

21164 Alpha Microprocessor

Hardware Reference Manual

Preliminary



ELECTRONICS

Notice

The information in this publication has been carefully checked and is believed to be entirely accurate at the time of publication. Samsung assumes no responsibility, however, for possible errors or omissions, or for any consequences resulting from the use of the information contained herein.

Samsung reserves the right to make changes in its products or product specifications with the intent to improve function or design at any time and without notice and is not required to update this documentation to reflect such changes.

This publication does not convey to a purchaser of semiconductor devices described herein any license under the patent rights of Samsung or others.

21164 Alpha™ Microprocessor Hardware Reference Manual

©1997 Samsung Electronics

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electric or mechanical, by photocopying, recording, or otherwise, without the prior written consent of Samsung Electronics.

Alpha, Alpha AXP, Digital, Open VMS, VAX, VAX DOCUMENT are trademarks of Digital Equipment Corporation.

GRAFOIL is a registered trademark of Union Carbide Corporation.

IEEE is a registered trademark of The Institute of Electrical and Electronics Engineers, Inc.

Microsoft is a registered trademark and Windows NT is a trademark of Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Ltd.

All other trademarks and registered trademarks are the property of their respective owners.

San #24 Nongseo-ri, Kiheung-eup
Yongin-city, Kyungki-do, Korea
449-900
FAX : 82-331-209-4492
TEL : 82-331-209-6138

Printed in the Republic of Korea

Contents

Preface

1 Introduction

1.1	The Architecture	1-1
1.1.1	Addressing	1-2
1.1.2	Integer Data Types	1-2
1.1.3	Floating-Point Data Types	1-3
1.2	21164 Microprocessor Features	1-3

2 Internal Architecture

2.1	21164 Microarchitecture	2-1
2.1.1	Instruction Fetch/Decode Unit and Branch Unit	2-3
2.1.1.1	Instruction Decode and Issue	2-4
2.1.1.2	Instruction Prefetch	2-4
2.1.1.3	Branch Execution	2-6
2.1.1.4	Instruction Translation Buffer	2-7
2.1.1.5	Interrupts	2-8
2.1.2	Integer Execution Unit	2-9
2.1.3	Floating-Point Execution Unit	2-10
2.1.4	Memory Address Translation Unit	2-10
2.1.4.1	Data Translation Buffer	2-11
2.1.4.2	Load Instruction and the Miss Address File	2-11
2.1.4.3	Dcache Control and Store Instructions	2-12
2.1.4.4	Write Buffer	2-12
2.1.5	Cache Control and Bus Interface Unit	2-13
2.1.6	Cache Organization	2-13
2.1.6.1	Data Cache	2-13
2.1.6.2	Instruction Cache	2-13
2.1.6.3	Second-Level Cache	2-14
2.1.6.4	External Cache	2-14

2.1.7	Serial Read-Only Memory Interface	2-14
2.2	Pipeline Organization	2-14
2.2.1	Pipeline Stages and Instruction Issue	2-18
2.2.2	Aborts and Exceptions	2-18
2.2.3	Nonissue Conditions	2-20
2.3	Scheduling and Issuing Rules	2-20
2.3.1	Instruction Class Definition and Instruction Slotting	2-20
2.3.2	Coding Guidelines	2-23
2.3.3	Instruction Latencies	2-24
2.3.3.1	Producer—Producer Latency	2-27
2.3.4	Issue Rules	2-28
2.4	Replay Traps	2-29
2.5	Miss Address File and Load-Merging Rules	2-30
2.5.1	Merging Rules	2-30
2.5.2	Read Requests to the CBU	2-31
2.5.3	Load Instructions to Noncacheable Space	2-31
2.5.4	MAF Entries and MAF Full Conditions	2-31
2.5.5	Fill Operation	2-32
2.6	MTU Store Instruction Execution	2-33
2.7	Write Buffer and the WMB Instruction	2-35
2.7.1	The Write Buffer	2-35
2.7.2	The Write Memory Barrier (WMB) Instruction	2-35
2.7.3	Entry-Pointer Queues	2-36
2.7.4	Write Buffer Entry Processing	2-36
2.7.5	Ordering of Noncacheable Space Write Instructions	2-37
2.8	Performance Measurement Support—Performance Counters	2-38
2.9	Floating-Point Control Register	2-39
2.10	Design Examples	2-41

3 Hardware Interface

3.1	21164 Microprocessor Logic Symbol	3-1
3.2	21164 Signal Names and Functions	3-3

4 Clocks, Cache, and External Interface

4.1	Introduction to the External Interface	4-2
4.1.1	System Interface	4-2
4.1.1.1	Commands and Addresses	4-3
4.1.2	Bcache Interface	4-4
4.2	Clocks	4-4
4.2.1	CPU Clock	4-4
4.2.2	System Clock	4-6
4.2.3	Delayed System Clock	4-7

