

Pentium™ Family
User's Manual



Volume 3: Architecture and
Programming Manual

One good thing

1979 - 8086 and 8088 CPU

leads to another...



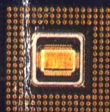
1982 - 80286 CPU

and another...



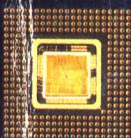
1985 - Intel386™ CPU

and another...



1989 - Intel486™ CPU

and another...



1993 - Pentium™ Processor

1994 - Pentium™ Processor
(90 & 100 MHz)



and another...and another...

奔腾™ 系列用户手册

第三卷 结构与程序设计

(英文版)

上海科学普及出版社

Pentium™ Processor Family User's Manual

Volume 3: Architecture and Programming Manual

NOTE: The *Pentium™ Processor Family User's Manual* consists of three books: *Pentium™ Processor Family Data Book*, Order Number 241428; the *82496/82497 Cache Controller and 82491/82492 Cache SRAM Data Book*, Order Number 241429; and the *Architecture and Programming Manual*, Order Number 241430. Please refer to all three volumes when evaluating your design needs.

1994

A thick, solid black vertical bar located at the bottom right of the page.

(沪)新登字第 305 号

责任编辑 胡名正

©上海科学普及出版社 1994

©1994 年 Intel 公司英文版的授权重印本。

本英文版重印本的出版和发行获得上述英文版的一切出版和发行权的所有者——Intel 公司的许可。上海科学普及出版社拥有上述英文版在中华人民共和国(除台湾省外)的独家重印、出版、发行权。任何人不得再对上述英文版进行翻印、改编、出版和发行。

奔腾™ 系列用户手册

第三卷 结构与程序设计手册

(英文版)

上海科学普及出版社出版发行

(上海曹杨路 500 号 邮政编码 200063)

上海竟成印刷厂印刷

开本 787×960 1/16 印张 65

1994 年 10 月第 1 版 1994 年 10 月第 1 次印刷

ISBN 7-5427-0932-1/TP·223 定价: 135.00 元

TABLE OF CONTENTS

CHAPTER 1

GETTING STARTED

	Page
1.1. HOW TO USE THIS MANUAL	1-1
1.1.1. Part I—Application and Numeric Programming	1-2
1.1.2. Part II—System Programming	1-2
1.1.3. Part III—Compatibility	1-4
1.1.4. Part IV—Optimization	1-4
1.1.5. Part V—Instruction Set	1-4
1.1.6. Appendices	1-5
1.2. RELATED LITERATURE	1-5
1.3. NOTATIONAL CONVENTIONS	1-5
1.3.1. Bit and Byte Order	1-6
1.3.2. Undefined Bits and Software Compatibility	1-6
1.3.3. Instruction Operands	1-7
1.3.4. Hexadecimal Numbers	1-7
1.3.5. Segmented Addressing	1-8
1.3.6. Exceptions	1-8

CHAPTER 2

INTRODUCTION TO THE INTEL PENTIUM PROCESSOR FAMILY

PART I — APPLICATION AND NUMERIC PROGRAMMING

CHAPTER 3

BASIC PROGRAMMING MODEL

3.1. MEMORY ORGANIZATION	3-1
3.1.1. Unsegmented or "Flat" Model	3-2
3.1.2. Segmented Model	3-3
3.2. DATA TYPES	3-4
3.3. REGISTERS	3-8
3.3.1. General Registers	3-8
3.3.2. Segment Registers	3-10
3.3.3. Stack Implementation	3-13
3.3.4. Flags Register	3-14
3.3.4.1. STATUS FLAGS	3-14
3.3.4.2. CONTROL FLAG	3-16
3.3.5. Instruction Pointer	3-16
3.4. INSTRUCTION FORMAT	3-16

	Page
3.5. OPERAND SELECTION	3-18
3.5.1. Immediate Operands	3-19
3.5.2. Register Operands	3-20
3.5.3. Memory Operands	3-20
3.5.3.1. SEGMENT SELECTION	3-21
3.5.3.2. EFFECTIVE-ADDRESS COMPUTATION	3-21
3.6. INTERRUPTS AND EXCEPTIONS	3-24

