation look-aside buffer (TLB) for fast translations. Two programmable address windows control PCI peripheral device access to system memory.

1.10 Obtaining Additional Information

To obtain more information about the DECchip 21066 microprocessor, the Alpha AXP architecture and instruction set, and the PCI, see Technical Support, Ordering, and Associated Literature.

2 Pinout

Sections 2.1 through 2.3 list the external signals and their associated pins, describe the external signals, and list the signals according to function.

2.1 Signal List

Table 1 lists the signal associated with each pin.

Table 1 Signal List

Pin	Signal	Pin	Signal	Pin	Signal
Index		B01	**Vdd**	C01	**ad17**
A02	**Vss**	B02	**ad12**	C02	**ad14**
A03	**Vdd**	B03	**ad11**	C03	**pci_clk_in**
A04	**Vss**	B04	**ad9**	C04	**ad10**
A05	**gnt_l**	B05	**par**	C05	**ad8**
A06	**Vdd**	B06	**perr_l**	C06	**req_l**
A07	**Vss**	B07	**devsel_l**	C07	**frame_l**
A08	**lock_l**	B08	**trdy_l**	C08	**stop_l**
A09	**Vdd**	B09	**c_be_l3**	C09	**irdy_l**
A10	**Vss**	B10	**c_be_l0**	C10	**c_be_l1**
A11	**ad5**	B11	**ad6**	C11	**ad7**
A12	**Vdd**	B12	**ad4**	C12	**ad3**
A13	**Vss**	B13	**ad2**	C13	**ad1**
A14	**ad0**	B14	**bc_cs_l**	C14	**bc_oe_l**
A15	**Vdd**	B15	**bc_tag7**	C15	**bc_tag6**
A16	**Vss**	B16	**bc_tag5**	C16	**bc_tag3**
A17	**bc_tag4**	B17	**bc_tag2**	C17	**bc_dirty**
A18	**Vdd**	B18	**bc_tag0**	C18	**bc_idx_tag0**
A19	**Vss**	B19	**bc_parity**	C19	**bc_idx_tag2**
A20	**bc_idx_tag1**	B20	**bc_idx_tag3**	C20	**bc_idx_tag4**
A21	**Vdd**	B21	**mem_addr0**	C21	**mem_addr1**
		B22	**mem_addr3**	C22	**Vdd**

(continued on next page)

Table 1 (Cont.) Signal List

Pin	Signal	Pin	Signal	Pin	Signal
D01	**Vss**	E01	**Vdd**	H19	**Vdd**
D02	**ad16**	E02	**ad18**	H20	**mem_wr_oe_l**
D03	**instr_ref**	E03	**ad15**	H21	**mem_write_l**
D04	**ad13**	E04	**Vss**	H22	**mem_rd_oe**
D05	**Vdd**	E19	**Vdd**	J01	**ad29**
D06	**Vss**	E20	**mem_addr5**	J02	**ad28**
D07	**rst_l**	E21	**mem_addr6**	J03	**ad27**
D08	**Vdd**	E22	**mem_addr9**	J04	**Vdd**
D09	**Vss**	F01	**ad23**	J19	**Vss**
D10	**c_be_l2**	F02	**ad21**	J20	**mem_data0**
D11	**Vdd**	F03	**ad19**	J21	**mem_data1**
D12	**Vss**	F04	**Vdd**	J22	**Vdd**
D13	**bc_we_l**	F19	**Vss**	K01	**Vss**
D14	**Vdd**	F20	**mem_addr7**	K02	**mem_data32**
D15	**Vss**	F21	**mem_addr10**	K03	**ad31**
D16	**bc_tag1**	F22	**Vdd**	K04	**ad30**
D17	**Vdd**	G01	**Vss**	K19	**mem_data2**
D18	**Vss**	G02	**ad24**	K20	**mem_data3**
D19	**bc_index**	G03	**ad22**	K21	**mem_data4**
D20	**mem_addr2**	G04	**ad20**	K22	**Vss**
D21	**mem_addr4**	G19	**mem_addr8**	L01	**Vdd**
D22	**Vss**	G20	**mem_addr11**	L02	**mem_data34**
		G21	**mem_cas_l**	L03	**mem_data33**
		G22	**Vss**	L04	**Vss**
		H01	**Vdd**	L19	**Vdd**
		H02	**ad26**	L20	**mem_data5**
		H03	**ad25**	L21	**mem_data6**
		H04	**Vss**	L22	**mem_data7**

(continued on next page)

Table 1 (Cont.) Signal List

Pin	Signal	Pin	Signal	Pin	Signal
M01	**mem_data35**	R19	**Vss**	W01	**Vss**
M02	**mem_data36**	R20	**mem_data17**	W02	**mem_data53**
M03	**mem_data37**	R21	**mem_data16**	W03	**mem_data57**
M04	**Vdd**	R22	**Vdd**	W04	**mem_data60**
M19	**Vss**	T01	**Vss**	W05	**Vss**
M20	**mem_data9**	T02	**mem_data46**	W06	**Vdd**
M21	**mem_data8**	T03	**mem_data47**	W07	**pll_filter**
M22	**Vdd**	T04	**mem_data51**	W08	**Vss**
N01	**Vss**	T19	**mem_data22**	W09	**Vdd**
N02	**mem_data38**	T20	**mem_data20**	W10	**irq1**
N03	**mem_data39**	T21	**mem_data18**	W11	**Vss**
N04	**mem_data40**	T22	**Vss**	W12	**Vdd**
N19	**mem_data12**	U01	**Vdd**	W13	**mem_dsf**
N20	**mem_data11**	U02	**mem_data48**	W14	**Vss**
N21	**mem_data10**	U03	**mem_data50**	W15	**Vdd**
N22	**Vss**	U04	**Vss**	W16	**mem_ecc1**
P01	**Vdd**	U19	**Vdd**	W17	**Vss**
P02	**mem_data41**	U20	**mem_data23**	W18	**Vdd**
P03	**mem_data42**	U21	**mem_data21**	W19	**mem_data31**
P04	**Vss**	U22	**mem_data19**	W20	**mem_data29**
P19	**Vdd**	V01	**mem_data49**	W21	**mem_data25**
P20	**mem_data15**	V02	**mem_data52**	W22	**Vss**
P21	**mem_data14**	V03	**mem_data54**		
P22	**mem_data13**	V04	**Vdd**		
R01	**mem_data43**	V19	**Vss**		
R02	**mem_data44**	V20	**mem_data26**		
R03	**mem_data45**	V21	**mem_data24**		
R04	**Vdd**	V22	**Vdd**		

(continued on next page)

Table 1 (Cont.) Signal List

Pin	Signal	Pin	Signal	Pin	Signal
Y01	**Vdd**	AA01	**mem_data55**	AB01	**Vss**
Y02	**mem_data56**	AA02	**mem_data58**	AB02	**Vdd**
Y03	**mem_data59**	AA03	**mem_data61**	AB03	**mem_data63**
Y04	**mem_data62**	AA04	**Vss**	AB04	**Vss**
Y05	**pll_clk_in**	AA05	**pll_i_ref**	AB05	**Vdd**
Y06	**pll_clk_in_l**	AA06	**pll_5v**	AB06	**reset_in_l**
Y07	**pll_bypass**	AA07	**Vss**	AB07	**Vss**
Y08	**test_clk_out**	AA08	**sromd**	AB08	**Vdd**
Y09	**sromoe_l**	AA09	**sromclk**	AB09	**irq2**
Y10	**irq0**	AA10	**memreq_l**	AB10	**Vss**
Y11	**trst_l**	AA11	**memack_l**	AB11	**Vdd**
Y12	**tdi**	AA12	**tms**	AB12	**tck**
Y13	**vrefresh_l**	AA13	**mem_dtoe_l**	AB13	**Vss**
Y14	**mem_rasa_l3**	AA14	**vframe_l**	AB14	**Vdd**
Y15	**mem_rasa_l0**	AA15	**mem_rasa_l2**	AB15	**tdo**
Y16	**mem_rasb_l2**	AA16	**mem_rasa_l1**	AB16	**Vss**
Y17	**mem_ecc0**	AA17	**mem_rasb_l1**	AB17	**Vdd**
Y18	**mem_ecc2**	AA18	**mem_rasb_l0**	AB18	**mem_rasb_l3**
Y19	**mem_ecc5**	AA19	**mem_ecc3**	AB19	**Vss**
Y20	**mem_ecc7**	AA20	**mem_ecc4**	AB20	**Vdd**
Y21	**mem_data28**	AA21	**mem_data30**	AB21	**mem_ecc6**
Y22	**mem_data27**	AA22	**Vdd**	AB22	**Vss**

2.2 Signal Descriptions

Table 2 describes the function of each external signal in alphabetical order.