4.2.4	Reference Clock	4-8
4.2.4.1	Reference Clock Examples	4-9
4.3	Physical Address Considerations	4-11
4.3.1	Physical Address Regions	4-11
4.3.2	Data Wrapping	4-12
4.3.3	Noncached Read Operations	4-13
4.3.4	Noncached Write Operations	4-13
4.4	Bcache Structure	4-14
4.4.1	Duplicate Tag Store	4-14
4.4.1.1	Full Duplicate Tag Store	4-15
4.4.1.2	Partial Scache Duplicate Tag Store	4-16
4.4.2	Bcache Victim Buffers	4-17
4.5	Systems Without a Bcache	4-17
4.6	Cache Coherency	4-18
4.6.1	Cache Coherency Basics	4-18
4.6.2	Write Invalidate Cache Coherency Protocol Systems	4-20
4.6.3	Write Invalidate Cache Coherency States	4-21
4.6.3.1	Write Invalidate Protocol State Machines	4-22
4.6.4	Flush Cache Coherency Protocol Systems	4-23
4.6.5	Flush-Based Protocol State Machines	4-25
4.6.6	Cache Coherency Transaction Conflicts	4-25
4.6.6.1	Case 1	4-26
4.6.6.2	Case 2	4-26
4.7	Lock Mechanisms	4-26
4.8	21164-to-Bcache Transactions	4-28
4.8.1	Bcache Timing	4-28
4.8.2	Bcache Read Transaction (Private Read Operation)	4-28
4.8.3	Wave Pipeline	4-29
4.8.4	Bcache Write Transaction (Private Write Operation)	4-30
4.8.5	Synchronous Cache Support	4-31
4.8.6	Selecting Bcache Options	4-34
4.9	21164-Initiated System Transactions	4-35
4.9.1	READ MISS—No Bcache	4-38
4.9.2	READ MISS—Bcache	4-39
4.9.3	FILL	4-41
4.9.4	READ MISS with Victim	4-42
4.9.4.1	READ MISS with Victim (Victim Buffer)	4-42
4.9.4.2	READ MISS with Victim (Without Victim Buffer)	4-43
4.9.5	WRITE BLOCK and WRITE BLOCK LOCK	4-45
4.9.6	SET DIRTY and LOCK	4-46
4.9.7	MEMORY BARRIER (MB)	4-48
4.9.7.1	When to Use a MEMORY BARRIER Command	4-48
4.9.8	FETCH	4-48
4.9.9	FETCH_M	4-48
4.10	System-Initiated Transactions	4-48
4.10.1	Sending Commands to the 21164	4-49
4.10.2	Write Invalidate Protocol Commands	4-51

4.10.2.1	21164 Responses to Write Invalidate Protocol Commands	4-53
4.10.2.2	READ DIRTY and READ DIRTY/INVALIDATE	4-54
4.10.2.3	INVALIDATE	4-55
4.10.2.4	SET SHARED	4-56
4.10.3	Flush-Based Cache Coherency Protocol Commands	4-58
4.10.3.1	21164 Responses to Flush-Based Protocol Commands	4-59
4.10.3.2	FLUSH	4-60
4.10.3.3	READ	4-61
4.11	Data Bus and Command/Address Bus Contention	4-62
4.11.1	Command/Address Bus	4-63
4.11.2	Read/Write Spacing—Data Bus Contention	4-63
4.11.3	Using idle_bc_h and fill_h	4-64
4.11.4	Using data_bus_req_h	4-66
4.11.5	Tristate Overlap	4-67
4.11.5.1	READ or WRITE to FILL	4-67
4.11.5.2	BCACHE VICTIM to FILL	4-67
4.11.5.3	System Bcache Command to FILL	4-69
4.11.5.4	FILL to Private Read or Write Operation	4-71
4.11.6	Auto DACK	4-72
4.11.7	Victim Write Back Under Miss	4-74
4.12	21164 Interface Restrictions	4-75
4.12.1	FILL Operations After Other Transactions	4-75
4.12.2	Command Acknowledge for WRITE BLOCK Commands	4-75
4.12.3	Systems Without a Bcache	4-75
4.12.4	Fast Probes with No Bcache	4-76
4.12.5	WRITE BLOCK LOCK	4-77
4.13	21164/System Race Conditions	4-78
4.13.1	Rules for 21164 and System Use of External Interface	4-78
4.13.2	READ MISS with Victim Example	4-79
4.13.3	idle_bc_h and cack_h Race Example	4-80
4.13.4	READ MISS with idle_bc_h Asserted Example	4-82
4.13.5	READ MISS with Victim Abort Example	4-83
4.13.6	Bcache Hit Under READ MISS Example	4-84
4.14	Data Integrity, Bcache Errors, and Command/Address Errors	4-84
4.14.1	Data ECC and Parity	4-85
4.14.2	Force Correction	4-87
4.14.3	Bcache Tag Data Parity	4-87
4.14.4	Bcache Tag Control Parity	4-87
4.14.5	Address and Command Parity	4-87
4.14.6	Fill Error	4-87
4.14.7	Forcing 21164 Reset	4-88
4.15	Interrupts	4-88
4.15.1	Interrupt Signals During Initialization	4-89
4.15.2	Interrupt Signals During Normal Operation	4-89
4.15.3	Interrupt Priority Level	4-89