CHAPTER 4

APPLICATION PROGRAMMING

4.1. DATA MOVEMENT INSTRUCTIONS	4-1
4.1.1. General-Purpose Data Movement Instructions	4-1
4.1.2. Stack Manipulation Instructions	4-2
4.1.3. Type Conversion Instructions	4-5
4.2. BINARY ARITHMETIC INSTRUCTIONS	4-6
4.2.1. Addition and Subtraction Instructions	4-7
4.2.2. Comparison and Sign Change Instruction	4-8
4.2.3. Multiplication Instructions	4-8
4.2.4. Division Instructions	4-9
4.3. DECIMAL ARITHMETIC INSTRUCTIONS	4-10
4.3.1. Packed BCD Adjustment Instructions	4-10
4.3.2. Unpacked BCD Adjustment Instructions	4-10
4.4. LOGICAL INSTRUCTIONS	4-11
4.4.1. Boolean Operation Instructions	4-11
4.4.2. Bit Test and Modify Instructions	4-12
4.4.3. Bit Scan Instructions	4-12
4.4.4. Shift and Rotate Instructions	4-13
4.4.4.1. SHIFT INSTRUCTIONS	4-13
4.4.4.2. DOUBLE-SHIFT INSTRUCTIONS	4-16
4.4.4.3. ROTATE INSTRUCTIONS	4-17
4.4.4.4. FAST "bit bit" USING DOUBLE-SHIFT INSTRUCTIONS	4-19
4.4.4.5. FAST BIT STRING INSERT AND EXTRACT	4-19
4.4.5. Byte-Set-On-Condition Instructions	4-23
4.4.6. Test Instruction	4-23
4.5. CONTROL TRANSFER INSTRUCTIONS	4-23
4.5.1. Unconditional Transfer Instructions	4-23
4.5.1.1. JUMP INSTRUCTION	4-24
4.5.1.2. CALL INSTRUCTIONS	4-24
4.5.1.3. RETURN AND RETURN-FROM-INTERRUPT INSTRUCTIONS	4-24
4.5.2. Conditional Transfer Instructions	4-25
4.5.2.1. CONDITIONAL JUMP INSTRUCTIONS	4-25
4.5.2.2. LOOP INSTRUCTIONS	4-26
4.5.2.3. EXECUTING A LOOP OR REPEAT ZERO TIMES	4-27
4.5.3. Software Interrupts	4-27
4.6. STRING OPERATIONS	4-28
4.6.1. Repeat Prefixes	4-29
4.6.2. Indexing and Direction Flag Control	4-30
4.6.3. String Instructions	4-30
4.7. INSTRUCTIONS FOR BLOCK-STRUCTURED LANGUAGES	4-31

	Page
4.8. FLAG CONTROL INSTRUCTIONS	4-38
4.8.1. Carry and Direction Flag Control Instructions	4-38
4.8.2. Flag Transfer Instructions	4-38
4.9. NUMERIC INSTRUCTIONS	4-40
4.10. SEGMENT REGISTER INSTRUCTIONS	4-40
4.10.1. Segment-Register Transfer Instructions	4-41
4.10.2. Far Control Transfer Instructions	4-41
4.10.3. Data Pointer Instructions	4-41
4.11. MISCELLANEOUS INSTRUCTIONS	4-42
4.11.1. Address Calculation Instruction	4-43
4.11.2. No-Operation Instruction	4-43
4.11.3. Translate Instruction	4-43
4.11.4. Byte Swap Instruction	4-43
4.11.5. Exchange-and-Add Instruction	4-45
4.11.6. Compare-and-Exchange Instructions	4-45
4.11.7. CPUID Instruction	4-46

CHAPTER 5

FEATURE DETERMINATION

5.1. CPU IDENTIFICATION	5-1
5.2. FPU DETECTION	5-2
5.3. SAMPLE CPUID IDENTIFICATION/FPU DETECTION CODE	5-2

CHAPTER 6

NUMERIC APPLICATIONS

6.1. INTRODUCTION TO NUMERIC APPLICATIONS	6-1
6.1.1. History	6-1
6.1.2. Performance	6-2
6.1.3. Ease of Use	6-2
6.1.4. Applications	6-4
6.1.5. Programming Interface	6-5
6.2. ARCHITECTURE OF THE FLOATING-POINT UNIT	6-7
6.2.1. Numerical Registers	6-7
6.2.1.1. THE FPU REGISTER STACK	6-8
6.2.1.2. THE FPU STATUS WORD	6-9
6.2.1.3. CONTROL WORD	6-11
6.2.1.4. THE FPU TAG WORD	6-14
6.2.1.5. OPCODE FIELD OF LAST INSTRUCTION	6-15
6.2.1.6. THE NUMERIC INSTRUCTION AND DATA POINTERS	6-16
6.2.2. Computation Fundamentals	6-19
6.2.2.1. NUMBER SYSTEM	6-19
6.2.2.2. DATA TYPES AND FORMATS	6-21
6.2.2.2.1. Binary Integers	6-21
6.2.2.2.2. Decimal Integers	6-21
6.2.2.2.3. Real Numbers	6-23
6.2.2.3. ROUNDING CONTROL	6-25
6.2.2.4. PRECISION CONTROL	6-26

