AlphaPC 164 Motherboard DIGITAL UNIX

User's Manual

Revision/Update Information: This manual supersedes the *AlphaPC 164 Motherboard DIGITAL UNIX User's Manual*, Revision A.

January 1997

Printed in U.S.A.

EC–QZT5B–TE

Contents

Figures

Tables

1 About This Manual

This manual describes the AlphaPC 164 motherboard, a module for computing systems based on the Digital Semiconductor 21164 Alpha microprocessor and the companion Digital Semiconductor 21172 core logic chipset. It describes the features of the motherboard and how to set the configuration jumpers. The manual is intended for users of the AlphaPC 164 motherboard to assist them in installing the motherboard and populating it with memory modules and peripheral cards.

1.1 Document Conventions

The following conventions are used in this document.

Caution: Cautions indicate potential damage to equipment, software, or data.

Note: Notes provide additional information about a topic.

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} and 19A are hexadecimal numbers.

Extents: 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. Multiple bit fields are shown as extents.

Register Figures: Register figures have bit and field position numbering starting at the right (low-order) and increasing to the left (high-order).

Signal Names: All signal names are printed in **boldface** type. Signals whose names originate in an industry-standard specification, such as PCI or IDE, are printed in the case used in the specification (usually uppercase). Active low signals have either a pound sign "#" appended, or a "not" overscore bar (for example; **DEVSEL#** and **$\overline{\text{RESET}}$**).

Italic Type: *Italic* type emphasizes important information and indicates complete titles of documents.

Terms: The following terms are used in this document:

This term...	Refers to...
Alpha SRM Console	The Alpha SRM Console firmware.
DIGITAL UNIX installation guide	The *DIGITAL UNIX Installation Guide*.
DIGITAL UNIX	The DIGITAL UNIX operating system.

2 Features of the AlphaPC 164 Motherboard

This section lists the AlphaPC 164 motherboard features and shows the location of major components, connectors, and jumpers.

AlphaPC 164 Features

The AlphaPC 164 motherboard uses a Digital Semiconductor 21164 Alpha microprocessor and companion Digital Semiconductor 21172 core logic chipset.

Table 1 lists the features of the AlphaPC 164 motherboard. Figure 1 shows the board outlines, and identifies the location of jumpers, connectors, and major components. Table 2 lists and defines these items. Refer to Section 3 for jumper configurations. Refer to Section 4 for connector pinouts.

Table 1 AlphaPC 164 Features

Feature	Description
Microprocessor	Digital Semiconductor 21164 Alpha microprocessor (64-bit RISC)
Chipset	Digital Semiconductor 21172 core logic chipset, consisting of one control chip and four data chips, that provides an interface to system memory and the PCI bus
DRAM memory	16MB to 512MB memory array -- One bank of either 4 (128-bit) or 8 (256-bit) commodity, 36-bit, 70-ns SIMMs
Caching	
L1 Icache	8KB, direct-mapped, instruction cache on the CPU chip
L1 Dcache	8KB, direct-mapped, data cache on the CPU chip
L2 Scache	96KB, three-way, set-associative, write-back, unified instruction and data cache on the CPU chip
L3 backup cache	Onboard 1MB, direct-mapped, synchronous SRAM backup cache with 128-bit data path
I/O and miscellaneous support	32-bit and 64-bit, 33-MHz PCI
	Four dedicated PCI expansion slots (two 64-bit)
	PCI/IDE control (CMD646)
	Intel 82378ZB PCI-to-ISA bridge chip
	Two dedicated ISA expansion slots
	SMC FDC37C935 combination chip provides control for diskettes, two UARTs with modem control, parallel port, keyboard, mouse, and time-of-year clock
	1MB flash ROM
Firmware	Alpha SRM Console firmware