5 Internal Processor Registers

5.1	Instruction Fetch/Decode Unit and Branch Unit (IDU) IPRs	5-5
5.1.1	Istream Translation Buffer Tag Register (ITB_TAG)	5-5
5.1.2	Instruction Translation Buffer Page Table Entry (ITB_PTE) Register	5-5
5.1.3	Instruction Translation Buffer Address Space Number (ITB_ASN) Register	5-7
5.1.4	Instruction Translation Buffer Page Table Entry Temporary (ITB_PTE_TEMP) Register	5-8
5.1.5	Instruction Translation Buffer Invalidate All Process (ITB_IAP) Register	5-8
5.1.6	Instruction Translation Buffer Invalidate All (ITB_IA) Register	5-8
5.1.7	Instruction Translation Buffer IS (ITB_IS) Register	5-9
5.1.8	Formatted Faulting Virtual Address (IFault_VA_FORM) Register	5-10
5.1.9	Virtual Page Table Base Register (IVPTBR)	5-11
5.1.10	Icache Parity Error Status (ICPERR_STAT) Register	5-12
5.1.11	Icache Flush Control (IC_FLUSH_CTL) Register	5-12
5.1.12	Exception Address (EXC_ADDR) Register	5-13
5.1.13	Exception Summary (EXC_SUM) Register	5-14
5.1.14	Exception Mask (EXC_MASK) Register	5-15
5.1.15	PAL Base Address (PAL_BASE) Register	5-16
5.1.16	IDU Current Mode (ICM) Register	5-16
5.1.17	IDU Control and Status Register (ICSR)	5-17
5.1.18	Interrupt Priority Level Register (IPLR)	5-19
5.1.19	Interrupt ID (INTID) Register	5-20
5.1.20	Asynchronous System Trap Request Register (ASTRR)	5-21
5.1.21	Asynchronous System Trap Enable Register (ASTER)	5-21
5.1.22	Software Interrupt Request Register (SIRR)	5-22
5.1.23	Hardware Interrupt Clear (HWINT_CLR) Register	5-23
5.1.24	Interrupt Summary Register (ISR)	5-24
5.1.25	Serial Line Transmit (SL_XMIT) Register	5-26
5.1.26	Serial Line Receive (SL_RCV) Register	5-27
5.1.27	Performance Counter (PMCTR) Register	5-28
5.2	Memory Address Translation Unit (MTU) IPRs	5-33
5.2.1	Dstream Translation Buffer Address Space Number (DTB_ASN) Register	5-33
5.2.2	Dstream Translation Buffer Current Mode (DTB_CM) Register	5-33
5.2.3	Dstream Translation Buffer Tag (DTB_TAG) Register	5-34
5.2.4	Dstream Translation Buffer Page Table Entry (DTB_PTE) Register	5-34
5.2.5	Dstream Translation Buffer Page Table Entry Temporary (DTB_PTE_TEMP) Register	5-36
5.2.6	Dstream Memory Management Fault Status (MM_STAT) Register	5-37
5.2.7	Faulting Virtual Address (VA) Register	5-38
5.2.8	Formatted Virtual Address (VA_FORM) Register	5-39
5.2.9	MTU Virtual Page Table Base Register (MVPTBR)	5-40
5.2.10	Dcache Parity Error Status (DC_PERR_STAT) Register	5-41
5.2.11	Dstream Translation Buffer Invalidate All Process (DTB_IAP) Register	5-42
5.2.12	Dstream Translation Buffer Invalidate All (DTB_IA) Register	5-42
5.2.13	Dstream Translation Buffer Invalidate Single (DTB_IS) Register	5-43

5.2.14	MTU Control Register (MCSR)	5-44
5.2.15	Dcache Mode (DC_MODE) Register	5-46
5.2.16	Miss Address File Mode (MAF_MODE) Register	5-48
5.2.17	Dcache Flush (DC_FLUSH) Register	5-50
5.2.18	Alternate Mode (ALT_MODE) Register	5-50
5.2.19	Cycle Counter (CC) Register	5-51
5.2.20	Cycle Counter Control (CC_CTL) Register	5-52
5.2.21	Dcache Test Tag Control (DC_TEST_CTL) Register	5-53
5.2.22	Dcache Test Tag (DC_TEST_TAG) Register	5-54
5.2.23	Dcache Test Tag Temporary (DC_TEST_TAG_TEMP) Register	5-56
5.3	External Interface Control (CBU) IPRs	5-58
5.3.1	Scache Control (SC_CTL) Register (FF FFF0 00A8)	5-59
5.3.2	Scache Status (SC_STAT) Register (FF FFF0 00E8)	5-62
5.3.3	Scache Address (SC_ADDR) Register (FF FFF0 0188)	5-65
5.3.4	Bcache Control (BC_CONTROL) Register (FF FFF0 0128)	5-68
5.3.5	Bcache Configuration (BC_CONFIG) Register (FF FFF0 01C8)	5-75
5.3.6	Bcache Tag Address (BC_TAG_ADDR) Register (FF FFF0 0108)	5-78
5.3.7	External Interface Status (EI_STAT) Register (FF FFF0 0168)	5-80
5.3.8	External Interface Address (EI_ADDR) Register (FF FFF0 0148)	5-83
5.3.9	Fill Syndrome (FILL_SYN) Register (FF FFF0 0068)	5-84
5.4	PALcode Storage Registers	5-87
5.5	Restrictions	5-88
5.5.1	CBU IPR PALcode Restrictions	5-88
5.5.2	PALcode Restrictions—Instruction Definitions	5-89

6 Privileged Architecture Library Code

6.1	PALcode Description	6-1
6.2	PALmode Environment	6-2
6.3	Invoking PALcode	6-3
6.4	PALcode Entry Points	6-5
6.4.1	CALL_PAL Entry	6-5
6.4.2	PALcode Trap Entry Points	6-6
6.5	Required PALcode Function Codes	6-7
6.6	21164 Implementation of the Architecturally Reserved Opcodes	6-7
6.6.1	HW_LD Instruction	6-8
6.6.2	HW_ST Instruction	6-10
6.6.3	HW_REI Instruction	6-11
6.6.4	HW_MFPR and HW_MTPR Instructions	6-11

7 Initialization and Configuration

7.1	Input Signals sys_reset_l and dc_ok_h and Booting	7-1
7.1.1	Pin State with dc_ok_h Not Asserted	7-6

7.2	Sysclk Ratio and Delay	7-6
7.3	Built-In Self-Test (BiSt)	7-6
7.4	Serial Read-Only Memory Interface Port	7-6
7.4.1	Serial Instruction Cache Load Operation	7-7
7.5	Serial Terminal Port	7-8
7.6	Cache Initialization	7-8
7.6.1	Icache Initialization	7-9
7.6.2	Flushing Dirty Blocks	7-9
7.7	External Interface Initialization	7-10
7.8	Internal Processor Register Reset State	7-10
7.9	Timeout Reset	7-13
7.10	IEEE 1149.1 Test Port Reset	7-14

