

---

---

# Contents

1	Introduction to Computers and Programming	1		
1.1	What is a Computer?	1		
1.2	Digital Computer Hardware	2		
1.3	Basic Computer Organization	6		
1.4	Computer System Software	8		
1.5	Algorithms	11		
1.6	Computer Data Types	12		
1.7	Classification of Computers	14		
	References	15		
2	The Programming Language Pascal	17		
2.1	Overview	18		
2.2	Vocabulary	21		
2.3	Program Structure	23		
2.4	Declarations	25		
2.5	Standard Data Types and Expressions	28		
	2.5.1 Integer	2.5.2 Real	2.5.3 Char	2.5.4 Boolean
2.6	User-Defined Types	30		
	2.6.1 Enumerated Types	2.6.2 Subrange Types		
2.7	Simple Statements	33		
	2.7.1 Assignment Statements	2.7.2 The GOTO Statement		
	2.7.3 Procedure Statements			
2.8	Structured Statements	36		
	2.8.1 Compound Statements	2.8.2 The IF Statement		
	2.8.3 The FOR Statement	2.8.4 The WHILE Statement		
	2.8.5 The REPEAT Statement	2.8.6 The CASE Statement		
	References	45		
	Exercises	47		
3	Data Structures in Pascal Programs	49		
3.1	One-Dimensional Arrays	50		
3.2	Multidimensional Arrays	54		
3.3	Records	55		

3.4	Stacks	59
3.5	Queues	64
3.6	Dynamic Variables and Pointers	68
3.7	One-Way Linked Lists	72
3.8	Two-Way Linked Lists	76
3.9	List Storage Management	77
	References	80
	Exercises	81
<b>4</b>	<b>Number Systems and Arithmetic</b>	<b>85</b>
4.1	Positional Number Systems	85
4.2	Octal and Hexadecimal Numbers	87
4.3	General Positional Number System Conversions	88
4.4	Addition and Subtraction of Nondecimal Numbers	90
4.5	Representation of Negative Numbers	92
	4.5.1 Signed-Magnitude Representation	
	4.5.2 Complement Number Systems	
	4.5.3 Radix-Complement Representation	
	4.5.4 Two's-Complement Representation	
	4.5.5 Diminished Radix-Complement Representation	
	4.5.6 Ones'-Complement Representation	4.5.7 Excess Representations
4.6	Two's-Complement Addition and Subtraction	97
	4.6.1 Addition Rules	4.6.2 A Graphical View
	4.6.4 Subtraction Rules	4.6.3 Overflow
	4.6.5 Two's-Complement and Unsigned Binary Numbers	
4.7	Ones'-Complement Addition and Subtraction	102
4.8	Binary Multiplication	103
	4.8.1 Unsigned Multiplication	4.8.2 Signed Multiplication
4.9	Binary Division	107
	4.9.1 Unsigned Division	4.9.2 Signed Division
4.10	Binary-Coded Decimal Representation	113
4.11	Fixed-Point and Floating-Point Representations	114
	4.11.1 Fixed-Point Representation	
	4.11.2 Basic Floating-Point Representation	
	4.11.3 IEEE Standard Floating-Point Representation	
	4.11.4 Floating-Point Operations	
4.12	Character Codes	120
	References	121
	Exercises	121

---

**PART 2 / GENERAL PRINCIPLES**
**125**

<b>5</b>	<b>Basic Computer Organization</b>	<b>125</b>
5.1	Memory	126
	5.1.1 Memory Organizations	
	5.1.2 Pascal Simulation of Processors and Memories	
	5.1.3 Memory Types	