Figure 1 AlphaPC 164 Jumper/Connector/Component Location

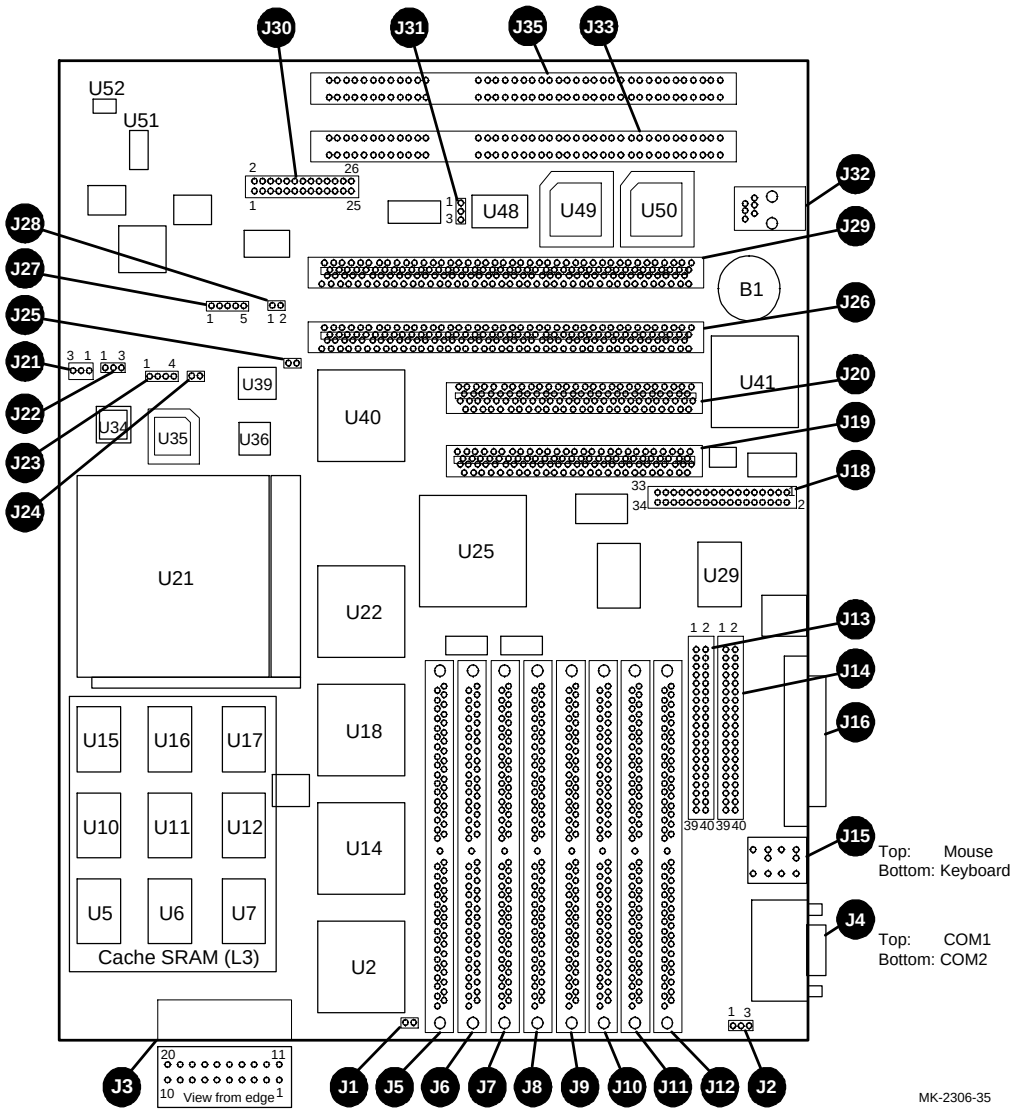


Table 2 AlphaPC 164 Jumper/Connector/Component List

Item Number	Description	Item Number	Description
B1	RTC battery (CR2032)	J1	Memory bus width jumper
J2	Fan power, enclosure (+12V)	J3	Power (+3V, +5V, -5V, +12V, -12V)
J4	COM1/COM2 (DB9) connectors	J5	DRAM SIMM 0 [35:0] connector
J6	DRAM SIMM 1 [71:36] connector	J7	DRAM SIMM 2 [107:72] connector
J8	DRAM SIMM 3 [143:108] connector	J9	DRAM SIMM 4 [179:144] connector
J10	DRAM SIMM 5 [215:180] connector	J11	DRAM SIMM 6 [251:216] connector
J12	DRAM SIMM 7 [287:252] connector	J13	IDE drive 2/3 connector
J14	IDE drive 0/1 connector	J15	Keyboard/mouse connectors
J16	Parallel I/O connector	J18	Diskette (floppy) drive connector
J19	PCI slot 3 (32-bit)	J20	PCI slot 2 (32-bit)
J21	Microprocessor fan/fan sense connector	J22	Enclosure fan +12V power connector
J23	Speaker connector	J24	Reset button connector
J25	Halt button connector	J26	PCI slot 1 (64-bit)
J27	Power LED connector	J28	Hard-drive LED connector
J29	PCI slot 0 (64-bit)	J30	Configuration jumpers
J31	Flash update enable/disable jumper	J32	SROM test port connector
J33	ISA slot 1	J35	ISA slot 0
U2	Data switch 0 (DSC 21172-BA)	U5 to U7	Cache SRAM (L3)
U10 to U12	Cache SRAM (L3)	U14	Data switch 1 (DSC 21172-BA)
U15 to U17	Cache SRAM (L3)	U18	Data switch 2 (DSC 21172-BA)
U21	Microprocessor, socketed (DSC 21164 Alpha)	U22	Data switch 3 (DSC 21172-BA)
U25	I/O interface and address control (DSC 21172-CA)	U29	IDE controller
U34	Microprocessor clock crystal, 36.66-MHz (default), socketed	U35	Microprocessor clock PLL (TriQuint TQ2061)
U36	System clock PLL (CDC 2586)	U39	Serial ROM, socketed (Xilinx XC17128D)
U40	PCI-to-ISA bridge (Intel 82378ZB)	U41	Combination controller, Super I/O (SMC FDC37C935)
U48	Flash ROM (1MB)	U49	PCI arbiter PAL
U50	PCI interrupt request PAL	U51	Power controller
U52	Power sense	—	—

2.1 Power Requirements

The AlphaPC 164 motherboard has a total power dissipation of 116 W, excluding any plug-in PCI and ISA devices. Table 3 lists the power requirement for each dc supply voltage.

The power supply must supply a **DCOK** signal to the system reset logic.

Table 3 Power Supply DC Current Requirements

Voltage/Tolerance	Current[1]
+3.3 V dc, ±5%	5.0 A
+5 V dc, ±5%	12.0 A
–5 V dc, ±5%	0 A
+12 V dc, ±5%	1.0 A
–12 V dc, ±5%	100.0 mA

[1] Values indicated are for an AlphaPC 164 motherboard (64MB DRAM) excluding adapter cards and disk drives.

Caution: **Fan sensor required**. The 21164 microprocessor cooling fan *must* have a built-in sensor that will drive a signal if the airflow stops. The sensor is connected to motherboard connector J21. When the signal is generated, it resets the system.

2.2 Environmental Requirements

The 21164 microprocessor is cooled by a small fan blowing directly into the chip's heat sink. The AlphaPC 164 motherboard is designed to run efficiently by using only this fan. Additional fans may be necessary depending upon cabinetry and the requirements of plug-in cards.

The AlphaPC 164 motherboard is specified to run within the environment listed in Table 4.

Table 4 AlphaPC 164 Motherboard Environmental Requirements

Parameter	Specification
Operating temperature	10°C to 40°C (50°F to 104°F)
Storage temperature	–55°C to 125°C (–67°F to 257°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 ±2°C/hour (20°F/hour ±4°F/hour)

2.3 Physical Parameters

The AlphaPC 164 motherboard is an ATX-size printed-wiring board (PWB) with the following dimensions:

- Length: 30.48 cm (12.0 in. ±0.0005 in.)
- Width: 24.38 cm (9.6 in. ±0.0005 in.)
- Height: 6.0 cm (2.4 in.)

The motherboard can be used in certain desktop and deskside systems that have adequate clearance for the 21164 microprocessor heat sink and fan. All ISA and PCI expansion slots are usable in standard desktop or deskside enclosures.

3 AlphaPC 164 Jumper Configuration

The AlphaPC 164 motherboard has three groups of jumpers at location J1, J30, and J31. These jumpers set the hardware configuration and boot options. Figure 1 shows the jumper location on the AlphaPC 164 motherboard. Figure 2 shows the jumper functions for each group. Section 3.1 through Section 3.7 describe the jumper configurations.

3.1 Memory Bus Width Jumper (J1)

The memory bus width can be either 128 bits (J5 through J8 populated with SIMMs and J9 through J12 empty) or 256 bits (J5 through J12 populated with SIMMs). When using a memory bus width of 128 bits, jumper J1 must be in. When using a memory bus width of 256 bits, jumper J1 must be out.

3.2 System Clock Divisor Jumpers (IRQ3 Through IRQ0)

The system clock divisor jumpers are located at J30–1/2 (IRQ3), J30–3/4 (IRQ2), J30–5/6 (IRQ1), and J30–7/8 (IRQ0). The jumper configuration set in IRQ3 through IRQ0 determines the frequency of the microprocessor's system clock output. These four jumpers set the speed at power-up as listed in Figure 2. The microprocessor frequency divided by the ratio determines the system clock frequency.

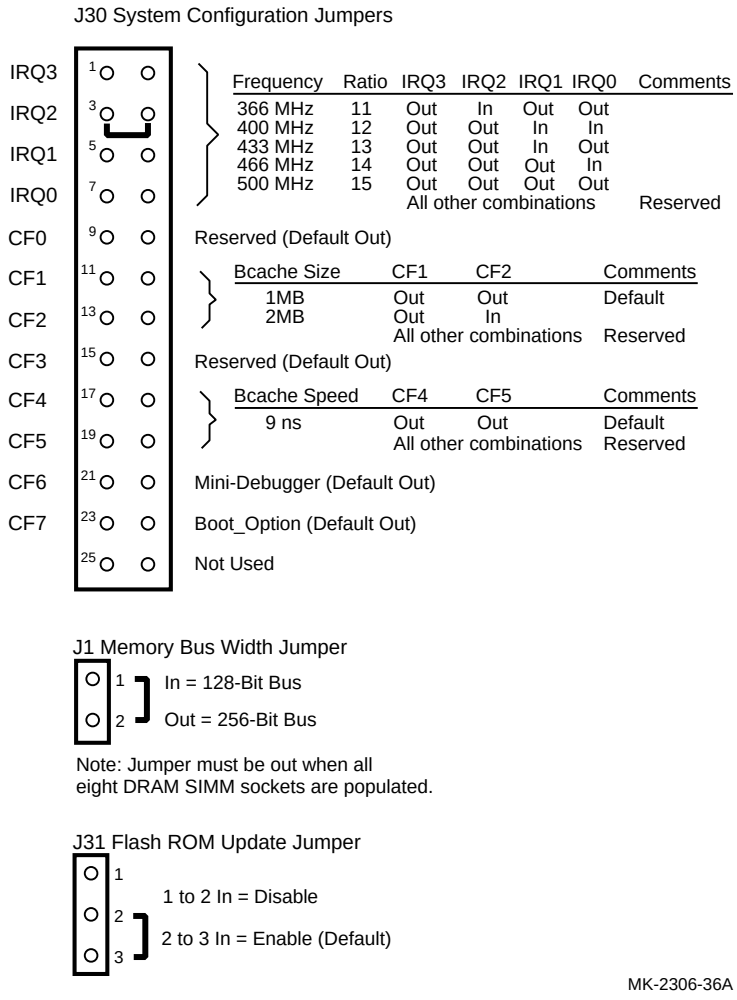
3.3 Bcache Size Jumpers (CF1 and CF2)