	Page
6.3. FLOATING-POINT INSTRUCTION SET	6-26
6.3.1. Source and Destination Operands	6-27
6.3.2. Data Transfer Instructions	6-27
6.3.3. Nontranscendental Instructions	6-28
6.3.4. Comparison Instructions	6-30
6.3.5. Transcendental Instructions	6-31
6.3.6. Constant Instructions	6-33
6.3.7. Control Instructions	6-33
6.4. NUMERIC APPLICATIONS	6-35
6.4.1. High-Level Languages	6-36
6.4.1.1. C PROGRAMS	6-36
6.4.1.2. PL/M-386/486	6-36
6.4.1.3. ASM386/486	6-38
6.4.1.3.1. Defining Data	6-38
6.4.1.3.2. Records and Structures	6-40
6.4.1.3.3. Addressing Methods	6-41
6.4.1.4. COMPARATIVE PROGRAMMING EXAMPLE	6-42
6.4.1.5. CONCURRENT PROCESSING	6-47
6.4.1.6. MANAGING CONCURRENCY	6-47
6.4.1.7. EXCEPTION SYNCHRONIZATION	6-49
6.4.1.8. PROPER EXCEPTION SYNCHRONIZATION	6-49

CHAPTER 7

SPECIAL COMPUTATIONAL SITUATIONS

7.1. SPECIAL NUMERIC VALUES	7-1
7.1.1. Denormal Real Numbers	7-7
7.1.2. Zeros	7-9
7.1.3. Infinity	7-9
7.1.4. NaN (Not-a-Number)	7-15
7.1.4.1. SIGNALING NaNs	7-16
7.1.4.2. QUIET NaNs	7-16
7.1.5. Indefinite	7-17
7.1.6. Encoding of Data Types	7-18
7.1.6.1. UNSUPPORTED FORMATS	7-18
7.1.7. Numeric Exceptions	7-18
7.1.8. Handling Numeric Exceptions	7-19
7.1.8.1. AUTOMATIC EXCEPTION HANDLING	7-19
7.1.8.2. SOFTWARE EXCEPTION HANDLING	7-20
7.1.9. Invalid Operation	7-21
7.1.9.1. STACK EXCEPTION	7-22
7.1.9.2. INVALID ARITHMETIC OPERATION	7-22
7.1.10. Division by Zero	7-23
7.1.11. Denormal Operand	7-23
7.1.12. Numeric Overflow and Underflow	7-24
7.1.12.1. OVERFLOW	7-24
7.1.12.2. UNDERFLOW	7-26
7.1.13. Inexact (Precision)	7-27
7.1.14. Exception Priority	7-28
7.1.15. Standard Underflow/Overflow Exception Handler	7-28

CHAPTER 8
NUMERIC PROGRAMMING EXAMPLES

	Page
8.1. CONDITIONAL BRANCHING EXAMPLE.....	8-1
8.2. EXCEPTION HANDLING EXAMPLES.....	8-4
8.3. FLOATING POINT TO ASCII CONVERSION EXAMPLES.....	8-7
8.3.1. Function Partitioning.....	8-22
8.3.2. Exception Considerations.....	8-22
8.3.3. Special Instructions.....	8-22
8.3.4. Description of Operation.....	8-23
8.3.5. Scaling the Value.....	8-24
8.3.5.1. INACCURACY IN SCALING.....	8-24
8.3.5.2. AVOIDING UNDERFLOW AND OVERFLOW.....	8-24
8.3.5.3. FINAL ADJUSTMENTS.....	8-25
8.3.6. Output Format.....	8-25
8.4. TRIGONOMETRIC CALCULATION EXAMPLES.....	8-25

PART II — SYSTEM PROGRAMMING

CHAPTER 9
REAL-ADDRESS MODE SYSTEM ARCHITECTURE

9.1. ADDRESS TRANSLATION.....	9-1
9.2. REGISTERS AND INSTRUCTIONS.....	9-2
9.3. INTERRUPT AND EXCEPTION HANDLING.....	9-3
9.4. REAL-ADDRESS MODE EXCEPTIONS.....	9-3

CHAPTER 10
PROTECTED-MODE SYSTEM ARCHITECTURE OVERVIEW

10.1. SYSTEM REGISTERS.....	10-1
10.1.1. System Flags.....	10-2
10.1.2. Memory-Management Registers.....	10-4
10.1.3. Control Registers.....	10-5
10.1.4. Debug Registers.....	10-9
10.2. SYSTEM INSTRUCTIONS.....	10-11

CHAPTER 11
PROTECTED-MODE MEMORY MANAGEMENT

11.1. SELECTING A SEGMENTATION MODEL.....	11-2
11.1.1. Flat Model.....	11-3
11.1.2. Protected Flat Model.....	11-4
11.1.3. Multisegment Model.....	11-5
11.2. SEGMENT TRANSLATION.....	11-6
11.2.1. Segment Registers.....	11-9
11.2.2. Segment Selectors.....	11-10
11.2.3. Segment Descriptors.....	11-11
11.2.4. Segment Descriptor Tables.....	11-15
11.2.5. Descriptor Table Base Registers.....	11-16