5.2	Processor Architectures	129
	5.2.1 <i>Accumulator Machines</i>	
	5.2.2 <i>General-Register Machines</i>	
	5.2.3 <i>Stack Machines</i>	
	5.2.4 <i>Other Processor Organizations and Classifications</i>	
5.3	68000-Family Processor Organization	133
5.4	Instruction Processing Cycle	137
5.5	Instruction Set	140
5.6	Operands and Operations	141
5.7	Addressing Modes	144
5.8	A Machine Language Program	148
5.9	Assembly Language	148
5.10	Operation of a Simple Program	151
5.11	Pascal Simulation of Instruction Execution	155
5.12	Indirect Addressing	160
5.13	Subroutines	163
	References	169
	Exercises	170
6	Assembly Language Programming	174
6.1	Assembly Language	175
	6.1.1 <i>Assembly Language Formats</i>	
	6.1.2 <i>Pseudo-operations and the PLC</i>	
	6.1.3 <i>Expressions</i>	
	6.1.4 <i>Assembly-Time, Load-Time, and Run-Time Operations</i>	
6.2	The Two-Pass Assembler	187
6.3	Object Modules	192
6.4	Relocating Assemblers and Loaders	194
	6.4.1 <i>Relocation</i>	
	6.4.2 <i>Linking</i>	
6.5	Macros	200
6.6	Structured Assembly Language	204
	References	208
	Exercises	209
7	Addressing	210
7.1	General Concepts	210
	7.1.1 <i>Address Size and Register Size</i>	
	7.1.2 <i>Address Size and Memory Size</i>	
	7.1.3 <i>Address Aliasing</i>	
	7.1.4 <i>Alignment</i>	
	7.1.5 <i>Addressing Modes</i>	
7.2	Single-Component Addressing Modes	215
	7.2.1 <i>Register Direct</i>	
	7.2.2 <i>Absolute</i>	
	7.2.3 <i>Immediate</i>	
	7.2.4 <i>Register Indirect</i>	
	7.2.5 <i>Auto-Increment and Auto-Decrement</i>	
7.3	Multicomponent Addressing Modes	225
	7.3.1 <i>Indexed</i>	
	7.3.2 <i>Based</i>	
	7.3.3 <i>Based Indexed</i>	
	7.3.4 <i>Relative</i>	
	7.3.5 <i>Relative Indexed</i>	
	7.3.6 <i>Paged</i>	
7.4	Position-Independent Code	247
7.5	Memory Mapping and Management	249
	7.5.1 <i>Memory Mapping</i>	
	7.5.2 <i>Memory Management</i>	
	7.5.3 <i>Virtual Memory</i>	
	References	256
	Exercises	257

8	Operations	260
8.1	Instruction Formats	262
8.2	Condition Bits	264
8.3	Processor Control and Status Bits	267
8.4	Data Moves	268
	8.4.1 68000 Quirks	
8.5	Addition and Subtraction	273
8.6	Some Single-Operand Instructions	277
8.7	Program Control	279
	8.7.1 Example Programs	
8.8	Logical Instructions	287
8.9	Rotates and Shifts	291
	8.9.1 Rotate Instructions	8.9.2 Logical Shift Instructions
	8.9.3 Arithmetic Shift Instructions	
8.10	Multiplication and Division	298
8.11	Decimal Arithmetic	300
8.12	Miscellaneous 68000 Control Instructions	303
	References	307
	Exercises	308
9	Subroutines and Parameters	312
9.1	Procedures and Functions in Pascal	313
	9.1.1 Procedures	9.1.2 Functions
9.2	Parameters in High-Level Language Programs	318
	9.2.1 Pascal Parameters	9.2.2 Value Parameters
	9.2.3 Variable Parameters	9.2.4 Procedure and Function Parameters
	9.2.5 Parameters in Other High-Level Languages	
9.3	Assembly Language Subroutines and Parameters	323
	9.3.1 Subroutine Calling Methods	9.3.2 Subroutine Parameters
	9.3.3 Passing Parameters in Registers and Memory Locations	
	9.3.4 Parameter Areas	9.3.5 Static and Dynamic Allocation
	9.3.6 Stack-Oriented Parameter-Passing Conventions	
	9.3.7 Another Example: Queue Manipulation Subroutines	
9.4	Recursion	337
	9.4.1 Recursive Procedures and Functions	9.4.2 Recursive Subroutines
9.5	Coroutines	343
	9.5.1 General Structure	9.5.2 Pascal Coroutines
	9.5.3 Assembly Language Coroutines	9.5.4 Coroutine Applications
	References	351
	Exercises	352
10	Input/Output	356
10.1	I/O Organization	356
	10.1.1 Buses	10.1.2 Devices and Interfaces
	10.1.3 Ports	
10.2	I/O Programming	360
	10.2.1 Isolated I/O	10.2.2 Memory-Mapped I/O
	10.2.3 The 68000 MOVEP Instruction	
10.3	I/O Protocols	366
	10.3.1 Input Operations	10.3.2 Overlapped I/O
	10.3.3 Output Operations	