The Bcache size jumpers are located at J30–11/12, CF1 and J30–13/14, CF2. These jumpers configure the Bcache as specified in Figure 2.

3.4 Bcache Speed Jumpers (CF4 and CF5)

The Bcache speed jumpers are located at J30–17/18, CF4 and J30–19/20, CF5. These jumpers select the Bcache timing parameters used to compute a value that is loaded into the microprocessor's Bcache configuration register at power-up time. Because the Bcache SRAMs are soldered onto the board, the default jumper configuration selecting an SRAM access time of 9 ns as shown in Figure 2 will *always* be used.

Figure 2 AlphaPC 164 Configuration Jumpers



3.5 Mini-Debugger Jumper (CF6)

The Mini-Debugger jumper is located at J30–21/22 (CF6). The default position for this jumper is out (Figure 2). The Alpha SROM Mini-Debugger is stored in the SROM. When this jumper is in, it causes the SROM initialization to trap to the Mini-Debugger (communication through connector J32) after all initialization is complete, but before starting the execution of the system flash ROM code.

3.6 Boot Option Jumper (CF7)

The boot option jumper is located at J30–23/24 (CF7). The default position for this jumper is out (Figure 2). This jumper selects the image to be loaded into memory from the system flash ROM. With the jumper out, the Alpha SRM Console firmware is loaded. With the jumper in, the fail-safe booter is loaded. For more information about the fail-safe booter, refer to Section 7.4.3.

3.7 Flash ROM Update Jumper (J31)

When J31–2/3 are jumpered together (default), the flash ROM is write-enabled. When J31–1/2 are jumpered together, the flash ROM is write-protected.

4 AlphaPC 164 Connector Pinouts

This section lists the pinouts of all connectors (see Table 5 through Table 21). See Figure 1 for connector locations.

Table 5 Peripheral Component Interface (PCI) Bus Connector Pinouts

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
32-Bit and 64-Bit PCI Connectors (J19, J20, J26, J29)							
A1	**TRST#**	A2	**+12V**	A3	**TMS**	A4	**TDI**
A5	**Vdd**	A6	**INTA**	A7	**INTC**	A8	**Vdd**
A9	—	A10	**Vdd**	A11	—	A12	**Gnd**
A13	**Gnd**	A14	—	A15	**RST#**	A16	**Vdd**
A17	**GNT#**	A18	**Gnd**	A19	—	A20	**AD[30]**
A21	**+3V**	A22	**AD[28]**	A23	**AD[26]**	A24	**Gnd**
A25	**AD[24]**	A26	**IDSEL**	A27	**+3V**	A28	**AD[22]**
A29	**AD[20]**	A30	**Gnd**	A31	**AD[18]**	A32	**AD[16]**
A33	**+3V**	A34	**FRAME#**	A35	**Gnd**	A36	**TRDY#**
A37	**STOP#**	A38	**STOP#**	A39	**+3V**	A40	**SDONE**
A41	**SBO#**	A42	**Gnd**	A43	**PAR**	A44	**AD[15]**
A45	**+3V**	A46	**AD[13]**	A47	**AD[11]**	A48	**Gnd**
A49	**AD[09]**	A50	Not used	A51	Not used	A52	**C/BE#[0]**
A53	**+3V**	A54	**AD[06]**	A55	**AD[04]**	A56	**Gnd**
A57	**AD[02]**	A58	**AD[00]**	A59	**Vdd**	A60	**REQ64#**
A61	**Vdd**	A62	**Vdd**	B1	**-12V**	B2	**TCK**
B3	**Gnd**	B4	**TDO**	B5	**Vdd**	B6	**Vdd**
B7	**INTB**	B8	**INTD**	B9	**PRSNT1#**	B10	—
B11	**PRSNT2#**	B12	**Gnd**	B13	**Gnd**	B14	—
B15	**Gnd**	B16	**CLK**	B17	**Gnd**	B18	**REQ#**
B19	**Vdd**	B20	**AD[31]**	B21	**AD[29]**	B22	**Gnd**
B23	**AD[27]**	B24	**AD[25]**	B25	**+3V**	B26	**C/BE#[3]**
B27	**AD[23]**	B28	**Gnd**	B29	**AD[21]**	B30	**AD[19]**
B31	**+3V**	B32	**AD[17]**	B33	**C/BE#[2]**	B34	**Gnd**
B35	**IRDY#**	B36	**+3V**	B37	**DEVSEL#**	B38	**Gnd**
B39	**LOCK#**	B40	**PERR#**	B41	**+3V**	B42	**SERR#**
B43	**+3V**	B44	**C/BE#[1]**	B45	**AD[14]**	B46	**Gnd**
B47	**AD[12]**	B48	**AD[10]**	B49	**Gnd**	B50	Not used
B51	Not used	B52	**AD[08]**	B53	**AD[07]**	B54	**+3V**
B55	**AD[05]**	B56	**AD[03]**	B57	**Gnd**	B58	**AD[01]**
B59	**Vdd**	B60	**ACK64#**	B61	**Vdd**	B62	**Vdd**

Table 5 (Continued) Peripheral Component Interface (PCI) Bus Connector Pinouts

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
64-Bit PCI Connectors Only (J26, J29)							
A63	**Gnd**	A64	**C/BE#[7]**	A65	**C/BE#[5]**	A66	**Vdd**
A67	**PAR64**	A68	**D[62]**	A69	**Gnd**	A70	**D[60]**
A71	**D[58]**	A72	**Gnd**	A73	**D[56]**	A74	**D[54]**
A75	**Vdd**	A76	**D[52]**	A77	**D[50]**	A78	**Gnd**
A79	**D[48]**	A80	**D[46]**	A81	**Gnd**	A82	**D[44]**
A83	**D[42]**	A84	**Vdd**	A85	**D[40]**	A86	**D[38]**
A87	**Gnd**	A88	**D[36]**	A89	**D[34]**	A90	**Gnd**
A91	**D[32]**	A92	—	A93	**Gnd**	A94	—
B63	—	B64	**Gnd**	B65	**C/BE#[6]**	B66	**C/BE#[4]**
B67	**Gnd**	B68	**D[63]**	B69	**D[61]**	B70	**Vdd**
B71	**D[59]**	B72	**D[57]**	B73	**Gnd**	B74	**D[55]**
B75	**D[53]**	B76	**Gnd**	B77	**D[51]**	B78	**D[49]**
B79	**Vdd**	B80	**D[47]**	B81	**D[45]**	B82	**Gnd**
B83	**D[43]**	B84	**D[41]**	B85	**Gnd**	B86	**D[39]**
B87	**D[37]**	B88	**Vdd**	B89	**D[35]**	B90	**D[33]**
B91	**Gnd**	B92	—	B93	—	B94	**Gnd**