	Page
11.3. PAGE TRANSLATION	11-17
11.3.1. Paging Options	11-18
11.3.2. Linear Address	11-18
11.3.3. Page Tables	11-19
11.3.4. Page-Table Entries	11-19
11.3.4.1. PAGE FRAME ADDRESS	11-20
11.3.4.2. PRESENT BIT	11-20
11.3.4.3. ACCESSED AND DIRTY BITS	11-21
11.3.4.4. READ/WRITE AND USER/SUPERVISOR BITS	11-22
11.3.4.5. PAGE-LEVEL CACHE CONTROL BITS	11-22
11.3.5. Translation Lookaside Buffers	11-22
11.4. COMBINING SEGMENT AND PAGE TRANSLATION	11-23
11.4.1. Flat Model	11-23
11.4.2. Segments Spanning Several Pages	11-23
11.4.3. Pages Spanning Several Segments	11-23
11.4.4. Non-Aligned Page and Segment Boundaries	11-24
11.4.5. Aligned Page and Segment Boundaries	11-24
11.4.6. Page-Table Per Segment	11-25

CHAPTER 12 PROTECTION

12.1. SEGMENT-LEVEL PROTECTION	12-1
12.2. SEGMENT DESCRIPTORS AND PROTECTION	12-2
12.2.1. Type Checking	12-2
12.2.2. Limit Checking	12-5
12.2.3. Privilege Levels	12-6
12.3. RESTRICTING ACCESS TO DATA	12-8
12.3.1. Accessing Data in Code Segments	12-10
12.4. RESTRICTING CONTROL TRANSFERS	12-10
12.5. GATE DESCRIPTORS	12-12
12.5.1. Stack Switching	12-15
12.5.2. Returning from a Procedure	12-18
12.6. INSTRUCTIONS RESERVED FOR THE OPERATING SYSTEM	12-20
12.6.1. Privileged Instructions	12-20
12.6.2. Sensitive Instructions	12-21
12.7. INSTRUCTIONS FOR POINTER VALIDATION	12-21
12.7.1. Descriptor Validation	12-23
12.7.2. Pointer Integrity and RPL	12-23
12.8. PAGE-LEVEL PROTECTION	12-24
12.8.1. Page-Table Entries Hold Protection Parameters	12-24
12.8.1.1. RESTRICTING ADDRESSABLE DOMAIN	12-25
12.8.1.2. TYPE CHECKING	12-25
12.8.2. Combining Protection of Both Levels of Page Tables	12-26
12.8.3. Overrides to Page Protection	12-26
12.9. COMBINING PAGE AND SEGMENT PROTECTION	12-27

CHAPTER 13
PROTECTED-MODE MULTITASKING

	Page
13.1. TASK STATE SEGMENT	13-2
13.2. TSS DESCRIPTOR	13-4
13.3. TASK REGISTER	13-5
13.4. TASK GATE DESCRIPTOR	13-6
13.5. TASK SWITCHING	13-8
13.6. TASK LINKING	13-11
13.6.1. Busy Bit Prevents Loops	13-13
13.6.2. Modifying Task Linkages	13-14
13.7. TASK ADDRESS SPACE	13-14
13.7.1. Task Linear-to-Physical Space Mapping	13-14
13.7.2. Task Logical Address Space	13-15

CHAPTER 14
PROTECTED-MODE EXCEPTIONS AND INTERRUPTS

14.1. EXCEPTION AND INTERRUPT VECTORS	14-1
14.2. INSTRUCTION RESTART	14-3
14.3. ENABLING AND DISABLING INTERRUPTS	14-3
14.3.1. NMI Masks Further NMIs	14-3
14.3.2. IF Masks INTR	14-4
14.3.3. RF Masks Debug Faults	14-4
14.3.4. MOV or POP to SS Masks Some Exceptions and Interrupts	14-5
14.4. PRIORITY AMONG SIMULTANEOUS EXCEPTIONS AND INTERRUPTS	14-5
14.5. INTERRUPT DESCRIPTOR TABLE	14-6
14.6. IDT DESCRIPTORS	14-8
14.7. INTERRUPT TASKS AND INTERRUPT PROCEDURES	14-9
14.7.1. Interrupt Procedures	14-10
14.7.1.1. STACK OF INTERRUPT PROCEDURE	14-11
14.7.1.2. RETURNING FROM AN INTERRUPT PROCEDURE	14-11
14.7.1.3. FLAG USAGE BY INTERRUPT PROCEDURE	14-12
14.7.1.4. PROTECTION IN INTERRUPT PROCEDURES	14-12
14.7.2. Interrupt Tasks	14-12
14.8. ERROR CODE	14-14
14.9. EXCEPTION CONDITIONS	14-14
14.9.1. Interrupt 0—Divide Error	14-15
14.9.2. Interrupt 1—Debug Exceptions	14-15
14.9.3. Interrupt 3—Breakpoint	14-15
14.9.4. Interrupt 4—Overflow	14-16
14.9.5. Interrupt 5—Bounds Check	14-16
14.9.6. Interrupt 6—Invalid Opcode	14-16
14.9.7. Interrupt 7—Device Not Available	14-16
14.9.8. Interrupt 8—Double Fault	14-17
14.9.9. Interrupt 9—(Intel reserved. Do not use.)	14-18
14.9.10. Interrupt 10—Invalid TSS	14-18
14.9.11. Interrupt 11—Segment Not Present	14-19
14.9.12. Interrupt 12—Stack Exception	14-20
14.9.13. Interrupt 13—General Protection	14-21
14.9.14. Interrupt 14—Page Fault	14-22
14.9.14.1. PAGE FAULT DURING TASK SWITCH	14-23
14.9.14.2. PAGE FAULT WITH INCONSISTENT STACK POINTER	14-23

