Alpha 21066A Microprocessor Evaluation Board (EB66+)

User’s Guide

Order Number: EC–QDVCB–TE

Revision/Update Information: This document supersedes the Alpha 21066A Microprocessor Evaluation Board (EB66+) User’s Guide (EC-QDVCA-TE).

Digital Equipment Corporation
Maynard, Massachusetts

February 1995

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

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

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

4 Power and Environmental Requirements

A Address Map

B SROM Initialization

C Technical Support and Ordering Information

Glossary

Index

Figures

Tables

About This Guide

This guide describes the Alpha 21066A Microprocessor Evaluation Board (also called the EB66+), an evaluation and development module for computing systems based on the 21066A microprocessor.

Audience

This guide is intended for system designers and others who use the EB66+ to design or evaluate computer systems based on the 21066A microprocessor.

Scope

This guide describes the features, configuration, and functional operation of the EB66+. This guide does not include specific bus specifications (for example, PCI, ISA, or SCSI buses). Additional information is available in the EB66+ schematics and appropriate vendor and IEEE specifications. See Appendix C for information about how to order related documentation and obtain additional technical support.

Content

This guide contains the following chapters and appendixes:

- Chapter 1, Introduction to the EB66+, is an overview of the EB66+.
- Chapter 2, System Configuration and Connections, provides EB66+ configuration information and identifies all connectors, sockets, and jumpers.
- Chapter 3, Functional Description, is a functional description of the EB66+.
- Chapter 4, Power and Environmental Requirements, describes the EB66+ power and environmental requirements.
- Appendix A, Address Map, provides a map of the EB66+ memory and I/O physical address space.
- Appendix B, SROM Initialization, describes the SROM bootstrap procedure.

- Appendix C, Technical Support and Ordering Information, provides information about technical support, ordering information, and associated literature.
- Glossary lists and defines terms associated with the EB66+.

Document Conventions

This section provides the conventions used in this guide.

Addresses

All addresses listed in this guide are hexadecimal. In case of ambiguity, the address is marked as hexadecimal (for example, 800_{16}).

Bit and Field Abbreviations

The following list describes the bit and field abbreviations:

Bit/Field Abbreviation	Description
RO (read only)	Bits and fields specified as RO can be read but not written.
RW (read/write)	Bits and fields specified as RW can be read and written.
WO (write only)	Bits and fields specified as WO can be written but not read.

Bit Notation

Multiple bit fields are shown as extents (see Ranges and Extents in this section).

Caution

Cautions indicate potential damage to equipment or data.

Data Units

The following data unit terminology, common within Digital, is used throughout this guide:

Term	Words	Bytes	Bits
Word	1	2	16
Longword	2	4	32
Quadword	4	8	64

Note

Notes provide additional information.

Numbering

All numbers are decimal or hexadecimal unless otherwise indicated. In case of ambiguity, a subscript indicates the radix of nondecimal numbers. For example, 19 is a decimal number, but 19_{16} is a hexadecimal number.

Ranges and Extents

Ranges are specified by a pair of numbers separated by two periods (..) and are inclusive. For example, a range of 0..4 includes the integers 0, 1, 2, 3, and 4.

Extents are specified by a single number, or a pair of numbers in angle brackets (< >) separated by a colon (:) and are inclusive. For example, bits <7:3> specify an extent including bits 7, 6, 5, 4, and 3.

Schematic References

Logic schematics are included in the EB66+ design package. In this guide, references to schematic pages are printed in italics. For example:

“ . . . transceivers (schematic page *eb66p.16*).”

Signal Names

Signal names in text are printed in boldface lowercase type. For example, “ . . . bits **adr<11:0>** are delivered to DRAM . . . ”

1

Introduction to the EB66+

The Alpha 21066A Microprocessor Evaluation Board (also called EB66+) is an evaluation and development board for computing systems based on the Alpha 21066A microprocessor (also called the 21066A). It gives the user a single-board platform for the design, integration, and analysis of supporting logic, subsystems, and software. The EB66+ is one of many example designs available for Digital microprocessors that implement the Alpha architecture. This chapter describes the EB66+, its uses, and its features. Figure 1–1 is a block diagram of the system.

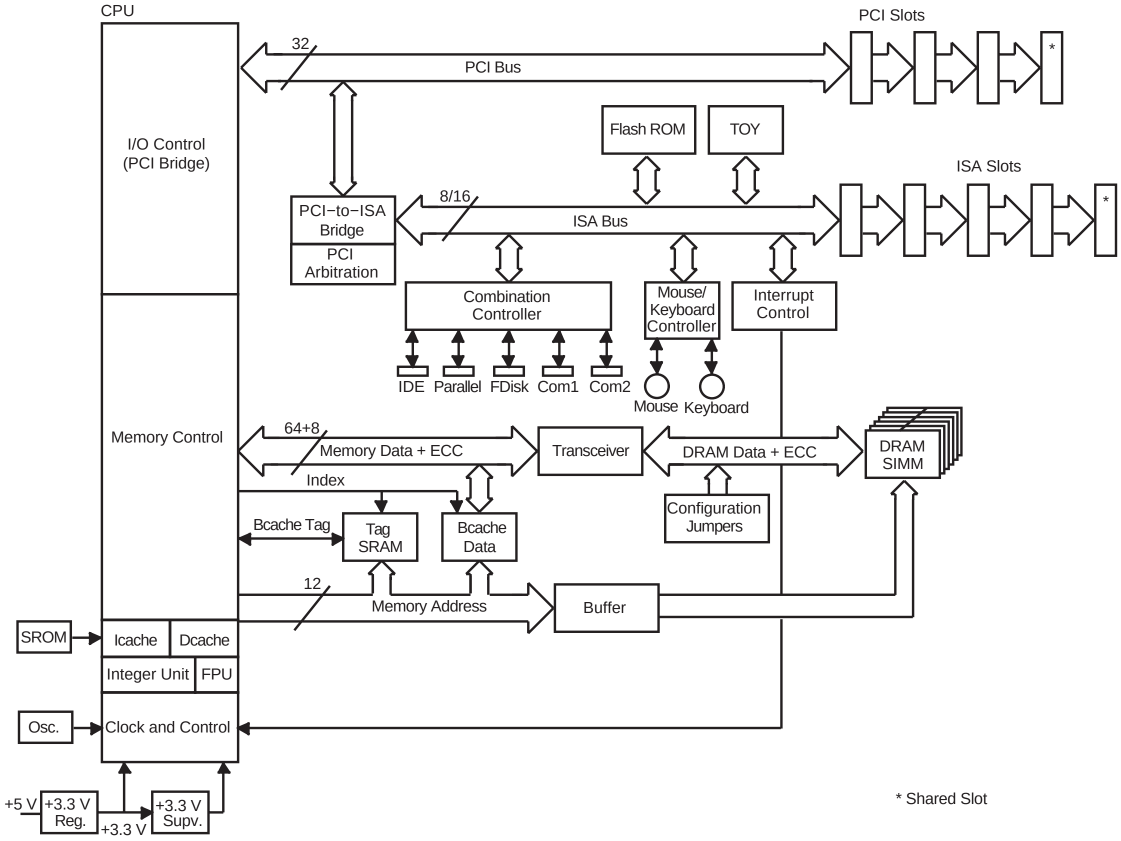
1.1 System Components

The Alpha 21066A Microprocessor Evaluation Board includes:

- A 21066A microprocessor running at frequencies of up to 233 MHz. Lower frequencies are available through onboard jumpers.
- An ECC-protected 64-bit-wide dynamic RAM (DRAM) main memory ranging in size from 8 MB to 384 MB.
- An onboard external backup cache (Bcache) subsystem comprising 1 MB of static RAM (SRAM). The Bcache is direct mapped, write-back, with a quadword block size. It contains ECC protection for data and parity protection for the tag.
- A serial boot ROM (SROM).
- A flash ROM for debugging and operating system support.
- A microprocessor containing an integrated bridge to peripheral component interconnect (PCI) local bus interface running at 33 MHz with up to four expansion slots. Three PCI slots are dedicated, and one slot is shared with the Industry Standard Architecture (ISA) bus.

1.1 System Components

Figure 1–1 EB66+ System Block Diagram



- An ISA (IEEE-P996) interface with up to five ISA expansion slots. Four ISA slots are dedicated, and one is shared with the PCI bus.
- An embedded port with control to integrated device electronics (IDE) devices.
- Two embedded serial interface ports with modem control.
- An embedded parallel interface port.
- An embedded port with control to a diskette drive.

- Latches, buffers, glue logic, power regulators, oscillators, decoupling capacitors, and associated components as needed to form a complete motherboard.
- Database files and user documentation that allow designers with no previous knowledge of the Alpha architecture to successfully create a working system using an Alpha microprocessor with minimal assistance.

1.2 Uses

The EB66+ has a wide range of uses. The following are a few examples:

- **System development**

  The EB66+ design can be used as the basis of a full computing system. Starting from the design database, you can customize the design to meet your requirements. For example, you could design or integrate off-the-shelf PCI and ISA supporting modules.

  The EB66+ itself can also be used as a platform for testing the design and software of new components. The EB66+ works under worst-case voltage, temperature, and process conditions, and can serve as the core of a high-volume product without making any significant design changes. The EB66+ is designed as a PC motherboard, and as such, forms a suitable basis for the design of personal computers and workstations.

  Its design is also an appropriate starting point for the design of embedded control products such as laser printers, communication engines (servers, bridges, and routers), and video products. The 21066A microprocessor runs at speeds of up to 233 MHz with onboard power regulation and cooling.

- **Software development**

  The EB66+ has a software debug monitor for loading code into the system and performing other software debugging functions, such as memory read, memory write, and instruction breakpoint. The debug monitor also allows users to debug remote source-level prototype software from a workstation running DECladebug. It can be used to debug new software modules on the EB66+. This is particularly useful when developing new drivers for hardware components, such as PCI or ISA modules. It is also designed to be ported to new hardware, making it easy to debug the software for user-designed systems.

- **DRAM memory evaluation**

  The EB66+ is a high-speed design and includes a full 64-bit data path for the DRAM main memory. The user can select DRAM access time individually by bank through memory controller registers.

1.2 Uses

- **Bcache subsystem evaluation**

  The EB66+ 64-bit data path extends to the onboard high-performance SRAM Bcache subsystem. The EB66+ includes a 1-MB Bcache SRAM, but it can emulate a 256-KB or 512-KB size by means of jumper placement (see Table 2–1).

  Two high-order address bits are manipulated to effect a size reduction. Such flexibility allows users to change cache size and speed characteristics and run performance benchmarks to determine the effect on actual programs. The available hardware configurations can also be combined and tested with different coding techniques to determine optimum system performance.

1.3 Features

The EB66+ is a fully populated baby AT board that acts as a PC motherboard. The major components of the EB66+ include:

- 21066A with supporting logic
- DRAM/cache memory subsystem
- PCI and ISA I/O bus support
- Software support

The major features of the EB66+ are described in Sections 1.3.1 through 1.3.7.

1.3.1 Microprocessor (Central Processing Unit)

The EB66+ includes the Alpha 21066A microprocessor, a CMOS/VLSI chip that implements the Alpha architecture and contains the following:

- Integral central processor
- Floating-point processor (IEEE and Digital formats)
- 8-KB primary instruction cache memory
- 8-KB primary data cache memory
- Secondary external backup cache control
- DRAM memory controller with fully programmable timing per bank
- Embedded graphics support functions
- PCI bus I/O controller
- Internal phase-locked loop (PLL) clock multiplier

- 287-pin PGA package with heat sink mounting studs

1.3.2 Memory Subsystem

The main memory subsystem accommodates 8 MB to 384 MB of DRAM, using up to six commodity, 72-pin, single inline memory module (SIMM) cards. Each SIMM card is 36 bits wide. The EB66+ supports three banks of memory. Each memory bank must be populated with two identical SIMM cards. Each of the three banks provides 8 MB to 128 MB and may be populated with identical or different DRAM SIMM card pairs. The following SIMM card sizes are supported on the EB66+:

1M × 36 — 70 ns or faster
2M × 36 — 70 ns or faster
4M × 36 — 70 ns or faster
8M × 36 — 70 ns or faster
16M × 36 — 70 ns or faster

The memory subsystem is a high-speed, 64-bit (plus 8-bit ECC) configuration. Timing for each DRAM bank is individually programmable to allow flexibility in memory design.

1.3.3 Bcache Subsystem

The microprocessor's integral Bcache control logic supports a board-level cache of up to 2 MB. The EB66+ implements a 1-MB board-level Bcache made up of 128K × 8 SRAMs.

1.3.4 PCI I/O Interface

The microprocessor provides an interface to the PCI local bus for onboard and external PCI devices. The EB66+ implements the following onboard PCI devices and functions:

- PCI-to-ISA bridge (SIO)—Intel 82378ZB

  The 82378ZB 208-pin PQFP VLSI chip provides a bridge between the EB66+'s PCI bus and the ISA bus. It offers PCI arbitration, a PCI master/slave interface running at a speed of 33 MHz, independent PCI/ISA addressing for concurrent operation, 8-bit or 16-bit DMA device support, and a speaker output.

- Expansion slots

  The PCI bus may be expanded by four individually addressable onboard connectors. Three connectors are dedicated, and one is a shared slot with the ISA bus.

1.3.5 ISA Interface

The 82378ZB PCI-to-ISA bridge implements standard IEEE-P996 bus functions. It manages ISA master/slave operations and DMA transfers, and supports the following ISA and utility bus devices and functions:

- A peripheral device interface—National PC87312

  The PC87312 100-pin PQFP VLSI chip incorporates an IDE device controller, a diskette drive controller, two universal asynchronous receiver–transmitters (UARTs) for serial device applications, and a bidirectional parallel port. All interfaces are available by onboard headers.

- Flash ROM

  A debug monitor in the flash ROM allows code to be developed in a host system and loaded into the EB66+. The debug monitor supports the source-level debugging of software that is running on the EB66+ from a remote workstation running DECladebug. Software can be loaded from a host system into the EB66+ through the combination chip communication port 1 (J32), the diskette, or the optional PCI Ethernet card.

- Mouse and keyboard control—Intel 8242 with Phoenix BIOS

  The 8242 VLSI custom microcontroller chip and associated connectors control the input from a standard keyboard and mouse. It contains an integral Phoenix keyboard BIOS.

- Time-of-year clock—Dallas Semiconductor DS1287

  The DS1287 chip with battery backup provides the time-of-year (TOY) function.

The bridge also drives five ISA bus connectors on the EB66+ board; four connectors are dedicated, and one is a shared slot with the PCI bus.

1.3.6 Serial ROM

The EB66+ uses an SROM for its initialization code. When the system power is turned on or reset, the contents of the SROM are read into the microprocessor's primary internal instruction cache. The code is then executed from the instruction cache. See Section 3.5 and Section 3.7 for more information on the SROM.

1.3.7 Programmable Array Logic Devices

Because of the combinatorial logic density, programmable array logic (PAL) devices are used for interrupt control logic. This reduces PC board real estate, manufacturing expense, and aggregate component expense.

Specific PAL devices used in the EB66+ for interrupt control are AMD MACH210-20 and AMD 22V10-25 chips.

2

System Configuration and Connections

To enhance its versatility, the EB66+ has jumpers to provide variations in timing and speed, software, and interrupt control. These jumpers must be configured for the user's environment. Onboard connectors and headers are provided, for the following:

- PCI/ISA I/O
- IDE devices
- Serial and parallel ports
- Diskette drive
- User interface
- DRAM memory SIMMs
- Power

The board is shipped with jumpers placed in default positions; it can be reconfigured for custom configurations. After the board is reconfigured and peripheral devices are connected, power can be applied, and the debug monitor can be run. The debug monitor and its commands are described in the *Alpha Microprocessors Evaluation Board Debug Monitor User's Guide*. For information about other software design tools, refer to the literature listed in Appendix C.

2.1 Configuration Jumpers

Figure 2–1 shows the system configuration jumpers and Table 2–1 describes them.

2.1 Configuration Jumpers

Figure 2–1 EB66+ Jumpers

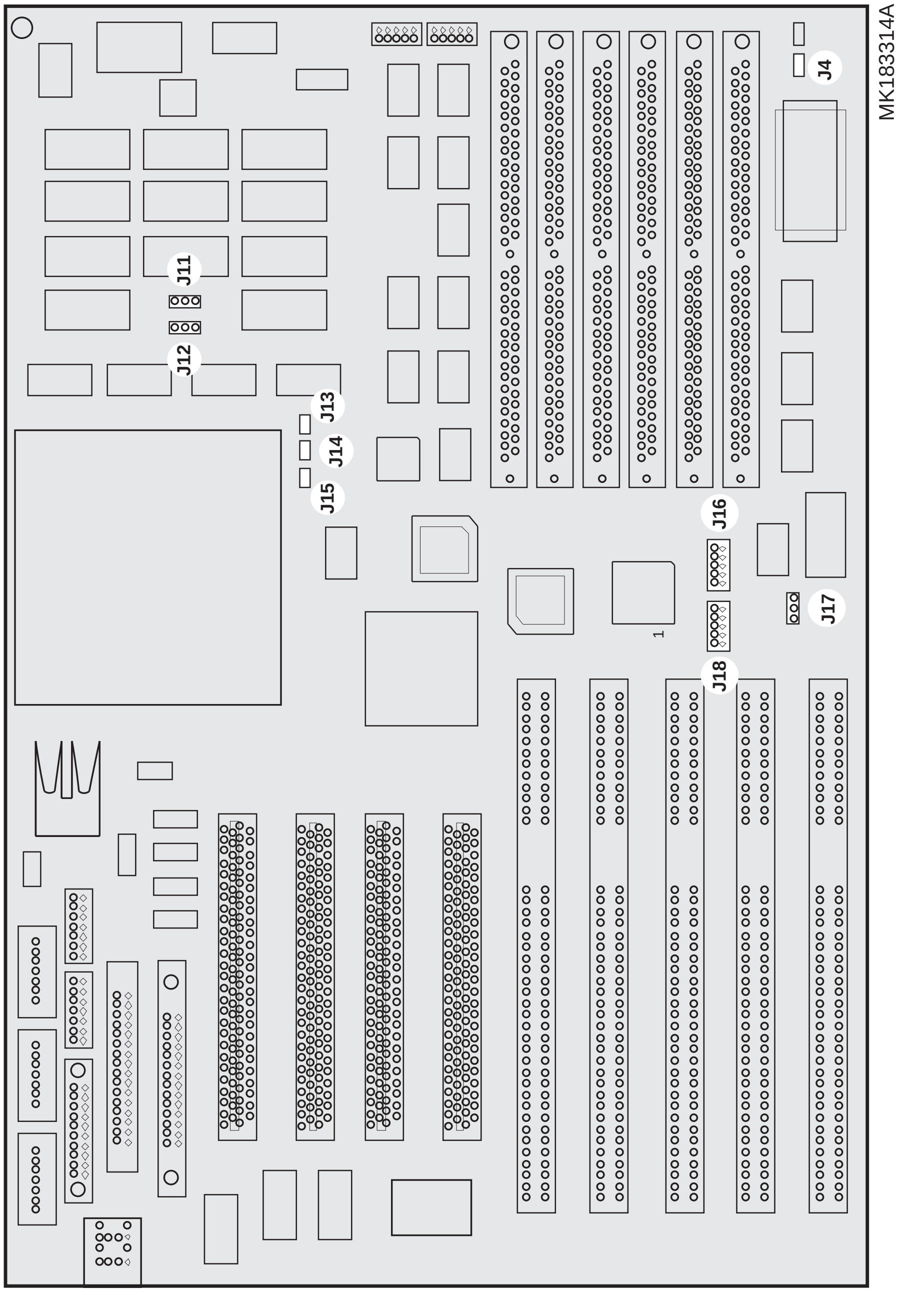
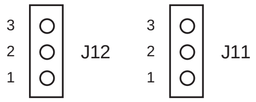


Table 2–1 EB66+ Jumpers Description

Jumper	Description
J4	Reserved
J11 J12	Bcache **idx_tag2** jumper (*eb66p.8*) Bcache **idx_tag3** jumper (*eb66p.8*)

Jumper J12 Pins	Jumper J11 Pins	Size
2 to 3	2 to 3	256-KB Bcache
2 to 3	1 to 2	512-KB Bcache
1 to 2	1 to 2	1-MB Bcache (default)
1 to 2	2 to 3	Illegal



Note: When effectively reducing the Bcache SRAM size from 1 MB to 256 KB, jumpers J16 (**sp_bit<1:0>**), J11, and J12 must be reconfigured.