Table 6 ISA Expansion Bus Connector Pinouts (J33, J35)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	**Gnd**	2	**IOCHCK#**	3	**RSTDRV**	4	**SD7**
5	**Vdd**	6	**SD6**	7	**IRQ9**	8	**SD5**
9	**–5V**	10	**SD4**	11	**DRQ2**	12	**SD3**
13	**–12V**	14	**SD2**	15	**ZEROWS#**	16	**SD1**
17	**+12V**	18	**SD0**	19	**Gnd**	20	**IOCHRDY**
21	**SMEMW#**	22	**AEN**	23	**SMEMR#**	24	**SA19**
25	**IOW#**	26	**SA18**	27	**IOR#**	28	**SA17**
29	**DACK3#**	30	**SA16**	31	**DRQ3**	32	**SA15**
33	**DACK1#**	34	**SA14**	35	**DRQ1**	36	**SA13**
37	**REFRESH#**	38	**SA12**	39	**SYSCLK**	40	**SA11**
41	**IRQ7**	42	**SA10**	43	**IRQ6**	44	**SA9**
45	**IRQ5**	46	**SA8**	47	**IRQ4**	48	**SA7**
49	**IRQ3**	50	**SA6**	51	**DACK2#**	52	**SA5**
53	**TC**	54	**SA4**	55	**BALE**	56	**SA3**
57	**Vdd**	58	**SA2**	59	**OSC**	60	**SA1**
61	**Gnd**	62	**SA0**	63	**MEMCS16#**	64	**SBHE#**
65	**IOCS16#**	66	**LA23**	67	**IRQ10**	68	**LA22**
69	**IRQ11**	70	**LA21**	71	**IRQ12**	72	**LA20**
73	**IRQ15**	74	**LA19**	75	**IRQ14**	76	**LA18**
77	**DACK0#**	78	**LA17**	79	**DRQ0**	80	**MEMR#**
81	**DACK5#**	82	**MEMW#**	83	**DRQ5**	84	**SD8**
85	**DACK6#**	86	**SD9**	87	**DRQ6**	88	**SD10**
89	**DACK7#**	90	**SD11**	91	**DRQ7**	92	**SD12**
93	**Vdd**	94	**SD13**	95	**MASTER#**	96	**SD14**
97	**Gnd**	98	**SD15**	—	—	—	—

Table 7 DRAM SIMM Connector Pinouts (J5 Through J12)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	**Gnd**	2	**DQ1**	3	**DQ2**	4	**DQ3**
5	**DQ4**	6	**DQ5**	7	**DQ6**	8	**DQ7**
9	**DQ8**	10	**Vdd**	11	**Gnd**	12	**A0**
13	**A1**	14	**A2**	15	**A3**	16	**A4**
17	**A5**	18	**A6**	19	**A10**	20	**DQ9**
21	**DQ10**	22	**DQ11**	23	**DQ12**	24	**DQ13**
25	**DQ14**	26	**DQ15**	27	**DQ16**	28	**A7**
29	**A11**	30	**Vdd**	31	**A8**	32	**A9**
33	**RAS3**	34	**RAS2**	35	**DQ17**	36	**DQ18**
37	**DQ19**	38	**DQ20**	39	**Gnd**	40	**CAS0**
41	**CAS2**	42	**CAS3**	43	**CAS1**	44	**RAS0**
45	**RAS1**	46	**Vdd**	47	**WE**	48	NC
49	**DQ21**	50	**DQ22**	51	**DQ23**	52	**DQ24**
53	**DQ25**	54	**DQ26**	55	**DQ27**	56	**DQ28**
57	**DQ29**	58	**DQ30**	59	**Vdd**	60	**DQ31**
61	**DQ32**	62	**DQ33**	63	**DQ34**	64	**DQ35**
65	**DQ36**	66	**Vdd**	67	NC	68	NC
69	NC	70	NC	71	**Gnd**	72	**Gnd**

Table 8 IDE Drive Bus Connector Pinouts (J13, J14)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	**$\overline{\text{RESET}}$**	2	**Gnd**	3	**IDE_D7**	4	**IDE_D8**
5	**IDE_D6**	6	**IDE_D9**	7	**IDE_D5**	8	**IDE_D10**
9	**IDE_D4**	10	**IDE_D11**	11	**IDE_D3**	12	**IDE_D12**
13	**IDE_D2**	14	**IDE_D13**	15	**IDE_D1**	16	**IDE_D14**
17	**IDE_D0**	18	**IDE_D15**	19	**Gnd**	20	NC (key pin)
21	**MARQ**	22	**Gnd**	23	**$\overline{\text{IOW}}$**	24	**Gnd**
25	**$\overline{\text{IOR}}$**	26	**Gnd**	27	**CHRDY**	28	**BALE**
29	**MACK**	30	**Gnd**	31	**IRQ**	32	**$\overline{\text{IOCS16}}$**
33	**ADDR1**	34	**NC**	35	**ADDR0**	36	**ADDR2**
37	**$\overline{\text{CS0}}$**	38	**$\overline{\text{CS1}}$**	39	**$\overline{\text{ACT}}$**	40	**Gnd**

Table 9 Diskette Drive Bus Connector Pinouts (J18)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	**Gnd**	2	**DEN0**	3	**Gnd**	4	NC
5	**Gnd**	6	**DEN1**	7	**Gnd**	8	**INDEX**
9	**Gnd**	10	**MTR0**	11	**Gnd**	12	**DR1**
13	**Gnd**	14	**DR0**	15	**Gnd**	16	**MTR1**
17	**Gnd**	18	**DIR**	19	**Gnd**	20	**STEP**
21	**Gnd**	22	**WDATA**	23	**Gnd**	24	**WGATE**
25	**Gnd**	26	**TRK0**	27	**Gnd**	28	**WRTPRT**
29	**ID0**	30	**RDATA**	31	**Gnd**	32	**HDSEL**
33	**ID1**	34	**DSKCHG**	—	—	—	—

Table 10 Parallel Bus Connector Pinouts (J16)

Pin	Signal	Pin	Signal	Pin	Signal	Pin	Signal
1	**STB**	2	**PD0**	3	**PD1**	4	**PD2**
5	**PD3**	6	**PD4**	7	**PD5**	8	**PD6**
9	**PD7**	10	**ACK**	11	**BUSY**	12	**PE**
13	**SLCT**	14	**AFD**	15	**ERR**	16	**INIT**
17	**SLIN**	18	**Gnd**	19	**Gnd**	20	**Gnd**
21	**Gnd**	22	**Gnd**	23	**Gnd**	24	**Gnd**
25	**Gnd**	—	—	—	—	—	—

Table 11 COM1/COM2 Serial Line Connector Pinouts (J4)