Table 2 Signal Description

Signal	Type	Description
ad<31:0>	I/O	Multiplexed PCI address and data. The byte address is driven during the first clock of a PCI transaction, and data is driven during subsequent clock cycles.
bc_cs_l	O	Bcache SRAM chip select. The SRAMs are enabled only when this signal is asserted. This signal provides a power saving feature when the Bcache is not being accessed.
bc_dirty	I/O	Indicates the status of the data stored in a cache block. A value of 0 indicates that the cache and memory contain the same data; a value of 1 indicates that the cache contains more recently written data than memory (in other words, memory data is stale). It has the same timing characteristics as **bc_tag<7:0>**.
bc_idx_tag<4:0>	I/O	An index bit or a tag bit, depending on the Bcache size. When the signal is an index bit, it is an output. When the signal is a tag bit, signal direction depends on cycle type — during a cache fill or write, the signal is an output; during a cache lookup, the signal is an input.
bc_index	O	Transmits Bcache index bit 12 to the cache.
bc_oe_l	O	Bcache output enable signal.
bc_parity	I/O	Transmits the Bcache tag parity to the tag array during cache fills and writes, and from the tag array during cache lookups. It has the same timing characteristics as **bc_tag<7:0>**.
bc_tag<7:0>	I/O	Represent the upper part of the physical address stored in a cache block. During a Bcache lookup, the cache drives these bits (and possibly **bc_idx_tag<4:0>**) with the tag that corresponds to the current index being driven on **mem_addr<11:0>**, **bc_index**, and the nontag bits of **bc_idx_tag<4:0>**. During a Bcache fill or write, the memory controller drives the most significant address bits on these lines, to be stored in the Bcache tag array.
bc_we_l	O	Bcache write enable signal.
c_be_l<3:0>	I/O	Multiplexed PCI bus command codes and byte enables. The command code is driven during the address cycle of a PCI transaction, and inverted byte enables are driven during data cycles.

(continued on next page)

Table 2 (Cont.) Signal Description

Signal	Type	Description
devsel_l	I/O	The device that is addressed by the current PCI transaction asserts this signal.
frame_l	I/O	Asserted at the beginning of a PCI transaction. It also controls the number of data transfers during the transaction (burst length). It is deasserted during the final data phase of a transaction.
gnt_l	I	External arbitration logic asserts this signal when the CPU is granted ownership of the PCI. The arbiter is expected to park (default grant) the PCI to the CPU when no other device is requesting ownership.
instr_ref	O	During normal operation, this signal indicates the type of current access (instruction or data) on the memory data bus. When high, this signal indicates an Istream reference; when low, this signal indicates a Dstream reference. During reset, the internal PCI clock is output on this pin for test purposes.
irdy_l	I/O	Asserted by the initiator of a PCI transaction to indicate that it can complete the current data phase of a PCI transaction. During a read cycle, this signal is asserted to indicate that the initiator is ready to accept read data. During a write cycle, this signal is asserted to indicate that the initiator is driving valid write data onto **ad<31:0>**. The current data phase completes when both **trdy_l** and **irdy_l** are asserted when sampled.
irq<2:0>	I	External interrupt requests. During reset, these pins are part of the clock system and are tristated to receive the power-up values (through external resistors tied to **Vss** or **Vdd**) that set the frequency ratio between the internal clock and external reference clock.
lock_l	I	Implements atomic (exclusive) access operations on the PCI. In test mode, this signal is used to test the 24-bit retry timeout counter at the chip tester. This test mode can be selected by programming the JTAG IR register (IR bit 3 = 0). During test mode, when this signal is high it selects the lower half of the counter for testing; when this signal is low it selects the upper half of the counter for testing.
memack_l	O	This PCI sideband signal is synchronous with **pci_clk_in**. The IOC asserts this signal when it has won arbitration (internal to the 21066) for access to memory requested by a **memreq_l**. This signal remains asserted until **memreq_l** is deasserted.

(continued on next page)

Table 2 (Cont.) Signal Description

Signal	Type	Description
mem_addr<11:0>	O	Transmit the row and column address to memory and the 12 least significant index bits to the Bcache. For memory reads and writes (that is, not refresh) signals **mem_addr<11:0>** contain a valid row address when **mem_ras_l** is asserted, and contain a valid column address when **mem_cas_l** is asserted.
mem_cas_l	O	When asserted during memory reads and writes, this signal indicates that **mem_addr<11:0>**, **mem_data<63:0>** (for writes), and **mem_write_l** contain valid information. During a memory refresh cycle, this signal is asserted before **mem_ras_l** is asserted.
mem_data<63:0>	I/O	Transmit data between the memory controller and either the memory or the Bcache. Memory (or an optional transceiver, see **mem_rd_oe** and **mem_wr_oe_l**) drives these signals during a read from memory, when **mem_cas_l** and **mem_rd_oe** are asserted and **mem_write_l** is not asserted. The Bcache drives these signals during a read from the Bcache, when **bc_oe_l** is asserted. The memory controller drives these signals during a write. Data is valid when either **bc_we_l** or **mem_cas_l**, **mem_wr_oe_l**, and **mem_write_l** are asserted. During a write-per-bit operation, the memory controller drives these signals with the write-per-bit mask. Data is valid when **mem_ras_l**, **mem_wr_oe_l**, and **mem_write_l** are asserted. *
mem_dsf	O	Selects a full or split VRAM shift-register load. This special function signal is valid before **mem_ras_l** is asserted when **mem_dtoe_l** is asserted.
mem_dtoe_l	O	Controls the memory output enable function or the VRAM shift-register load function. For normal memory operations, this multifunction signal is deasserted when **mem_ras_l** is asserted. During a read cycle, this signal is asserted before **mem_cas_l** is asserted to enable the memory data output drivers. To indicate a VRAM shift-register load sequence, this signal is asserted before **mem_ras_l** is asserted. The value of **mem_dsf** determines whether a full or split VRAM shift-register load sequence is done.

*The term **mem_ras_l** represents the **mem_rasa_l<3:0>** and **mem_rasb_l<3:0>** signal for the selected bank.

(continued on next page)

Table 2 (Cont.) Signal Description

Signal	Type	Description
mem_ecc<7:0>	I/O	Transmit the error correction codes (ECC) between the memory controller and either the memory or the Bcache. The memory controller generates ECC for writes and checks it for reads. These signals have the same external timing as **mem_data<63:0>**. (Memory storage for ECC is optional, and ECC checking can be disabled using the bank configuration registers.) These signals also transmit a write byte mask for banks that have byte write enabled in the bank configuration register (external logic is required to gate the DRAM write signals). When used this way, **mem_ecc0** corresponds to **mem_data<7:0>**, **mem_ecc1** corresponds to **mem_data<15:8>**, and so on.
mem_rasa_l<3:0>, **mem_rasb_l<3:0>**	O	Each pair of these signals is associated with a bank of memory. During normal reads and writes, the assertion of **mem_ras_l*** indicates that **mem_addr<11:0>** contains a valid row address. Which RAS is asserted depends on all of the following: • Which memory bank is addressed • Whether split bank is enabled in the bank configuration register • A row address bit During a memory refresh cycle, if the refresh enable bit is set, all eight of these signals are asserted together. Refresh cycles are the CAS-before-RAS type. During VRAM shift-register loads, **mem_rasa_l*n*** and **mem_rasb_l*n*** for bank *n* are asserted.
mem_rd_oe	O	Enables an optional, external, memory transceiver to drive data from the memory parts onto **mem_data<63:0>** and **mem_ecc<7:0>**. It is asserted during read cycles when **mem_cas_l** is asserted. This signal should be ignored if a transceiver is not used.
memreq_l	I	This PCI sideband signal is synchronous with **pci_clk_in**. When the IOC samples this signal asserted, it arbitrates (internal to the 21066) for access to memory. When arbitration to memory has been won, the IOC asserts **memack_l**.

*The term **mem_ras_l** represents the **mem_rasa_l<3:0>** and **mem_rasb_l<3:0>** signal for the selected bank.

(continued on next page)

Table 2 (Cont.) Signal Description

Signal	Type	Description
mem_write_l	O	Provides read and write control for memory and enables loading of the write-per-bit mask. For writes, **mem_write_l** is asserted before **mem_cas_l** is asserted; for reads, **mem_write_l** is deasserted before **mem_cas_l** is asserted. The write-per-bit function is activated when **mem_write_l** is asserted before **mem_ras_l**.
mem_wr_oe_l	O	Enables an optional, external, memory transceiver to drive data from **mem_data<63:0>** and **mem_ecc<7:0>** to the memory parts. It is asserted during write cycles when **mem_data<63:0>** is driven. This pin should be ignored if a transceiver is not used.
par	I/O	This PCI signal is the even parity bit for **ad<31:0>** and **c_be_l<3:0>**.
pci_clk_in	I	Provides timing for all transactions on the PCI. All of the IOC's PCI signals except **rst_l** are synchronous with this signal. Inputs are sampled on, and outputs change state as a result of the rising edge of this signal.
perr_l	I/O	This PCI signal is asserted when a data parity error has been detected.
pll_bypass	I	When this signal is asserted, the external clock input **pll_clk_in** directly drives the internal logic, and the internal clock and external reference frequencies are equal.
pll_clk_in, **pll_clk_in_l**	I	For normal operation, a low-speed (less than 50 MHz), single-ended clock and an appropriate reference or bias voltage are supplied to **pll_clk_in** and **pll_clk_in_l**, respectively. To minimize jitter induced by module and package noise, a high-speed (greater than 50 MHz) differential reference clock (logically-complementary, nominal square waves) is supplied to **pll_clk_in** and **pll_clk_in_l**.
pll_filter	I	A capacitor connected between this signal and **Vss** maintains stable operation by setting the correct feedback-loop time constant. The time constant regulates the speed with which the PLL responds to changes in frequency or operating conditions.
pll_i_ref	I	The constant current flowing in a resistor connected between this signal and **Vss** is the reference for analog PLL circuits. It reduces the variations in the speed of CMOS devices over a wide range of process and operating conditions.