10.4	I/O Drivers	373
	10.4.1 Terminal I/O	10.4.2 Shared I/O Drivers
	References	379
	Exercises	379
11	Interrupts and Direct Memory Access	382
11.1	Basic Interrupt System Structure and Programming	382
	11.1.1 General Considerations	11.1.2 A Simple Interrupt System
	11.1.3 An I/O Program Using Interrupts	
11.2	Variations in Interrupt Systems and Programs	392
	11.2.1 Interrupt Levels and Enabling	11.2.2 Interrupt Priority
	11.2.3 Interrupt Identification and Polling	11.2.4 Vectored Interrupts
	11.2.5 Nonmaskable Interrupts	
11.3	Traps and Software Interrupts	398
	11.3.1 Traps	11.3.2 Software Interrupts and Trap Instructions
11.4	Summary of 68000 Interrupt System and Traps	399
	11.4.1 Exception Vectors	11.4.2 Interrupt Handling
	11.4.3 Pascal Simulation of 68000 Interrupt Handling	11.4.4 Traps
	11.4.5 Bus and Address Errors	11.4.6 Reset
11.5	Direct Memory Access	406
	11.5.1 Motivation	11.5.2 DMA Channel Programming
	11.5.3 Memory-Mapped Displays	
	References	411
	Exercises	411
12	Processes and Concurrency	413
12.1	Interrupt Processes	414
	12.1.1 Processes	12.1.2 A Simplifying Restriction
12.2	Concurrency	415
	12.2.1 Multiple Interrupt Processes	12.2.2 Implementation Details
	12.2.3 Waking and Sleeping	
12.3	Timing-Dependent Process Behavior	420
	12.3.1 Deadlocks and Critical Sections	12.3.2 Detecting Critical Sections
	12.3.3 Locking and Semaphores	12.3.4 Multiple Processors
12.4	Shared Data and Instructions	425
	12.4.1 Shared Variables and Data Structures	
	12.4.2 Shared Instructions (Reentrant Programs)	
12.5	Assembly Language Program Example	430
	References	436
	Exercises	436
13	Program Development	439
13.1	The Stages of Program Development	440
13.2	Specification and Design	442
	13.2.1 Program Structure	13.2.2 Detailed Module Design
	13.2.3 Design Iterations	13.2.4 Data Structures
13.3	Documentation	447
	13.3.1 Prologues	13.3.2 Program Module Prologues
	13.3.3 Data Module Prologues	
	13.3.4 Ownership of Global Data Structures	13.3.5 Documenting Code

13.4	Coding	452	
	13.4.1	Coding Rules	13.4.2 Coding Tricks
13.5	Testing and Debugging	457	
	13.5.1	Development Approach	13.5.2 Testing 13.5.3 Debugging
	References	463	
	Exercises	464	