Jumper	Description
J13 J14 J15	PLL clock multiplier (IRQ0) (*eb66p.5*) PLL clock multiplier (IRQ1) (*eb66p.5*) PLL clock multiplier (IRQ2) (*eb66p.5*)

J15	J14	J13		
IRQ2	**IRQ1**	**IRQ0**	**Multiplier**	**Frequency with 33-MHz clock**
In	In	In	Multiply × 2	66.6 MHz
In	In	Out	Multiply × 3	99.9 MHz
In	Out	In	Multiply × 4	133.2 MHz
In	Out	Out	Multiply × 5	166.5 MHz
Out	In	In	Multiply × 6	199.8 MHz
Out	In	Out	Multiply × 7	233.1 MHz (default)
Out	Out	In	Multiply × 8	Illegal
Out	Out	Out	Multiply × 9	Illegal

(continued on next page)

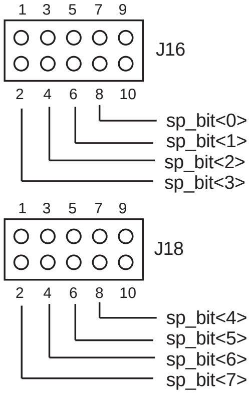
Table 2–1 (Cont.) EB66+ Jumpers Description

Jumper	Description
J16 and J18	System configuration jumpers **sp_bit<7:0>** (**ramdata<15:08>**) (*eb66p.18*)

sp_bit<#>/Name	Jumpers	Function
sp_bit<1:0>	In, in	Bcache disabled
Bcache size 1, 0	In, out	256 KB
	Out, in	512 KB
	Out, out	1 MB (default)
sp_bit<3:2>	In, in	6-ns RAM
SRAM speed	In, out	8-ns RAM
	Out, in	10-ns RAM (default)
	Out, out	12-ns RAM
sp_bit<4>	In	SROM init traps to Mini-Debugger
Mini-Debugger	Out	No trap—boot from flash ROM (default)
sp_bit<5>	In	Boot alternate image from system ROM
Boot option	Out	Boot first image (debug monitor) from system ROM (default)
sp_bit<6>	In	Read speed one cycle faster
Bcache read cycle	Out	Nominal read speed (default)
sp_bit<7>	In	Disables ECC in memory and Bcache (if enabled)
	Out	Enable ECC (default)

(continued on next page)

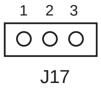
Table 2–1 (Cont.) EB66+ Jumpers Description

<table>
<tr><th>Jumper</th><th>Description</th></tr>
<tr><td></td><td>


Note: There are no connections to pins 9 and 10 on J16 and J18.

Note: When effectively reducing the Bcache SRAM size from 1 MB to 512 KB or 256 KB, jumpers J16 (**sp_bit<1:0>**), J11, and J12 must be reconfigured.
</td></tr>
<tr><td>J17</td><td>
Flash ROM write-enable (*eb66p.34*)

<table>
<tr><th>Jumper Pins</th><th>Function</th></tr>
<tr><td>1 to 2</td><td>Flash protected (not writable)</td></tr>
<tr><td>2 to 3</td><td>Write-enable (flash writable by system software) (default)</td></tr>
</table>



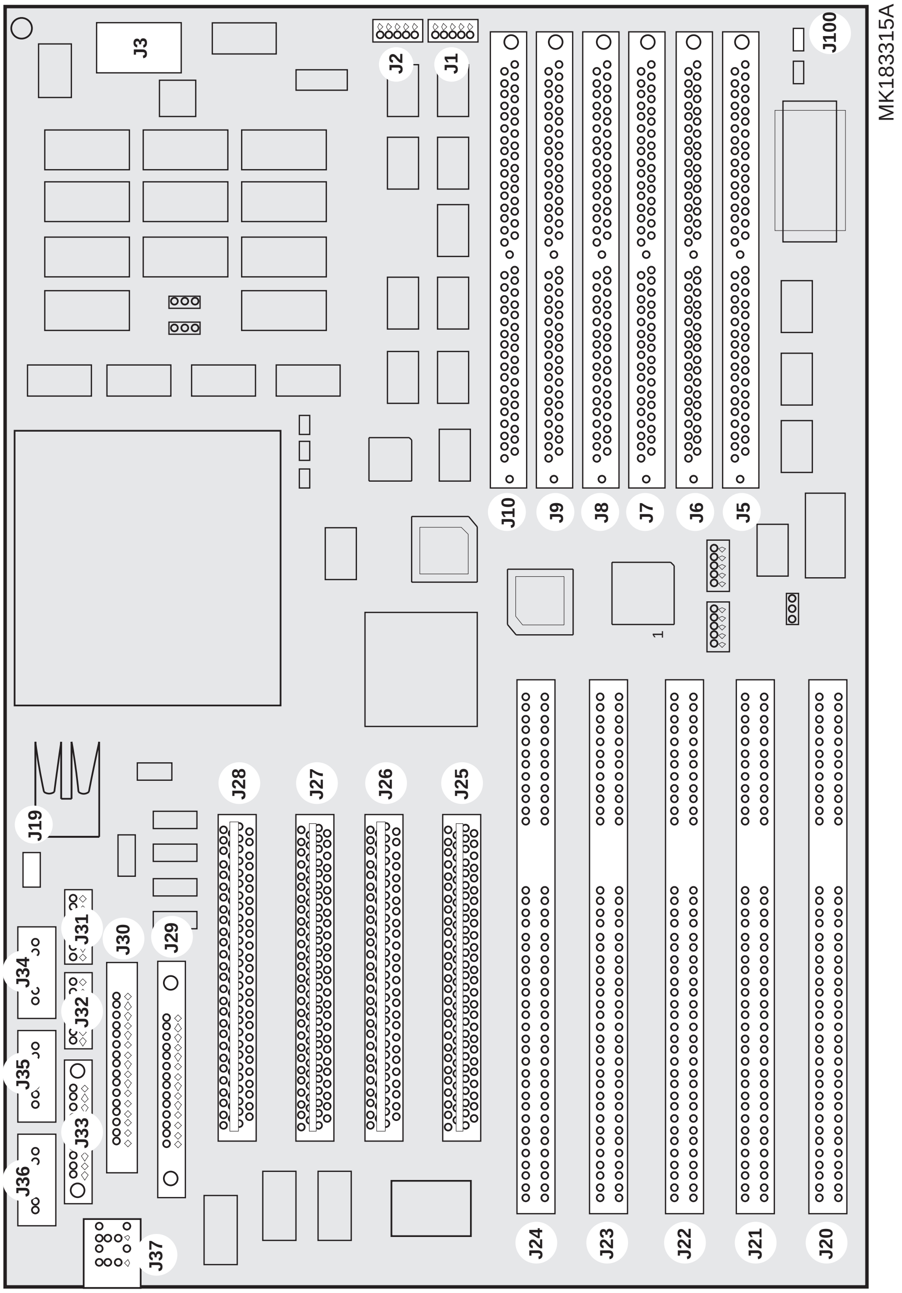
</td></tr>
</table>

2.2 Connectors and Headers

Figure 2–2 shows the connectors and headers for user-supplied power, peripheral devices, and memory. Table 2–2 describes the EB66+ connectors and headers.

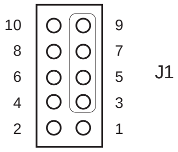
2.2 Connectors and Headers

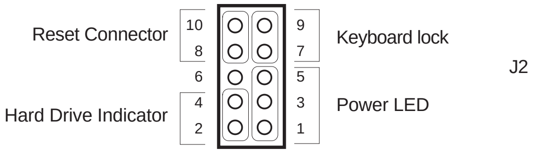
Figure 2–2 EB66+ Connectors and Headers



2.2 Connectors and Headers

Table 2–2 EB66+ Connectors and Headers Description

<table>
<tr><th>Connector</th><th>Pins</th><th>Description</th></tr>
<tr><td>J1</td><td>10</td><td>Speaker connector (<i>eb66p.6</i>)
<table>
<tr><th>Pin</th><th>Voltage/Signal</th></tr>
<tr><td>1</td><td>NC</td></tr>
<tr><td>2,5,9</td><td>Vdd (+5 V)</td></tr>
<tr><td>3</td><td>Speaker</td></tr>
<tr><td>4,6,7,8,10</td><td>Ground</td></tr>
</table>


<b>Note:</b> Place a standard speaker connection as shown.</td></tr>
<tr><td>J2</td><td>10</td><td>Miscellaneous connector (<i>eb66p.6</i>)
<table>
<tr><th>Pin</th><th>Voltage/Signal</th></tr>
<tr><td>1</td><td><b>power_led_l</b></td></tr>
<tr><td>2</td><td><b>hd_led_l</b></td></tr>
<tr><td>3,5,6,9,10</td><td>Ground</td></tr>
<tr><td>4</td><td><b>hd_act_l</b></td></tr>
<tr><td>7</td><td><b>key_lock</b></td></tr>
<tr><td>8</td><td><b>reset_button</b></td></tr>
</table>

</td></tr>
<tr><td>J3</td><td>6</td><td>SROM test data and Mini-Debugger serial port input header (<i>eb66p.5</i>)</td></tr>
</table>

(continued on next page)

2.2 Connectors and Headers

Table 2–2 (Cont.) EB66+ Connectors and Headers Description

Connector	Pins	Description
J9	72	Bank 0, DRAM 0 SIMM connector (*eb66p.19*)
J10	72	Bank 0, DRAM 1 SIMM connector (*eb66p.19*)
J7	72	Bank 1, DRAM 0 SIMM connector (*eb66p.20*)
J8	72	Bank 1, DRAM 1 SIMM connector (*eb66p.20*)
J5	72	Bank 2, DRAM 0 SIMM connector (*eb66p.21*)
J6	72	Bank 2, DRAM 1 SIMM connector (*eb66p.21*)
		Note: To fill the 64-bit and 8-bit ECC data path, SIMM connectors must be populated *in pairs*. Both J9 and J10 must be used together for Bank 0. Likewise, both J7 and J8 must be used together for Bank 1, and J5 and J6 must be used together for Bank 3. Figure 2–3 shows the DRAM connector for each configuration.
J19	3	CPU fan power and sensor connector (*eb66p.36*)
		Caution: The CPU cooling fan *must* have a built-in fan sensor that drives a signal if the airflow stops. The sensor is connected to J19. The fan supplied with the EB66+ includes an airflow sensor. (Refer to Section 3.6 for more information.)
J20	98	ISA bus 0 connector (*eb66p.26*)
J21	98	ISA bus 1 connector (*eb66p.26*)
J22	98	ISA bus 2 connector (*eb66p.26*)
J23	98	ISA bus 3 connector (*eb66p.27*)
J24	98	ISA bus 4 connector (*eb66p.27*)
J25	120	PCI bus slot 0 connector (*eb66p.22*)
J26	120	PCI bus slot 1 connector (*eb66p.22*)
J27	120	PCI bus slot 2 connector (*eb66p.23*)
J28	120	PCI bus slot 3 connector (*eb66p.23*)
J29	40	Combination chip (PC87312) IDE device header (*eb66p.25*)
J30	34	Combination chip (PC87312) diskette drive header (*eb66p.31*)
J31	10	Combination chip (PC87312) communication port 2 (serial) header (*eb66p.31*)

(continued on next page)

Table 2–2 (Cont.) EB66+ Connectors and Headers Description

<table>
<tr><th>Connector</th><th>Pins</th><th>Description</th></tr>
<tr><td>J32</td><td>10</td><td>Combination chip (PC87312) communication port 1 (serial) header (<i>eb66p.31</i>).
This header is used for the debug monitor terminal interface.</td></tr>
<tr><td>J33</td><td>26</td><td>Combination chip (PC87312) parallel port header (<i>eb66p.30</i>)</td></tr>
<tr><td>J34</td><td>6</td><td>Board power connector (<i>eb66p.4</i>) (optional)
<b>Note:</b> The EB66+ has an onboard 5-V-to-3.3-V regulator. This power connector is used if the regulator is not used.
<table>
<tr><th>Pin</th><th>Voltage/Signal</th></tr>
<tr><td>1</td><td>+3.3 V</td></tr>
<tr><td>2</td><td>+3.3 V</td></tr>
<tr><td>3</td><td>+3.3 V</td></tr>
<tr><td>4</td><td>Ground</td></tr>
<tr><td>5</td><td>Ground</td></tr>
<tr><td>6</td><td>Ground</td></tr>
</table></td></tr>
<tr><td>J35</td><td>6</td><td>Board power connector (<i>eb66p.36</i>)
<table>
<tr><th>Pin</th><th>Voltage/Signal</th></tr>
<tr><td>1</td><td>Ground</td></tr>
<tr><td>2</td><td>Ground</td></tr>
<tr><td>3</td><td>–5 V</td></tr>
<tr><td>4</td><td>Vdd (+5 V)</td></tr>
<tr><td>5</td><td>Vdd (+5 V)</td></tr>
<tr><td>6</td><td>Vdd (+5 V)</td></tr>
</table></td></tr>
</table>

(continued on next page)

Table 2–2 (Cont.) EB66+ Connectors and Headers Description

<table>
<tr><th>Connector</th><th>Pins</th><th>Description</th></tr>
<tr><td>J36</td><td>6</td><td>Board power connector (<i>eb66p.36</i>)
<table>
<tr><th>Pin</th><th>Voltage/Signal</th></tr>
<tr><td>1</td><td><b>p_dcok</b></td></tr>
<tr><td>2</td><td>Vdd (+5 V)</td></tr>
<tr><td>3</td><td>+12 V</td></tr>
<tr><td>4</td><td>–12 V</td></tr>
<tr><td>5</td><td>Ground</td></tr>
<tr><td>6</td><td>Ground</td></tr>
</table>
<b>Note:</b> Power for the EB66+ is provided by a user-supplied, standard, PC power supply. Digital does not provide this power supply. (Refer to Section 3.6 and Chapter 4 for more information.)</td></tr>
<tr><td>J37</td><td>6</td><td>Keyboard and mouse connector (<i>eb66p.32</i>)
<b>Note:</b> The mouse and keyboard are connected with the included Y cable.</td></tr>
<tr><td>J100</td><td>2</td><td>Optional external fan connector (<i>eb66p.36</i>)</td></tr>
</table>

2.3 DRAM Configuration

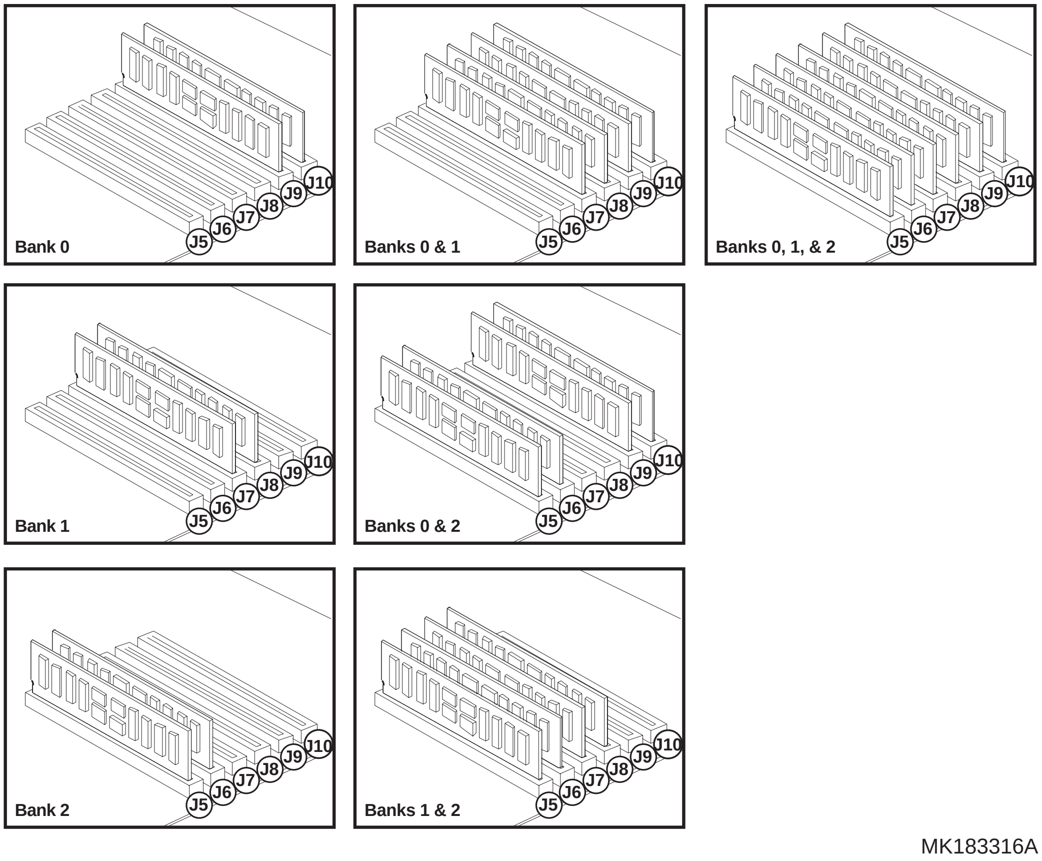
The main memory subsystem can accommodate from 8 MB to 384 MB of DRAM, using either two, four, or six SIMM cards. Each SIMM card is 36 bits wide.

The EB66+ supports three banks of memory, with each memory bank populated with two SIMM cards of *identical size and speed*. Each of the three banks can provide from 8 MB to 128 MB, and the banks can be populated with identical or different DRAM SIMM card pairs.

Figure 2–3 shows the DRAM SIMM memory bank configuration.