(continued on next page)

Table 2 (Cont.) Signal Description

Signal	Type	Description
pll_5v	I	This *clean* +5 V signal is regulated internally to source the nominal +3.3 V used by the PLL and associated logic. This on-chip isolation is necessary to reduce phase jitter. Connect 5-volt decoupling capacitors as close as possible to this pin and the **Vss** pins.
req_l	O	The CPU asserts this signal when it needs to initiate a PCI transfer. External arbitration logic is required.
reset_in_l	I	Master reset input for the 21066; should be asserted when power is first applied to the chip. When it is asserted, certain internal chip logic is immediately initialized (some internal state is not reset and must be handled by software when the chip boots). Internal chip activity starts 31 cycles after **reset_in_l** is negated in synchronism with the internal clock.
rst_l	O	PCI reset signal generated by the CPU. The RST bit in the IOC PCI soft reset register allows this signal to be asserted under software control. This signal is automatically asserted when **reset_in_l** is asserted.
sromclk	O	SROM clock signal when **sromoe_l** is asserted. When **sromoe_l** is not asserted, this is software-controlled serial port output data.
sromd	I	SROM data when **sromoe_l** is asserted. When **sromoe_l** is not asserted, this is software-controlled serial port input data.
sromoe_l	O	Asserted after **rese_in_l** is asserted, and enables the SROM for initialization. Following initialization, this signal is deasserted, enabling the SROM port to be used as a software-controlled serial port.
stop_l	I/O	The target of a PCI transaction drives this signal to request that the initiator stop the current transaction.
tck	I	JTAG boundary scan clock.
tdi	I	JTAG serial boundary scan data in signal.
tdo	O	JTAG serial boundary scan data out signal.
test_clk_out	O	An output reference clock to be used only for testing the 21066. Its rising edge into a 40 pF load nominally coincides with the start of an internal microcycle. The relationship between this signal and **pll_clk_in** can be determined following the second negation of **reset_in_l** after power is turned on or the clock frequency ratio is changed. This pin should not be used to drive module-level logic.
tms	I	JTAG test mode select signal.

(continued on next page)

Table 2 (Cont.) Signal Description

Signal	Type	Description
trdy_l	I/O	Asserted by the target of a PCI transaction to indicate that it can complete the current data phase of a PCI transaction. During a read cycle, this signal is asserted to indicate that the selected device is driving valid data onto **ad<31:0>**. During a write cycle, this signal is asserted to indicate that the selected device is ready to accept write data. The current data phase completes when both **trdy_l** and **irdy_l** are asserted when sampled.
trst_l	I	JTAG test access port (TAP) reset signal.
vframe_l	I	When this signal is asserted, the memory controller uses the video and graphics control register fields to: 1. Reload the video display pointer with the start-of-video-frame value. 2. Perform a full VRAM shift-register load cycle to the bank selected by the start-of-video-frame value. 3. Increment the video display pointer twice, as specified by the address increment value.
vrefresh_l	I	When this signal is asserted, the memory controller uses the video and graphics control register fields to: 1. Perform a split VRAM shift-register load cycle to the bank selected by the start-of-video-frame value. 2. Increment the video display pointer twice as specified by the address increment value.

2.3 Quick Reference to Signals by Function and Direction

Table 3 is a quick reference to the signals, grouped by function.

Table 3 Signals by Function

Name	Qty	Type	Purpose	Value at Reset
Memory Controller Signals				
bc_cs_l	1	O	Bcache chip select	Driven, asserted
bc_dirty	1	I/O	Bcache valid	Tristate
bc_idx_tag<4:0>	5	I/O	Bcache index or tag	Tristate
bc_index	1	O	Bcache index (bit 12)	Driven, undefined
bc_oe_l	1	O	Bcache output enable	Driven, asserted
bc_parity	1	I/O	Bcache tag parity	Tristate
bc_tag<7:0>	8	I/O	Bcache tag	Tristate
bc_we_l	1	O	Bcache write enable	Driven, deasserted
mem_addr<11:0>	12	O	Row/column address, Bcache index	Driven, undefined
mem_cas_l	1	O	Column address strobe	Driven, deasserted
mem_data<63:0>	64	I/O	Memory/Bcache data	Tristate
mem_dsf	1	O	Disable special function	Driven, deasserted
mem_dtoe_l	1	O	Data transfer/output enable	Driven, deasserted
mem_ecc<7:0>	8	I/O	Memory/Bcache error correcting code	Tristate
mem_rasa_l<3:0> **mem_rasb_l<3:0>**	8	O	Row address strobes	Driven, deasserted
mem_rd_oe	1	O	Memory read transceiver output enable	Driven, deasserted
mem_write_l	1	O	Write enable	Driven, deasserted
mem_wr_oe_l	1	O	Memory write transceiver output enable	Driven, asserted
vframe_l	1	I	Load video display pointer and load VRAM shift register	Not applicable
vrefresh_l	1	I	Increment video display pointer and load VRAM shift register	Not applicable

(continued on next page)

Table 3 (Cont.) Signals by Function

Name	Qty	Type	Purpose	Value at Reset
PCI Signals				
ad<31:0>	32	I/O	PCI multiplexed address and data bus	Tristate when **gnt_l** is deasserted; otherwise, undefined.
c_be_l<3:0>	4	I/O	PCI multiplexed cycle command and byte enables	Tristate when **gnt_l** is deasserted; otherwise, undefined.
devsel_l	1	I/O	PCI device select	Tristate
frame_l	1	I/O	PCI cycle frame	Tristate
gnt_l	1	I	PCI bus grant	Not applicable
irdy_l	1	I/O	PCI initiator ready	Tristate
lock_l	1	I	PCI lock	Not applicable
memack_l	1	O	Grant for IOC access to 21066 memory	Tristate
memreq_l	1	I	Request for IOC access to 21066 memory	Not applicable
par	1	I/O	PCI even parity bit	Tristate when **gnt_l** is deasserted; otherwise, undefined.
pci_clk_in	1	I	PCI clock input	Not applicable
perr_l	1	I/O	PCI parity error	Tristate
req_l	1	O	PCI bus request	Tristate
rst	1	O	PCI reset	Asserted
stop_l	1	I/O	PCI target stop	Tristate
trdy_l	1	I/O	PCI target ready	Tristate

(continued on next page)

Table 3 (Cont.) Signals by Function

Name	Qty	Type	Purpose	Value at Reset
Clock Signals				
pll_bypass	1	I	PLL bypass select	Not applicable
pll_clk_in	1	I	PLL clock input	Not applicable
pll_clk_in_l	1	I	PLL clock input low	Not applicable
pll_i_ref	1	I	PLL reference current	Not applicable
pll_filter	1	I	PLL low-pass filter capacitor	Not applicable
pll_5v	1	I	PLL voltage supply	Not applicable
reset_in_l	1	I	Master reset input	Not applicable
test_clk_out	1	O	Output clock	Driven, clocking
JTAG Signals				
tck	1	I	JTAG boundary scan clock	Not applicable
tdi	1	I	JTAG serial boundary scan data in	Not applicable
tdo	1	O	JTAG serial boundary scan data out	Determined by the state of the JTAG controller
tms	1	I	JTAG test mode select	Not applicable
trst_l	1	I	JTAG TAP reset	Not applicable

(continued on next page)

Table 3 (Cont.) Signals by Function

Name	Qty	Type	Purpose	Value at Reset
Interrupt, SROM Interface, and instr_ref Signals				
instr_ref	1	O	Istream or Dstream reference	If **lock_l** is deasserted and the mode is PCI_SYNC_MODE, this pin is driven with **pci_clk_in**; otherwise, this pin is driven with the chip internal clock.
irq<2:0>	3	I	External interrupt request	Not applicable
sromclk	1	O	SROM clock or transmit serial data	Driven high
sromd	1	I	SROM data or receive serial data	Not applicable
sromoe_l	1	O	SROM output enable	Driven high

Table 4 is a quick reference to the signals, grouped by direction.

Table 4 Signals by Direction

Signal	Active Level	Signal	Active Level	Signal	Active Level
Input Signals					
gnt_l	Low	**pll_clk_in_l**	Low	**tck**	High
irq<2:0>	High	**pll_filter**	High	**tdi**	High
lock_l	Low	**pll_i_ref**	High	**tms**	High
memreq_l	Low	**pll_5v**	High	**trst_l**	Low
pci_clk_in	High	**reset_in_l**	Low	**vframe_l**	Low
pll_bypass	High	**sromd**	High	**vrefresh_l**	Low
pll_clk_in	High				

(continued on next page)

Table 4 (Cont.) Signals by Direction

Signal	Active Level	Signal	Active Level	Signal	Active Level
Output Signals					
bc_cs_l	Low	**mem_cas_l**	Low	**mem_wr_oe_l**	Low
bc_index	High	**mem_dsf**	High	**req_l**	Low
bc_oe_l	Low	**mem_dtoe_l**	Low	**rst_l**	Low
bc_we_l	Low	**mem_rasa_l<3:0>**	Low	**sromclk**	High
instr_ref	High	**mem_rasb_l<3:0>**	Low	**sromoe_l**	Low
memack_l	Low	**mem_rd_oe**	High	**tdo**	High
mem_addr<11:0>	High	**mem_write_l**	Low	**test_clk_out**	High
I/O Signals					
ad<31:0>	High	**c_be_l<3:0>**	Low	**mem_ecc<7:0>**	High
bc_dirty	High	**devsel_l**	Low	**par**	High
bc_idx_tag<4:0>	High	**frame_l**	Low	**perr_l**	Low
bc_parity	High	**irdy_l**	Low	**stop_l**	Low
bc_tag<7:0>	High	**mem_data<63:0>**	High	**trdy_l**	Low

3 Electrical Specifications

This section specifies:

- PCI electrical conformance
- Absolute maximum ratings
- Supply current and power dissipation
- dc and ac specifications

3.1 PCI Electrical Specification Conformance

The 21066 IOC PCI pins conform to the basic set of PCI electrical specifications in the *PCI Local Bus Specification, Revision 2.0,* including:

- Standard signaling

  Logic levels follow standard TTL thresholds to accommodate PCI drivers and receivers implemented with existing CMOS and TTL devices and processes.