COM1 Pin (Top)	COM1 Signal	COM2 Pin (Bottom)	COM2 Signal
1	**DCD1**	1	**DCD2**
2	**RxD1**	2	**RxD2**
3	**TxD1**	3	**TxD2**
4	**DTR1**	4	**DTR2**
5	**SG1**	5	**SG2**
6	**DSR1**	6	**DSR2**
7	**RTS1**	7	**RTS2**
8	**CTS1**	8	**CTS2**
9	**RI1**	9	**RI2**

Table 12 Keyboard/Mouse Connector Pinouts (J15)

Keyboard Pin (Top)	Keyboard Signal	Mouse Pin (Bottom)	Mouse Signal
1	**KBDATA**	1	**MSDATA**
2	NC	2	NC
3	**Gnd**	3	**Gnd**
4	**Vdd**	4	**Vdd**
5	**KBCLK**	5	**MSCLK**
6	NC	6	NC

Table 13 SROM Test Data Input Connector Pinouts (J32)

Pin	Signal	Name
1	NC	—
2	**SROM_CLK_L**	Clock out
3	**Gnd**	—
4	NC	—
5	**TEST_SROM_D_L**	SROM serial data in
6	NC	—

Table 14 Input Power Connector Pinouts (J3)

Pin	Voltage	Pin	Voltage	Pin	Voltage	Pin	Voltage
1	+3.3 V dc	2	+3.3 V dc	3	Ground	4	+5 V dc
5	Ground	6	+5 V dc	7	Ground	8	**P_DCOK**
9	NC	10	+12 V dc	11	+3.3 V dc	12	–12 V dc
13	Ground	14	NC	15	Ground	16	Ground
17	Ground	18	–5 V dc	19	+5 V dc	20	+5 V dc

Table 15 Enclosure Fan (+12 V dc) Power Connector Pinouts (J2, J22)

Pin	Voltage
1	Ground
2	+12 V dc
3	Ground

Table 16 Speaker Connector Pinouts (J23)

Pin	Signal	Name
1	**SPKR**	Speaker output
2	**Gnd**	—
3	**Gnd**	—
4	**Gnd**	—

Table 17 Microprocessor Fan Power Connector Pinouts (J21)

Pin	Signal	Name
1	**+12V**	—
2	**FAN_CONN_L**	Fan connected
3	**Gnd**	—

Table 18 Power LED Connector Pinouts (J27)

Pin	Signal	Name
1	**POWER_LED_L**	Pull-up to **Vdd**
2	**Gnd**	—
3	NC	—
4	NC	—
5	NC	—

Table 19 IDE Drive LED Connector Pinouts (J28)

Pin	Signal	Name
1	**HD_ACT_L**	Hard drive active
2	**HD_LED_L**	Pull-up to **Vdd**

Table 20 Reset Button Connector Pinouts (J24)

Pin	Signal	Name
1	**RESET_BUTTON**	Reset system
2	**Gnd**	—

Table 21 Halt Button Connector Pinouts (J25)

Pin	Signal	Name
1	**HALT_BUTTON**	Halt system
2	**Gnd**	—

5 Configuring and Upgrading DRAM Memory

For higher system speed or greater throughput, DRAM memory can be upgraded either by replacing SIMMs with those of greater size, or by widening the memory bus from 128 bits to 256 bits by adding more SIMMs.

Note: When configuring or upgrading DRAM, the following rules must be observed:

- All SIMMs must be 36-bit and have a 70-ns or faster access time.
- All SIMMs must be of equal size.

5.1 Configuring DRAM Memory

Table 22 lists the DRAM memory configurations available. Refer to Figure 1 for SIMM connector location.

Table 22 AlphaPC 164 DRAM Memory Configurations

Total Memory	128-Bit Memory Mode (J1 In) J5 Through J8 Populated with SIMM Sizes...
16MB	1Mb x 36
32MB	2Mb X 36
64MB	4Mb x 36
128MB	8Mb X 36
256MB	16Mb X 36
Total Memory	**256-Bit Memory Mode (J1 Out) J5 Through J12 Populated with SIMM Sizes...**
32MB	1Mb X 36
64MB	2Mb X 36
128MB	4Mb X 36
256MB	8Mb X 36
512MB	16Mb X 36

5.2 Upgrading DRAM Memory

There are three options for upgrading DRAM memory (Table 23).

Table 23 Memory Upgrade Options

Option	Memory Bus Width Before	Memory Bus Width After	Upgrade Possibilities
1	128-bit	128-bit	Replace the 4 SIMMs in sockets J5 through J8 with SIMMs of greater size, thus retaining the 128-bit memory bus width.
2	128-bit	256-bit	Add 4 SIMMs in sockets J9 through J12 with sizes equal to those in sockets J5 through J8, thus widening the memory bus width to 256 bits.
3	256-bit	256-bit	Replace the 8 SIMMs in sockets J5 through J12 with SIMMs of greater size.

To widen the memory bus to its 256-bit maximum (upgrade option 2), add four SIMMs and make a jumper change (remove J1). The SIMMs that you add must be of the same size (*n*Mb X 36-bit) and have an access time equal to or less than the four SIMMs already in the system. Refer to Figure 1 for SIMM connector and jumper location.

1. *Observe antistatic precautions.* Handle SIMMs only at the edges to prevent damage.
2. Remove power from the system.
3. Hold the SIMM at an angle with the notch facing the key in the socket.
4. Firmly push the module into the connector and stand the module upright. Ensure that the SIMM snaps into the metal locking clips on both ends.
5. For 128-bit memory bus width, jumper J1 must be in. For 256-bit memory bus width, jumper J1 must be out.
6. Restore power to the system.

6 Interrupts and ISA Bus Addresses

This section lists the system and I/O interrupt assignments. It also lists the physical AlphaPC 164 I/O space assignments.

6.1 Interrupts

Table 24 lists each AlphaPC 164 ISA interrupt and its source.

Table 24 ISA Interrupts

Interrupt Number	Interrupt Source
IRQ0	Internal timer 1
IRQ1	Keyboard
IRQ2	Interrupt from controller 2
IRQ3	COM2
IRQ4	COM1
IRQ5	Available
IRQ6	Diskette
IRQ7	Parallel port
IRQ8#[1]	Reserved
IRQ9	Available
IRQ10	Available
IRQ11	Available
IRQ12	Mouse
IRQ13	Available
IRQ14	IDE
IRQ15	IDE

[1] The # symbol indicates an active low signal.

6.2 ISA I/O Address Map

Table 25 lists the AlphaPC 164 ISA I/O space address mapping.

Table 25 ISA I/O Address Map

Range (hex)	Usage
000-00F	8237 DMA #1
020-021	8259 PIC #1
040-043	8253 timer
060-061	Ubus IRQ12 and NMI control
070	CMOS RAM address and NMI mask register
080-08F	DMA page registers
0A0-0A1	8259 PIC #2
0C0-0DF	8237 DMA #2
2F8-2FF	Serial port—COM2
370-377	Secondary diskette
3BC-3BF	Parallel port—LPT1
3F0-3F7	Primary diskette
3F8-3FF	Serial port—COM1
800	FLASH_ADR19 register
801	AlphaPC 164 configuration register
804-806	PCI interrupt registers

6.3 Flash ROM Address Map

The address range for the flash ROM is FFF8.0000–FFFF.FFFF. Flash space of 1MB is obtained by double mapping this 512KB space. FLASH_ADR19 register at I/O location 800h provides this function. Writing a 0 to this location enables the lower 512KB of flash. Writing a 1 to this location enables the upper 512KB of flash.