2.3 DRAM Configuration

Figure 2–3 DRAM SIMM Memory Bank Configuration



3

Functional Description

This chapter describes the functional design and operation of the EB66+, including the specific implementation of the microprocessor with its supporting memory and I/O devices. Bus timing and protocol information is not included in this document but can be found in other documentation. Appendix C provides a list of supporting documents and their order numbers.

3.1 Address Space and Mapping

This section describes the mapping of the 34-bit processor physical address space into memory and I/O space addresses, the translation of the processor-initiated address into a PCI address, and the translation of PCI-initiated addresses into physical memory addresses.

Table 3–1 shows the division of the 34-bit address space.

Table 3–1 EB66+ System Address Map

Start Address <33:0>	End Address <33:0>	Address Region	Size
0 0000 0000	0 1FFF FFFF	Cacheable memory	0.5 GB
0 2000 0000	0 3FFF FFFF	Noncacheable memory	0.5 GB
0 4000 0000	0 FFFF FFFF	Nonexistent memory	3.0 GB
1 0000 0000	1 1FFF FFFF	Graphics memory	0.5 GB
1 2000 0000	1 7FFF FFFF	Memory controller CSRs	1.5 GB
1 8000 0000	1 9FFF FFFF	IOC CSRs	0.5 GB
1 A000 0000	1 BFFF FFFF	PCI interrupt acknowledge/special cycle	0.5 GB
1 C000 0000	1 DFFF FFFF	PCI I/O	0.5 GB
1 E000 0000	1 FFFF FFFF	PCI configuration	0.5 GB
2 0000 0000	2 FFFF FFFF	PCI sparse memory	4.0 GB
3 0000 0000	3 FFFF FFFF	PCI dense memory	4.0 GB

3.1 Address Space and Mapping

The microprocessor's I/O controller (IOC) functions as an interface bridge between the I/O peripheral devices, CPU, and system memory. All EB66+ I/O peripheral devices are connected to the microprocessor and system memory through the IOC. The IOC interface protocol is compatible with the *PCI Local Bus Specification, Revision 2.0*. EB66+ peripherals that interface to PCI can be connected to the microprocessor without any additional glue logic. However, external logic is needed on the board for interrupt generation.

The microprocessor is not a PCI peripheral. The microprocessor's IOC implements the functions of a bridge between the PCI, CPU, and system memory.

3.1.1 CPU-Initiated PCI Cycles

The IOC operates as a PCI master when the CPU executes a load or store instruction that addresses a PCI peripheral. The IOC provides address windows to the CPU that allow access to the memory, I/O, and configuration address spaces of the PCI. The IOC provides address space that, when read, generates a PCI interrupt acknowledge cycle or, when written, generates a PCI special cycle.

To support various transfer size (byte enable) combinations, the addresses used by the CPU to reference PCI address spaces are encoded. The encoding uses bits <4:3> of the physical address as size bits. Bits <31:7> of the physical address are used for PCI address bits <26:2>. Physical address bits <2:0> must be zero for all CPU-to-PCI-initiated transfers.

Address Extension for Sparse Space

Due to the encoding of the CPU address used to generate the PCI byte enables, only 27 physical address bits, CPU address bits <31:5>, are used to generate PCI address bits **pci_ad<26:0>**. This results in an effective PCI address space of 128 MB. This 128-MB address space is subdivided into a 16-MB region and a 112-MB region. PCI address bits **pci_ad<31:27>** are generated differently depending on which of these regions are referenced (see Table 3–2).

Table 3–2 lists the host address extensions.

Table 3–2 Host Address Extension

CPU A<33:32>	CPU A<31:29>	PCI AD<31:27>
10	000	00000
10	001	HAE<31:27>
10	010	HAE<31:27>
10	011	HAE<31:27>
10	100	HAE<31:27>
10	101	HAE<31:27>
10	110	HAE<31:27>
10	111	HAE<31:27>

The 16-MB region is always mapped to the first 16 MB of the PCI address space. This 16-MB region is referenced whenever CPU address bits <31:29> are zero. During CPU-initiated PCI transactions to this 16-MB region, PCI address bits **pci_ad<31:27>** are zero.

The 112-MB region is referenced whenever CPU address bits <31:29> are non-zero. During CPU-initiated PCI transactions to this 112-MB region, PCI address bits **pci_ad<31:27>** are generated using the host address extension (HAE) bit field in the IOC_CTRL register.

Table 3–3 shows the complete address translation for different cycle types.

Table 3–3 CPU-to-PCI Address Translation

CPU R/W	CPU A <33:32>	CPU A <31:29>	PCI AD <31:27>	PCI AD <26:24>	PCI AD <23:02>	PCI AD <01:00>	PCI CMD <3:0>	Transaction
Write	10	Non-zero	HAE<31:27>	A<31:29>	A<28:07>	00	0111	Sparse memory write
Write	10	000	00000	A<31:29>	A<28:07>	00	0111	Sparse memory write
Read	10	Non-zero	HAE<31:27>	A<31:29>	A<28:07>	00	0110	Sparse memory read
Read	10	000	00000	A<31:29>	A<28:07>	00	0110	Sparse memory read

(continued on next page)

Table 3–3 (Cont.) CPU-to-PCI Address Translation

CPU R/W	CPU A <33:32>	CPU A <31:29>	PCI AD <31:27>	PCI AD <26:24>	PCI AD <23:02>	PCI AD <01:00>	PCI CMD <3:0>	Transaction
Write	11	NA*	A<31:27>	A<26:24>	A<23:02>	00	0111	Dense memory write
Read	11	NA*	A<31:27>	A<26:24>	A<23:02>	00	0110	Dense memory read
Write	01	110	00000	000	A<28:07>	A<06:05>	0011	I/O write
Read	01	110	00000	000	A<28:07>	A<06:05>	0010	I/O read
Write	01	111	00000	000	A<28:07>	cfg<01:00>†	1011	Configuration write
Read	01	111	00000	000	A<28:07>	cfg<01:00>†	1010	Configuration read
Write	01	101	HAE<31:27>	A<31:29>	A<28:07>	00	0001	Special cycle
Read	01	101	HAE<31:27>	A<31:29>	A<28:07>	00	0000	Interrupt acknowledge

*NA = not applicable.

†cfg is the IOC configuration cycle type register <1:0>.

Table 3–4 shows the byte-enable encoding for CPU-initiated transfers in sparse space.

Table 3–4 Byte-Enable Encoding for CPU-Initiated Transfers in Sparse Space

CPU A<06:05>	CPU A <04:03>	PCI Byte Enables C_BE_L<03:00>	PCI AD<01:00> Memory Cycles	PCI AD<01:00> I/O Cycles	Transfer Type	Size
00	00	1110	00	00	Masked	Byte
01	00	1101	00	01	Masked	Byte
10	00	1011	00	10	Masked	Byte
11	00	0111	00	11	Masked	Byte
00	01	1100	00	00	Masked	Word
01	01	1001	00	01	Masked	Word
10	01	0011	00	10	Masked	Word
00	10	1000	00	00	Masked	Tribyte
01	10	0001	00	01	Masked	Tribyte
00	11	0000	00	00	Unmasked	Longword
11	11	0000	00	00	Unmasked	Quadword

3.1.2 Address Translation of Peripheral-Initiated PCI Cycles

Because the PCI uses a 32-bit address and the microprocessor uses a 34-bit address, PCI addresses to which the IOC responds must be translated to an equivalent address in the CPU's address space. The IOC provides two programmable address windows that control access by PCI peripherals to system memory. These address windows are referred to as PCI target windows. A set of three registers is associated with each PCI target window. When an address match occurs with a PCI target window, the CPU translates the 32-bit PCI address to a 34-bit CPU address. The translated address is generated in one of two ways as determined by the scatter-gather (SG) bit of the PCI target window's CSR:

- If the SG bit = 0, direct-mapped address translation is used.
- If the SG bit = 1, scatter-gather mapped address translation is used.

3.1.3 Physical Address Space

The microprocessor uses a 34-bit address space as shown in Table 3–1. The two most significant bits of the address <33:32> determine the quadrant of the address space. Table 3–5 shows the address space quadrant.

Table 3–5 Address Space Quadrant

Address <33:32>	Region
00	Memory (cacheable and noncacheable)
01	Graphics memory, memory controller and IOC CSRs, PCI interrupt acknowledge/special cycle, PCI I/O, and PCI configuration space
10	PCI sparse memory space
11	PCI dense memory space (includes PCI operational registers)

The memory region is further divided into cacheable and noncacheable regions. The graphics memory region, memory controller CSR region, IOC CSR region, and the PCI regions are all noncacheable. Furthermore, PCI peripherals can access only the memory region because the IOC maps the 32-bit PCI address to the internal address that can access only the memory space.

For additional information on I/O controller (IOC) memory mapping and address translation (including generating addresses and address extensions for sparse and dense space), refer to the *Alpha 21066, 21066A, and 21068 Microprocessors Hardware Reference Manual*.

3.1.4 Sparse and Dense Memory Space

The microprocessor's IOC implements both sparse and dense memory space. Sparse memory space is a region in which the lower bits of the CPU address indicate transfer size. The address is shifted down by 5 bits in the translation from CPU address to PCI address. The host address extension (HAE) register extends the shifted address to cover 32 bits. Different-sized transfers, such as, byte, word, tribyte, longword, and quadword are supported in this space. The CPU references sparse address space by generating addresses with CPU **a<33:32>**=10.

Dense memory space is a region in which unmasked longword and quadword transfers are supported. The CPU address is transparently mapped into the PCI address. This allows more efficient use of the CPU write buffers but reduces the capability for different-sized transfers. The write buffers condense transfers that span multiple quadwords into a single burst. The CPU references dense address space by generating addresses with CPU

a<33:32>=11. Because the dense address is not shifted, the HAE is neither needed nor used in dense space.

3.1.5 PCI Configuration Address Space

The PCI configuration address space comprises 0.5 GB and covers the address range of 1 E000 0000 through 1 FFFF FFFF. The PCI configuration register set occupies this space. The following table identifies the EB66+ PCI devices and the corresponding PCI address bit that drives the device **idsel** pin:

PCI Device	PCI Address Bit Driving idsel Pin	Physical Address
PCI expansion slot 0	**pci_ad<17>**	1 E040 0000
PCI expansion slot 1	**pci_ad<18>**	1 E080 0000
System I/O (SIO)	**pci_ad<19>**	1 E100 0000
PCI expansion slot 2	**pci_ad<20>**	1 E200 0000
PCI expansion slot 3	**pci_ad<21>**	1 E400 0000

3.2 Memory Subsystem (DRAM, Bcache)

The microprocessor's memory controller controls EB66+ DRAM and Bcache memory. It can directly control up to 384 MB of memory organized into three banks of DRAM, with each bank containing two memory DRAM SIMMs. The memory data path is 64 bits (8 bytes) wide.

EB66+ memory DRAM SIMMs have a 36-bit-wide configuration. The supported SIMM sizes are as follows:

- 1M × 36 DRAM SIMM (70 ns or faster)
- 2M × 36 DRAM SIMM (70 ns or faster)
- 4M × 36 DRAM SIMM (70 ns or faster)
- 8M × 36 DRAM SIMM (70 ns or faster)
- 16M × 36 DRAM SIMM (70 ns or faster)

With the three banks of SIMMs, the EB66+ supports a minimum memory size of 8 MB and a maximum memory size of 384 MB.

The memory subsystem is a high-speed, 64-bit, and 8-bit ECC configuration, supporting three banks of memory. It can accommodate from 8 MB to 384 MB of DRAM, using either two, four, or six 36-bit-wide SIMM cards. Each memory bank must be populated with two SIMM cards of identical size and speed. DRAM SIMM speed must be a minimum of 70 ns. Each of the three banks provides from 8 MB to 128 MB and may be populated with identical or different DRAM SIMM card pairs. SROM routines configure DRAM size and speed at power-up.

All memory interface signals are provided by the microprocessor. System designers need only determine if additional external buffering of these signals is needed.

The memory controller can perform read, write, longword write, byte write (with additional logic), and write-per-bit operations on the DRAMs. The memory controller can also read and write the Bcache. It optimizes fast page-mode access, controls the memory refresh for the DRAMs, and includes support for embedded graphics control.

Refer to the *Alpha 21066, 21066A, and 21068 Microprocessors Hardware Reference Manual* for memory controller (DRAM, Bcache, and graphics) signal descriptions.

3.2.1 Memory/Bcache and Data Path Logic

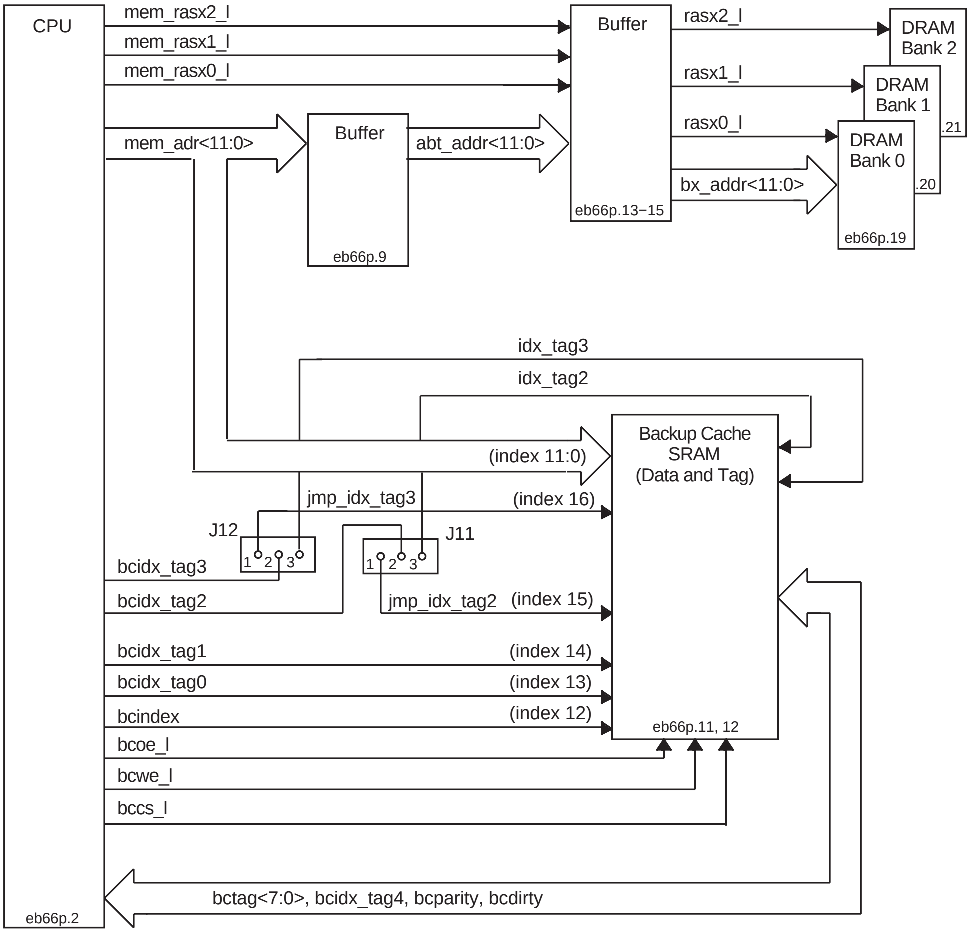
Figure 3–1 shows a simplified view of the memory/Bcache address logic as found in the EB66+ schematics. Figure 3–2 shows a simplified view of the memory/Bcache data path logic.

CPU address bits **mem_addr<11:0>** are driven directly to the Bcache and to buffers. The buffered address bits **abt_addr<11:0>** are then driven to DRAM as **bx_addr<11:0>**. Microprocessor signals drive higher-order Bcache address bits (<16:12>) as shown in Figure 3–1. This figure shows the SRAM address index values in parentheses.

The memory controller multiplexes the DRAM SIMM addresses to present the row and column addresses at the proper time. Row address bits are strobed by **rasx** while column address bits are strobed by **cas**. The address multiplex definitions do not change with SIMM size; SIMMs that do not have a particular input address bit ignore the upper address lines.

The EB66+ supports split-bank addressing; **mem_rasax_l** drives bank set 0, and **mem_rasbx_l** drives bank set 1. SIMM row and column sizes are programmed through the memory controller BCRx and BMRx registers.

Figure 3–1 Memory/Bcache Addressing



MK183317A

Table 3–6 lists the microprocessor **mem_rasx_l** signals, the resulting buffered **rasx** signals, and their destinations. The **ras** lines 0, 1, and 2 strobe the DRAM memory. The **ras** line 3 strobes the system configuration jumper array J16 and J18 on read transactions only.

3.2 Memory Subsystem (DRAM, Bcache)

Figure 3–2 Memory/Bcache Data Paths

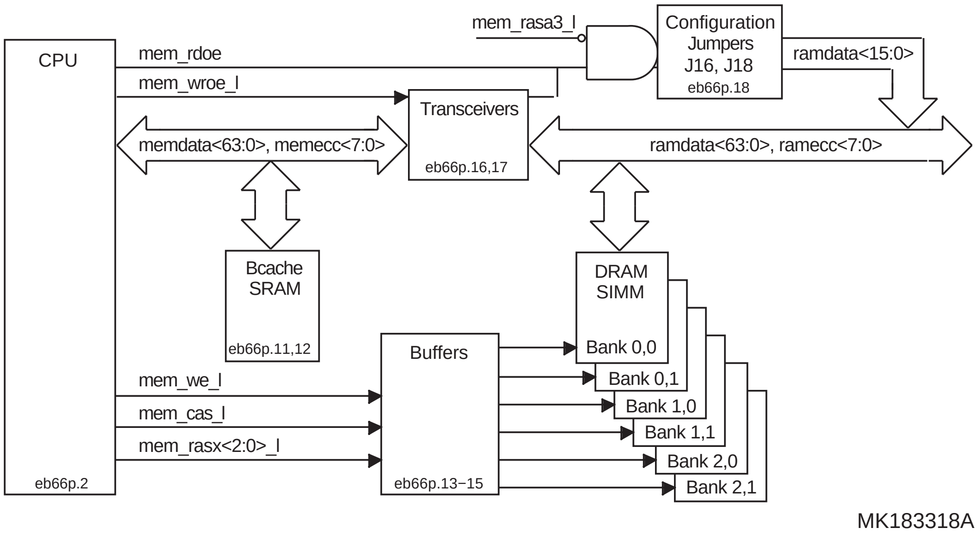


Table 3–6 Memory Controller RAS Signal Distribution

CPU Memory Controller Signal Name	Buffered Signal Name	Destination
mem_rasa0_l	**rasa0_l**	Memory bank 0, bank set 0 (DRAM ras0 and ras2)
mem_rasb0_l	**rasb0_l**	Memory bank 0, bank set 1 (DRAM ras1 and ras3)
mem_rasa1_l	**rasa1_l**	Memory bank 1, bank set 0 (DRAM ras0 and ras2)
mem_rasb1_l	**rasb1_l**	Memory bank 1, bank set 1 (DRAM ras1 and ras3)
mem_rasa2_l	**rasa2_l**	Memory bank 2, bank set 0 (DRAM ras0 and ras2)
mem_rasb2_l	**rasb2_l**	Memory bank 2, bank set 1 (DRAM ras1 and ras3)
mem_rasa3_l	—	Configuration jumpers, presence detect bits

The DRAM memory subsystem has a 64-bit (and 8-bit ECC) configuration. The EB66+ supports several DRAM SIMM memory sizes, from 8 MB to 384 MB, in six industry-standard DRAM SIMM connectors (J5, J6, J7, J8, J9, and J10), as shown in Figure 2–2 and described in Table 2–2. To fill the 64-bit and 8-bit ECC data path width, two (one bank), four (two banks), or six (three banks) SIMM connectors must be populated.