- 33-10 support

  The 21066 supports a 33-MHz interconnection of up to ten PCI devices.

3.2 Absolute Maximum Ratings

Table 5 lists the absolute maximum ratings for the 21066. These are stress ratings only; extended exposure to the maximum ratings may affect the reliability of the device.

Caution

Although the 21066 incorporates protective circuitry to resist damage from static electric discharge, Digital recommends avoiding high-static voltages or electric fields.

Table 5 Absolute Maximum Ratings

Parameter	Minimum	Maximum
Storage temperature range	–55°C	+125°C
Active temperature range (junction)	0°C	+100°C
Supply voltage **Vdd**	–0.5 V	3.6 V
Supply voltage **Vcc** (**pll_5v**)	–0.5 V	5.5 V
ESD protection voltage	Not applicable	1500.0 V
Overshoot (5-V-safe pins)	Not applicable	See Notes for Table 6
Overshoot (5-V-nonsafe pins)	Not applicable	See Notes for Table 6
Undershoot	Not applicable	–1.0 V

3.3 Supply Current and Power Dissipation

The supply current and power dissipation are as follows:

Idd	6.1 A
Power	21 W (maximum)

Test Conditions

The supply current and power dissipation test conditions are as follows:

Package temperature with heat sink	85°C
Vdd	3.465 V
pll_5v	5.250 V
Clock frequency	166.67 MHz

3.4 Chip Power Supply Sequencing

The **Vdd** (3.3-V) and **pll_5v** (5-V) supply voltages should ramp up and ramp down simultaneously, but the two ramps need not be perfectly aligned. As shown in the following relationship, the rule is that the **pll_5v** supply must never exceed the value of the **Vdd** supply by more than 3.6 V; that is, **Vdd** cannot be less than ground or more than 3.465 V.

Vdd $\geq$ (**pll_5v** $-$ 3.6 V) for 0 V $\leq$ **Vdd** $\leq$ 3.465 V

In other words, when the 5-V supply is 3.6 V or less, the 3.3-V supply can be zero. But after the 5-V supply exceeds 3.6 V, the 3.3-V supply must match the rise in the 5-V supply, volt-for-volt. For example, when the 5-V supply reaches 4.5 V, the 3.3-V supply must be 0.9 V or more ($4.5 - 3.6 = 0.9$).

The ramp rates of the two supplies are not part of the equation and only the difference in voltages need be considered. However, power supplies with long ramp rates (several tens of milliseconds or longer) should be avoided because such slow ramp rates are likely to cause excessive die heating.

If the 3.3-V supply ramps up before the 5-V supply, there are no voltage-differential restrictions and the value of the 3.3-V supply can lead the value of the 5-V supply by any amount. However, because die power dissipation is high in the absence of clocks, and the phase-locked loop (PLL) that generates the clocks runs on 5 V, timing is restricted.

On power-up, if the **Vdd** supply leads the **pll_5v** supply, the **pll_5v** supply must reach 4.5 V no more than 1 second after the **Vdd** supply has reached 2 V. Generally, there is no problem during power-down provided that the clocks are stopped for no longer than 1 second while the 3.3-V supply remains applied. This power-down timing restriction can be satisfied by ensuring that the value of the **Vdd** supply will be 2 V or less within 1 second after the value of the **pll_5v** supply is less than 3 V.

Because the rules for the **Vdd** supply leading the **pll_5v** supply are more difficult to implement, Digital recommends that the **pll_5v** supply be applied and removed before the **Vdd** supply according to the guidelines in this section.

The **pll_5v** pin must be connected to a 5-V supply if either the 5-V PCI clamps or the PLL are used.

3.5 DC Specifications

Table 6 lists the pin characteristics.

Table 6 Pin Characteristics

Signals	Type	Notes	Internal* Pull-up or Pull-down	Signals	Type	Notes	Internal* Pull-up or Pull-down
mem_data<63:0>	I/O	❶	Pull-down	**perr_l**	I/O	❶	—
mem_ecc<7:0>	I/O	❶	Pull-down	**devsel_l**	I/O	❶	—
mem_addr<11:0>	O	❷	—	**req_l**†	O	❷	—
mem_write_l	O	❷	Pull-up	**gnt_l**	I	❶	—
mem_rasa_l<3:0>	O	❷	Pull-up	**rst_l**	O	❷	—
mem_rasb_l<3:0>	O	❷	Pull-up	**lock_l**	I	❶	—
mem_cas_l	O	❷	Pull-up	**pci_clk_in**	I	❶	—
mem_dtoe_l	O	❷	Pull-up	**memreq_l**	I	❶	—
mem_dsf	O	❷	Pull-up	**memack_l**	O	❷	—
mem_rd_oe	O	❷	Pull-down	**pll_clk_in**	I	❶	—
mem_wr_oe_l	O	❷	Pull-up	**pll_clk_in_l**	I	❶	—
bc_oe_l	O	❷	—	**pll_bypass**	I	—	—
bc_tag<7:0>	I/O	❶	Pull-down	**pll_filter**	I	—	—
bc_parity	I/O	❶	Pull-down	**pll_i_ref**	I	—	—
bc_index	O	❷	—	**pll_i_ref_ret**	I	—	—
bc_idx_tag<4:0>	I/O	❶	Pull-down	**test_clk_out**	O	❷	—
bc_cs_l	O	❷	Pull-up	**reset_in_l**	I	❶	—
bc_we_l	O	❷	—	**tdi**	I	❶	Pull-up
bc_dirty	I/O	❶	Pull-down	**tdo**	O	❷	—
vframe_l	I	❶	—	**tms**	I	❶	Pull-up
vrefresh_l	I	❶	—	**tck**	I	❶	Pull-down
ad<31:0>	I/O	❶	—	**trst_l**	I	❶	Pull-up
c_be_l<3:0>	I/O	❶	—	**irq<2:0>**	I	❶	—
frame_l	I/O	❶	—	**sromoe_l**	O	❷	—
trdy_l	I/O	❶	—	**sromd**	I	❶	—
irdy_l	I/O	❶	—	**sromclk**	O	❷	—
stop_l	I/O	❶	—	**instr_ref**	O	❷	—
par	I/O	❶	—	—	—	—	—

*Internal pull-up and pull-down are on during reset and JTAG operations.

†The **req_l** pin is tristated during chip or PCI reset.

Notes for Table 6:

❶ The I/O and input-only pins are *5-V-safe*. This means that as long as **Vdd** is 3.3 V ±5%, these pins can be safely exposed to voltages up to 5.7 V indefinitely. In addition, overshoots up to 6.4 V are allowed for up to 5% of the duty cycle, 11 ns per pulse.

The I/O and input-only pins are not 5-V-safe when **Vdd** is less than 3.135 V. Under these conditions, the pins can be exposed to voltages up to the greater of 3.6 V or **Vdd** + 2.6 V for any length of time. Overshoots up to the greater of 4.3 V or **Vdd** + 3.3 V are allowed for up to 5% of the duty cycle, 11 ns per pulse. These overshoot limits do not apply to the PCI pins unless **Vdd** < 3.1 V and **pll_5v** > 4.75 V.

❷ The output-only pins are not 5-V-safe. This means that when driving, these pins can be exposed to voltages up to 3.6 V indefinitely, but cannot be exposed to higher voltages. Therefore, if returned reflections are present, they must be limited to 3.6 V; and if test equipment is used to overdrive an output-only pin, the test equipment must not expose the pin to voltages greater than 3.6 V.

3.5.1 DC Operating Specifications

Table 7 lists the functional operating dc parameters for the 21066. The functional operating range is as follows:

Vdd = 3.3 V ±5%
Tcase = 0°C to 85°C (package temperature with heat sink) except as noted

Note

In Table 7, currents into the chip (chip sinking) are denoted as positive (+) current. Currents from the chip (chip sourcing) are denoted as negative (−) current.

Table 7 DC Parameters

Symbol	Parameter	Min	Max	Unit	Comments
Vil	Low-level input voltage	—	0.8	V	—
Vih	High-level input voltage	2.0	—	V	—
Vol	Low-level output voltage	—	0.4	V	Iol = 6 mA
Voh	High-level output voltage	2.4	—	V	Ioh = −2 mA
Iil	Input leakage current	—	±30	µA	For pins without internal pull-up or pull-down @ 0.4 V or 2.7 V
Ihl	Input leakage current	—	+175	µA	For pins with internal pull-down Vih = **Vdd**
Ill	Input leakage current	—	−260	µA	For pins with internal pull-up Vil = **Vss**
Ioz	Tristate leakage current	—	±70	µA	Pins without pull-up or pull-down @ 0.4 V or 2.7 V
		—	+195	µA	Pins with pull-down @ 2.7 V
		—	−195	µA	Pins with pull-up @ 0.4 V
Cin	Input capacitance	—	6	pF	Frequency = 1 MHz by design
Co	I/O or output-only pin capacitance	—	14	pF	Frequency = 1 MHz by design

Min = minimum, Max = maximum

(continued on next page)

Table 7 (Cont.) DC Parameters

Symbol	Parameter	Min	Max	Unit	Comments
Cclk	CLK capacitance:				
	pll_clk_in	—	10	pF	Frequency = 1 MHz by design
	pll_clk_in_l	—	10	pF	Frequency = 1 MHz by design
	pci_clk_in	—	16	pF	Frequency = 1 MHz by design
Viclk	Differential voltage:				Vcclk = 1.2 V nominal clock differential center voltage
	pll_clk_in	±0.6	—	V	
	pll_clk_in_l	±0.6	—	V	
Vbclk	Inactive clock bias voltage:				For single-ended clock operation
	pll_clk_in	1.2	—	V	
	pll_clk_in_l	1.2	—	V	
VOS	Externally driven pin voltage	—	**Vdd** + 2.6	V	Also applies to PCI pins if **Vdd** < 3.1 V and **pll_5v** > 4.75 V
Vos	Externally driven pin voltage	—	**Vdd** + 3.3	V	5% duty cycle, 11 ns maximum pulse width, also applies to PCI pins if **Vdd** < 3.1 V and **pll_5v** > 4.75 V

Min = minimum, Max = maximum

3.6 AC Specifications

The ac specifications consist of input requirements and output responses. The input requirements are rise and fall times, pulse widths, and setup and hold times. Output responses are delays from clock to signal.