8 Error Detection and Error Handling

8.1	Error Flows	8-1
8.1.1	Icache Data or Tag Parity Error	8-1
8.1.2	Scache Data Parity Error—Istream	8-2
8.1.3	Scache Tag Parity Error—Istream	8-2
8.1.4	Scache Data Parity Error—Dstream Read/Write, READ_DIRTY	8-3
8.1.5	Scache Tag Parity Error—Dstream or System Commands	8-3
8.1.6	Dcache Data Parity Error	8-4
8.1.7	Dcache Tag Parity Error	8-4
8.1.8	Istream Uncorrectable ECC or Data Parity Errors (Bcache or Memory)	8-5
8.1.9	Dstream Uncorrectable ECC or Data Parity Errors (Bcache or Memory)	8-5
8.1.10	Bcache Tag Parity Errors—Istream	8-6
8.1.11	Bcache Tag Parity Errors—Dstream	8-7
8.1.12	System Command/Address Parity Error	8-7
8.1.13	System Read Operations of the Bcache	8-8
8.1.14	Istream or Dstream Correctable ECC Error (Bcache or Memory)	8-8
8.1.15	Fill Timeout (FILL_ERROR_H)	8-9
8.1.16	System Machine Check	8-9
8.1.17	IDU Timeout	8-9
8.1.18	cfail_h and Not cack_h	8-10
8.2	MCHK Flow	8-10
8.3	Processor-Correctable Error Interrupt Flow (IPL 31)	8-12
8.4	MCK_INTERRUPT Flow	8-13
8.5	System-Correctable Error Interrupt Flow (IPL 20)	8-13

9 Electrical Data

9.1	Electrical Characteristics	9-1
9.2	DC Characteristics	9-2
9.2.1	Power Supply	9-2

9.2.2	Input Signal Pins	9-2
9.2.3	Output Signal Pins	9-2
9.3	Clocking Scheme	9-5
9.3.1	Input Clocks	9-5
9.3.2	Clock Termination and Impedance Levels	9-7
9.3.3	AC Coupling	9-8
9.4	AC Characteristics	9-9
9.4.1	Test Configuration	9-9
9.4.2	Pin Timing	9-11
9.4.2.1	Backup Cache Loop Timing	9-11
9.4.2.2	sys_clk-Based Systems	9-14
9.4.2.3	Reference Clock-Based Systems	9-17
9.4.3	Digital Phase-Locked Loop	9-19
9.4.4	Timing—Additional Signals	9-20
9.4.5	Timing of Test Features	9-24
9.4.5.1	Icache BiSt Operation Timing	9-25
9.4.5.2	Automatic SROM Load Timing	9-26
9.4.6	Clock Test Modes	9-27
9.4.6.1	Normal (1× Clock) Mode	9-27
9.4.6.2	2× Clock Mode	9-28
9.4.6.3	Chip Test Mode	9-28
9.4.6.4	Module Test Mode	9-28
9.4.6.5	Clock Test Reset Mode	9-28
9.4.7	IEEE 1149.1 (JTAG) Performance	9-29
9.5	Power Supply Considerations	9-29
9.5.1	Decoupling	9-30
9.5.1.1	Vdd Decoupling	9-30
9.5.1.2	Vddi Decoupling	9-30
9.5.2	Power Supply Sequencing	9-31

10 Thermal Management

10.1	Operating Temperature	10-1
10.2	Heat Sink Specifications	10-3
10.3	Thermal Design Considerations	10-4

11 Mechanical Data and Packaging Information

11.1	Mechanical Specifications	11-1
11.2	Signal Descriptions and Pin Assignment	11-3
11.2.1	Signal Pin Lists	11-3
11.2.2	Pin Assignment	11-8

12 Testability and Diagnostics

12.1	Test Port Pins	12-1
12.2	Test Interface	12-2
12.2.1	IEEE 1149.1 Test Access Port	12-2
12.2.2	Test Status Pins	12-5
12.3	Boundary-Scan Register	12-6

A Alpha Instruction Set

A.1	Alpha Instruction Summary	A-1
A.1.1	Opcodes Reserved	A-9
A.1.2	Opcodes Reserved for PALcode	A-9
A.2	IEEE Floating-Point Instructions	A-10
A.3	VAX Floating-Point Instructions	A-12
A.4	Opcode Summary	A-12
A.5	Required PALcode Function Codes	A-14
A.6	21164 Microprocessor IEEE Floating-Point Conformance	A-14