3.2.2 Backup Cache (Bcache)

The standard EB66+ Bcache is 1 MB of SRAM. The Bcache is direct mapped, write-back, with a quadword block size. It contains both data ECC and tag parity protection.

The EB66+ supports a Bcache size of 1 MB, and can emulate sizes of 512 KB, and 256 KB by means of jumper placement (see Table 2–1 and Figure 2–1). Users may change the effective Bcache memory size and speed characteristics and run performance benchmarks to determine the affect on actual programs.

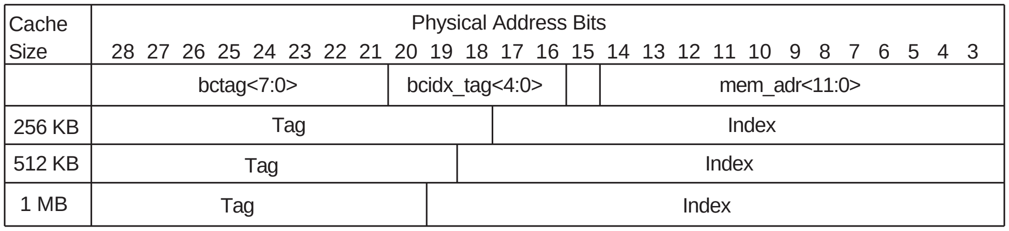
When effectively reducing the Bcache SRAM size from 1 MB to either 512 KB or 256 KB, jumpers J11, J12, and J16 must be reconfigured. The two high-order Bcache SRAM address bits are manipulated to effect the size reduction. In the EB66+ implementation, both microprocessor signals **bcidx_tag<0>** and **bcidx_tag<1>** are index bits, and **bcidx_tag<4>** is a tag bit. Signals

bcidx_tag<3:2> act as either index bits or tag bits depending on the desired Bcache size as configured in jumpers J11 and J12. The function of these bits is shown in Table 3–7. Figure 3–3 shows the available EB66+ Bcache sizes and the address partitioning.

Table 3–7 Using bcidx_tag<4:0> Bits

Bcache	bcidx_tag<4:0>				
Size	<4>	<3>	<2>	<1>	<0>
256 KB	Tag 8	Tag 9	Tag 10	Index 14	Index 13
512 KB	Tag 8	Tag 9	Index 15	Index 14	Index 13
1 MB	Tag 8	Index 16	Index 15	Index 14	Index 13

Figure 3–3 EB66+ Bcache Address Partitioning



Notes:
Address bits <2:0> are not used because cache reference is quadword aligned.
Address bit <15> is driven on signal bcindex.
Address bits <19:18> are used as index or tag depending upon the configuration of jumpers J9 and J10.

MK183327A

3.2.2.1 Implementation

The microprocessor's integral Bcache control logic supports a 1-MB Bcache implemented in SRAM (128K × 8). The tag SRAMs are also 128K × 8.

3.2.2.2 Latency

The memory controller attempts to minimize overall memory latency in the way that it "uses" the Bcache.

- For a read hit, the cache data will be used and the DRAM will not be read.
- For a read miss, the DRAM will be read and a cache block will be allocated.

On write transactions (hit or miss), if the Bcache is enabled and the address is in cacheable space (address bits <33:32>=0), the cache and the cache tag are written.

If the Bcache is not enabled or the address is not in cacheable space, the memory controller accesses the DRAM immediately on read transactions (omitting the cache access) and writes the DRAM only, not the Bcache, on write transactions.

Refer to the *Alpha 21066, 21066A, and 21068 Microprocessors Hardware Reference Manual* for specific information on cache operation during masked write transactions and page mode operation.

3.3 I/O Subsystem

The microprocessor's input/output controller (IOC) provides a direct interface to the peripheral component interconnect (PCI) bus. This section describes the EB66+ onboard PCI devices and system support functions. Figure 1–1 shows the EB66+ PCI devices.

Refer to the *PCI Local Bus Specification, Revision 2.0* and *PCI System Design Guide* for details on PCI protocol, ac and dc considerations, and timing. Refer to the *Alpha 21066, 21066A, and 21068 Microprocessors Hardware Reference Manual* for specific PCI implementation.

3.3.1 PCI Devices

The EB66+ uses the PCI bus as the main I/O bus for the majority of peripheral functions. EB66+ implements ISA as an expansion bus for other devices and for system support functions.

3.3.1.1 Intel System I/O (SIO) Chip

The Intel 82378ZB System I/O chip (SIO) provides the bridge between the PCI bus and the ISA bus. It is packaged in a 208-pin PQFP. The SIO incorporates the logic for:

- PCI interface (master and slave)
- ISA interface (master and slave)
- Enhanced 7-channel DMA controller that supports fast DMA transfers and scatter-gather data buffers to isolate the PCI bus from the ISA bus
- PCI and ISA arbitration
- A 14-level interrupt controller
- A 16-bit BIOS timer
- Three programmable timer/counters
- Nonmaskable interrupt (NMI) control logic

- Decoding and control for utility bus peripheral devices
- Speaker driver

Refer to the Intel *82420/82430 PCIset ISA and EISA Bridges* document for additional information on SIO pin assignments and signal descriptions, register descriptions, and a detailed functional description including timing, electrical characteristics, and mechanical data.

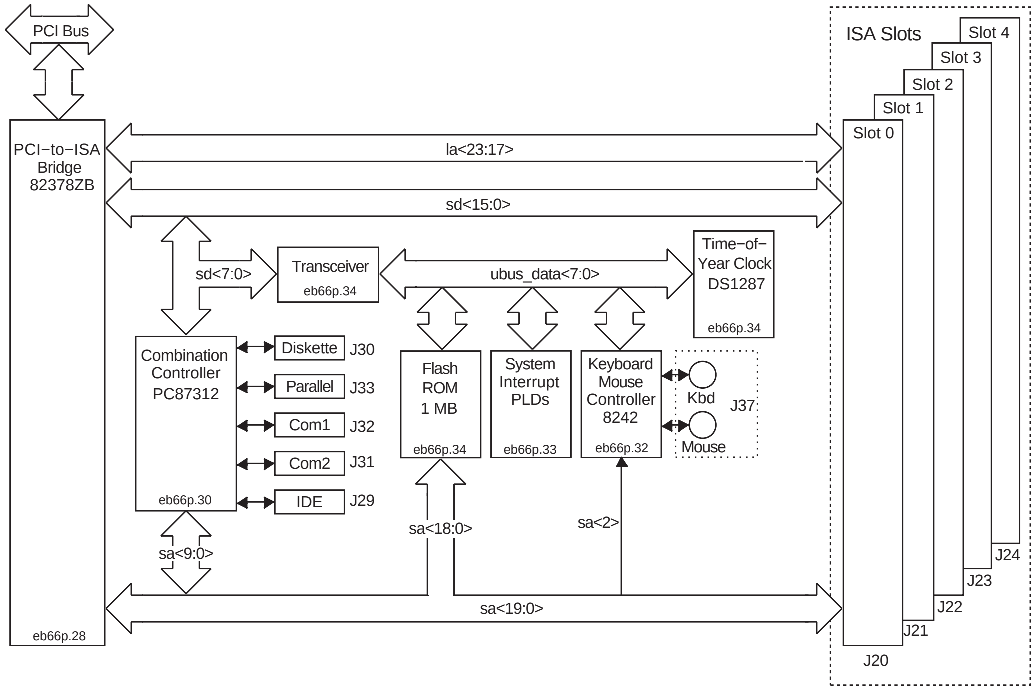
3.3.1.2 PCI Expansion Slots

Four PCI bus expansion slots are available on the EB66+; one is shared with the ISA. They use the standard 5-V PCI connector and pinout for 32-bit implementation. This allows the system designer or user to add additional PCI options.

3.3.2 ISA Devices

Figure 3–4 shows the EB66+ ISA bus implementation with peripheral devices and expansion connectors. It also shows the utility bus with system support devices.

Figure 3–4 ISA Bus Devices



MK183319B

3.3.2.1 Combination Controller

The EB66+ uses the National Semiconductor PC87312 as the combination controller chip. It is packaged in a 100-pin PQFP configuration. This chip provides the following ISA peripheral features:

- Diskette controller — The software is compatible with the Intel PC8477 chip, which contains a superset of the Intel DP8473, NEC μPD765, and Intel N82077 flexible diskette controller (FDC) functions. The onchip analog data separator requires no external filter components and supports the 4-MB drive format as well as other standard diskette drives used with 5.25-in and 3.5-in media. FDC data and control lines are brought out to a standard 34-pin header connector. A ribbon cable connects the header to one or two diskette drives.
- Two serial ports — Two full-function UARTs with modem control, compatible with the NS16450 or PC16550, are brought out to separate onboard 10-pin header connectors, which can then be brought out through 9-pin male D-sub connectors on the bulkhead of a standard PC enclosure.

- Parallel port — The bidirectional (software-controlled) parallel port is brought out to an onboard 26-pin header connector and can be brought out through a 25-pin female D-sub connector on the bulkhead of a standard PC enclosure.
- IDE interface control — The IDE control logic provides a complete IDE interface, including external signal buffers.

A 24-MHz clock supplies a reference for the diskette data separator and serial ports (Section 3.4).

Refer to the National Semiconductor *PC87311/PC87312 SuperI/O II/III Floppy Disk Controller with Dual UARTs, Parallel Port, and IDE Interface* document for additional information on pin assignments and signal descriptions, register descriptions, and a detailed functional description including timing, electrical characteristics, and mechanical data.

3.3.2.2 Keyboard and Mouse Controller

The Intel 8242 device, located on the EB66+ utility bus, provides the keyboard and mouse controller function. It is packaged in a 44-pin PLCC configuration.

The 8242 is an Intel UPI-42AH universal peripheral interface. Architecturally, it is an 8-bit slave microcontroller with 2 KB of ROM and 256 bytes of RAM that has been preprogrammed with a Phoenix keyboard BIOS for standard scan codes.

Refer to the Intel *UPI-41AH/42AH Universal Peripheral Interface 8-Bit Slave Microcontroller* document for additional information on pin assignments and signal descriptions, register descriptions, and a detailed functional description including timing, electrical characteristics, and mechanical data.

3.3.2.3 Time-of-Year (TOY) Clock

Dallas Semiconductor’s DS1287 chip, located on the EB66+ utility bus, provides the time-of-year (TOY) function. It is packaged in a plastic 24-pin DIP configuration. The DS1287 is designed with an onchip RAM, a lithium energy source, a quartz crystal, and write-protection circuitry. The device is, therefore, a complete subsystem replacing as many as 16 components in a typical application. The functions available to the user include a nonvolatile time-of-day clock, an alarm, a 100-year calendar, programmable interrupt, a square-wave generator, and 50 bytes of nonvolatile static RAM. The time-of-day and memory are maintained even in the absence of power.

The multiplexed, bidirectional address/data bus, driven by the utility bus, reduces the device pin count because address information and data information time share the same signal paths. Address/data multiplexing does not slow the access time of the device because the bus change from address to data occurs during the internal RAM access time.

The DS1287 includes three separate, fully automatic sources of interrupt for a processor. The alarm interrupt may be programmed to occur at rates from once per second to once per day. The periodic interrupt may be selected for rates from 122 μs to 500 ms. The update-ended interrupt may be used to indicate to the program that an update cycle is complete. The device's interrupt signal **rtc_irq_l** drives CPU interrupt input **cpu_irq<1>** (see Section 3.3.3 and Figure 3–5).

3.3.2.4 Flash ROM

The Intel 28F008SA Flash Memory is located on the EB66+ utility bus. This 1-MB, versatile flash memory provides nonvolatile RAM for operating system support as well as ROM for firmware. The flash ROM is split up into two 512-KB segments. Selection between the two segments is determined by the value of **flash_adr19**. This signal is latched and driven by the interrupt PALs that reside on the ISA bus. Writing a 0 to ISA location 800_{16} selects the lower 512 KB, writing a 1 selects the upper 512 KB.

In order for the flash ROM to be written, 12 V must be present on the V_{pp} pin of the flash ROM. Jumper J17 controls the voltage to this pin. With the jumper across pins 2 and 3, the contents of the flash ROM can be modified. With the jumper across pins 1 and 2, it is protected from write operations (see Table 2–1).

Refer to Intel's *Memory Products Data Book* for additional information on pin assignments and signal descriptions, register descriptions, and a functional description (including timing, electrical characteristics, and mechanical data).

3.3.2.5 ISA Expansion Slots

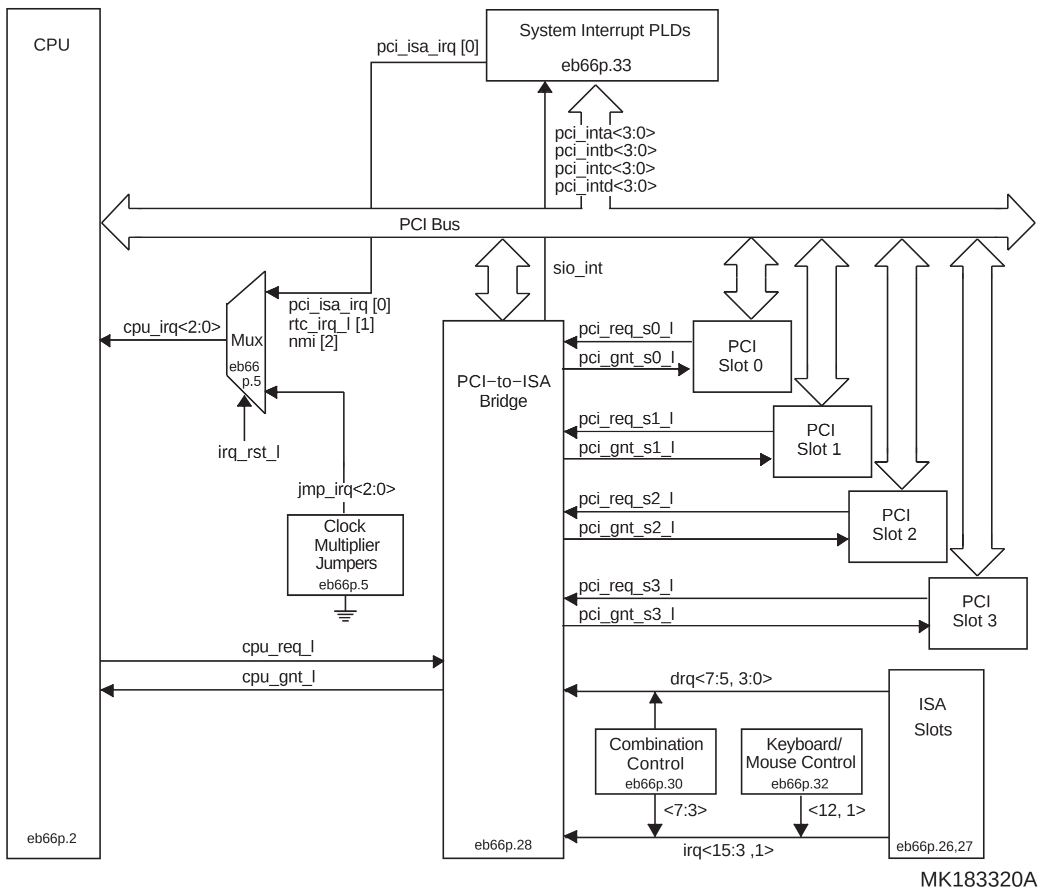
Five ISA expansion slots are provided for plug-in ISA peripherals. One of these is a shared slot and may be used for a PCI or an ISA device.

3.3.3 System Interrupts

Figure 3–5 shows the EB66+ interrupt logic. Interrupt logic is implemented in MACH210–20 and AMD22V10–25 programmable logic devices (PLDs) (schematic page *eb66p.33*). The PLDs allow each PCI and SIO interrupt to be individually masked. The PLDs also allow the current state of the interrupt lines to be read.

3.3 I/O Subsystem

Figure 3–5 Interrupt Control and PCI Arbitration



The interrupt controller has 17 interrupts as follows:

- Four from each of the four PCI slots = 16
- One from the SIO bridge

All PCI interrupts are combined in the PLD and drive a single output signal, **pci_isa_irq**. This signal drives CPU input **cpu_irq0** through a multiplexer. There is also a memory controller error interrupt and an I/O controller error interrupt within the CPU.

During normal operation, the CPU interrupt assignment is as follows:

Interrupt Source	CPU Interrupt	Description
pci_isa_irq	**cpu_irq0**	Combined output of the interrupt PLD
rtc_irq_l	**cpu_irq1**	Realtime clock interrupt from DS1287
nmi	**cpu_irq2**	Nonmaskable interrupt from the SIO bridge

Three jumpers (J13, J14, and J15) connect to one side of the multiplexer. The jumper configuration sets the CPU clock multiplier value through the IRQ inputs during reset.

The ISA bus interrupts (IRQ0 through IRQ15) are all nested through the SIO and then into the CPU. The interrupt assignment can be configured as needed but is normally used as follows:

Interrupt Level	Interrupt Source
IRQ0	Interval timer
IRQ1	Keyboard
IRQ2	Chains interrupts from a slave programmable interrupt controller (PIC)
IRQ3	8-bit ISA from serial port COM2
IRQ4	8-bit ISA from serial port COM1
IRQ5	8-bit ISA from parallel port (or IRQ7)
IRQ6	8-bit ISA from diskette controller
IRQ7	8-bit ISA from parallel port (or IRQ5)
IRQ8	Unused (RTC internal to the SIO)
IRQ9	16-bit ISA
IRQ10	16-bit ISA
IRQ11	16-bit ISA
IRQ12	Mouse
IRQ13	Unused
IRQ14	IDE
IRQ15	16-bit ISA

The EB66+ timer interrupt is generated by the realtime clock by means of **cpu_irq1**, rather than the timer within the SIO, which would route the interrupt through the ISA bus interrupts.

Interrupt PLDs' Function

The MACH210 and AMD22V10 PLDs act as 8-bit I/O slaves on the ISA bus at addresses 804, 805, and 806. This is accomplished by a decode of the three ISA address bits **sa<2:0>** and the three **ecas_addr<2:0>** bits.

Each interrupt can be individually masked by setting the appropriate bit in the mask register. An interrupt is disabled by writing a 1 to the desired position in the mask register. An interrupt is enabled by writing a 0. For example, bit <7> set in interrupt mask register 1 indicates that the INTB2 interrupt is disabled. There are three mask registers located at ISA addresses 804, 805, and 806.