Test Conditions

The test conditions for the ac parameters specified in this section (except in Table 11) are as follows:

Package temperature with heat sink	+85°C
Vss	0 V
Vdd	3.3 V ±5%
Cload	50 pF unless otherwise specified

Figure 3 defines the ac parameter measurements.

Figure 3 AC Timing Measurement

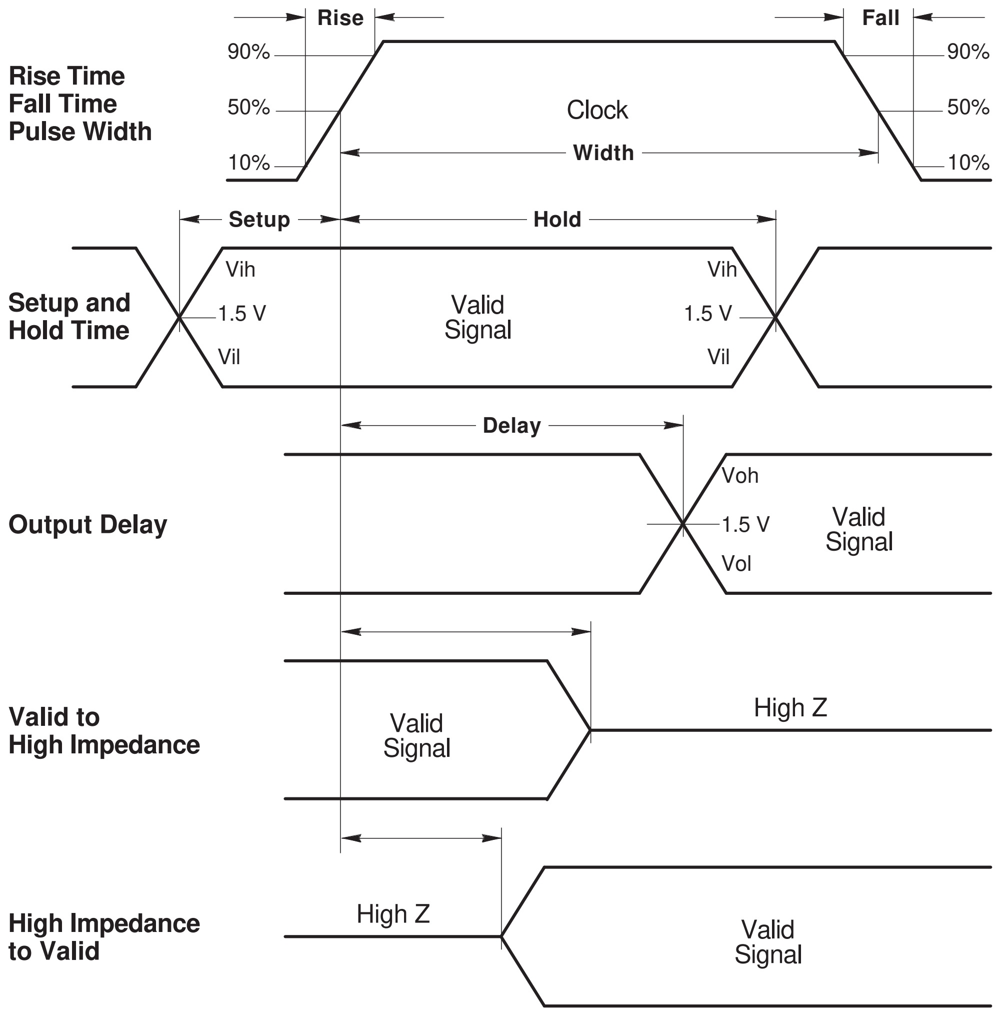


Table 8 lists the clock and reset pin ac parameters.

Table 8 Clock and Reset AC Parameters

Symbol	Parameter	Signals	Min*	Max*	Unit	Notes
Tif	Internal clock frequency		16.67	166.67	MHz	❷
Tfreq	Frequency	**pll_clk** †	TBD	—	MHz	❶
		pll_clk ‡	TBD	—		❶
		pci_clk	16.67	33		❷
Tcyc	Clock cycle time	**pll_clk**	—	—	ns	❶
		pll_clk ‡	—	—		❶
		pci_clk	30	Infinity		—
Thigh	Clock high time	**pll_clk**	0.4 ×Tcyc	—	ns	> 2.0 V
		pll_clk ‡	2.5	—		
		pci_clk	0.4×Tcyc	—		
Tlow	Clock low time	**pll_clk**	0.4×Tcyc	—	ns	< 0.8 V
		pll_clk ‡	2.5	—		
		pci_clk	0.4×Tcyc	—		
Tr	Clock rise time	**pll_clk**	—	3.0	ns	0.8 V to 2.0 V
		pll_clk ‡	—	0.8		
		pci_clk	—	3.0		
Tf	Clock fall time	**pll_clk**	—	3.0	ns	2.0 V to 0.8 V
		pll_clk ‡	—	0.8		
		pci_clk	—	3.0		
Trst	Reset pulse width	**reset_in_l**	10×Tcyc	—	ms	—
		rst (PCI)	1	—		❸
Trss	Setup time	**reset_in_l**	5	—	ns	❹
Trst_clk	Clock active time to end of **rst**		100	—	μs	—
Tclkout	Delay time	**test_clk_out**	—	25	ns	❺

*Min = minimum, Max = maximum.

†Bypass = 0

‡Bypass = 1

Notes for Table 8:

❶ **pll_clk** refers to both **pll_clk_in** and **pll_clk_in_l**, the differential clock inputs to the chip. The maximum internal clock frequency (Tif) is 166.67 MHz. The internal high-frequency clock is a function of the programmed multiplier value on the **irq<2:0>** pins during reset. The **pll_clk** maximum frequency must be chosen such that the maximum internal clock frequency (Tif) is not exceeded. When **pll_bypass** is asserted, the internal clock frequency equals the external clock frequency. When **pll_bypass** is deasserted, the typical internal clock frequencies are as follows:

pll_clk_in (MHz) =	**16.67**	**25.00**	**33.33**
Multiplier Values	**Internal Clock Frequency (MHz)**		
9	150.00	225.00 *	300.00*
8	133.33	200.00*	266.67*
7	116.67	175.00*	233.33*
6	100.00	150.00	200.00*
5	83.33	125.00	166.67
4	66.67	100.00	133.33
3	50.00	75.00	100.00
2	33.33	50.00	66.67

*The maximum internal frequency (Tif) limits the range of multiplier values for a given **pll_clk_in** frequency. Although these multiplier values are available, they should not be used at this **pll_clk_in** frequency.

❷ The PCI clock input is asynchronous to the internal CPU clock. For correct operation, the PCI clock frequency must be less than or equal to the internal CPU clock frequency.

❸ Trst typical = 100 ms.

❹ **reset_in_l** is an asynchronous input. The setup time is only for the tester.

❺ **test_clk_out** delay is measured with respect to the rising edge of **pll_clk_in** by design.

To maintain stable operation, a 0.01 µF capacitor is connected between the **pll_filter** pin and **Vss**. It sets the feedback loop time constant needed to regulate the speed with which the PLL responds to changes in frequency or operating conditions. Additionally, a 4 k Ω resistor is connected between the **pll_i_ref** pin and **Vss**. The constant current that flows in the resistor provides the reference to the analog PLL circuits.

Table 9 lists the memory controller pin ac parameters. The parameters are relative to the core clock.

Table 9 Memory Controller AC Parameters

Symbol	Parameter	Signals	Minimum ns	Maximum ns	Notes
Tsv1	Setup time	**mem_data<63:0>** **mem_ecc<7:0>**	1.50	—	❶
Th1	Hold time	**mem_data<63:0>** **mem_ecc<7:0>**	0.50	—	❶
Td1	Valid delay time	**mem_data<63:0>** **mem_ecc<7:0>**	—	4.50	❷
Tvz1	Valid to high-Z delay time	**mem_data<63:0>** **mem_ecc<7:0>**	—	4.00	❸
Tzv1	High-Z to valid delay time	**mem_data<63:0>** **mem_ecc<7:0>**	1.50	—	❷
Tsv2	Setup time	**bc_tag<7:0>** **bc_parity** **bc_idx_tag<4:0>** **bc_dirty**	2.00	—	❶
Th2	Hold time	**bc_tag<7:0>** **bc_parity** **bc_idx_tag<4:0>** **bc_dirty**	0.00	—	❶
Td2	Valid delay time	**bc_tag<7:0>** **bc_parity** **bc_idx_tag<4:0>** **bc_dirty** **mem_addr<11:0>** **bc_index**	—	4.25	❷
Tvz2	Valid to high-Z delay time	**bc_tag<7:0>** **bc_parity** **bc_idx_tag<4:0>** **bc_dirty** **bc_index**	—	3.75	❸
Tzv2	High-Z to valid delay time	**bc_tag<7:0>** **bc_parity** **bc_idx_tag<4:0>** **bc_dirty** **bc_index**	1.50	—	❷

(continued on next page)

Table 9 (Cont.) Memory Controller AC Parameters

Symbol	Parameter	Signals	Minimum ns	Maximum ns	Notes
Td3	Valid delay time	**mem_cas_l** **mem_rd_oe** **mem_wr_oe_l**	—	4.00	➋
Td4	Valid delay time	**mem_rasa_l<3:0>** **mem_rasb_l<3:0>** **mem_dtoe_l** **mem_dsf** **mem_write_l**	—	3.75	➋
Td5	Valid delay time	**bc_we_l** **bc_oe_l** **bc_cs_l**	—	4.50	➋
Tpwl	Pulse width low time	**vrefresh_l** **vframe_l**	10.00	85.00	—
Tpwh	Pulse width high	**vrefresh_l** **vframe_l**	1000.00	—	—
Tno	Time by which the assertion of **vrefresh_l** and **vframe_l** must be separated	**vrefresh_l** **vframe_l**	1000.00	—	—

Notes for Table 9:

➊ Setup and hold times are measured with respect to the rising edge of the **test_clk_out** pin. The **test_clk_out** pin, when loaded with a lumped 40 pF load, imitates the internal clock. The **test_clk_out** pin is intended for test purposes only.