	Page
14.9.15. Interrupt 16—Floating-Point Error	14-24
14.9.15.1. NUMERICS EXCEPTION HANDLING	14-25
14.9.15.2. SIMULTANEOUS EXCEPTION RESPONSE	14-26
14.9.16. Interrupt 17—Alignment Check	14-27
14.9.17. Interrupt 18—Machine Check	14-28
14.10. EXCEPTION SUMMARY	14-28
14.11. ERROR CODE SUMMARY	14-30

CHAPTER 15

INPUT/OUTPUT

15.1. I/O ADDRESSING	15-1
15.1.1. I/O Address Space	15-1
15.1.2. Memory-Mapped I/O	15-3
15.2. I/O INSTRUCTIONS	15-4
15.2.1. Register I/O Instructions	15-4
15.2.2. Block I/O Instructions	15-4
15.3. PROTECTED-MODE I/O	15-5
15.3.1. I/O Privilege Level	15-6
15.3.2. I/O Permission Bit Map	15-6
15.3.3. Paging and Caching	15-8
15.4. ORDERING OF I/O	15-8

CHAPTER 16

INITIALIZATION AND MODE SWITCHING

16.1. PROCESSOR INITIALIZATION	16-1
16.1.1. Processor State after Reset	16-2
16.1.2. First Instruction Executed	16-5
16.2. FPU INITIALIZATION	16-5
16.2.1. Configuring the Numerics Environment	16-6
16.2.2. FPU Software Emulation	16-8
16.3. CACHE ENABLING	16-9
16.4. SOFTWARE INITIALIZATION IN REAL-ADDRESS MODE	16-9
16.4.1. System Tables	16-10
16.4.2. NMI Interrupt	16-10
16.5. SOFTWARE INITIALIZATION IN PROTECTED MODE	16-10
16.5.1. System Tables	16-10
16.5.2. Interrupts	16-11
16.5.3. Paging	16-11
16.5.4. Tasks	16-12
16.5.5. TLB, BTB and Cache Testing	16-12
16.6. MODE SWITCHING	16-12
16.6.1. Switching to Protected Mode	16-13
16.6.2. Switching Back to Real-Address Mode	16-13
16.7. INITIALIZATION AND MODE SWITCHING EXAMPLE	16-14
16.7.1. Goal of this Example	16-14
16.7.2. Memory Layout Following Reset	16-14
16.7.3. The Algorithm	16-15
16.7.4. Tool Usage	16-17
16.7.5. STARTUP.ASM Listing	16-18
16.7.6. MAIN.ASM Source Code	16-26
16.7.7. Supporting Files	16-29

CHAPTER 17 DEBUGGING

17.1. DEBUGGING SUPPORT	17-1
17.2. DEBUG REGISTERS	17-2
17.2.1. Debug Address Registers (DR17-DR3)	17-3
17.2.2. Debug Control Register (DR7)	17-3
17.2.3. Debug Status Register (DR6)	17-4
17.2.4. Debug Registers DR4 and DR5	17-5
17.2.5. Breakpoint Field Recognition	17-5
17.3. DEBUG EXCEPTIONS	17-6
17.3.1. Interrupt 1—Debug Exceptions	17-6
17.3.1.1. INSTRUCTION-BREAKPOINT FAULT	17-7
17.3.1.2. DATA MEMORY AND I/O BREAKPOINTS	17-8
17.3.1.3. GENERAL-DETECT FAULT	17-8
17.3.1.4. SINGLE-STEP TRAP	17-9
17.3.1.5. TASK-SWITCH TRAP	17-9
17.3.2. Interrupt 3—Breakpoint Instruction	17-10