An I/O read at ISA addresses 804, 805, and 806 returns the state of the 17 interrupts rather than the state of the masked interrupts. On read transactions, a 1 means that the interrupt source shown in Figure 3–6 has asserted its interrupt. The mask register can be updated by writing addresses 804, 805, or 806. The mask register is write-only.

Figure 3–6 Interrupt and Interrupt Mask Registers

Interrupt and Interrupt Mask Register 1 (ISA Address 804h)

7	6	5	4	3	2	1	0
intb2	intb1	intb0	sio	inta3	inta2	inta1	inta0

Interrupt and Interrupt Mask Register 2 (ISA Address 805h)

7	6	5	4	3	2	1	0
intd2	intd1	intd0	intc3	intc2	intc1	intc0	intb3

Interrupt and Interrupt Mask Register 3 (ISA Address 806h)

7	6	5	4	3	2	1	0
RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	RAZ	intd3

Notes: RAZ = read−as−zero, read−only.
Interrupt mask register is write−only.

MK183321A

3.3.4 PCI/ISA Arbitration

Arbitration logic is implemented in the Intel 82378ZB chip (SIO). The arbitration scheme is flexible and software programmable. Refer to the Intel *82420/82430 PCIset ISA and EISA Bridges* document for more information on programmable arbitration.

3.4 System Clocks

Figure 3–7 shows the EB66+ clock generation and distribution scheme.

The EB66+ system clocking includes input clocks to the microprocessor as well as clock distribution for the various PCI I/O devices. There are other miscellaneous clocks for the combination controller and ISA bus support. The heart of the clock generation comes from an external crystal feeding an ICS9158 chip, an integrated buffer, and motherboard frequency generator.

System clocking can be divided into three main areas:

- Microprocessor input clocks, **clk_in_h** and **clk_in_l** — These provide a fundamental clock input to the microprocessor and determine its operating frequency. Operating frequencies that are supported include the fundamental frequency of 33.3 MHz or a multiple of 2 through 9 times the fundamental frequency.
- Clock distribution to PCI components — The ICS9158 chip distributes buffered clock signals to all PCI slots, the SIO, and the PCI clock input of the CPU.
- Miscellaneous oscillating signals — These signals are needed for the ISA bus and the combination controller, and are provided by the ICS9158 chip.

The fundamental microprocessor input clock, signal **clk_in_h**, is 33.3 MHz. The 21066A microprocessor can run at a 4.29-ns cycle time (233.3 MHz). This frequency is arrived at by an onchip phase-locked loop (PLL) multiplier. The PLL synthesizes a higher frequency CPU clock from a lower frequency external clock. In this case, the multiplier would be set to ×7. This synthesized clock is only used internally within the microprocessor and has no timing relationship to PCI bus activity.

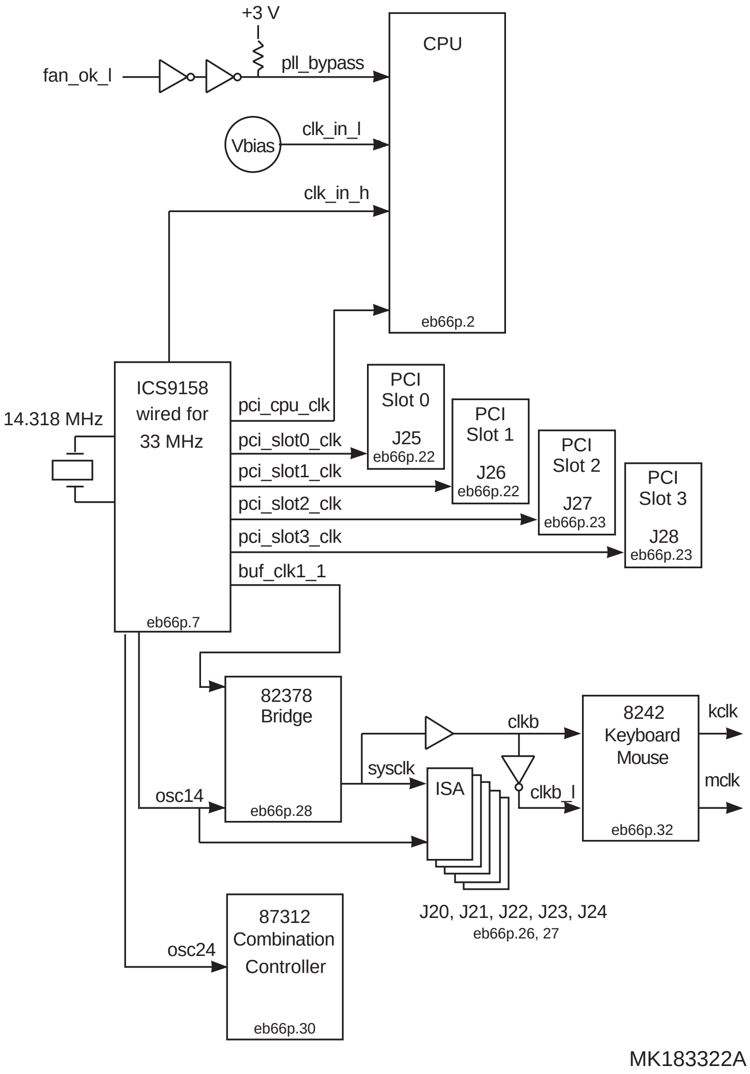
The microprocessor's **cpu_irq<2:0>** pins (see Figure 3–5) are for interrupt requests and are not normally part of the clock logic. However, during reset, these pins receive the clock multiplier values set in jumpers J13, J14, and J15 through a multiplexer (Figure 3–5). The pins are either jumpered to ground, or pulled up if no jumper, in a specific combination so as to set the frequency ratio of the internal clock to the external reference clock. (Refer to Table 2–1 for a list of jumper combinations.)

Note

If the microprocessor heat sink fan sensor detects no airflow, it disables the internal PLL clock multiplier, making the chip run slower. This causes the 21066A to use less power and therefore run cooler.

3.4 System Clocks

Figure 3–7 System Clocks and Distribution



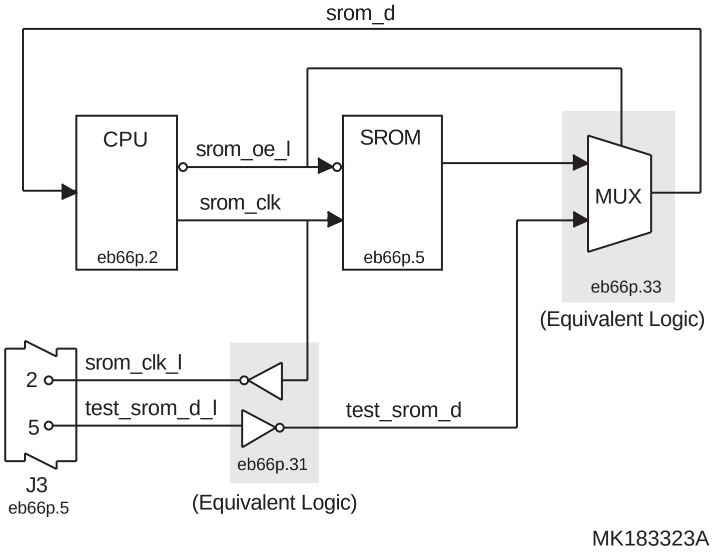
3.5 Serial ROM

The serial ROM (SROM) provides the following functions:

- Initializes the microprocessor's internal processor registers (IPRs)
- Sets up DRAM memory refresh and starts the refresh timer
- Configures the microprocessor's CSR for external Bcache accesses
- Initializes the Bcache
- Writes good ECC by copying the contents of memory to itself
- Copies the contents of the selected flash ROM image to memory, flushes the Icache, and jumps to the ROM image in memory.

Figure 3–8 is a simplified diagram of the SROM serial port logic.

Figure 3–8 SROM Serial Port



Signal **srom_oe_l** selects the input to the multiplexer. The multiplexer selects either the output of the Xilinx XC1765D SROM (**real_srom_d**) or a user-supplied input through the test SROM port (**test_srom_d**). The multiplexer output (**srom_d**) provides data input to the microprocessor. The multiplexer is implemented in the interrupt PALs.

3.5 Serial ROM

After the initial SROM code has been read into the microprocessor's instruction cache, the microprocessor's test SROM port can be used as a software-controlled serial port. This serial port can be used for diagnosing system problems when the only working devices are the microprocessor, the SROM, and the circuits needed for the direct support of the microprocessor and SROM. Connector J3 (see Figure 2–2 and Table 2–2) supports an RS232 or RS422 terminal connection to this port. Additional external logic is not required.

3.6 dc Power Distribution

The dc power from the dedicated PC-type supply is delivered to the PCB's logic on dedicated power planes within the board's 6-layer structure. Figure 3–9 shows the dc power distribution within the EB66+ printed circuit board (PCB).

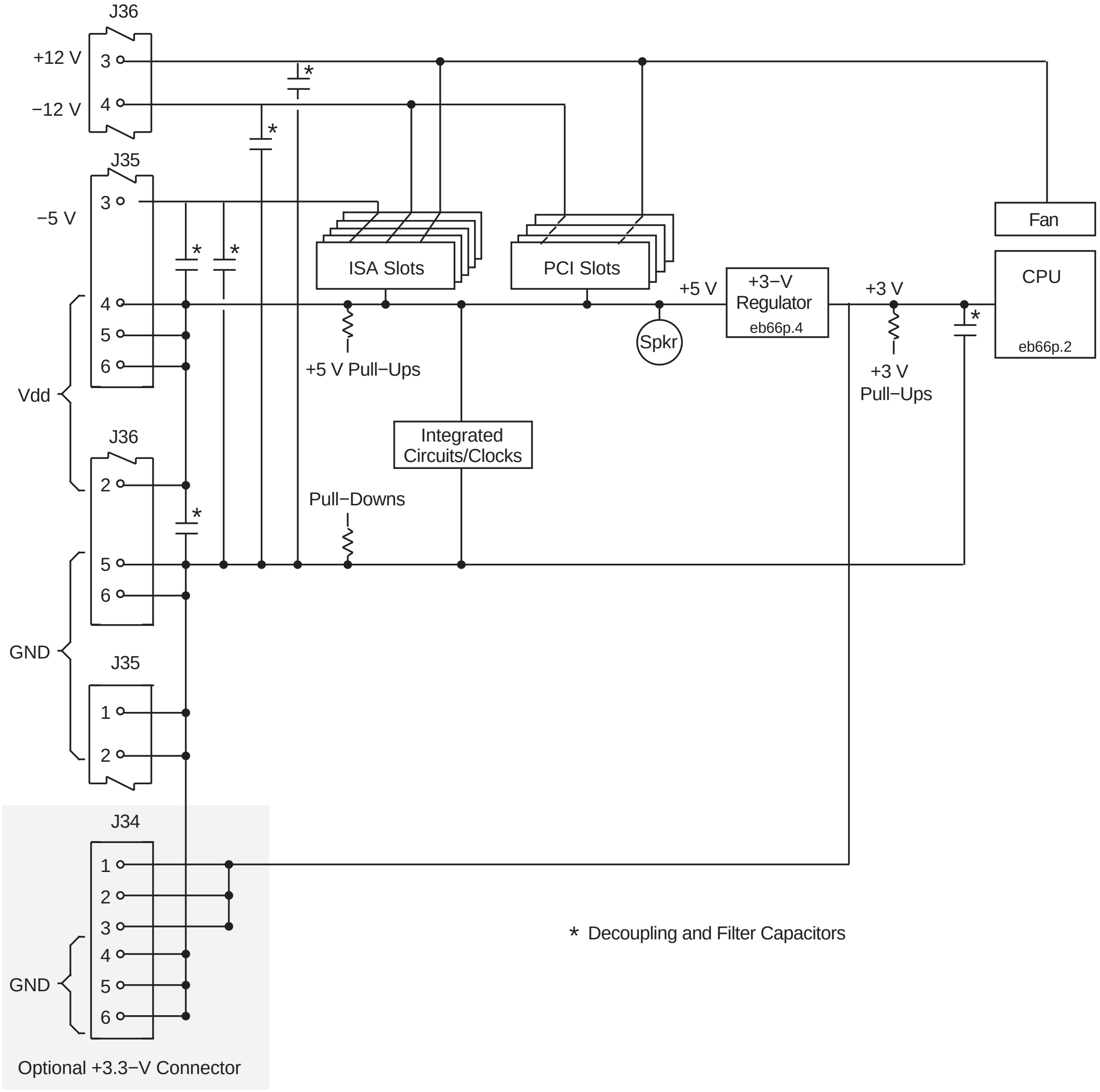
The 21066A requires +3.3 V dc. This voltage is provided by an LT1083 7.5 A linear regulator driven by the +5-V supply. A TI TL7702B power monitor senses the microprocessor's +3.3-V level to ensure that it is stable before the microprocessor inputs are driven.

Refer to Section 3.7 for more information about reset and initialization.

The board has been set up to use the linear regulator, but it can support a PC power supply that provides 3.3 V. If such a PC power supply is used, the 3.3-V regulator (U48) must be be omitted. Then, J34 is populated in the correct orientation and used. Other hardware changes to the board are necessary to allow use of the 3.3-V supply and are shown in the schematics (eb66p.4,5,6).

Refer to Chapter 4 for information about dc power consumption and environmental considerations.

Figure 3–9 dc Power Distribution



MK183324A

3.7 Reset and Initialization

Figure 3–10 shows the EB66+ reset logic.

An external reset switch may be connected to the EB66+ through header connector J2 (see Figure 2–2 and Table 2–2). This low true signal is wire-ORed with the **p_dcok** signal from the power supply. Both serve to keep the **cpu_reset_l** signal to the CPU true until both the reset switch is released and the power supply ramps up and releases the clamp on **p_dcok** so that it can be pulled up. The signal **cpu_reset_l** is also asserted when the microprocessor's heat sink fan fails.

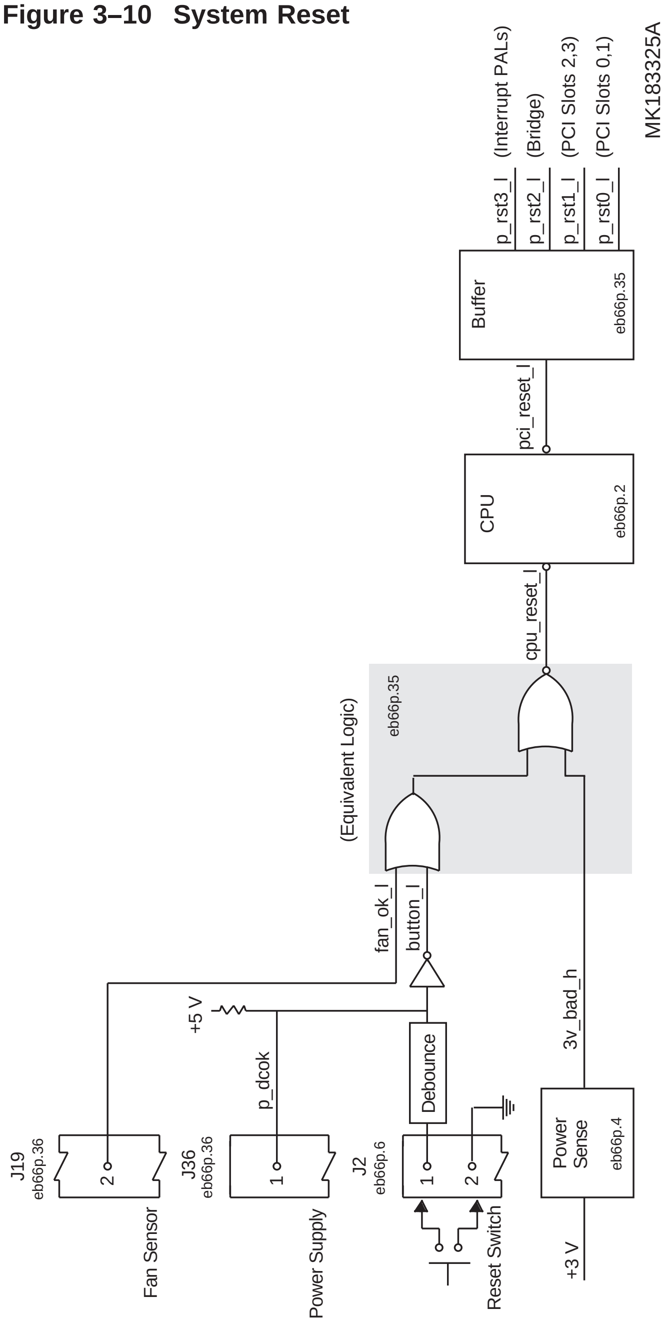
The +5-V dc power level is regulated by a linear device to produce a +3.3-V dc level for the microprocessor (**+3v**). This level is sensed by a TL7702B device that develops a **dcok**-type signal for the +3.3-V levels. Because +3.3 V is not generated as part of the power supply, this circuit is needed to sense this level. The high true output of the TL7702B device, **3v_bad_h**, resets the CPU through the signal **cpu_reset_l**.

Signal **cpu_reset_l** initializes the microprocessor, which, in turn, asserts the signal **pci_reset_l**. Signal **pci_reset_l** is buffered to produce signals **p_rstx_l** that initialize the PCI and ISA I/O subsystem control logic.

The microprocessor uses a serial ROM (SROM) for its initialization code. When **cpu_reset_l** is deasserted, the contents of the SROM are read into the microprocessor's instruction cache. The code is then executed from the instruction cache. For more information on the SROM, refer to Section 3.5.

Refer to the *Alpha 21066, 21066A, and 21068 Microprocessors Hardware Reference Manual* for more information on microprocessor power sequencing.

Figure 3–10 System Reset



3.8 Physical Characteristics

The EB66+ module consists of a 6-layer printed-wiring board (PWB) populated with integrated circuit packages along with supporting active and passive components.