➋ The drive times assume a lumped 50 pF load and are measured with respect to the **test_clk_out** pin.

➌ The pin is defined to be in tristate when a 2 mA current source changes the output voltage by 50 mV. The pin is assumed to be connected to a lumped 40 pF load for this test.

Table 10 lists the I/O controller pin ac parameters. The parameters are relative to the PCI clock signal **pci_clk_in**.

Table 10 IOC Pin AC Parameters

Symbol	Parameter	Signals	Minimum	Maximum
Tval	Clock to signal valid delay time	**ad<31:0>** **c_be_l<3:0>** **frame** **trdy_l** **irdy_l** **stop_l** **par** **perr_l** **devsel_l** **lock_l**	—	11 ns *
Tival	Clock to signal invalid delay time	**ad<31:0>** **c_be_l<3:0>** **frame** **trdy_l** **irdy_l** **stop_l** **par** **perr_l** **devsel_l** **lock_l**	2.0 ns	—
Ton	High-Z to active delay time	**ad<31:0>** **c_be_l<3:0>** **frame** **trdy_l** **irdy_l** **stop_l** **par** **perr_l** **devsel_l** **lock_l**	2.0 ns	—

*Cload = 50 pF

(continued on next page)

Table 10 (Cont.) IOC Pin AC Parameters

Symbol	Parameter	Signals	Minimum	Maximum
Toff	Active to high-Z delay time	**ad<31:0>** **c_be_l<3:0>** **frame** **trdy_l** **irdy_l** **stop_l** **par** **perr_l** **devsel_l** **lock_l**	2.0 ns	28 ns
Tsu	Input signal valid setup time	**ad<31:0>** **c_be_l<3:0>** **frame** **trdy_l** **irdy_l** **stop_l** **par** **perr_l** **devsel_l** **lock_l** **memreq_l**	7.0 ns	—
Th	Input signal hold time	**ad<31:0>** **c_be_l<3:0>** **frame** **trdy_l** **irdy_l** **stop_l** **par** **perr_l** **devsel_l** **lock_l** **memreq_l**	0.0 ns	—
Tackv	Valid delay time from clock rising edge	**memack_l**	3.6 ns	15 ns *
Tval-side	Signal valid delay time	**req_l**	—	12 ns
Tsu-side	Signal valid setup time	**gnt_l**	12 ns	—

*Cload = 50 pF

(continued on next page)

Table 10 (Cont.) IOC Pin AC Parameters

Symbol	Parameter	Signals	Minimum	Maximum
Vclamp	PCI pin clamp voltage†	—	6.8 V	@ Iih = 64 mA **Vdd** = 3.3 V − 5% **pll_5v** = 5.0 V + 5%

†PCI pin clamps protect only the 21066 and not other devices on the bus.

If a 5-V signaling environment is used on the PCI bus, the **pll_5v** pin must be connected to a 5-V supply. Table 11 (abridged from the *PCI Local Bus Specification, Revision 2.0*) specifies the ac parameters for 5-V signaling.

Table 11 AC Specifications for 5 V Signaling

Symbol	Parameter	Condition	Minimum	Maximum	Unit
Ioh(ac)	Switching current high	$0 < Vout \leq 1.4$ $1.4 < Vout < 2.4$	-44 $-44 + (Vout - 1.4) / 0.024$	— Equation A*	mA mA
	Test point	Vout = 3.1	—	-142	mA
Iol(ac)	Switching current low	$Vout \geq 2.2$ $2.2 > Vout > 0.55$	95 Vout / 0.023	— Equation B*	mA mA
	Test point	Vout = 0.71	—	206	mA
Icl	Low clamp current	$-5 < Vin \leq -1$	$-25 + (Vin + 1) / 0.015$	—	mA
Tr	Unloaded output rise time	0.4 V to 2.4 V	1	—	V/ns
Tf	Unloaded output fall time	2.4 V to 0.4 V	1	—	V/ns

*See the *PCI Local Bus Specification, Revision 2.0*.

Table 12 lists the JTAG pin ac parameters.

Table 12 JTAG Pin AC Parameters

Symbol	Parameter	Signals	Minimum	Maximum	Unit	Comments
Tjf	Frequency	**tck**	0	10	MHz	—
Tjp	Period	**tck**	100	—	ns	—
Tjht	High time	**tck**	45	—	ns	—
Tjlt	Low time	**tck**	45	—	ns	—
Tjrt	Rise time	**tck**	—	10	ns	Measured between 0.8 V and 2.0 V
Tjft	Fall time	**tck**	—	10	ns	Measured between 2.0 V and 0.8 V
Tjs	Setup time	**tdi** **tms**	10	—	ns	With respect to **tck** rising edge
Tjh	Hold time	**tdi** **tms**	25	—	ns	With respect to **tck** rising edge
Tjd	Valid delay	**tdo**	—	30	ns	With respect to **tck** falling edge Cload = 50 pF
Tjfd	Float delay	**tdo**	—	30	ns	With respect to **tck** falling edge

Table 13 lists the ac parameters for the miscellaneous pins. The parameters specified are for test purposes only and are measured with respect to the **test_clk_out** signal.

Table 13 Miscellaneous Pin AC Parameters

Symbol	Parameter	Signals	Minimum	Maximum
Tmst	Setup time	**irq<2:0>** **sromd**	5 ns	—
Tmht	Hold time	**irq<2:0>** **sromd**	0 ns	—

4 Mechanical Specifications

Figures 4 and 5 show the 287-pin standard pin grid array (PGA) package and its dimensions.

Figure 4 21066 Package — Top and Side

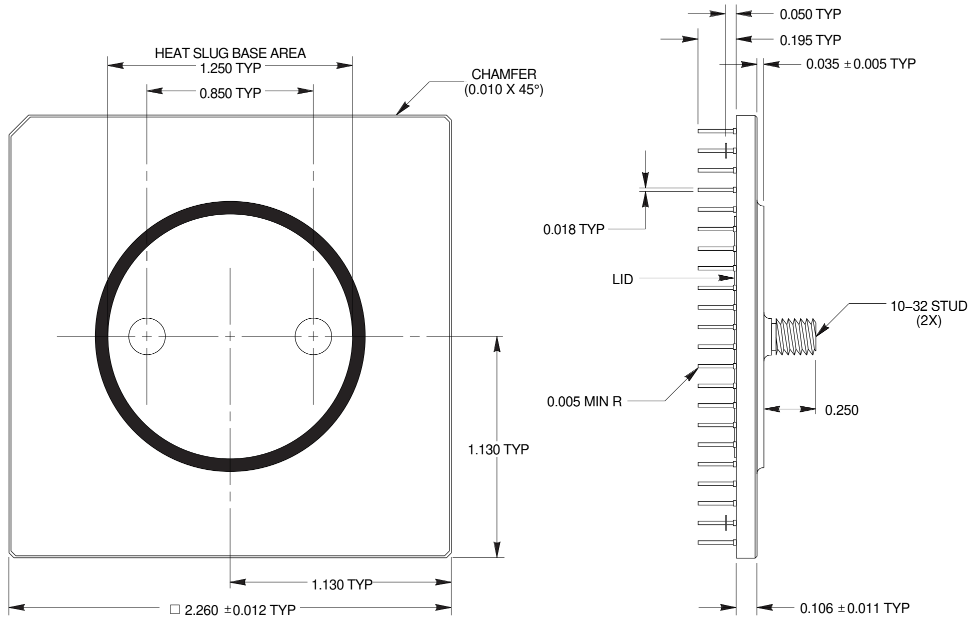
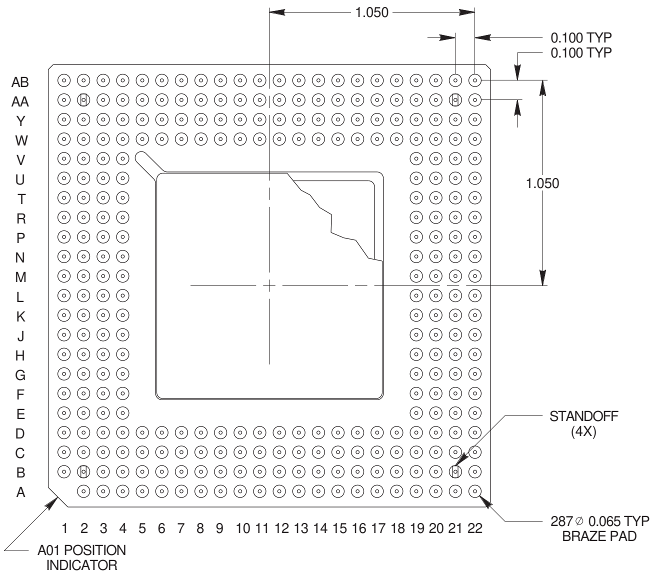


Figure 5 21066 Package — Bottom



5 Thermal Specifications

Sections 5.1 and 5.2 specify the DECchip 21066 operating temperature and thermal resistance.