CHAPTER 18 CACHING, PIPELINING AND BUFFERING

18.1. INTERNAL INSTRUCTION AND DATA CACHES	18-1
18.1.1. Data Cache	18-2
18.1.2. Data Cache Update Policies	18-3
18.1.3. Instruction Cache	18-3
18.2. OPERATION OF THE INTERNAL CACHES	18-3
18.2.1. Cache Control Bits	18-4
18.2.2. Cache Management Instructions	18-4
18.2.3. Self-Modifying Code	18-5
18.3. PAGE-LEVEL CACHE MANAGEMENT	18-5
18.3.1. PCD Bit	18-6
18.3.2. PWT Bit	18-6
18.4. ADDRESS TRANSLATION CACHES	18-6
18.5. CACHE REPLACEMENT ALGORITHM	18-7
18.6. EXECUTION PIPELINING AND PAIRING	18-7
18.7. WRITE BUFFERS	18-7
18.8. SERIALIZING INSTRUCTIONS	18-7

CHAPTER 19 MULTIPROCESSING

19.1. LOCKED BUS CYCLES	19-1
19.1.1. LOCK Prefix and the LOCK# Signal	19-2
19.1.2. Automatic Locking	19-2
19.2. MEMORY ACCESS ORDERING	19-3
19.3. Pentium Processor (735/90, 815/100) INTEGRATED APIC	19-4
19.3.1. Interrupt Control Mechanism	19-6
19.3.1.1. VALID INTERRUPTS	19-6
19.3.1.2. INTERRUPT SOURCES	19-6
19.3.1.3. BUS ARBITRATION	19-6
19.3.1.4. THE LOCAL APIC STRUCTURE	19-7
19.3.1.5. INTERRUPT DESTINATION AND APIC ID	19-9

	Page
19.3.1.6. INTERRUPT DISTRIBUTION MODES	19-11
19.3.1.7. LOCAL VECTOR TABLE	19-12
19.3.1.8. INTER-PROCESSOR AND SELF INTERRUPTS	19-14
19.3.1.9. INTERRUPT ACCEPTANCE	19-18
19.3.1.9.1. Interrupt Acceptance Decision Flow Chart	19-19
19.3.1.9.2. Task Priority Register	19-21
19.3.1.9.3. Processor Priority Register (PPR)	19-21
19.3.1.9.4. Arbitration Priority Register (APR)	19-21
19.3.1.9.5. Spurious Interrupt	19-22
19.3.1.9.6. End-Of-Interrupt (EOI)	19-22
19.3.1.10. READING REMOTE APIC REGISTERS	19-22
19.3.1.11. LOCAL APIC STATE	19-23
19.3.1.11.1. Spurious Interrupt Vector Register	19-23
19.3.1.11.2. Local APIC Initialization	19-24
19.3.1.11.3. Local APIC State After Power-Up Reset and Init	19-24
19.3.1.12. LOCAL APIC VERSION REGISTER	19-25
19.3.2. APIC Bus And Inter-APIC Communication Protocol	19-26
19.3.2.1. BUS ARBITRATION	19-26
19.3.2.2. BUS MESSAGE FORMATS	19-26
19.3.3. Error Handling In APIC	19-33
19.3.4. Timer	19-34
19.3.4.1. OVERVIEW	19-34
19.3.5. APIC Valid/Invalid Programming Combination	19-38
19.3.6. Software Visible Differences Between APIC and 82489DX	19-39
19.3.7. Dual Processing Bootup Handshake Protocol Sequence With Examples	19-40

CHAPTER 20

POWER MANAGEMENT

20.1. PENTIUM™ PROCESSOR (510\60,567\66) POWER MANAGEMENT	20-1
20.1.1. Introduction to System Management Mode Architecture	20-1
20.1.2. Terminology	20-1
20.1.3. Pentium Processor System Management Interrupt Processing	20-2
20.1.3.1. SMRAM	20-2
20.1.3.2. SMRAM STATE SAVE MAP	20-4
20.1.3.3. EXIT FROM SMM	20-6
20.1.4. System Management Mode Programming Model	20-7
20.1.4.1. SMM ENTRY	20-7
20.1.4.2. PROCESSOR ENVIRONMENT	20-9
20.1.4.3. EXECUTING SYSTEM MANAGEMENT MODE HANDLER	20-10
20.1.4.4. EXCEPTIONS AND INTERRUPTS WITHIN SMM	20-10
20.1.5. SMM Features	20-11
20.1.5.1. SMM REVISION IDENTIFIER	20-11
20.1.5.2. HALT AUTO RESTART	20-12
20.1.5.3. SMM BASE RELOCATION	20-12
20.1.6. Pentium Processor SMM - Software Considerations	20-14
20.1.6.1. SMM CODE CONSIDERATIONS	20-14
20.1.6.2. EXCEPTION HANDLING	20-15
20.1.6.3. HALT DURING SMM	20-15
20.1.6.4. RELOCATING SMRAM TO AN ADDRESS ABOVE ONE MEGABYTE	20-15