B 21164 Microprocessor Specifications

C Serial Icache Load Predecode Values

Glossary

Index

Figures

2-1	21164 Microprocessor Block/Pipe Flow Diagram	2-2
2-2	Instruction Pipeline Stages	2-15
2-3	Floating-Point Control Register (FPCR) Format	2-39
2-4	Typical Uniprocessor Configuration	2-41
2-5	Typical Multiprocessor Configuration	2-42
2-6	Cacheless Multiprocessor Configuration	2-43
3-1	21164 Microprocessor Logic Symbol	3-2
4-1	21164 System/Bcache Interface	4-3
4-2	Clock Signals and Functions	4-6
4-3	21164 Uniprocessor Clock	4-7
4-4	21164 Reference Clock for Multiprocessor Systems	4-9
4-5	ref_clk_in_h Initially Sampled Low	4-10
4-6	ref_clk_in_h Initially Sampled High	4-11
4-7	Full Scache Duplicate Tag Store	4-15
4-8	Duplicate Tag Store Algorithm	4-16
4-9	Partial Scache Duplicate Tag Store	4-17
4-10	Cache Subset Hierarchy	4-18
4-11	Write Invalidate Protocol: 21164 State Transitions	4-22
4-12	Write Invalidate Protocol: System/Bus State Transitions	4-23
4-13	Flush-Based Protocol 21164 States	4-25
4-14	Flush-Based Protocol System/Bus States	4-25
4-15	Bcache Read Transaction	4-29
4-16	Wave Pipeline Timing Diagram	4-30
4-17	Bcache Write Transaction	4-30
4-18	Synchronous Read Timing Diagram	4-33
4-19	Synchronous Write Timing Diagram	4-33
4-20	READ MISS—No Bcache Timing Diagram	4-38
4-21	READ MISS MOD—Bcache Timing Diagram	4-40
4-22	READ MISS with Victim (Victim Buffer) Timing Diagram	4-43
4-23	READ MISS with Victim (Without Victim Buffer) Timing Diagram	4-44
4-24	WRITE BLOCK Timing Diagram	4-46
4-25	SET DIRTY and LOCK Timing Diagram	4-47
4-26	Algorithm for System Sending Commands to the 21164	4-50
4-27	READ DIRTY Timing Diagram (Scache Hit)	4-55
4-28	INVALIDATE Timing Diagram (Bcache Hit)	4-56
4-29	SET SHARED Timing Diagram	4-57
4-30	FLUSH Timing Diagram (Scache Hit)	4-61
4-31	Read Timing Diagram (Scache Hit)	4-62
4-32	Driving the Command/Address Bus	4-63
4-33	Example of Using idle_bc_h and fill_h	4-65
4-34	Using data_bus_req_h	4-66
4-35	READ MISS Completed First—Victim Buffer	4-68
4-36	READ MISS Second—No Victim Buffer	4-69
4-37	System Command to FILL Example 1	4-70

4-38	System Command to FILL Example 2	4-71
4-39	FILL to Private Read or Write Operation	4-72
4-40	Two Commands, Auto DACK Disabled	4-73
4-41	Two Commands, Auto DACK Enabled	4-73
4-42	sysclk Ratio = 4	4-74
4-43	sysclk Ratio = 3	4-75
4-44	READ MISS with Victim Example	4-80
4-45	idle_bc_h and cack_h Race Examples	4-81
4-46	READ MISS with idle_bc_h Asserted Example	4-82
4-47	READ MISS with Victim Abort Example	4-83
4-48	Bcache Hit Under READ MISS Example	4-84
4-49	ECC Code	4-85
4-50	21164 Interrupt Signals	4-88
5-1	Istream Translation Buffer Tag Register (ITB_TAG)	5-5
5-2	Instruction Translation Buffer Page Table Entry (ITB_PTE) Register Write Format	5-6
5-3	Instruction Translation Buffer Page Table Entry (ITB_PTE) Register Read Format	5-7
5-4	Instruction Translation Buffer Address Space Number (ITB_ASN) Register	5-7
5-5	Instruction Translation Buffer IS (ITB_IS) Register	5-9
5-6	Formatted Faulting Virtual Address (IFault_VA_Form) Register (NT_Mode=0)	5-10
5-7	Formatted Faulting Virtual Address (IFault_VA_Form) Register (NT_Mode=1)	5-10
5-8	Virtual Page Table Base Register (IVPTBR) (NT_Mode=0)	5-11
5-9	Virtual Page Table Base Register (IVPTBR) (NT_Mode=1)	5-11
5-10	Icache Parity Error Status (ICPERR_STAT) Register	5-12
5-11	Exception Address (EXC_ADDR) Register	5-13
5-12	Exception Summary (EXC_SUM) Register	5-14
5-13	Exception Mask (EXC_MASK) Register	5-15
5-14	PAL Base Address (PAL_BASE) Register	5-16
5-15	IDU Current Mode (ICM) Register	5-16
5-16	IDU Control and Status Register (ICSR)	5-17
5-17	Interrupt Priority Level Register (IPLR)	5-19
5-18	Interrupt ID (INTID) Register	5-20
5-19	Asynchronous System Trap Request Register (ASTRR)	5-21
5-20	Asynchronous System Trap Enable Register (ASTER)	5-21
5-21	Software Interrupt Request Register (SIRR)	5-22
5-22	Hardware Interrupt Clear (HWINT_CLR) Register	5-23
5-23	Interrupt Summary Register (ISR)	5-24
5-24	Serial Line Transmit (SL_XMIT) Register	5-26
5-25	Serial Line Receive (SL_RCV) Register	5-27
5-26	Performance Counter (PMCTR) Register	5-28
5-27	Dstream Translation Buffer Address Space Number (DTB_ASN) Register	5-33
5-28	Dstream Translation Buffer Current Mode (DTB_CM) Register	5-33
5-29	Dstream Translation Buffer Tag (DTB_TAG) Register	5-34
5-30	Dstream Translation Buffer Page Table Entry (DTB_PTE) Register—Write Format	5-35
5-31	Dstream Translation Buffer Page Table Entry Temporary (DTB_PTE_TEMP) Register	5-36
5-32	Dstream Memory Management Fault Status (MM_STAT) Register	5-37
5-33	Faulting Virtual Address (VA) Register	5-38