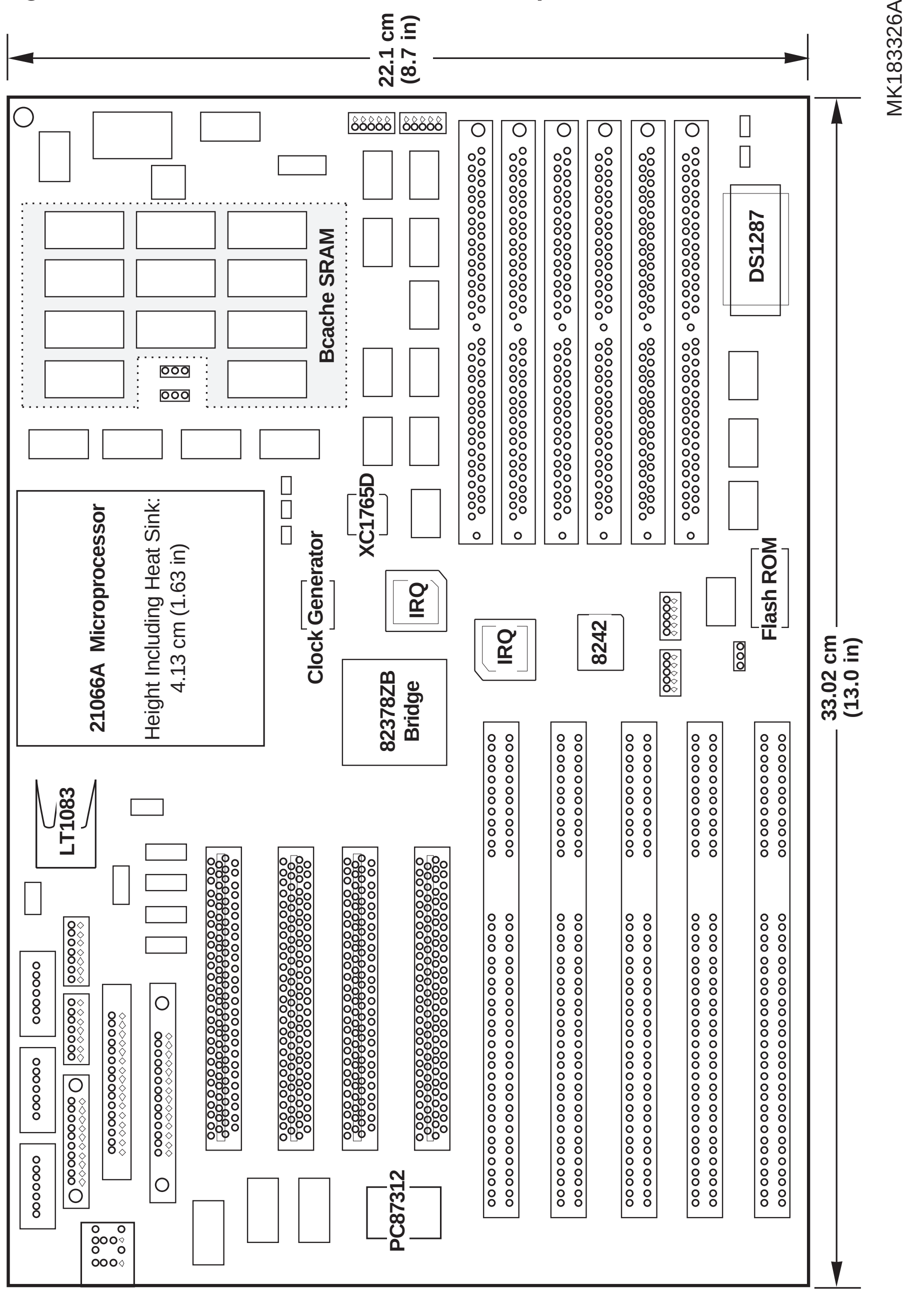
Figure 3–11 shows the EB66+ board dimensions and identifies major functional components. Refer to Chapter 2 for jumper and connector locations.

The EB66+ module has the following dimensions:

- Width: 22.1 cm ±0.13 mm (8.7 in ±0.005 in)
- Length: 33.02 cm ±0.13 mm (13.0 in ±0.005 in)

The EB66+ can be installed in most desktop or deskside/tower system enclosures that accommodate a baby AT module. It may not be usable in certain desktop systems that do not have adequate clearance for the microprocessor and regulator heat sinks.

Figure 3–11 EB66+ Dimensions and Component Outlines



3.9 System Software

EB66+ software consists of the following:

- Serial ROM code
- Mini-Debugger code
- Debug monitor ROM code
- Operating systems

The serial ROM code, Mini-Debugger code, and debug monitor code are all included with the EB66+ and do not require a license. Operating systems are available as licensed products. Refer to Appendix C for a list of related documentation.

3.9.1 Serial ROM Code

The serial ROM code is contained in the Xilinx XC1765D serial configuration PROM. This code is executed by the microprocessor when system power is turned on (Section 3.5). The serial ROM code initializes the system, then transfers control to either the Mini-Debugger, the debug monitor, or the operating system, depending upon the setting of jumpers J18 **sp_bit<4>** and **sp_bit<5>** (Table 2–1).

3.9.2 Mini-Debugger Code

The SROM Mini-Debugger provides basic hardware debugging capability through a serial connector interface to the serial ROM port of the 21066A. Using only an SROM containing this program, a clock source, a CPU chip, and a few gates, you can exercise the device connected to the CPU to debug caches, main memory, and I/O subsystems until the board is functional enough to support a more fully featured monitor.

The Mini-Debugger provides:

- Basic hardware debugging capability
- A monitor that can point to hardware addresses and exercise registers and devices at those locations
- The ability to examine and deposit memory locations
- A case-independent command language
- Support for variable CPU speeds and communication baud rates

For additional information, refer to the *Alpha SROM Mini-Debugger User's Guide*.

3.9.3 Debug Monitor ROM Code

The first image contained in the flash ROM is the debug monitor code. Other ROM images can be stored in the flash ROM for user-specific code development and can be selected by board jumper J18 **sp_bit<5>** (see Table 2–1). The user can develop code on a host system, load the code into the EB66+ system through the SROM serial test port, the optional Ethernet card, the ISA COM1 port, or the diskette, and then save it to nonvolatile flash ROM memory.

The functions provided by the debug monitor include:

- File load
- Read/write memory and registers
- Memory image dump
- Transfer control to program
- Breakpoints
- Single stepping
- Disassembly
- Source-level remote debugging
- Alternate ROM image selecting

The user kit includes the full source code listing for all SROM and debug monitor ROM software. For additional information, refer to the *Alpha Microprocessors Evaluation Board Debug Monitor User's Guide*.

3.9.4 Booting Other ROM Images

The SROM contains support for booting multiple images from the flash ROM. This allows for multiple firmwares for multiple platforms to be supported simultaneously on one module.

When the jumper for J18 **sp_bit<5>** is in, an alternate ROM image is loaded and executed from the flash ROM. Which image is booted depends on the contents of a byte of nonvolatile RAM that resides in the TOY chip (U3–DS1287). The value in location 3F of the TOY chip determines which image is booted. Table 3–8 shows the values.

3.9 System Software

Table 3–8 TOY Chip Values

Value in Location 3F	Image That Is Booted
00	Debug monitor (included)
01	NT firmware (included)
02	OpenVMS firmware (not included)
03	OSF firmware (not included)
80	Load the whole flash ROM to memory, starting at 0_{16}
8*n*	The *n*th image, where *n*=1 means the first image (for $1 \leq n \leq F$)

If an image is specified and not found, the SROM will load the first image with a valid header that is found in the flash ROM. If no valid header is found, the whole flash ROM image is loaded at address 0000 0000.

3.9.5 Operating Systems

The EB66+ is designed to run a wide range of licensed operating systems including OSF/1, Windows NT, OpenVMS, and VxWorks for Alpha. For additional information, contact the Digital Semiconductor Information Line (see Appendix C).

4

Power and Environmental Requirements

This chapter describes the EB66+ power requirements and environmental requirements.

4.1 Power Requirements

The EB66+ derives its main dc system power from a user-supplied, industry-standard PC power supply. The EB66+ consumes 90.6 W of power, excluding plug-in PCI, ISA, and frame buffer devices. Table 4–1 lists the current requirements for each dc supply voltage. The power supply must also supply a **dcok** signal to the system reset logic. (Refer to Section 3.6 and schematic pages *eb66p.4* and *eb66p.36* for more information.)

Table 4–1 Power Supply dc Current Requirements

	Minimum Current*	
Voltage	**With 3.3-V Regulator†**	**With 3.3-V Power Supply**
+5 V dc	15.0 A	7.5 A
-5 V dc	0 A	0 A
+12 V dc	1 A	1 A
-12 V dc	300 mA	300 mA
+3.3 V dc	0 A	7.5 A

*Values indicated are for a fully populated EB66+ system module, excluding plug-in PCI and ISA, with a CPU clock speed of 233 MHz.

†The EB66+ comes with the onboard 3.3-V regulator.

Caution

The 21066A cooling fan *must* have a built-in sensor that drives a signal if the airflow stops. The sensor is connected to EB66+ board connector J19 (see Figure 2–2 and Table 2–1). When the signal is generated, it puts the system into reset. It also asserts **pll_bypass**, which causes the internal PLL to be disabled. This protects the CPU under fan-failure conditions by reducing its speed, and therefore its heat dissipation.

4.2 Environmental Requirements

The EB66+ is designed to run without any additional fans. The 21066A microprocessor running in its upper speed ranges is cooled by a small fan blowing directly into the chip's heat sink.

The EB66+ is specified to run within the following environment:

Parameter	Specification
Temperature:	15°C (59°F) to 32°C (89.6°F)
Relative humidity:	10% to 90% with maximum wet bulb temperature 28°C (82°F) and minimum dew point 2°C (36°F)
Rate of (dry bulb) temperature change:	11°C/hour (20°F/hour) ±2°C/hour (4°F/hour)

A

Address Map

This appendix lists all EB66+ memory and I/O addresses.

A.1 Physical Memory and Non-IOC Space Address Space

Table A–1 is a map of EB66+ physical memory address space and other non-IOC space addresses.

Table A–1 Physical Memory and Non-IOC Address Space Map

Address	Description
0 0000 0000–0 1FFF FFFF	Cacheable memory space (0.5 GB)
0 2000 0000–0 3FFF FFFF	Noncacheable memory space (0.5 GB)
0 4000 0000–0 FFFF FFFF	Nonexistent memory (3.0 GB)
1 0000 0000–1 1FFF FFFF	Graphics memory (0.5 GB)

A.2 Memory Controller Address Space

Table A–2 is a map of memory controller address space.

Table A–2 Memory Controller Address Space Map

Address	Register
1 2000 0000–1 7FFF FFFF	Memory controller CSRs (1.5 GB)
1 2000 0000	Bank configuration 0
1 2000 0008	Bank configuration 1
1 2000 0010	Bank configuration 2
1 2000 0018	Bank configuration 3
1 2000 0020	Bank mask 0
1 2000 0028	Bank mask 1
1 2000 0030	Bank mask 2
1 2000 0038	Bank mask 3
1 2000 0040	Bank timing 0
1 2000 0048	Bank timing 1
1 2000 0050	Bank timing 2
1 2000 0058	Bank timing 3
1 2000 0060	Global timing
1 2000 0068	Error status
1 2000 0070	Error address
1 2000 0078	Cache control
1 2000 0080	Video and graphics control
1 2000 0088	Plane mask
1 2000 0090	Foreground

A.3 IOC Control and Status Register Address Space

Table A–3 is a map of the microprocessor's IOC control and status register (CSR) address space.

Table A–3 IOC CSR Address Space Map

Address	Register
1 8000 0000–1 9FFF FFFF	IOC CSR space (0.5 GB)
1 8000 0000	Host address extension
1 8000 0020	Configuration cycle type
1 8000 0040	Status 0
1 8000 0060	Status 1
1 8000 0080	IOC translation base invalidate all
1 8000 00A0	Translation base enable
1 8000 00C0	Soft reset
1 8000 00E0	Parity disable
1 8000 0100	Window base 0
1 8000 0120	Window base 1
1 8000 0140	Window mask 0
1 8000 0160	Window mask 1
1 8000 0180	Translated base 0
1 8000 01A0	Translated base 1
1 8100 0000	TB tag 0
1 8100 0020	TB tag 1
1 8100 0040	TB tag 2
1 8100 0060	TB tag 3
1 8100 0080	TB tag 4
1 8100 00A0	TB tag 5
1 8100 00C0	TB tag 6
1 8100 00E0	TB tag 7

A.4 PCI Interrupt Acknowledge/Special Cycle Address Space

The PCI interrupt acknowledge/special cycle address space comprises 0.5 GB and covers the address range of 1 A000 0000 through 1 BFFF FFFF.

A.5 PCI I/O Address Space

The PCI I/O address space comprises 0.5 GB and covers the address range of 1 C000 0000 through 1 DFFF FFFF. The PCI operating register set occupies this space.

A.6 SIO PCI-to-ISA Bridge Operating Register Address Space

Table A–4 is a map of the SIO PCI-to-ISA bridge operating address space.

Table A–4 SIO PCI-to-ISA Bridge Operating Register Address Space Map

Offset	Address	Register
000	1 C000 0000	DMA1 CH0 Base and Current Address
001	1 C000 0020	DMA1 CH0 Base and Current Count
002	1 C000 0040	DMA1 CH1 Base and Current Address
003	1 C000 0060	DMA1 CH1 Base and Current Count
004	1 C000 0080	DMA1 CH2 Base and Current Address
005	1 C000 00A0	DMA1 CH2 Base and Current Count
006	1 C000 00C0	DMA1 CH3 Base and Current Address
007	1 C000 00E0	DMA1 CH3 Base and Current Count
008	1 C000 0100	DMA1 Status and Command
009	1 C000 0120	DMA1 Write Request
00A	1 C000 0140	DMA1 Write Single Mask Bit
00B	1 C000 0160	DMA1 Write Mode
00C	1 C000 0180	DMA1 Clear Byte Pointer
00D	1 C000 01A0	DMA1 Master Clear
00E	1 C000 01C0	DMA1 Clear Mask
00F	1 C000 01E0	DMA1 Read/Write All Mask Register Bits
020	1 C000 0400	INT 1 Control
021	1 C000 0420	INT 1 Mask

(continued on next page)

Table A–4 (Cont.) SIO PCI-to-ISA Bridge Operating Register Address Space Map

Offset	Address	Register
040	1 C000 0800	Timer Counter 1 - Counter 0 Count
041	1 C000 0820	Timer Counter 1 - Counter 1 Count
042	1 C000 0840	Timer Counter 1 - Counter 2 Count
043	1 C000 0860	Timer Counter 1 - Command Mode
060	1 C000 0C00	Reset UBus IRQ12
061	1 C000 0C20	NMI Status and Control
070	1 C000 0E00	CMOS RAM Address and NMI Mask
078–07B	1 C000 0F18	BIOS Timer
080	1 C000 1000	DMA Page Register Reserved
081	1 C000 1020	DMA Channel 2 Page
082	1 C000 1040	DMA Channel 3 Page
083	1 C000 1060	DMA Channel 1 Page
084	1 C000 1080	DMA Page Register Reserved
085	1 C000 10A0	DMA Page Register Reserved
086	1 C000 10C0	DMA Page Register Reserved
087	1 C000 10E0	DMA Channel 0 Page
088	1 C000 1100	DMA Page Register Reserved
089	1 C000 1120	DMA Channel 6 Page
08A	1 C000 1140	DMA Channel 7 Page
08B	1 C000 1160	DMA Channel 5 Page
08C	1 C000 1180	DMA Page Register Reserved
08D	1 C000 11A0	DMA Page Register Reserved
08E	1 C000 11C0	DMA Page Register Reserved
08F	1 C000 11E0	DMA Low Page Register Refresh
090	1 C000 1200	DMA Page Register Reserved
092	1 C000 1240	Port 92
094	1 C000 1280	DMA Page Register Reserved
095	1 C000 12A0	DMA Page Register Reserved

(continued on next page)

A.6 SIO PCI-to-ISA Bridge Operating Register Address Space

Table A–4 (Cont.) SIO PCI-to-ISA Bridge Operating Register Address Space Map

Offset	Address	Register
096	1 C000 12C0	DMA Page Register Reserved
098	1 C000 1300	DMA Page Register Reserved
09C	1 C000 1380	DMA Page Register Reserved
09D	1 C000 13A0	DMA Page Register Reserved
09E	1 C000 13C0	DMA Page Register Reserved
09F	1 C000 13E0	DMA Low Page Register Refresh
0A0	1 C000 1400	INT2 Control
0A1	1 C000 1420	INT2 Mask
0C0	1 C000 1800	DMA2 CH0 Base and Current Address
0C2	1 C000 1840	DMA2 CH0 Base and Current Count
0C4	1 C000 1880	DMA2 CH1 Base and Current Address
0C6	1 C000 18C0	DMA2 CH1 Base and Current Count
0C8	1 C000 1900	DMA2 CH2 Base and Current Address
0CA	1 C000 1940	DMA2 CH2 Base and Current Count
0CC	1 C000 1980	DMA2 CH3 Base and Current Address
0CE	1 C000 19C0	DMA2 CH3 Base and Current Count
0D0	1 C000 1A00	DMA2 Status(r) and Command(w)
0D2	1 C000 1A40	DMA2 Write Request
0D4	1 C000 1A80	DMA2 Write Single Mask Bit
0D6	1 C000 1AC0	DMA2 Write Mode
0D8	1 C000 1B00	DMA2 Clear Byte Pointer
0DA	1 C000 1B40	DMA2 Master Clear
0DC	1 C000 1B80	DMA2 Clear Mask
0DE	1 C000 1BC0	DMA2 Read/Write All Mask Register Bits
0F0	1 C000 1E00	Coprocessor Error
372	1 C000 6E40	Secondary Floppy Disk Digital Output
3F2	1 C000 7E40	Primary Floppy Disk Digital Output
40A	1 C000 8140	Scatter-Gather Interrupt Status

(continued on next page)

Table A–4 (Cont.) SIO PCI-to-ISA Bridge Operating Register Address Space Map

Offset	Address	Register
40B	1 C000 8160	DMA1 Extended Mode
410	1 C000 8200	CH0 Scatter-Gather Command
411	1 C000 8220	CH1 Scatter-Gather Command
412	1 C000 8240	CH2 Scatter-Gather Command
413	1 C000 8260	CH3 Scatter-Gather Command
415	1 C000 82A0	CH5 Scatter-Gather Command
416	1 C000 82C0	CH6 Scatter-Gather Command
417	1 C000 82E0	CH7 Scatter-Gather Command
418	1 C000 8300	CH0 Scatter-Gather Status
419	1 C000 8320	CH1 Scatter-Gather Status
41A	1 C000 8340	CH2 Scatter-Gather Status
41B	1 C000 8360	CH3 Scatter-Gather Status
41D	1 C000 83A0	CH5 Scatter-Gather Status
41E	1 C000 83C0	CH6 Scatter-Gather Status
41F	1 C000 83E0	CH7 Scatter-Gather Status
420–423	1 C000 8418	CH0 Scatter-Gather Descriptor Table Pointer
424–427	1 C000 8498	CH1 Scatter-Gather Descriptor Table Pointer
428–42B	1 C000 8518	CH2 Scatter-Gather Descriptor Table Pointer
42C–42F	1 C000 8598	CH3 Scatter-Gather Descriptor Table Pointer
434–437	1 C000 8698	CH5 Scatter-Gather Descriptor Table Pointer
438–43B	1 C000 8718	CH6 Scatter-Gather Descriptor Table Pointer
43C–43F	1 C000 8798	CH7 Scatter-Gather Descriptor Table Pointer
481	1 C000 9020	DMA CH2 High Page
482	1 C000 9040	DMA CH3 High Page
483	1 C000 9060	DMA CH1 High Page
487	1 C000 90E0	DMA CH0 High Page
489	1 C000 9120	DMA CH6 High Page
48A	1 C000 9140	DMA CH7 High Page

(continued on next page)

Table A–4 (Cont.) SIO PCI-to-ISA Bridge Operating Register Address Space Map

Offset	Address	Register
48B	1 C000 9160	DMA CH5 High Page
4D6	1 C000 9AC0	DMA2 Extended Mode

A.7 PCI Configuration Address Space

The PCI configuration address space comprises 0.5 GB and covers the address range of 1 E000 0000 through 1 FFFF FFFF. The PCI configuration register set occupies this space. Table A–5 identifies the EB66+ PCI devices and the corresponding PCI address bit that drives the device **idsel** pin.