7 Alpha SRM Console Firmware

The Alpha SRM Console firmware initializes the system and enables you to install and boot the DIGITAL UNIX operating system. This firmware resides in the flash ROM on the AlphaPC 164 motherboard.

7.1 Alpha SRM Console Firmware Conventions

The following conventions are used in this section:

Convention	Description
>>>	Alpha SRM Console prompt.
Backslash (\) at the end of a line	Continuation symbol to continue long commands on the next line.
_>	Continuation line prompt.
Maximum command length	255 characters.
Multiple contiguous spaces or tabs	Treated as a single space.
Command abbreviations	Allowed, if not ambiguous.
Command qualifiers or options	Prefix with a space and a dash (-).
Numbers	Hexadecimal, unless otherwise specified. (Registers, such as R0–R31, are shown in decimal notation.)

The following table lists Alpha SRM Console special keys and their functions. These special keys, also referred to as shortcut keys, provide command recall, line editing, and basic input/output control flow.

Shortcut Key	Function
`Enter`	Terminate the command line input.
`Backspace or Delete`	Delete one character to the left of the cursor.
`Ctrl/A`	Toggles insert/overstrike mode. (Overstrike is the default.)
`Ctrl/B` `Up arrow` `Down arrow`	Recall previous commands. (The last 16 commands are stored.)
`Ctrl/C`	Terminate the foreground process.
`Ctrl/D` `Left arrow`	Move the cursor one position to the left.
`Ctrl/E`	Move the cursor to the end of the line.
`Ctrl/F` `Right arrow`	Move the cursor one position to the right.
`Ctrl/H`	Move the cursor to the beginning of the line.
`Ctrl/O`	Suppress or resume (toggle) console output.
`Ctrl/Q`	Resume the flow (XON) of data to the console.
`Ctrl/R`	Retype the current command line.
`Ctrl/S`	Stop the flow (XOFF) of data to the console.
`Ctrl/U`	Delete the entire line.

7.2 Basic Alpha SRM Console Command Descriptions

This section describes the following basic Alpha SRM Console commands that are necessary to boot the DIGITAL UNIX operating system:

- arc
- boot
- deposit
- examine
- fwupdate
- set
- show

The Alpha SRM Console offers additional commands. For a complete list of Alpha SRM Console commands, enter **`help`** at the Alpha SRM Console prompt (>>>).

arc

Loads and runs the Windows NT ARC firmware from a diskette.

Syntax

```
arc
nt
```

Arguments

None

Options

None

Examples

Either of the following commands load and run the Windows NT ARC firmware from a diskette:

```
>>>arc
```

or

```
>>>nt
```

boot

Initializes the processor, loads a program image from the specified boot device, and transfers control to the loaded image.

Syntax

```
boot [-file <filename>] [-flags
<longword>[,<longword>]]
[-protocols <enet_protocol>] [-halt]
[<boot_device>]
```

Arguments

<boot_device>
A device path or list of devices from which the firmware will attempt to boot. Use the **set bootdef_dev** command to set an environment variable that specifies a default boot device.

Options

boot Command Option	Description
-file <filename>	Specifies the name of a file to load into the system. Use the **set boot_file** command to set the environment variable that specifies a default boot file.
-flags <longword> [,<longword>]	Specifies additional information for the operating system. For DIGITAL UNIX systems, the following values may be used: i = Interactive boot s = Boot to single user a = Autoboot to multiuser Use the **set boot_osflags** command to set an environment variable that specifies a default boot flag value.
-protocols <enet_protocol>	Specifies the Ethernet protocols that will be used for a network boot. Values may be **mop** or **bootp**.
-halt	Forces the bootstrap operation to halt and invoke the console program after the image is loaded and the page tables and other data structures are set up.

Examples

boot Command Example	Description
>>>**boot**	Boots the system from the default boot device.
>>>**boot ewa0**	Boots the system from Ethernet port ewa0.
>>>**boot -file dec2.sys ewa0**	Boots the file named dec2.sys from Ethernet port ewa0.
>>>**boot -protocol bootp ewa0**	Boots the system using the TCP/IP BOOTP protocol from Ethernet port ewa0.
>>>**boot -flags 0,1**	Boots the system from the default boot device using flag setting 0,1.
>>>**boot -halt dka0**	Loads the bootstrap image from disk dka0, halts the bootstrap operation, and invokes the console program. Subsequently, you can enter **continue** to transfer control to the operating system.

deposit

Writes data to the specified address.

Syntax

```
deposit [-{b,w,l,q,o,h}] [{physical, virtual, gpr,
fpr, ipr}] [-n <count>] [-s <step>]
[<device>:]<address> <data>
```

Arguments

`<device>:`

The optional device name (or address space) selects the device to access. The following platform-independent devices are supported:

- pmem
  Physical memory.
- vmem
  Virtual memory. All access and protection checking occur. If the access is not allowed to a program running with the current processor status (PS), the console issues an error message. If memory mapping is not enabled, virtual addresses are equal to physical addresses.

`<address>`
An address that specifies the offset within a device into which data is deposited. The address may be any legal symbolic address.

Valid symbolic addresses are shown in the following table.

Symbolic Address	Description
gpr-name	Represents general-purpose register.
ipr-name	Represents internal processor register.
PC	Program counter.
+	The location immediately following the last location referenced by **examine** or **deposit**.
-	The location immediately preceding the last location referenced by **examine** or **deposit**.

Symbolic Address	Description
*	The location last referenced by **examine** or **deposit**.
@	The location addressed by the last location referenced by **examine** or **deposit**.

<data>
The data to be deposited.

Options

deposit Command Option	Description
-b	Specifies data type is byte.
-w	Specifies data type is word.
-l	Specifies data type is longword.
-q	Specifies data type is quadword.
-o	Specifies data type is octaword.
-h	Specifies data type is hexword.
-physical	References physical address space.
-virtual	References virtual address space.
-gpr	References general-purpose register address space.
-fpr	References floating-point register address space.
-ipr	References internal processor register address space.
-n <count>	Specifies the number of consecutive locations to examine.
-s <step>	Specifies the address increment as a hexadecimal value. This option allows you to override the increment that is normally derived from the data size.

Examples

deposit Command Example	Description
>>>**d -n 1ff pmem:0 0**	Clears the first 512 bytes of physical memory.
>>>**d -l -n 3 pmem:1234 5**	Writes the value 5 into four longwords, starting at physical memory address 1234.
>>>**d -n 8 r0 ffffffff**	Loads GPRs R0 through R8 with -1.
>>>**d -l -n 10 -s 200 pmem:0 8**	Writes the value 8 in the first longword of the first 17 pages in physical memory.

examine

Displays the contents of the specified address.

Syntax

examine [-{b,w,l,q,o,h,d}] [-{physical, virtual, gpr, fpr, ipr}] [-n <count>] [-s <step>] [<device>:]<address>

Arguments

<device>:
The optional device name (or address space) selects the device to access.

<address>
The address specifies the first location to examine within the current device. The address can be any legal address specified.