5-34	Formatted Virtual Address (VA_FORM) Register (NT_Mode=1)	5-39
5-35	Formatted Virtual Address (VA_FORM) Register (NT_Mode=0)	5-39
5-36	MTU Virtual Page Table Base Register (MVPTBR)	5-40
5-37	Dcache Parity Error Status (DC_PERR_STAT) Register	5-41
5-38	Dstream Translation Buffer Invalidate Single (DTB_IS) Register	5-43
5-39	MTU Control Register (MCSR)	5-44
5-40	Dcache Mode (DC_MODE) Register	5-46
5-41	Miss Address File Mode (MAF_MODE) Register	5-48
5-42	Alternate Mode (ALT_MODE) Register	5-50
5-43	Cycle Counter (CC) Register	5-51
5-44	Cycle Counter Control (CC_CTL) Register	5-52
5-45	Dcache Test Tag Control (DC_TEST_CTL) Register	5-53
5-46	Dcache Test Tag (DC_TEST_TAG) Register	5-54
5-47	Dcache Test Tag Temporary (DC_TEST_TAG_TEMP) Register	5-56
5-48	Scache Control (SC_CTL) Register	5-59
5-49	Scache Status (SC_STAT) Register	5-62
5-50	Scache Address (SC_ADDR) Register	5-66
5-51	Bcache Control (BC_CONTROL) Register	5-69
5-52	Bcache Configuration (BC_CONFIG) Register	5-75
5-53	Bcache Tag Address (BC_TAG_ADDR) Register	5-78
5-54	External Interface Status (EI_STAT) Register	5-81
5-55	External Interface Address (EI_ADDR) Register	5-83
5-56	Fill Syndrome (FILL_SYN) Register	5-84
6-1	HW_LD Instruction Format	6-9
6-2	HW_ST Instruction Format	6-10
6-3	HW_REI Instruction Format	6-11
6-4	HW_MTPR and HW_MFPR Instruction Format	6-12
9-1	osc_clk_in_h,l Input Network and Terminations	9-6
9-2	Impedance vs Clock Input Frequency	9-8
9-3	Input/Output Pin Timing	9-10
9-4	Bcache Timing	9-14
9-5	sys_clk System Timing	9-17
9-6	ref_clk System Timing	9-19
9-7	BiSt Timing Event—Time Line	9-25
9-8	SROM Load Timing Event—Time Line	9-26
9-9	Serial ROM Load Timing	9-27
10-1	Type 1 Heat Sink	10-3
10-2	Type 2 Heat Sink	10-4
11-1	Package Dimensions	11-2
11-2	21164 Top View (Pin Down)	11-8
11-3	21164 Bottom View (Pin Up)	11-9
12-1	IEEE1149.1 Test Access Port	12-3
12-2	TAP Controller State Machine	12-4

Tables

2-1	Effect of Branching Instructions on the Branch—Prediction Stack	2-7
2-2	Pipeline Examples—All Cases	2-16
2-3	Pipeline Examples—Integer Add	2-16
2-4	Pipeline Examples—Floating Add.	2-16
2-5	Pipeline Examples—Load (Dcache Hit)	2-17
2-6	Pipeline Examples—Load (Dcache Miss).	2-17
2-7	Pipeline Examples—Store (Dcache Hit)	2-18
2-8	Instruction Classes and Slotting	2-20
2-9	Instruction Latencies	2-25
2-10	Floating-Point Control Register Bit Descriptions	2-39
3-1	21164 Signal Descriptions	3-4
3-2	21164 Signal Descriptions by Function.	3-16
4-1	CPU Clock Generation Control	4-5
4-2	System Clock Divisor	4-6
4-3	System Clock Delay	4-8
4-4	Physical Memory Regions.	4-12
4-5	Components for 21164 Write Invalidate Systems.	4-20
4-6	Bcache States for Cache Coherency Protocols	4-21
4-7	Components for 21164 Flush Cache Protocol Systems	4-23
4-8	Bcache Options.	4-34
4-9	21164-Initiated Interface Commands	4-36
4-10	System-Initiated Interface Commands (Write Invalidate Protocol)	4-51
4-11	21164 Responses on addr_res_h<1:0> to Write Invalidate Protocol Commands	4-53
4-12	21164 Responses on addr_res_h<2> to 21164 Commands	4-53
4-13	21164 Minimum Response Time to Write Invalidate Protocol Commands.	4-54
4-14	System-Initiated Interface Commands (Flush Protocol)	4-58
4-15	21164 Responses to Flush-Based Protocol Commands	4-59
4-16	21164 Responses on addr_res_h<2> to 21164 Commands	4-59
4-17	Minimum 21164 Response Time to Flush Protocol Commands.	4-60
4-18	Data Check Bit Correspondence to CBn.	4-86
4-19	Interrupt Priority Level Effect.	4-89
5-1	IDU, MTU, Dcache, and PALtemp IPR Encodings	5-1
5-2	Granularity Hint Bits in ITB_PTE_TEMP Read Format.	5-8
5-3	Icache Parity Error Status Register Fields	5-12
5-4	Exception Summary Register Fields	5-14
5-5	IDU Control and Status Register Fields	5-17
5-6	Software Interrupt Request Register Fields	5-22
5-7	Hardware Interrupt Clear Register Fields	5-23
5-8	Interrupt Summary Register Fields	5-24
5-9	Serial Line Transmit Register Fields	5-26
5-10	Serial Line Receive Register Fields	5-27
5-11	Performance Counter Register Fields	5-29
5-12	PMCTR Counter Select Options	5-30
5-13	Measurement Mode Control	5-32