Table A–5 Address Bits and PCI Device idsel Pins

PCI Device	PCI Address Bit Driving idsel Pin	Physical Address
PCI expansion slot 0	**pci_ad<17>**	1 E040 0000
PCI expansion slot 1	**pci_ad<18>**	1 E080 0000
System I/O (SIO)	**pci_ad<19>**	1 E100 0000
PCI expansion slot 2	**pci_ad<20>**	1 E200 0000
PCI expansion slot 3	**pci_ad<21>**	1 E400 0000

A.8 SIO PCI-to-ISA Bridge Configuration Address Space

Table A–6 is a map of SIO PCI-to-ISA bridge configuration address space. PCI address bit **pci_ad19** drives the **idsel** chip select pin for access to the configuration register space.

Table A–6 SIO PCI-to-ISA Bridge Configuration Address Space Map

Offset	Address	Register
00–01	1 E100 0008	Vendor ID
02–03	1 E100 0048	Device ID
04–05	1 E100 0088	Command
06–07	1 E100 00C8	Device Status
08	1 E100 0100	Revision ID
40	1 E100 0800	PCI Control
41	1 E100 0820	PCI Arbiter Control
42	1 E100 0840	PCI Arbiter Priority Control
44	1 E100 0880	MEMCS# Control
45	1 E100 08A0	MEMCS# Bottom of Hole
46	1 E100 08C0	MEMCS# Top of Hole
47	1 E100 08E0	MEMCS# Top of Memory
48	1 E100 0900	ISA Address Decoder Control
49	1 E100 0920	ISA Address Decoder ROM Block Enable
4A	1 E100 0940	ISA Address Decoder Bottom of Hole
4B	1 E100 0960	ISA Address Decoder Top of Hole
4C	1 E100 0980	ISA Controller Recovery Timer
4D	1 E100 09A0	ISA Clock Divisor
4E	1 E100 09C0	Utility Bus Chip Select Enable A
4F	1 E100 09E0	Utility Bus Chip Select Enable B
54	1 E100 0A80	MEMCS# Attribute Register #1
55	1 E100 0AA0	MEMCS# Attribute Register #2
56	1 E100 0AC0	MEMCS# Attribute Register #3
57	1 E100 0AE0	Scatter-Gather Relocation Base Address
80–81	1 E100 1008	BIOS Timer Base Address

A.9 PCI Sparse Memory Address Space

The PCI sparse memory address space comprises 4.0 GB and covers the address range of 2 0000 0000 through 2 FFFF FFFF.

A.10 PCI Dense Memory Address Space

The PCI dense memory address space comprises 4.0 GB and covers the address range of 3 0000 0000 through 3 FFFF FFFF.

A.11 PC87312 Combination Controller Register Address Space

Table A–7 lists the base address values for the PC87312 combination diskette, serial port, and parallel port controller.

The general registers are located at addresses 398 (index address) and 399 (data address). Writing an index value of “1” to address 398 selects the Function Address Register. If a read from address 399 follows, the data associated with the Function Address Register is returned. If a write to address 399 follows, the Function Address Register will be updated.

Table A–7 PC87312 Combination Controller Register Address Space Map

Address Offset Read/Write	Physical Address	Register
General Registers		
398	1 C000 7300	Index Address
399	1 C000 7320	Data Address
	Index	**Register**
	0	Function Enable
	1	Function Address
	2	Power and Test

(continued on next page)

A.11 PC87312 Combination Controller Register Address Space

Table A–7 (Cont.) PC87312 Combination Controller Register Address Space Map

Address Offset Read/Write	Physical Address	Register
COM2 Serial Port Registers		
2F8-R 0DLAB=0	1 C000 5F00	COM2 Receiver Buffer
2F8-W 0DLAB=0	1 C000 5F00	COM2 Transmitter Holding
2F8 0DLAB=1	1 C000 5F00	COM2 Divisor Latch Register (LSB)
2F9 1DLAB=0	1 C000 5F20	COM2 Interrupt Enable
2F9 1DLAB=1	1 C000 5F20	COM2 Divisor Latch Register (MSB)
2FA-R	1 C000 5F40	COM2 Interrupt Identification
2FA-W	1 C000 5F40	COM2 FIFO Control
2FB	1 C000 5F60	COM2 Line Control
2FC	1 C000 5F80	COM2 Modem Control
2FD	1 C000 5FA0	COM2 Line Status
2FE	1 C000 5FC0	COM2 Modem Status
2FF	1 C000 5FE0	COM2 Scratch Pad
Diskette Registers		
3F0-R	1 C000 7E00	Status A
3F1-R	1 C000 7E20	Status B
3F2-R/W	1 C000 7E40	Digital Output
3F3-R/W	1 C000 7E60	Tape Drive
3F4-R	1 C000 7E80	Main Status
3F4-W	1 C000 7E80	Data Rate Select
3F5-R/W	1 C000 7EA0	Data (FIFO)
3F6	1 C000 7EC0	None (tristate bus)
3F7-R	1 C000 7EE0	Digital Input
3F7-W	1 C000 7EE0	Configuration Control

(continued on next page)

A.11 PC87312 Combination Controller Register Address Space

Table A–7 (Cont.) PC87312 Combination Controller Register Address Space Map

Address Offset Read/Write	Physical Address	Register
COM1 Serial Port Registers		
3F8-R 0DLAB=0	1 C000 7F00	COM1 Receiver Buffer
3F8-W 0DLAB=0	1 C000 7F00	COM1 Transmitter Holding
3F8 0DLAB=1	1 C000 7F00	COM1 Divisor Latch Register (LSB)
3F9 1DLAB=0	1 C000 7F20	COM1 Interrupt Enable
3F9 1DLAB=1	1 C000 7F20	COM1 Divisor Latch Register (MSB)
3FA-R	1 C000 7F40	COM1 Interrupt Identification
3FA-W	1 C000 7F40	COM1 FIFO Control
3FB	1 C000 7F60	COM1 Line Control
3FC	1 C000 7F80	COM1 Modem Control
3FD	1 C000 7FA0	COM1 Line Status
3FE	1 C000 7FC0	COM1 Modem Status
3FF	1 C000 7FE0	COM1 Scratch Pad
Parallel Port Registers		
3BC-R/W	1 C000 7780	Data
3BD-R	1 C000 77A0	Status
3BE-R/W	1 C000 77C0	Control
3BF	1 C000 77E0	None (tristate)

Table A–8 shows the IDE register addresses and their functions.

Table A–8 IDE Register Addresses

Address Offset	Physical Address	Read Function	Write Function
1F0	1 C000 3E00	Data	Data
1F1	1 C000 3E20	Error	Features (write Precomp)

(continued on next page)

Table A–8 (Cont.) IDE Register Addresses

Address Offset	Physical Address	Read Function	Write Function
1F2	1 C000 3E40	Sector count	Sector count
1F3	1 C000 3E60	Sector number	Sector number
1F4	1 C000 3E80	Cylinder low	Cylinder low
1F5	1 C000 3EA0	Cylinder high	Cylinder high
1F6	1 C000 3EC0	Drive/head	Drive/head
1F7	1 C000 3EE0	Status	Command
3F6	1 C000 7EC0	Alternate status	Device control
3F7	1 C000 7EE0	Drive address	Not used

A.12 Utility Bus Device Addresses

Table A–9 shows the decoding for utility bus devices driven from the SIO bridge.

Table A–9 Utility Bus Device Decode

	Device Address Select Bits					
ec-sen_l	ecas-addr_2	ecas-addr_1	ecas-addr_0	Device Select Signal	Device Selected	Address Decode
0	0	0	0	**rtcale_l**	TOY address	70, 72, 74, 76
0	0	0	1	**rtccs_l**	TOY data	71, 73, 75, 77
0	0	1	0	**kbcs_l**	Mouse/kbd enable	60, 62, 64, 66
0	0	1	1	**flashcs_l**	Flash ROM*	FFF8 0000–FFFF FFFF
0	1	0	0	—	Unused	—
0	1	0	1	—	Unused	—

*The encoded chip select signal for **flashcs_l** will always be generated for accesses to the upper 64-KB segment at the top of the 1-MB (F0000–FFFFF) segment, and its aliases at the top of the 4-GB segment and 4-GB-to-1-MB segment. Access to the lower 64-KB (E0000–EFFFF) segment and its aliases at the top of the 4-GB and 4-GB to 1-MB segment can be enabled or disabled through the SIO. An additional 384 KB of BIOS memory at the top of the 4-GB (FFFD0000–FFFDFFFF) segment can be enabled for BIOS use.

(continued on next page)

A.12 Utility Bus Device Addresses

Table A–9 (Cont.) Utility Bus Device Decode

	Device Address Select Bits					
ec-sen_l	ecas-addr_2	ecas-addr_1	ecas-addr_0	Device Select Signal	Device Selected	Address Decode
0	1	1	0	—	Unused	—
0	1	1	1	—	Unused	—
1	0	0	0	—	Unused	—
1	0	0	1	—	Unused	—
1	0	1	0	**flash_adr_19**	Flash ROM	0800
1	0	1	1	—	Unused	—
1	1	0	0	—	Unused	—
1	1	0	1	—	Unused	—
1	1	1	0	—	Unused	—
1	1	1	1	—	Unused	—

A.13 Interrupt Control PLD Addresses

Table A–10 lists the registers and memory addresses for the interrupt control programmable logic device (PLD).

Table A–10 Interrupt Control PLD Addresses

Offset	Physical Address	Register
804	1 C001 0080	Interrupt status/interrupt mask register 1
805	1 C001 00A0	Interrupt status/interrupt mask register 2
806	1 C001 00C0	Interrupt status/interrupt mask register 3

A.14 8242PC Keyboard and Mouse Controller Addresses

Table A–11 lists the register and memory addresses for the keyboard and mouse controller.

A.14 8242PC Keyboard and Mouse Controller Addresses

Table A–11 Keyboard and Mouse Controller Addresses

Offset	Physical Address	Register
60-R	1 C000 0C00	Auxiliary/keyboard data
60-W	1 C000 0C00	Command data
64-R	1 C000 0C80	Read status
64-W	1 C000 0C80	Command

A.15 Time-of-Year Clock Device Addresses

Table A–12 lists the register and memory addresses for the TOY clock device (Dallas Semiconductor DS1287).

The TOY clock register is accessed by writing to address 70 with the latched index. Then, reading from, or writing to, address 71 reads or writes the register. For example, writing an “8” to address 70 followed by a read from address 71 returns the value of the month. Writing a “4” to address 70 followed by a write to address 71 updates the hour register.

Table A–12 Time-of-Year Clock Device Addresses

Offset	Index (Latched)	Physical Address	Register
70	0	1 C000 0E00	Seconds
70	1	1 C000 0E00	Seconds alarm
70	2	1 C000 0E00	Minutes
70	3	1 C000 0E00	Minutes alarm
70	4	1 C000 0E00	Hour
70	5	1 C000 0E00	Hour alarm
70	6	1 C000 0E00	Day of week
70	7	1 C000 0E00	Day of month
70	8	1 C000 0E00	Month
70	9	1 C000 0E00	Year
70	A	1 C000 0E00	Register A
70	B	1 C000 0E00	Register B

(continued on next page)

Table A–12 (Cont.) Time-of-Year Clock Device Addresses

Offset	Index (Latched)	Physical Address	Register
70	C	1 C000 0E00	Register C
70	D	1 C000 0E00	Register D
70	3F	1 C000 0E00	Boot alternate image number
71	—	1 C000 0E20	TOY clock chip select

A.16 Flash ROM Memory Segment Select Register

Table A–13 lists the registers addresses for the flash ROM. The flash ROM is partitioned into two 512-KB segments. To select the first 512-KB segment, write a value of 0 to ISA port address 800_{16}. To access the second 512-KB segment, write a value of 1 to this register. This register is write-only.

Table A–13 Flash Memory Segment Select Register

Offset	Physical Address	Register
800	1 C001 0000	Flash ROM segment select

A.17 Flash ROM Memory Addresses

Table A–14 lists the address range for the flash ROM.

Table A–14 Flash ROM Memory Addresses (Within Segment)

Offset	Physical Address	Capacity
0 0000—7 FFFF	3 FFF8 0000—3 FFFF FFFF	512 KB

A.18 Flash ROM Configuration Registers

Table A–15 lists the configuration registers for the Intel 28F00SA 1-MB flash ROM. A read operation is performed by simply reading from the appropriate address.

A.18 Flash ROM Configuration Registers

To write data, the flash ROM must first be erased. The internal workings only allow the flash ROM to be erased in 64-KB blocks (see Section A.19). In order to change 1 byte, read the whole 64-KB block into memory, change the byte in memory, erase that 64-KB block in the flash ROM, and finally write the whole 64 KB to the flash ROM.

All accesses to the flash ROM (except for read operations) require two bus cycles. First register data is written to set up the registers, then the subsequent read or write operation performs the operation desired. For more information on the reading, erasing, and writing the flash ROM, see the *Intel Memory Products Data Book*.

Table A–15 Flash ROM Configuration Registers

Offset	Data Written on First Access	Physical Address	Register
x	FF	3 FFF8 xxxx	Read array/reset
x	90	3 FFF8 xxxx	Intelligent identifier
x	70	3 FFF8 xxxx	Read status
x	50	3 FFF8 xxxx	Clear status
BA	20	3 FFFz zzzz	Erase setup/Confirm
x	B0	3 FFF8 xxxx	Erase suspend/resume
WA	40	3 FFFz zzzz	Byte write setup/write
WA	10	3 FFFz zzzz	Alt byte write setup/write

Key to Physical Address and Offset Notation

x = Don't care
z zzzz = Corresponds to the value of BA or WA
BA = Address within the block being erased
WA = Address of memory location to be written to

A.19 Memory Map of Flash ROM Memory

Table A–16 provides a map of flash ROM memory.

Table A–16 Memory Map of Flash ROM Memory

Offset	Physical Address	Block Number*	Capacity
0 0000—0 FFFF	3 FFF8 0000—3 FFF8 FFFF	0,8	64 KB
1 0000—1 FFFF	3 FFF9 0000—3 FFF9 FFFF	1,9	64 KB
2 0000—2 FFFF	3 FFFA 0000—3 FFFA FFFF	2,10	64 KB
3 0000—3 FFFF	3 FFFB 0000—3 FFFB FFFF	3,11	64 KB
4 0000—4 FFFF	3 FFFC 0000—3 FFFC FFFF	4,12	64 KB
5 0000—5 FFFF	3 FFFD 0000—3 FFFD FFFF	5,13	64 KB
6 0000—6 FFFF	3 FFFE 0000—3 FFFE FFFF	6,14	64 KB
7 0000—7 FFFF	3 FFFF 0000—3 FFFF FFFF	7,15	64 KB

*The block number is determined by the value in the flash ROM memory segment select register (see Section A.16).

B
SROM Initialization

The 21066A microprocessor provides a mechanism for loading the initial instruction stream (Istream) from a compact serial ROM (SROM) to start the bootstrap procedure. The SROM executable image is limited to the size of the CPU instruction cache (Icache). Because the image is running only in the Icache, it is relatively difficult to debug. Therefore, Digital suggests that the scope and purpose of this code be limited to performing the system initialization necessary to boot the next level of firmware contained in the larger system flash ROM.

However, trade-offs between simplicity and convenience were made to support the EB66+ in various configurations. The source code for the EB66+ SROM is available with free licensing for use and modification.

B.1 SROM Initialization Procedures

After reset, the contents of the SROM is loaded into the Icache. After loading the Icache, the CPU begins execution at location zero. Execution is performed in the CPU PALmode environment with privileged access to the computer hardware. To initialize the SROM:

1. Initialize the CPU’s internal processor registers (IPRs).
2. Perform the minimum I/O subsystem initialization necessary to access the realtime clock (RTC) and the system flash ROM.
3. Detect CPU speed by polling the periodic interrupt flag (PIF) in the RTC.
4. Set up memory and/or backup cache (Bcache) parameters based on the speed of the CPU, the size and speed of the cache, and the configuration jumpers.
5. Wake up the DRAMs.
6. Initialize the Bcache if it is turned on.
7. Copy the contents of the entire memory to itself to ensure good memory data error correction code (ECC).

8. Scan system flash ROM for special header that specifies where and how system flash ROM firmware should be loaded.
9. Copy the contents of the system flash ROM to memory and begin code execution.
10. Pass parameters up to the next level of firmware to provide a predictable firmware interface.

B.2 Firmware Interface

A firmware interface provides a mechanism for passing critical information about the state of the system and CPU up to the next level of firmware. This interface is achieved through the use of a set of defined SROM output parameters as described in Table B–1.

This particular firmware interface serves the 21066A microprocessor. Other Alpha implementations would likely require a different firmware interface.

Table B–1 Output Parameter Descriptions

Output Parameter	Description
r1 (t0)—AboxCtl value	The AboxCtl value allows the next-level software to preserve any system-specific Dcache configuration information. This register also contains the superpage enables that could be modified by both the next-level firmware and/or operating system PALcodes.
r2 (t1)—bankConfig[0..1]	Specifies bankConfig0 in lower 32 bits.
r3 (t2)—bankConfig[2..3]	Specifies bankConfig2 in lower 32 bits.
r4 (t3)—bankMask[0..1]	Specifies bankMask0 in lower 32 bits.
r5 (t4)—bankMask[2..3]	Specifies bankMask2 in lower 32 bits.