Options

examine Command Option	Description
-b	Specifies data type is byte.
-w	Specifies data type is word.
-l	Specifies data type is longword.
-q	Specifies data type is quadword.
-o	Specifies data type is octaword.
-h	Specifies data type is hexword.
-d	Specifies the data displayed is the decoded macro instruction. The Alpha instruction decode (-d) does not recognize machine-specific PALcode instructions.
-physical	References physical address space.
-virtual	References virtual address space.
-gpr	References general-purpose register address space.
-fpr	References floating-point register address space.

examine Command Option	Description
-ipr	References internal processor register address space.
-n <count>	Specifies the number of consecutive locations to examine.
-s <step>	Specifies the address increment as a hexadecimal value. This option allows you to override the increment that is normally derived from the data size.

Examples

examine Command Example	Display	Description
>>>e r0	gpr: 0 (R0) 0000000000000002	Examines the contents of R0, using a symbolic address.
>>>e -g 0	gpr: 0 (R0) 0000000000000002	Examines the contents of R0, using address space.
>>>e grp:0	gpr: 0 (R0) 0000000000000002	Examines the contents of R0, using a device name.
>>>examine -n 5 r7	gpr: 38 (R7) 0000000000000000 gpr: 40 (R8) 0000000000000000 gpr: 48 (R9) 0000000000000000 gpr: 50 (R10) 000000007FFBF800 gpr: 58 (R11) 000000007FF781A2 gpr: 60 (R12) 0000000000000000	Examines the contents of R7 and the next five registers.
>>>examine ipr:11	ipr 11 (KSP) FFFFFFFF8228DFD0	Examines the contents of internal processor register 11.

fwupdate

Loads and runs the firmware update utility from a diskette.

Syntax

fwupdate

Arguments

None

Options

None

Examples

The following fwupdate script command loads and runs the firmware update utility from a diskette:

```
>>>fwupdate
```

set

Sets or modifies the value of an environment variable.

Syntax

set <envar> <value> [-default] [-integer] [-string]

Arguments

<envar>

The environment variable to be assigned a new value.

<value>

The value that is assigned to the environment variable. It can be either a numeric value or an ASCII string.

Options

set Command Option	Description
-default	Restores an environment variable to its default value.
-integer	Creates an environment variable as an integer.
-string	Creates an environment variable as a string.

Examples

set Command Example	Description
>>>set bootdef_dev ewa0	Modifies the default boot device to ewa0.
>>>set auto_action boot	Attempts to boot the operating system following an error, halt, or power-up.
>>>set boot_osflags 0,1	Modifies the default boot flags to 0,1.
>>>set foobar 5	Creates an environment variable called foobar and gives it a value of 5.

show

Displays the current value of the specified environment variable or information about the system.

Syntax

```
show [{config, device [device_name], iobq, hwrpb,
map, memory, pal, version, <envar>...}]
```

Arguments

show Command Argument	Description
config	Displays the current memory configuration, PCI logical slots, and ISA logical slots.
device [device name]	Displays the devices and controllers in the system. Specifying a device name returns information on that device only.
iobq	Displays the input/output counter blocks.
hwrpb	Displays the hardware restart parameter block.
map	Displays the system virtual memory map.
memory	Displays the memory module configuration.
pal	Displays the version of DIGITAL UNIX PALcode.
version	Displays the version of the console.
<envar>	Displays the current value of a specified environment variable.

Examples

show Command Example	Description
>>>**show device** dka0.0.0.6.0 DKA0 RZ26L 441A dka400.4.0.6.0 DKA400 RRD43 3213 dva0.0.0.0.1 DVA0 ewa0.0.0.12.0 EWA0 08-00-2B-E2-1C-25 pka0.7.0.6.0 PKA0 SCSI Bus ID 7	Lists device information, such as system designation, drive model, or Ethernet address.

show Command Example	Description
>>>**show memory** 48 Meg of System Memory	Lists system random-access memory (RAM) size.
>>>**show *** (refer to Section 7.3)	Lists all environment variables and their settings.
>>>**show boot*** (refer to Section 7.3)	Lists all environment variables, beginning with boot.

7.3 Environment Variables for Alpha SRM Console Commands

This section describes environment variables that are used to define the system operational state and to pass information between the firmware and the operating system.

7.3.1 Environment Variable Descriptions

Environment variables are classified as either Alpha SRM Console architecture-required or system-defined.

7.3.1.1 Architecture-Required Environment Variables

The following table shows common Alpha SRM Console architecture-required environment variables and their descriptions. For a complete list, enter **show *** at the Alpha SRM Console prompt.

Architecture-Required Environment Variable	Description
auto_action	When used with the **set** or **show** command, this variable modifies or displays the console action that follows an error, halt, or power-up. The action can be halt, boot, or restart. The default is halt.
boot_file	When used with the **set** or **show** command, this variable modifies or displays the file name to be used when a bootstrap requires a file name. The default is null.

Architecture-Required Environment Variable	Description
boot_osflags	When used with the **set** or **show** command, this variable modifies or displays the additional parameters to be passed to system software. The default is 0.
bootdef_dev	When used with the **set** or **show** command, this variable modifies or displays the default device or device list from which the system will attempt to boot. If the system software is preloaded, the variable is preset to point to the device containing the preloaded software. The default is null.

7.3.1.2 System-Defined Environment Variables

The following table shows common Alpha SRM Console system-defined environment variables and their descriptions. For a complete list, enter **show *** at the Alpha SRM Console prompt.

System-Defined Environment Variable	Description
console	When used with the **set** command, this variable modifies the console output to either the serial port or the graphics controller.
ewa0_mode	This variable determines if the AUI (ThinWire) or the twisted-pair Ethernet ports will be enabled. AUI is the default. (Autosensing is not supported.)
os_type	When used with the **set** or **show** command, this variable modifies or displays the specified firmware that will be loaded on the next power cycle. Specify the value **osf** or **UNIX** to select the Alpha SRM Console.
pci_parity	This variable controls PCI parity checking. The possible values are: **on** = Parity checking is enabled. **off** = Parity checking is disabled; this is the default. **sniff** = Parity checking is enabled or disabled depending on the PCI device.
oem_string	When used with the **set** or **show** command, this variable modifies or displays a text string that identifies the product name in the Alpha SRM Console banner.

System-Defined Environment Variable	Description
language n	The language environment variable assigns language n to the system (where n is the option number of a language listed in the menu that follows). Use the following procedure to select the language: 1. At the Alpha SRM Console prompt, enter the following commands: **>>>set language 0** **>>>init** The following menu and prompt are displayed: n Language n Language ========== 0 none (display menu) 40 Francais (Suisse Romande) 30 Dansk 42 Italiano 32 Deutsch 44 Nederlands 34 Deutsch (Schweiz) 46 Norsk 36 English (American) 48 Portugues 38 English (British/Irish) 4A Suomi 3A Espanol 4C Svenska 3C Francais 4E Vlaams 3E Francais (Canadian) (1..16): 2. Enter the number that corresponds to the language that you want to use. The following example shows how to assign the English (American) language to the system: (1..16):**36** 3. When you receive a message to reset the system, power cycle the system.

7.4 Using the Firmware Update Utility

Use the firmware update utility to update the firmware in a flash ROM.