5-14	Dstream Memory Management Fault Status Register Fields	5-37
5-15	Formatted Virtual Address Register Fields	5-40
5-16	Dcache Parity Error Status Register Fields.	5-42
5-17	MTU Control Register Fields.	5-45
5-18	Dcache Mode Register Fields.	5-47
5-19	Miss Address File Mode Register Fields.	5-49
5-20	Alternate Mode Register Settings	5-50
5-21	Cycle Counter Control Register Fields	5-52
5-22	Dcache Test Tag Control Register Fields.	5-53
5-23	Dcache Test Tag Register Fields	5-55
5-24	Dcache Test Tag Temporary Register Fields	5-57
5-25	CBU Internal Processor Register Descriptions	5-58
5-26	Scache Control Register Fields.	5-60
5-27	Scache Status Register Fields	5-63
5-28	SC_CMD Field Descriptions	5-64
5-29	Scache Address Register Fields.	5-67
5-30	Bcache Control Register Fields.	5-70
5-31	Bcache Configuration Register Fields.	5-75
5-32	Bcache Tag Address Register Fields	5-79
5-33	Loading and Locking Rules for External Interface Registers	5-81
5-34	EI_STAT Register Fields	5-82
5-35	Syndromes for Single-Bit Errors	5-85
5-36	CBU IPR PALcode Restrictions	5-88
5-37	PALcode Restrictions Table	5-89
6-1	PALcode Trap Entry Points.	6-6
6-2	Required PALcode Function Codes	6-7
6-3	Opcodes Reserved for PALcode	6-8
6-4	HW_LD Format Description	6-9
6-5	HW_ST Format Description	6-10
6-6	HW_REI Format Description.	6-11
6-7	HW_MTPR and HW_MFPR Format Description	6-12
7-1	21164 Signal Pin Reset State	7-3
7-2	Internal Processor Register Reset State.	7-10
9-1	21164 Absolute Maximum Ratings	9-1
9-2	Operating Voltages	9-2
9-3	CMOS DC Input/Output Characteristics	9-3
9-4	Input Clock Specification.	9-9
9-5	Bcache Loop Timing	9-12
9-6	Normal Output Driver Characteristics	9-13
9-7	Big Output Driver Characteristics	9-13
9-8	21164 System Clock Output Timing (sysclk=T _o)	9-15
9-9	21164 Reference Clock Input Timing	9-18
9-10	ref_clk System Timing Stages.	9-19
9-11	Input Timing for sys_clk_out- or ref_clk_in-Based Systems	9-21
9-12	Output Timing for sys_clk_out- or ref_clk_in-Based Systems.	9-22
9-13	Bcache Control Signal Timing.	9-24
9-14	BiSt Timing for Some System Clock Ratios, Port Mode=Normal (System Cycles)	9-25

9-15	BiSt Timing for Some System Clock Ratios, Port Mode=Normal (CPU Cycles) . .	9-26
9-16	SRAM Load Timing for Some System Clock Ratios (System Cycles)	9-26
9-17	SRAM Load Timing for Some System Clock Ratios (CPU Cycles)	9-27
9-18	Clock Test Modes	9-28
9-19	IEEE 1149.1 Circuit Performance Specifications	9-29
10-1	Θ_{ca} at Various Airflows	10-1
10-2	Maximum T_a at Various Airflows	10-2
11-1	Alphabetic Signal Pin List	11-3
12-1	21164 Test Port Pins	12-1
12-2	Compliance Enable Inputs	12-2
12-3	Instruction Register	12-5
12-4	Boundary-Scan Register Organization	12-7
A-1	Instruction Format and Opcode Notation	A-1
A-2	Architecture Instructions	A-2
A-3	Opcodes Reserved	A-9
A-4	Opcodes Reserved for PALcode	A-9
A-5	IEEE Floating-Point Instruction Function Codes	A-10
A-6	VAX Floating-Point Instruction Function Codes	A-12
A-7	Opcode Summary	A-13
A-8	Required PALcode Function Codes	A-14
B-1	21164 Microprocessor Specifications	B-1

Preface

This manual provides information about the architecture, internal design, external interface, and specifications of the 21164 Alpha microprocessor (referred to as the 21164) and its associated software.

Audience

This reference manual is for system designers and programmers who use the 21164 .

Manual Organization

This manual includes the following chapters and appendixes, and an index.

- Chapter 1, Introduction, introduces the 21164 and provides an overview of the Alpha architecture.
- Chapter 2, Internal Architecture, describes the major hardware functions and the internal chip architecture. It describes performance measurement facilities, coding rules, and design examples.
- Chapter 3, Hardware Interface, lists and describes the external hardware interface signals.
- Chapter 4, Clocks, Cache, and External Interface, describes the external bus functions and transactions, lists bus commands, and describes the clock functions.
- Chapter 5, Internal Processor Registers, lists and describes the 21164 internal processor register set.
- Chapter 6, Privileged Architecture Library Code, describes the privileged architecture library code (PALcode).
- Chapter 7, Initialization and Configuration, describes the initialization and configuration sequence.
- Chapter 8, Error Detection and Error Handling, describes error detection and error handling.

- Chapter 9, Electrical Data, provides electrical data and describes signal integrity issues.
- Chapter 10, Thermal Management, provides information about thermal management.
- Chapter 11, Mechanical Data and Packaging Information, provides mechanical data and packaging information, including signal pin lists.
- Chapter 12, Testability and Diagnostics, describes chip and system testability features.
- Appendix A, Alpha Instruction Set, summarizes the Alpha instruction set.
- Appendix B, 21164 Microprocessor Specifications, summarizes the 21164 specifications.
- Appendix C, Serial Icache Load Predecode Values, provides a C code example that calculates the predecode values of a serial Icache load.
- Appendix D, Errata Sheet, lists changes and revisions to this manual.
- The Glossary lists and defines terms associated with the 21164.

The companion volume to this manual, the *Alpha AXP Architecture Reference Manual*, contains the Alpha architecture information.

Conventions

This section defines product-specific terminology, abbreviations, and other conventions used throughout this manual.

Abbreviations

- Binary Multiples

The abbreviations K, M, and G (kilo, mega, and giga) represent binary multiples and have the following values.

K	=	2^{10} (1024)
M	=	2^{20} (1,048,576)
G	=	2^{30} (1,073,741,824)

For example:

2KB = 2 kilobytes = 2×2^{10} bytes
4MB = 4 megabytes = 4×2^{20} bytes
8GB = 8 gigabytes = 8×2^{30} bytes

- Register Access

The abbreviations used to indicate the type of access to register fields and bits have the following definitions:

IGN — Ignore

Register bits specified as IGN are ignored when written and are UNPREDICTABLE when read if not otherwise specified.