	Page
20.2. PENTIUM™ PROCESSOR (735\90,815\100) POWER MANAGEMENT	20-16
20.2.1. System Management Mode Architecture	20-16
20.2.2. Pentium Processor (735\90, 815\100) Power Management Differences vs. Pentium Processor (510\60, 567\66)	20-16
20.2.3. System Management Interrupt Via APIC	20-17
20.2.4. I/O Instruction Restart	20-17
20.2.4.1. ENABLING I/O INSTRUCTION RESTART	20-17
20.2.4.2. SMRAM STATE SAVE MAP	20-17
20.2.4.2.1. I/O Instruction Restart Slot	20-19
20.2.4.3. BACK-TO-BACK SMI# AND I/O INSTRUCTION RESTART	20-20
20.2.4.4. EXIT FROM SMM	20-20
20.2.5. System Management Mode Revision Identifier	20-20
20.2.6. SMM — Dual Processing Considerations	20-21
20.2.6.1. DP SMI DELIVERY	20-22
20.2.6.2. DP SMRAM	20-22
20.2.6.3. DP SMI ACT#	20-22

PART III — COMPATIBILITY

CHAPTER 21

MIXING 16-BIT AND 32-BIT CODE

21.1. USING 16-BIT AND 32-BIT ENVIRONMENTS	21-1
21.2. MIXING 16-BIT AND 32-BIT OPERATIONS	21-2
21.3. SHARING DATA AMONG MIXED-SIZE CODE SEGMENTS	21-3
21.4. TRANSFERRING CONTROL AMONG MIXED-SIZE CODE SEGMENTS	21-4
21.4.1. Size of Code-Segment Pointer	21-4
21.4.2. Stack Management for Control Transfer	21-4
21.4.2.1. CONTROLLING THE OPERAND SIZE FOR A CALL	21-7
21.4.2.2. CHANGING SIZE OF A CALL	21-7
21.4.3. Interrupt Control Transfers	21-7
21.4.4. Parameter Translation	21-8
21.4.5. The Interface Procedure	21-8

CHAPTER 22

VIRTUAL-8086 MODE

22.1. EXECUTING 8086 CPU CODE	22-1
22.1.1. Registers and Instructions	22-2
22.1.2. Address Translation	22-3
22.2. STRUCTURE OF A VIRTUAL-8086 TASK	22-4
22.2.1. Paging for Virtual-8086 Tasks	22-4
22.2.2. Protection within a Virtual-8086 Task	22-5
22.3. ENTERING AND LEAVING VIRTUAL-8086 MODE	22-5
22.3.1. Transitions Through Task Switches	22-7
22.3.2. Transitions Through Trap Gates and Interrupt Gates	22-8
22.4. SENSITIVE INSTRUCTIONS	22-9
22.5. VIRTUAL INTERRUPT SUPPORT	22-9
22.6. EMULATING 8086 OPERATING SYSTEM CALLS	22-10

	Page
22.7. VIRTUAL I/O.....	22-10
22.7.1. I/O-Mapped I/O.....	22-11
22.7.2. Memory-Mapped I/O.....	22-11
22.7.3. Special I/O Buffers.....	22-12
22.8. DIFFERENCES FROM 8086 CPU.....	22-12
22.9. DIFFERENCES FROM INTEL 286 CPU.....	22-15
22.9.1. Privilege Level.....	22-15
22.9.2. Bus Lock.....	22-16
22.10. DIFFERENCES FROM Intel386™ AND Intel486™ CPUs.....	22-16

CHAPTER 23 COMPATIBILITY

23.1. RESERVED BITS	23-1
23.2. INTEGER UNIT.....	23-2
23.2.1. New Functions and Modes	23-2
23.2.2. Serializing Instructions.....	23-2
23.2.3. Detecting the Presence of New Features.....	23-2
23.2.4. Undefined Opcodes.....	23-3
23.2.5. Clock Counts.....	23-3
23.2.6. Initialization and Reset	23-3
23.2.6.1. INTEGER UNIT INITIALIZATION AND RESET	23-3
23.2.6.2. FPU/NPX INITIALIZATION AND RESET.....	23-4
23.2.6.3. Intel486 SX MICROPROCESSOR AND Intel487™ SX MATH COPROCESSOR INITIALIZATION	23-7
23.2.7. New Instructions	23-8
23.2.7.1. NEW PENTIUM PROCESSOR INSTRUCTIONS	23-8
23.2.7.2. NEW Intel486 PROCESSOR INSTRUCTIONS	23-9
23.2.7.3. NEW Intel386 PROCESSOR INSTRUCTIONS	23-9
23.2.8. Obsolete Instructions	23-9
23.2.9. Flags.....	23-9
23.2.9.1. NEW PENTIUM PROCESSOR FLAGS.....	23-10
23.2.9.2. NEW Intel486 PROCESSOR FLAGS.....	23-10
23.2.10. Control Registers	23-11
23.2.10.1. PENTIUM PROCESSOR CONTROL REGISTERS	23-11
23.2.10.2. Intel486 PROCESSOR CONTROL REGISTERS	23-12
23.2.11. Debug Registers	23-14
23.2.11.1. DIFFERENCES IN DR6.....	23-14
23.2.11.2. DIFFERENCES IN DR7.....	23-14
23.2.11.3. DEBUG REGISTERS 4 AND 5	23-14
23.2.12. Test Registers.....	23-15
23.2.13. Model Specific Registers	23-15
23.2.14. Exceptions	23-15
23.2.14.1. NEW PENTIUM PROCESSOR EXCEPTIONS.....	23-15
23.2.14.2. NEW Intel486 PROCESSOR EXCEPTIONS.....	23-16
23.2.14.3. NEW Intel386 PROCESSOR EXCEPTIONS.....	23-16
23.2.14.4. INTERRUPT PROPAGATION DELAY	23-16
23.2.14.5. PRIORITY OF EXCEPTIONS	23-16
23.2.14.6. DIVIDE-ERROR EXCEPTIONS	23-17
23.2.14.7. WRITES USING THE CS REGISTER PREFIX.....	23-17
23.2.14.8. NMI INTERRUPTS	23-17
23.2.14.9. INTERRUPT VECTOR TABLE LIMIT.....	23-18