(continued on next page)

Table B–1 (Cont.) Output Parameter Descriptions

Output Parameter	Description		
	Minor Type <63:32>	Major Type <31:0>	CPU
r16 (a0)—Processor identification		1	—
		2	21064
	0		Pass 2 or 2.1
	1		Pass 3 (21064–150 or 21064/200)
		3	Simulation
		4	21066/21066A/21068
	0		Reserved
	1		Pass 1 or 1.1 (21066)
	2		Pass 2 (21066)
	3		Pass 1 or 1.1 (21068)
	4		Pass 2 (21068)
	5		21066A–233
	6		21066A–100
		5	21164
		6	21064A
	0		Reserved
	1		Pass 1
	2		Pass 1.1
	3		Pass 2

(continued on next page)

Table B–1 (Cont.) Output Parameter Descriptions

Output Parameter	Description
r17 (a1)—Memory size	This value is an unsigned quadword count of the number of contiguous bytes of good memory in the system starting at physical address zero. This simple mechanism will be sufficient for simple systems. Systems that need to communicate more detailed memory configuration may do so through the system context value (see System context value table entry).
r18 (a2)—Cycle count in picoseconds	This value is the number of picoseconds that elapse for each increment of the processor cycle count (as read by the RPCC instruction). Note that this may be a multiple of the actual internal cycle count of the microprocessor as specified in the *Alpha Architecture Reference Manual* (a microprocessor will increment the processor cycle count by a multiple of the microprocessor clock, where the multiple is a power of 2, including $2^0 = 1$).
r19 (a3)—Signature and system revision ID	This register includes a signature, which specifies that the transfer is following the standard protocol and that the other values may be trusted. In addition, the signature can identify which version of the protocol is being followed. The system revision is a 16-bit field that communicates system revisions that would be significant to operating system software. The register has the following format: Bits <63:32> = Ignore Bits <31:16> = Signature Bits <15:00> = System revision Valid signatures have the following values: 0xdeca—V1 (previous version of this specification) 0xdecb—V2 (current version of this specification)

(continued on next page)

Table B–1 (Cont.) Output Parameter Descriptions

Output Parameter	Description
r20 (a4)—Active processor mask	The processor mask identifies each processor that is present on the current system. Each mask bit corresponds to a processor number associated by the bit number (for example, bit 0 corresponds to processor 0). A value of 1 in the mask indicates that the processor is present, a value of 0 indicates that the processor is not present. To qualify as present, a processor must be: • Physically present • Functioning normally • Capable of sending and receiving interprocessor interrupt requests Uniprocessor systems pass a value of 1 to this register.
r21 (a5)—System context value	The context value is interpreted in a system-specific manner. If the system needs to pass more than one system-specific parameter, it may pass a context value, which is a physical address pointer to a data structure of many system-specific values.

B.3 Automatic CPU Speed Detection

The EB66+ RTC detects the speed of the CPU. This allows a somewhat generic SROM to support EB66+ systems configured for different CPU speeds. The CPU speed is measured by the number of CPU cycles executed while polling the RTC’s periodic interrupt flag (PIF).

B.4 Bcache Timing

Set up the CPU and Bcache timing. The EB66+ Bcache timing is determined based on CPU speed and fixed delays associated with the Bcache subsystem. The pertinent Bcache delays used in the calculations result from logic devices used in the Bcache subsystem, SRAM specifications, and board etch delays. This data is used to calculate the appropriate cache register (CAR) setting. Tables B–2 and B–3 describe the EB66+ fixed delays.

B.4 Bcache Timing

Table B–2 EB66+ Cache Loop Delay Characteristics

Function	Delay	Description
Tadr	11.0 ns	Address delay
skew	2.0 ns	Skew in Bcache circuit

Table B–3 Typical SRAM Specifications

Function	Specification	Description
Tacc	6 8 10 ns	Access from address valid to data valid
Twc	6 8 10 ns	Write cycle time
Twp	6 8 10 ns	Write pulse width
Tdw	3 4 5 ns	Data setup to write pulse deassertion
Tdh	0 0 0 ns	Data hold from write pulse deassertion
Taw	6 8 10 ns	Address setup to write pulse deassertion
Twr	0 0 0 ns	Address hold from write pulse deassertion
Tas	0 0 0 ns	Address setup to write pulse assertion

B.5 Read and Write Calculations

The read speed for the Bcache is determined as follows:

$$\text{Read} = \text{Tadr} + \text{Tacc}$$

The write speed for the Bcache is determined as follows:

$$\text{Write Pulse} = \text{Twp} + \text{skew}$$

B.6 Memory Initialization

The memory banks must be configured so that they are naturally aligned. For example, a bank configured with 32 MB must have a base address of zero or some multiple of 32 MB. Therefore, to ensure that all banks are contiguous (no gaps), the largest bank should be set to a base of zero. The larger of the two smaller banks should be set to the address immediately following the last location in the largest bank, and the last bank should be set to the address immediately following the last location of the middle bank.

If two of the banks are the same size, the bank with the lower bank number is assigned the lower address range.

This still holds true if the banks are the same size. There is no requirement for which bank must be the largest one. Therefore, the following algorithm is used to determine the base addresses of the banks:

```
Given banks a, b, and c
initialize base(a) = 0, base(b) = 0, base(c) = 0
if size(a) < size(b)
   base(a) = size(b)
if size(a) < size(c)
   base(a) = base(a) + size(c)
if size(b) ≤  size(a)
   base(b) = size(a)
if size(b) < size(c)
   base(b) = base(b) + size(c)
if size(c) ≤  size(a)
   base(c) = size(a)
if size(c) ≤  size(b)
   base(c) = base(c) + size(b)
```

Eight consecutive RAS cycles are performed for each memory bank to “wake up” the DRAMs, by reading from each bank eight times. The caches are disabled at this point so the read operations go directly to the DRAMs.

Good data ECC is ensured by writing all memory locations. This is done by rewriting the full contents of memory with the same data. Reading before writing memory lengthens the time to initialize data parity; however, it conserves the memory state for debugging purposes.

B.7 Bcache Initialization

To initialize the Bcache:

1. Temporarily configure the memory banks for the largest memory size supported by the memory controller (512 MB)[1].
2. Load the CAR to set timing and size and to disable ECC and parity.
3. Sweep through the Bcache with stores to initialize ECC and parity. There are no valid bits in the Bcache; therefore, it cannot be marked invalid. The sweep through writes the correct ECC and parity, even though they are disabled, to prevent errors.
4. Load the CAR to enable ECC and parity.

[1] 512 MB = 128 MB per bank × 4 banks

B.7 Bcache Initialization

When the system power is turned on, the Bcache contains unpredictable data in the tag RAMs. As the Bcache is swept for initialization, the old blocks (called the dirty victim blocks) are written back to main memory. These victim write operations occur based on the tags that store the upper part of the address location for the dirty blocks of memory. Because the tags are unpredictable, the victim write operations occur to unpredictable addresses in memory without regard to the actual system memory size. Therefore, configuring the banks for the fully supported memory avoids any illegal addresses to nonexistent memory. Accesses to nonexistent memory could cause the memory subsystem to fail to initialize all Bcache entries.

B.8 Special ROM Header

The MAKEROM tool is used to place a special header on ROM image files. The SROM allows the system flash ROM to contain two different ROM images, each with its own header. The header informs the SROM where to load the image, and whether or not it has been compressed with the MAKEROM tool. The header is optional for system ROMs containing a single image. If the header does not exist, the complete 1 MB system ROM is loaded and executed at physical address zero. Figure B–1 shows the header content.

Figure B–1 Special Header Content

31 24	23 16	15 8	7 0	
Validation Pattern 0x5A5AC3C3				0x00
Inverse Validation Pattern 0xA5A53C3C				0x04
Header Size (Bytes)				0x08
Image Checksum				0x0C
Image Size (Memory Footprint)				0x10
Decompression Flag				0x14
Destination Address Upper Longword				0x18
Destination Address Lower Longword				0x1C
Reserved		Firmware ID	Header Rev	0x20
ROM Image Size				0x24
Optional Firmware ID<31:00>				0x28
Optional Firmware ID<63:32>				0x2C
Header Checksum				0x30

MK230619A

Table B–4 describes each entry in the special header.

Table B–4 Special Header Entry Descriptions

Entry	Description
Validation and inverse validation pattern	This quadword contains a special signature pattern used to validate that the special ROM header has been located. The pattern is 0x5A5AC3C3A5A53C3C.
Header size (bytes)	This longword is provided to allow for some backward compatibility in the event that the header is extended in the future. When the header is located, current versions of SROM code determine where the image begins based on the header size. Additional data added to the header in the future will be ignored by current SROM code.

(continued on next page)

B.8 Special ROM Header

Table B–4 (Cont.) Special Header Entry Descriptions

<table>
<tr><th>Entry</th><th>Description</th></tr>
<tr><td>Image checksum</td><td>This longword is used to verify the integrity of the ROM.</td></tr>
<tr><td>Image size (memory footprint)</td><td>The image size is used by the SROM code to determine how much of the system ROM should be loaded.</td></tr>
<tr><td>Decompression flag</td><td>The decompression flag informs the SROM code whether the MAKEROM tool was used to compress the ROM image with a repeating byte algorithm. The SROM code contains routines that execute the decompression algorithm. Other compression and decompression schemes, which work independently from this scheme, may be employed.</td></tr>
<tr><td>Destination address</td><td>This quadword contains the destination address for the image. The SROM code will load the image at this address and begin execution.</td></tr>
<tr><td>Header rev</td><td>The revision of the header specifications used in this header. This is necessary to provide compatibility to future changes to this header specification. Version 0 headers are identified by the size of the header.</td></tr>
<tr><td>Firmware ID</td><td>The firmware ID is a byte that specifies the firmware type. This facilitates image boot options necessary to boot different operating systems.
<table>
<tr><th>Firmware Mnemonic</th><th>Firmware Type</th><th>Name</th></tr>
<tr><td>DBM</td><td>0</td><td>Alpha Microprocessors Evaluation Boards Debug Monitor</td></tr>
<tr><td>WNT</td><td>1</td><td>Windows NT Firmware</td></tr>
<tr><td>SRM</td><td>2</td><td>Alpha System Reference Manual Console</td></tr>
</table>
</td></tr>
<tr><td>ROM image size</td><td>The ROM image size reflects the size of the image as it is contained in the ROM.</td></tr>
<tr><td>Optional firmware ID</td><td>This is an optional field that can be used to provide additional firmware information such as firmware revision or a character descriptive string up to 8 characters.</td></tr>
<tr><td>Header checksum</td><td>The checksum of the header. This is used to validate the presence of a header beyond the validation provided by the validation pattern.</td></tr>
</table>

B.9 Icache Flush Code

The following code is loaded into memory after the system ROM image. It is then executed to flush the SROM initialization code from the Icache. The SROM initialization code is loaded into the Icache and maps to memory beginning at address zero.

```
77FF0055   mt     r31, flushIc
C0000001   br     r0,  +4
xxxxxxxx   .long  destination (found in r25)!addr filled at run time
6C008000   ldl_p  r0,  0(0)
47FF041F   bis    r31, 31, 31
47FF041F   bis    r31, 31, 31
47FF041F   bis    r31, 31, 31
47FF041F   bis    r31, 31, 31
47FF041F   bis    r31, 31, 31
47FF041F   bis    r31, 31, 31
6BE00000   jmp    r31, (r0)
```

In an attempt to transfer execution to the first page in memory, execution would just continue in the SROM initialization code at that address. Therefore, execution must be transferred to some address that does not hit in the Icache where other code can flush the Icache.

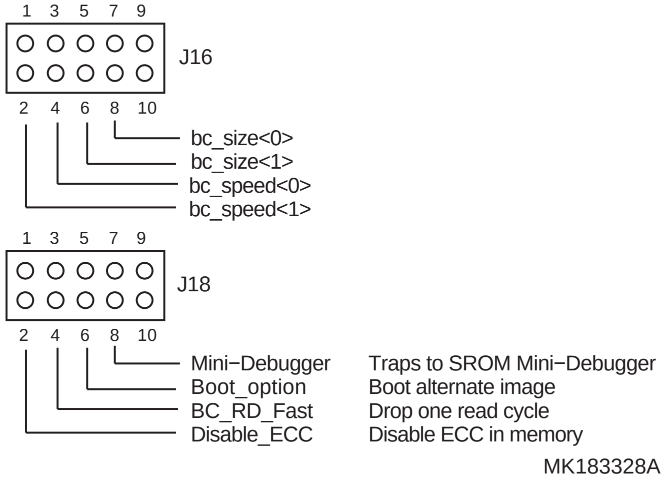
The NOPs following the Icache flush allow the instructions that were fetched before the Icache was updated, to be cleared from the pipeline. Execution will ultimately continue at the address contained in r25. At this point, r25 contains the starting address where the system ROM image was loaded into memory.

B.10 EB66+ Configuration Jumpers

The software configuration jumpers are completely programmable. The SROM code defines the software configuration jumpers as shown in Figure B–2.

B.10 EB66+ Configuration Jumpers

Figure B–2 Software Configuration Jumpers



Note that there are no connections to pins 9 and 10 on both J16 and J18.

Each jumper position is described in Table B–5.

Table B–5 Jumper Position Descriptions

Jumper Position	Description
BC_Size<1:0>	These jumpers force the Bcache to emulate a smaller Bcache. When both positions have jumpers in, the Bcache is disabled. These jumpers are changed in conjunction with the appropriate hardware jumpers as specified in Table 2–1.

BC_Size<1:0>	Bcache
In, in	Disabled
In, out	256 KB
Out, in	512 KB
Out, out	1 MB (default)

(continued on next page)

Table B–5 (Cont.) Jumper Position Descriptions

<table>
<tr><th>Jumper Position</th><th>Description</th></tr>
<tr><td>BC_Speed<1:0></td><td>These jumpers select different Bcache timing parameters used to compute the CAR value.
<table>
<tr><th>BC_Speed<1:0></th><th>Bcache Speed</th></tr>
<tr><td>In, in</td><td>6-ns RAM</td></tr>
<tr><td>In, out</td><td>8-ns RAM</td></tr>
<tr><td>Out, in</td><td>10-ns RAM (default)</td></tr>
<tr><td>Out, out</td><td>12-ns RAM</td></tr>
</table></td></tr>
<tr><td>Mini-Debugger</td><td>The SROM Mini-Debugger is provided in the SROM. This jumper (in) causes the SROM initialization to trap to the Mini-Debugger after all initialization is complete, but before starting the execution of the system ROM code.</td></tr>
<tr><td>Boot_option</td><td>When this jumper is in, an alternate image from the system flash ROM is selected. By default, the first image is loaded. The SROM allows the system flash ROM to contain multiple ROM images, each with its own header. The header informs the SROM where to load the image.</td></tr>
<tr><td>BC_RD_Fast</td><td>When in, a read speed setting of 1 cycle faster than nominal is forced.
<table>
<tr><th>BC_RD_Fast</th><th>Bcache Speed</th></tr>
<tr><td>In</td><td>Make read speed 1 cycle faster</td></tr>
<tr><td>Out</td><td>Nominal read speed (default)</td></tr>
</table></td></tr>
<tr><td>Disable_ECC</td><td>When in, the Disable_ECC jumper disables ECC in both the DRAM memory banks and the Bcache. Bcache error detection and correction is disabled for read operations that hit in the Bcache. For DRAM memory bank accesses, ECC is not checked or generated.</td></tr>
</table>

C

Technical Support and Ordering Information

Technical Support

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

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

Ordering Digital Semiconductor Products

To order the Alpha 21066A Microprocessor Evaluation Board, contact your local distributor.

You can order the following semiconductor products from Digital:

Product	Order Number
Alpha 21066 Microprocessor	21066–AA
Alpha 21068 Microprocessor	21068–AA
Alpha 21066A–233 Microprocessor	21066–AB
Alpha 21066A–100 Microprocessor	21066–CB
Alpha 21066A Evaluation Board	21A03–02

Ordering Associated Literature

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

Title	Order Number
Alpha Architecture Reference Manual[1]	EY–L520E–DP–YCH
Alpha 21066A Microprocessor Evaluation Board (EB66+) Product Brief	EC–QDVEA–TE
Alpha 21066A Microprocessor Product Brief	EC–QDV9B–TE
Alpha 21066, 21066A, and 21068 Microprocessors Hardware Reference Manual	EC–QC4GA–TE
Alpha 21066/21066A Microprocessors Data Sheet	EC–QC4HA–TE
Alpha Microprocessors Evaluation Board Debug Monitor User's Guide	EC–QHUVA-TE
Alpha SROM Mini-Debugger User's Guide	EC–QHUXA–TE

[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
82420/82430 PCIset ISA and EISA Bridges (includes 82378IB/ZB SIO) Order number: 290483	Intel Corporation Literature Sales P.O. Box 7641 Mt. Prospect, IL 60056 USA 1–800–628–8686 1–800–548–4725 FaxBACK® Service 1–800–628–2283 BBS 916–356–3600
UPI-41AH/42AH Universal Peripheral Interface 8-Bit Slave Microcontroller Order number: 210393	Intel Corporation (See previous entry.)
Memory Products Data Book Order number: 210830	Intel Corporation (See previous entry.)

Title	Vendor
PCI Local Bus Specification, Revision 2.0	PCI Special Interest Group N/S HH3–15A 5200 N.E. Elam Young Pkwy Hillsboro, Oregon 97124–6497 503–696–2000
PCI System Design Guide Order number: T159351	PCI Special Interest Group (See previous entry.)
PC87311/PC87312 (SuperI/O II/III) Floppy Disk Controller with Dual UARTs, Parallel Port, and IDE Interface Order number: 11362	National Semiconductor Corporation 2900 Semiconductor Drive P.O. Box 58090 Santa Clara, CA 95052 USA 1–800–272–9959

Glossary

This glossary provides definitions for terms and acronyms associated with the EB66+ and chips, specifically as applied to Alpha architecture.

cache memory

A small, high-speed memory placed between slower main memory and the processor. A cache increases effective memory transfer rates and processor speed. It contains copies of data recently used by the processor and fetches several bytes of data from memory, anticipating that the processor will access the next sequential series of bytes. The 21066A microprocessor contains two onchip internal caches; one for instructions, and one for data. *See also* write-back cache.

CAS

Column address strobe.

DRAM

Dynamic random-access memory. Read/write memory that must be refreshed (read from or written to) periodically to maintain the storage of information.

EB66+

An evaluation board. A hardware/software applications development platform for the low-cost Alpha program and a debugging platform for the 21066A microprocessor.

ISA

Industry Standard Architecture. An 8-/16-bit interface for interconnecting data storage, data processing, and peripheral control devices in a closely coupled configuration.

local bus

A bus that is in close proximity to the CPU and shares its speed. PCI is a local bus.

NVRAM

Nonvolatile random-access memory.

PAL

Programmable array logic.

PCI

Peripheral component interconnect. The 32-bit local bus developed by Intel.

PGA

Pin grid array.

PLA

Programmable logic array.

PLD

Programmable logic device.

PLL

Phase-locked loop.

primary cache

The cache that is the fastest and closest to the processor. The 21066A microprocessor contains an instruction cache and a data cache.

PROM

Programmable read-only memory.

RAM

Random-access memory.

RAS

Row address strobe.

region

One of four areas in physical memory space based on the two most significant, implemented, physical address bits.

RISC

Reduced instruction set computer. A computer with an instruction set that is paired down and reduced in complexity so that most instructions can be performed in a single processor cycle. High-level compilers synthesize the more complex, least frequently used instructions by breaking them down into simpler instructions. This approach allows the RISC architecture to implement a small, hardware-assisted instruction set, thus eliminating the need for microcode.

SCSI

Small computer system interface. A standard interface for peripheral devices in which the device contains most of the controller circuitry, leaving the interface free to communicate with other peripherals.

SRAM

Static random-access memory.

SROM

Serial read-only memory.

UVPROM

Ultraviolet (erasable) programmable read-only memory.

VRAM

Video random-access memory.

write-back cache

A cache in which copies are kept of any data in the region. Read and write transactions may use the copies, and write transactions use additional states to determine whether there are other copies to invalidate or update.

Index