5.1 Operating Temperature

The 21066 is specified to operate when the temperature at the center of the heat sink (T_c) is within the range of 0°C to 85°C. Temperature T_c should be measured at the center of the heat sink (between the two package studs).

5.2 Thermal Resistance

The following equations define the junction-to-ambient and junction-to-heat-sink thermal resistance values:

$$\Theta_j a = \frac{(T_j - T_a)}{P}$$

$$\Theta_j hs = \frac{(T_j - T_c)}{P}$$

$$\Theta_j a = \Theta_j hs + \Theta_h sa$$

$$T_j = T_a + P * \Theta_j a$$

The symbols in the previous equations are defined as follows:

$\Theta_j a$ is the junction-to-ambient thermal resistance (°C/W)
$\Theta_j hs$ is the junction-to-heat-sink thermal resistance (°C/W)
$\Theta_h sa$ is the heat-sink-to-ambient thermal resistance (°C/W)
T_j is the maximum junction temperature (°C)
T_a is the ambient temperature (°C)
T_c is the case temperature measured at a predefined location on the heat sink (°C)
P is the power dissipation (W)

Table 14 lists the values for the junction-to-ambient thermal resistance $\Theta_j a$ and junction-to-heat-sink thermal resistance $\Theta_j hs$ for the 287-pin ceramic PGA. Table 15 shows the allowable T_a (without exceeding T_c) at various airflows and operating frequencies.

Note

The heat sink greatly improves the ambient temperature requirement and Digital recommends its use.

The unidirectional heat sink (aluminum alloy 6063) approximate dimensions (l $\times$ w $\times$ h) are 6.6 $\times$ 6.6 $\times$ 3.0 cm (2.6 $\times$ 2.6 $\times$ 1.2 in). The Grafoil pad is the interface material between the package and the heat sink.

Table 14 $\Theta_j a$ at Various Airflows

Airflow: (ft/min)	50	100	200	400	600	800	1000
$\Theta_j a$ with heat sink: (°C/W)	3.35	2.50	2.10	1.70	1.50	1.35	1.30

Frequency: 166.67 MHz
Junction-to-heat-sink thermal resistance: 0.70 (°C/W) *

*If the Grafoil pad is not used as the interface material, the value of junction-to-heat-sink thermal resistance will go up by 0.45 (°C/W) to 1.15 (°C/W). This will also increase the value of the junction-to-ambient thermal resistance by 0.45 (°C/W) at various airflow points.

Table 15 Maximum T_a at Various Airflows

Airflow: (ft/min)	50	100	200	400	600	800	1000
T_a with heat sink: (°C)*	29.5	47.5	56.0	64.5	68.5	71.5	72.5

Frequency: 166.67 MHz
Power: 21 W

*If the Grafoil pad is not used as the interface material, the maximum ambient temperature will decrease approximately 9.5°C.

6 Register Summary

The tables in this section give a summary of the 21066 implementation-specific internal processor registers (IPRs), memory controller registers, and I/O controller (IOC) registers. For information about the architecturally specified IPRs, see the *Alpha Architecture Reference Manual*.

Table 16 21066-Specific Internal Processor Registers

Mnemonic	Register Name	Field*	Index*
Instruction Fetch and Decode Unit Registers			
ASTER	Asynchronous system trap interrupt enable	IBX	18
ASTRR	Asynchronous system trap request	IBX	14
EXC_ADDR	Exception Address	IBX	4
EXC_SUM	Exception Summary	IBX	10
HIER	Hardware interrupt enable	IBX	16
HIRR	Hardware interrupt request	IBX	12
ICCSR	Instruction cache control and status	IBX	2
ITBASM	Instruction translation buffer address space match	IBX	7
ITBIS	Instruction translation buffer initial state	IBX	8
ITB_PTE	Instruction translation buffer page table entry	IBX	1
ITB_PTE_TEMP	Instruction translation buffer page table entry temporary	IBX	3
ITBZAP	Instruction translation buffer ZAP	IBX	6
PAL_BASE	PAL base address	IBX	11
PS	Processor status	IBX	9
SIER	Software interrupt enable	IBX	17
SIRR	Software interrupt request	IBX	13
SL_CLR	Clear serial line interrupt	IBX	19
SL_RCV	Serial line receive	IBX	5
SL_XMIT	Serial line transmit	IBX	22
TB_TAG	Translation buffer tag	IBX	0
Load and Store Unit Registers			
ABOX_CTL	Load and store unit (Abox) control	ABX	14
ALT_MODE	Alternate processor mode	ABX	15
CC	Cycle counter	ABX	16
CC_CTL	Cycle counter control	ABX	17

*HW_MFPR and HW_MTPR instruction fields: PAL, ABX, IBX, and Index (<7,6,5,4:0>).

(continued on next page)

Table 16 (Cont.) 21066-Specific Internal Processor Registers

Mnemonic	Register Name	Field*	Index*
Load and Store Unit Registers			
DC_STAT	Data cache status	ABX	12
DTBASM	Data translation buffer address space match	ABX	7
DTB_CTL	Data translation buffer control	ABX	0
DTBIS	Data translation buffer invalidate single	ABX	8
DTB_PTE	Data translation buffer page table entry	ABX	2
DTB_PTE_TEMP	Data translation buffer page table entry temporary	ABX	3
DTBZAP	Data translation buffer ZAP	ABX	6
FLUSH_IC	Flush instruction cache	ABX	21
FLUSH_IC_ASM	Flush instruction cache address space match	ABX	23
MM_CSR	Memory management control and status	ABX	4
VA	Virtual address	ABX	5
PAL Temporary Registers			
PAL_TEMP<31:0>	PAL_TEMP internal processor	PAL	31..0

*HW_MFPR and HW_MTPR instruction fields: PAL, ABX, IBX, and Index (<7,6,5,4:0>).

Table 17 Memory Controller Registers

Mnemonic	Register Name	Address (hexadecimal)
BCR0	Bank configuration 0	1 2000 0000
BCR1	Bank configuration 1	1 2000 0008
BCR2	Bank configuration 2	1 2000 0010
BCR3	Bank configuration 3	1 2000 0018
BMR0	Bank mask 0	1 2000 0020
BMR1	Bank mask 1	1 2000 0028
BMR2	Bank mask 2	1 2000 0030
BMR3	Bank mask 3	1 2000 0038
BTR0	Bank timing 0	1 2000 0040
BTR1	Bank timing 1	1 2000 0048
BTR2	Bank timing 2	1 2000 0050
BTR3	Bank timing 3	1 2000 0058
GTR	Global timing	1 2000 0060

(continued on next page)

Table 17 (Cont.) Memory Controller Registers

Mnemonic	Register Name	Address (hexadecimal)
ESR	Error status	1 2000 0068
EAR	Error address	1 2000 0070
CAR	Cache control	1 2000 0078
VGR	Video and graphics control	1 2000 0080
PLM	Plane mask	1 2000 0088
FOR	Foreground	1 2000 0090

Table 18 I/O Controller Registers

Mnemonic	Register Name	Address (hexadecimal)
IOC_HAE	Host address extension	1 8000 0000
IOC_CFG	Configuration cycle type	1 8000 0020
IOC_STAT0	Status 0	1 8000 0040
IOC_STAT1	Status 1	1 8000 0060
IOC_TBIA	Translation buffer invalidate all	1 8000 0080
IOC_TB_ENA	Translation buffer enable	1 8000 00A0
IOC_SFT_RST	PCI soft reset	1 8000 00C0
IOC_PAR_DIS	Parity disable	1 8000 00E0
IOC_W_BASE0	Window base 0	1 8000 0100
IOC_W_BASE1	Window base 1	1 8000 0120
IOC_W_MASK0	Window mask 0	1 8000 0140
IOC_W_MASK1	Window mask 1	1 8000 0160
IOC_T_BASE0	Translated base 0	1 8000 0180
IOC_T_BASE1	Translated base 1	1 8000 01A0
IOC_TB_TAG0	Translation buffer tag 0	1 8100 0000
IOC_TB_TAG1	Translation buffer tag 1	1 8100 0020
IOC_TB_TAG2	Translation buffer tag 2	1 8100 0040
IOC_TB_TAG3	Translation buffer tag 3	1 8100 0060
IOC_TB_TAG4	Translation buffer tag 4	1 8100 0080
IOC_TB_TAG5	Translation buffer tag 5	1 8100 00A0
IOC_TB_TAG6	Translation buffer tag 6	1 8100 00C0
IOC_TB_TAG7	Translation buffer tag 7	1 8100 00E0
IOC_IACK_SC	Interrupt vector and special cycle	*

*Any quadword aligned address in the range 1A0000000..1BFFFFFE0.