	Page
23.2.15. Descriptor Types and Contents	23-18
23.2.16. Changes in Segment Descriptor Loads	23-18
23.2.17. Task Switching and Task State Segments	23-18
23.2.17.1. PENTIUM PROCESSOR TASK STATE SEGMENTS	23-18
23.2.17.2. TSS SELECTOR WRITES	23-18
23.2.17.3. ORDER OF READS/Writes TO THE TSS	23-19
23.2.17.4. USING A 16-BIT TSS WITH 32-BIT CONSTRUCTS	23-19
23.2.17.4.1. Differences in I/O Map Base Addresses	23-19
23.2.17.4.2. Caching, Pipelining, Prefetching	23-20
23.2.17.5. SELF MODIFYING CODE WITH CACHE ENABLED	23-21
23.2.18. Paging	23-21
23.2.18.1. PENTIUM PROCESSOR PAGING	23-21
23.2.18.2. Intel486 PROCESSOR PAGING	23-22
23.2.18.3. ENABLING AND DISABLING PAGING	23-22
23.2.19. Stack Operations	23-22
23.2.19.1. PUSH SP	23-22
23.2.19.2. FLAGS PUSHED ON THE STACK	23-23
23.2.19.3. SELECTOR PUSHES/POPS	23-23
23.2.19.4. ERROR CODE PUSHES	23-23
23.2.19.5. FAULT HANDLING EFFECTS ON THE STACK	23-24
23.2.19.6. INTERLEVEL RET/IRET FROM A 16-BIT INTERRUPT OR CALL GATE	23-24
23.2.20. Mixing 16- and 32-Bit Segments	23-24
23.2.21. Segment and Address Wraparound	23-25
23.2.21.1. SEGMENT WRAPAROUND	23-25
23.2.22. Write Buffers and Memory Ordering	23-26
23.2.23. Bus Locking	23-26
23.2.24. Bus Hold	23-27
23.2.25. Two Ways to Run Intel 286 CPU Tasks	23-27
23.3. FLOATING-POINT UNIT	23-28
23.3.1. Control Register Bits	23-28
23.3.1.1. EXTENSION TYPE (ET) BIT	23-28
23.3.1.2. NUMERIC EXCEPTION (NE) BIT	23-28
23.3.1.3. MONITOR COPROCESSOR (MP) BIT	23-28
23.3.1.4. FPU STATUS WORD	23-29
23.3.1.5. CONTROL WORD	23-29
23.3.1.6. TAG WORD	23-30
23.3.2. Data Types	23-30
23.3.2.1. NaNs	23-30
23.3.2.2. PSEUDOZERO, PSEUDO-NaN, PSEUDEOINFINITY, AND UNNORMAL FORMATS	23-31
23.3.3. Exceptions	23-31
23.3.3.1. DENORMAL EXCEPTIONS	23-31
23.3.3.2. OVERFLOW EXCEPTIONS	23-32
23.3.3.3. UNDERFLOW EXCEPTIONS	23-32
23.3.3.4. EXCEPTION PRECEDENCE	23-33
23.3.3.5. CS AND IP FOR FPU EXCEPTIONS	23-33
23.3.3.6. FPU ERROR SIGNALS	23-33
23.3.3.7. INVALID OPERATION ON DENORMALS	23-33
23.3.3.8. ALIGNMENT EXCEPTIONS	23-34
23.3.3.9. SEGMENT FAULT DURING FLDENV	23-34
23.3.3.10. INTERRUPT 7 — DEVICE NOT AVAILABLE	23-34