MBZ — Must Be Zero

Software must never place a nonzero value in bits and fields specified as MBZ. Reads return unpredictable values. Such fields are reserved for future use.

RAO — Read As One

Register bits specified as RAO return a 1 when read.

RAZ — Read As Zero

Register bits specified as RAZ return a 0 when read.

RC — Read To Clear

A register field specified as RC is written by hardware and remains unchanged until read. The value may be read by software, at which point, hardware may write a new value into the field.

RES — Reserved

Bits and fields specified as RES are reserved and should not be used; however, zeros can be written to reserved fields that cannot be masked.

RO — Read Only

Bits and fields specified as RO can be read and are ignored (not written) on writes.

RW — Read/Write

Bits and fields specified as RW can be read and written.

W0C — Write Zero to Clear

Bits and fields specified as W0C can be read. Writing a zero clears these bits for the duration of the write; writing a one has no effect.

W1C — Write One to Clear

Bits and fields specified as W1C can be read. Writing a one clears these bits for the duration of the write; writing a zero has no effect.

WO — Write Only

Bits and fields specified as WO can be written but not read.

Addresses

Unless otherwise noted, all addresses and offsets are hexadecimal.

Aligned and Unaligned

The terms *aligned* and *naturally aligned* are interchangeable and refer to data objects that are powers of two in size. An aligned datum of size 2^n is stored in memory at a byte address that is a multiple of 2^n ; that is, one that has n low-order zeros. For example, an aligned 64-byte stack frame has a memory address that is a multiple of 64.

A datum of size 2^n is *unaligned* if it is stored in a byte address that is not a multiple of 2^n .

Bit Notation

Multiple-bit fields can include contiguous and noncontiguous bits contained in angle brackets (<>). Multiple contiguous bits are indicated by a pair of numbers separated by a colon (:). For example, <9:7,5,2:0> specifies bits 9,8,7,5,2,1, and 0. Similarly, single bits are frequently indicated with angle brackets. For example, <27> specifies bit 27.

Caution

Cautions indicate potential damage to equipment or loss of data.

Data Units

The following data unit terminology is used throughout this manual.

Term	Words	Bytes	Bits	Other
Byte	1/2	1	8	—
Word	1	2	16	—
Dword	2	4	32	Longword
Quadword	4	8	64	2 Dwords

External

Unless otherwise stated, external means not contained in the 21164.

Numbering

All numbers are decimal or hexadecimal unless otherwise indicated. The prefix 0x indicates a hexadecimal number. For example, 19 is decimal, but 0x19 and 0x19A are hexadecimal (also see Addresses). Otherwise, the base is indicated by a subscript; for example, 100₂ is a binary number.

Ranges and Extents

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

Extents are specified by a pair of numbers in angle brackets (<>) separated by a colon (:), and are inclusive. Bit fields are often specified as extents. For example, bits <7:3> specifies bits 7, 6, 5, 4, and 3.

Security Holes

Security holes exist when unprivileged software (that is, software that is running outside of kernel mode) can:

- Affect the operation of another process without authorization from the operating system.
- Amplify its privilege without authorization from the operating system.
- Communicate with another process, either overtly or covertly, without authorization from the operating system.

Signal Names

Signal names are printed in lowercase, boldface type. Low-asserted signals are indicated by the **_l** suffix, while high-asserted signals have the **_h** suffix. For example, **osc_clk_in_h** is a high-asserted signal, and **osc_clk_in_l** is a low-asserted signal.

Unpredictable and Undefined

Throughout this manual, the terms UNPREDICTABLE and UNDEFINED are used. Their meanings are quite different and must be carefully distinguished.

In particular, only privileged software (that is, software running in kernel mode) can trigger UNDEFINED operations. Unprivileged software cannot trigger UNDEFINED operations. However, either privileged or unprivileged software can trigger UNPREDICTABLE results or occurrences.

UNPREDICTABLE results or occurrences do not disrupt the basic operation of the processor. The processor continues to execute instructions in its normal manner. In contrast, UNDEFINED operations can halt the processor or cause it to lose information.

The terms UNPREDICTABLE and UNDEFINED can be further described as follows:

Unpredictable

- Results or occurrences specified as UNPREDICTABLE may vary from moment to moment, implementation to implementation, and instruction to instruction within implementations. Software can never depend on results specified as UNPREDICTABLE.
- An UNPREDICTABLE result may acquire an arbitrary value subject to a few constraints. Such a result may be an arbitrary function of the input operands or of any state information that is accessible to the process in its current access mode. UNPREDICTABLE results may be unchanged from their previous values.

Operations that produce UNPREDICTABLE results may also produce exceptions.

- An occurrence specified as UNPREDICTABLE may happen or not based on an arbitrary choice function. The choice function is subject to the same constraints as are UNPREDICTABLE results and, in particular, must not constitute a security hole.

Specifically, UNPREDICTABLE results must not depend upon, or be a function of the contents of memory locations or registers that are inaccessible to the current process in the current access mode.

Also, operations that may produce UNPREDICTABLE results must not:

- Write or modify the contents of memory locations or registers to which the current process in the current access mode does not have access.
- Halt or hang the system or any of its components.

For example, a security hole would exist if some UNPREDICTABLE result depended on the value of a register in another process, on the contents of processor temporary registers left behind by some previously running process, or on a sequence of actions of different processes.

Undefined

- Operations specified as UNDEFINED may vary from moment to moment, implementation to implementation, and instruction to instruction within implementations. The operation may vary in effect from nothing, to stopping system operation.
- UNDEFINED operations may halt the processor or cause it to lose information. However, UNDEFINED operations must not cause the processor to hang, that is, reach an unhalted state from which there is no transition to a normal state in which the machine executes instructions. Only privileged software (that is, software running in kernel mode) may trigger UNDEFINED operations.