7.4.1 Starting the Firmware Update Utility

To start the firmware update utility from the Alpha SRM Console firmware update diskette, follow this procedure:

1. Insert the Alpha SRM Console firmware update diskette into the diskette drive.
2. At the Alpha SRM Console prompt, enter the following command:

   ```
   >>>fwupdate
   ```

   Note: Because the firmware update utility reinitializes some system components, it may appear as if your system is restarting.

3. Proceed to the Section 7.4.2 .

7.4.2 Running the Firmware Update Utility

To run the firmware update utility, follow this procedure:

1. From the Firmware Update menu, choose whichever selection appears:
   - **`Update SRM Console Firmware`**
   - **`Update Firmware`**
2. When you are prompted to continue the update, choose **`Yes`**.
3. If the console selection *does not* match the firmware you flashed, you will be prompted to update the console selection. If you are prompted to update the console selection, choose **`Yes.`**
4. Restart the AlphaPC 164 motherboard system.

   Note: Depending on the version of firmware that you are updating from, an error condition may occur. If an error condition occurs, power cycle the system.

5. Wait for the Alpha SRM Console prompt (>>>) to appear on the terminal attached to the COM1 serial port and on the graphics display unit.

Note: If you do not receive the Alpha SRM Console prompt (>>>), press the Enter key on the terminal attached to COM1 or on the console keyboard.

6. To specify the default console device, use the following Alpha SRM Console commands:

Default Console Device	Commands
Terminal attached to the COM1 serial port	**>>>set console serial** **>>>init**
Graphics display unit	**>>>set console graphics** **>>>init**

Refer to Section 7.2 and Section 7.3 for more information about Alpha SRM Console commands and environment variables.

7.4.3 Troubleshooting the Firmware Update Utility Procedure

The fail-safe booter provides an emergency recovery mechanism when the primary firmware image contained in flash memory has been corrupted. When flash memory has been corrupted, such that no image safely loads from the flash, the fail-safe booter can be run to facilitate booting another image from a diskette that is capable of reprogramming the flash.

7.4.3.1 Running the Fail-Safe Booter

The fail-safe booter can be started in one of two ways:

- If the primary firmware image is unavailable when the system is powered on or reset, the fail-safe booter automatically runs.

  When the fail-safe booter runs, the system emits a series of beeps through the speaker as beep code 1-2-3, that is, one beep and a pause followed by two beeps and a pause followed by three beeps. Then the diskette activity light flashes.

  1. To start the firmware update utility, you must insert the Alpha SRM Console firmware update diskette into the diskette drive.The diskette contains the file `FWUPDATE.EXE` and the Alpha SRM Console ROM image.

2. Return to Section 7.4.2 .

- The fail-safe booter can also be started manually as follows:
  1. Add jumper CF7 as described in Section 3.6.
  2. Insert the Alpha SRM Console firmware update diskette into the diskette drive.
  3. Return to Section 7.4.2 .

7.5 Installing the DIGITAL UNIX Operating System

This section supplements the *DIGITAL UNIX Installation Guide* for installing the DIGITAL UNIX operating system on an AlphaPC 164 motherboard system.

Note: If you already have a previous version of DIGITAL UNIX installed, see the *DIGITAL UNIX Installation Guide* for information on how to upgrade to a new version of the operating system. Be sure to review the preinstallation tasks, which are covered in the *DIGITAL UNIX Installation Guide*.

7.5.1 Requirements

You need the following hardware and software to install the DIGITAL UNIX operating system on an AlphaPC 164 motherboard system:

- A minimum of 32MB of main memory; 64MB is recommended
- A SCSI hard disk capable of storing the supported software subsets
  - A minimum of 425MB disk space for a default installation (that is, the mandatory subsets only)
  - A minimum of 680MB of disk space for an advanced installation (that is, all BASE software subsets); a 1GB (or larger) SCSI hard disk is recommended

    Note: IDE disks are not supported.

- Supported load devices
  - SCSI CD–ROM drives capable of reliably reading in 512-byte block mode

    or
  - A network interface

- A console terminal with ASCII capability or a supported graphics display console
- DIGITAL UNIX Version 4.0 Operating System Volume 1 compact disc
- Alpha SRM Console Version 4.5 or higher

7.5.2 Special Instructions

When booting DIGITAL UNIX Version 4.0 on an Alpha PC 164 motherboard system, the following message is displayed:

```
Module 1095:646 not in pci option table, can't
configure it.
```

This message is caused by the on-board IDE controller not being recognized and supported by DIGITAL UNIX Version 4.0. This message is normal and can be ignored.

8 Battery Recycle/Disposal Information

NOTICE

Recycle or dispose of batteries promptly in accordance with your organization's environmental policies. If this is a LITHIUM battery, the following additional precautions may apply:

- Replace batteries correctly to prevent possible explosion.
- Replace batteries with the same or equivalent type.
- Prior to disposal or recycling, protect all batteries against accidental short circuiting by affixing nonconductive tape across battery terminals or conductive surfaces.
- Keep small batteries away from children.

9 Ordering Associated Documentation

The following table lists some of the available third-party documentation that supports the AlphaPC 164 motherboard. You can order documentation directly from the vendor.

Title	Vendor
Alpha AXP Architecture Reference Manual (PN EY–T132E–DP)	Call 1–800–344–4825 from the U.S. or Canada, or call Butterworth-Heinemann (Digital Press) at 1–800–366–2665.
Alpha AXP Architecture Handbook (PN EC–QD2KB–TE)	Digital Equipment Corporation Digital Semiconductor 75 Reed Road Hudson MA 01749 USA Call the Digital Semiconductor Information Line: United States and Canada 1–800–332–2717 Outside North America +1–510–490–4753 Visit the Digital Semiconductor World Wide Web Internet site: http://www.digital.com/semiconductor
Digital Semiconductor 21164 Alpha Microprocessor Data Sheet (PN EC–QP98B–TE)	Digital Equipment Corporation (See previous entry.)
Digital Semiconductor 21164 Alpha Microprocessor Hardware Reference Manual (PN EC–QP99B–TE)	Digital Equipment Corporation (See previous entry.)
Digital Semiconductor 21172 Core Logic Chipset Technical Reference Manual (PN EC–QUQJA–TE)	Digital Equipment Corporation (See previous entry.)
DIGITAL UNIX Installation Guide (PN AA-QTLGA-TE)	Digital Equipment Corporation (See previous entry.)

Title	Vendor
Hardware Compatibility List	Software Product Description DIGITAL UNIX Operating System Version 4.0 (SPD 41.61.13) Digital Equipment Corporation (See previous entry.)
PCI System Design Guide	PCI Special Interest Group 1–800–433–5177 (U.S.) 1–503–797–4207 (International) 1–503–234–6762 (FAX)
PCI Local Bus Specification, Rev 2.1	PCI Special Interest Group (See previous entry.)
82420/82430 PCIset ISA and EISA Bridges (includes 82378IB/ZB SIO) (PN 290483)	Intel Corporation Literature Sales P.O. Box 7641 Mt. Prospect IL 60056 USA 1–800–628–8686 FaxBACK Service 1–800–628–2283 BBS 1–916–356–3600
Super I/O combination Controller (FDC37C935) Data Sheet	Standard Microsystems Corporation 80 Arkay Drive Hauppauge NY 11788 USA Phone: 1–516–435–6000 FAX: 1–516–231–6004