7 Instruction Summary

The tables in this section summarize the common instructions implemented by the Alpha AXP architecture, the PALmode instructions required by all Alpha AXP implementations, and the architecturally reserved PALmode instructions implemented in the DECchip 21066 microprocessor. The instruction summaries are contained in the following tables:

Instructions	Table
Memory integer load and store	19
Integer control	20
Integer arithmetic	21
Logical and shift	22
Byte manipulation	23
Memory format floating-point	24
Floating-point branch	25
Floating-point operate	26
Miscellaneous	27
VAX compatibility instructions	28
Required PALmode	29
Architecturally reserved PALmode	30

Table 19 Memory Integer Load and Store Instructions

Mnemonic	Operation
LDA	Load address
LDAH	Load address high
LDL	Load sign-extended longword
LDL_L	Load sign-extended longword locked
LDQ	Load quadword
LDQ_L	Load quadword locked
LDQ_U	Load quadword unaligned
STL	Store longword
STL_C	Store longword conditional
STQ	Store quadword
STQ_C	Store quadword conditional
STQ_U	Store quadword unaligned

Table 20 Integer Control Instructions

Mnemonic	Operation
BEQ	Branch if register equal to zero
BGE	Branch if register greater than or equal to zero
BGT	Branch if register greater than zero
BLBC	Branch if register low bit is clear
BLBS	Branch if register low bit is set
BLE	Branch if register less than or equal to zero
BLT	Branch if register less than zero
BNE	Branch if register not equal to zero
BR	Unconditional branch
BSR	Branch to subroutine
JMP	Jump
JSR	Jump to subroutine
RET	Return from subroutine
JSR_COROUTINE	Jump to subroutine return

Table 21 Integer Arithmetic Instructions

Mnemonic	Operation
ADD	Add quadword/longword
S4ADD	Scaled add by 4
S8ADD	Scaled add by 8
CMPEQ	Compare signed quadword equal
CMPLT	Compare signed quadword less than
CMPLE	Compare signed quadword less than or equal
CMPULT	Compare unsigned quadword less than
CMPULE	Compare unsigned quadword less than or equal
MUL	Multiply quadword/longword
UMULH	Multiply quadword unsigned high
SUB	Subtract quadword/longword
S4SUB	Scaled subtract by 4
S8SUB	Scaled subtract by 8

Table 22 Logical and Shift Instructions

Mnemonic	Operation
AND	Logical product
BIC	Logical product with complement
BIS	Logical sum (OR)
EQV	Logical equivalence (XORNOT)
ORNOT	Logical sum with complement
XOR	Logical difference
CMOVxx	Conditional move integer
SLL	Shift left logical
SRA	Shift right arithmetic
SRL	Shift right logical

Table 23 Byte-Manipulation Instructions

Mnemonic	Operation
CMPBGE	Compare byte
EXTBL	Extract byte low
EXTWL	Extract word low
EXTLL	Extract longword low
EXTQL	Extract quadword low
EXTWH	Extract word high
EXTLH	Extract longword high
EXTQH	Extract quadword high
INSBL	Insert byte low
INSWL	Insert word low
INSLL	Insert longword low
INSQL	Insert quadword low
INSWH	Insert word high
INSLH	Insert longword high
INSQH	Insert quadword high
MSKBL	Mask byte low
MSKWL	Mask word low
MSKLL	Mask longword low
MSKQL	Mask quadword low

(continued on next page)

Table 23 (Cont.) Byte-Manipulation Instructions

Mnemonic	Operation
MSKWH	Mask word high
MSKLH	Mask longword high
MSKQH	Mask quadword high
ZAP	Zero bytes
ZAPNOT	Zero bytes not

Table 24 Memory Format Floating-Point Instructions

Mnemonic	Operation	Subset
LDF	Load F_floating	VAX
LDG	Load G_floating (load D_floating)	VAX
LDS	Load S_floating (load longword integer)	IEEE and VAX
LDT	Load T_floating (load quadword integer)	IEEE and VAX
STF	Store F_floating	VAX
STG	Store G_floating (store D_floating)	VAX
STS	Store S_floating (store longword integer)	IEEE and VAX
STT	Store T_floating (store quadword integer)	IEEE and VAX

Table 25 Floating-Point Branch Instructions

Mnemonic	Operation	Subset
FBEQ	Floating branch equal	IEEE and VAX
FBGE	Floating branch greater than or equal	IEEE and VAX
FBGT	Floating branch greater than	IEEE and VAX
FBLE	Floating branch less than or equal	IEEE and VAX
FBLT	Floating branch less than	IEEE and VAX
FBNE	Floating branch not equal	IEEE and VAX

Table 26 Floating-Point Operate Instructions

Mnemonic	Operation	Subset
Arithmetic Operations		
ADDF	Add F_floating	VAX
ADDG	Add G_floating	VAX
ADDS	Add S_floating	IEEE
ADDT	Add T_floating	IEEE
CMPGxx	Compare G_floating	VAX
CMPTxx	Compare T_floating	IEEE
CVTDG	Convert D_floating to G_floating	VAX
CVTGD	Convert G_floating to D_floating	VAX
CVTGF	Convert G_floating to F_floating	VAX
CVTGQ	Convert G_floating to quadword	VAX
CVTQF	Convert quadword to F_floating	VAX
CVTQG	Convert quadword to G_floating	VAX
CVTQS	Convert quadword to S_floating	IEEE
CVTQT	Convert quadword to T_floating	IEEE
CVTST	Convert S_floating to T_floating	IEEE
CVTTQ	Convert T_floating to quadword	IEEE
CVTTS	Convert T_floating to S_floating	IEEE
DIVF	Divide F_floating	VAX
DIVG	Divide G_floating	VAX
DIVS	Divide S_floating	IEEE
DIVT	Divide T_floating	IEEE
MULF	Multiply F_floating	VAX
MULG	Multiply G_floating	VAX
MULS	Multiply S_floating	IEEE
MULT	Multiply T_floating	IEEE
SUBF	Subtract F_floating	VAX
SUBG	Subtract G_floating	VAX
SUBS	Subtract S_floating	IEEE
SUBT	Subtract T_floating	IEEE

(continued on next page)

Table 26 (Cont.) Floating-Point Operate Instructions

Mnemonic	Operation	Subset
Bit and FPCR Operations		
CPYS	Copy sign	IEEE and VAX
CPYSE	Copy sign and exponent	IEEE and VAX
CPYSN	Copy sign negate	IEEE and VAX
CVTLQ	Convert longword to quadword	IEEE and VAX
CVTQL	Convert quadword to longword	IEEE and VAX
FCMOVxx	Floating conditional move	IEEE and VAX
MF_FPCR	Move from floating-point control register	IEEE and VAX
MT_FPCR	Move to floating-point control register	IEEE and VAX

Table 27 Miscellaneous Instructions

Mnemonic	Operation
CALL_PAL	Call privileged architecture library routine
EXCB	Exception barrier
FETCH	Prefetch data
FETCH_M	Prefetch data, modify intent
MB	Memory barrier
RPCC	Read process cycle counter
TRAPB	Trap barrier
WMB	Write memory barrier

Table 28 VAX Compatibility Instructions

Mnemonic	Operation
RC	Read and clear
RS	Read and set

Table 29 Required PALmode Instructions

Mnemonic	Operation
HALT	Halt processor
IMB	Instruction stream memory barrier

Table 30 Architecturally Reserved PALmode Instructions

Mnemonic	Operation
HW_MTPR	Move data to processor register
HW_MFPR	Move data from processor register
HW_LD	Move data from memory
HW_ST	Move data to memory
HW_REI	Return from PALmode exception

Technical Support, Ordering, and Associated Literature

This section describes how to obtain DECchip information and technical support, as well as how to order DECchip products and associated literature.

Calling the DECchip Information Line for Information and Technical Support

Call the DECchip Information Line for information and technical support:

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

Ordering DECchip Products

To order the DECchip 21066 Alpha AXP Microprocessor and DECchip 21066 and DECchip 21068 Alpha AXP Evaluation Board, contact your local Digital sales office. When working with your sales representative, you may be able to take advantage of discounts and volume pricing.

You can order the following DECchip products from Digital.

Product	Order Number
DECchip 21066 Alpha AXP Microprocessor	21066–AA
DECchip 21066 and DECchip 21068 Alpha AXP Evaluation Board (150 MHz)	21A03–01
DECchip 21066 and DECchip 21068 Alpha AXP Evaluation Board (166 MHz)	21A03–02
DECchip 21066 and DECchip 21068 Alpha AXP Evaluation Board (200 MHz)	21A03–03

Ordering Associated DECchip Literature

The following table lists some of the DECchip literature that is available. For a complete list, and for information about ordering, contact the DECchip Information Line.

Title	Order Number
Alpha Architecture Reference Manual[1]	EY–L520E–DP–YCH
DECchip 21066 and DECchip 21068 Alpha AXP Evaluation Board Product Brief	EC–N0623–72
DECchip 21066 Alpha AXP Microprocessor Product Brief	EC–N0250–72
DECchip 21066 and DECchip 21068 Alpha AXP Evaluation Board User's Guide	EC–N0626–72
DECchip 21066 and DECchip 21068 Alpha AXP Microprocessors Hardware Reference Manual	EC–N2681–72
DECchip 21066 Alpha AXP Microprocessor Product Brief	EC–N0250–72
DECchip 21068 Alpha AXP Microprocessor Product Brief	EC–N0638–72

[1]To order and purchase the *Alpha Architecture Reference Manual*, call **1–800–DIGITAL** from the U.S. or Canada, or contact your local Digital office, or technical or reference bookstore where Digital Press books are distributed by Prentice Hall.

Ordering Third-Party Literature

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

Title	Vendor
PCI Local Bus Specification, Revision 2.0	PCI Special Interest Group M/S HF3–15A 5200 N.E. Elam Young Pkwy Hillsboro, Oregon 97124–6497 1–503–696–2000