**IT 3 / ADDITIONAL TOPICS****465**

14	Advanced 68000-Family Processors	465	
14.1	Virtual Memory and Virtual Machine Support	466	
14.2	68010 Programming Model and Control Operations	467	
14.3	68010 Extended Operations	470	
14.4	68010 Exception Handling	471	
14.5	String Operations and 68010 Loop Mode	473	
14.6	MC68020 and MC68030 Extensions	475	
14.7	68020 Programming Model	476	
14.8	68020 Extended Addressing Modes	478	
	14.8.1	Extended Based	14.8.2 Memory Indirect 14.8.3 Index Scaling
	14.8.4	Extended Based Indexed	14.8.5 Memory Indirect Preindexed
	14.8.6	Memory Indirect Postindexed	14.8.7 PC-Relative Extended Modes
	14.8.8	Summary	
14.9	68020 Extended Operations	492	
	14.9.1	Data Movement	
	14.9.2	Extended Displacements, Operand Sizes, and Types	
	14.9.3	Bit Field Instructions	
	14.9.4	Other Data Manipulation and Program Control Instructions	
	14.9.5	Traps	14.9.6 Coprocessor Instructions
	14.9.7	MC68020 Module Call and Return	
14.10	68020 Exception Handling	517	
	References	518	
	Exercises	519	
15	68000-Family System Elements	521	
15.1	MC68020 On-Chip Instruction Cache	522	
	15.1.1	Cache Basics	15.1.2 Direct-Mapped and Set-Associative Caches
	15.1.3	Write Operations	15.1.4 MC68020 Instruction Cache
15.2	MC68030 On-Chip Instruction and Data Caches	529	
15.3	MC68881 Floating-Point Coprocessor	532	
	15.3.1	Programming Model	15.3.2 Data Types 15.3.3 Instructions
	15.3.4	Programming	
15.4	MC68851 Paged Memory Management Unit	544	
	15.4.1	Basic Organization and Function	15.4.2 Programming Model
	15.4.3	Address Translation	15.4.4 Root Pointers and Descriptors
	15.4.5	Protection	15.4.6 Instruction Set
15.5	MC68030 On-Chip Memory Management Unit	565	

15.6	MC6850 Asynchronous Communications Interface Adapter	567
	15.6.1 Control Port	
	15.6.2 Transmitter Data Port	
	15.6.3 Receiver Data Port	
	15.6.4 Status Port	
	15.6.5 Modem Control Lines	
15.7	MC68230 Parallel Interface and Timer	573
	References	577
	Exercises	578
16	An Accumulator Architecture: The Motorola 6809	580
16.1	Basic Organization	581
	16.1.1 Computer Structure	
	16.1.2 Processor Programming Model	
	16.1.3 Instruction Formats	
16.2	Assembly Language	583
16.3	Addressing	584
	16.3.1 Immediate	
	16.3.2 Absolute	
	16.3.3 Direct Page	
	16.3.4 Indexed Addressing Modes	
	16.3.5 Register Indirect	
	16.3.6 Based or Indexed	
	16.3.7 Based Indexed	
	16.3.8 Auto-Increment and Auto-Decrement	
	16.3.9 Relative	
	16.3.10 Indirect	
16.4	Operations	594
	16.4.1 Memory Reference	
	16.4.2 Register Reference	
	16.4.3 Program Control	
16.5	Sample Programs	599
16.6	Input/Output, Interrupts, and Traps	607
	16.6.1 Input/Output	
	16.6.2 Interrupts	
	16.6.3 Traps and Reset	
	References	612
	Exercises	613
17	A Classic Mini/Micro Architecture: The DEC PDP-11 and LSI-11	614
17.1	BASIC ORGANIZATION	615
	17.1.1 Computer Structure	
	17.1.2 Processor Programming Model	
	17.1.3 Instruction Formats	
17.2	ASSEMBLY LANGUAGE	616
17.3	ADDRESSING	617
	17.3.1 Basic Addressing Modes	
	17.3.2 PC Addressing Modes	
	17.3.3 Addressing Mode Summary	
	17.3.4 Pascal Simulation of Addressing Modes	
	17.3.5 Memory Mapping and Management	
17.4	OPERATION TYPES	623
	17.4.1 Double Operand	
	17.4.2 Single Operand	
	17.4.3 Program Control	
	17.4.4 Miscellaneous	
	17.4.5 Extended	
17.5	SAMPLE PROGRAMS	628
17.6	INPUT/OUTPUT, INTERRUPTS, AND TRAPS	636
	17.6.1 Input/Output	
	17.6.2 Interrupts	
	17.6.3 Traps	
	References	645
	Exercises	646

## 18 An Evolved Architecture: The Intel 80x86 Family 647

- 18.1 Basic Organization 650
  - 18.1.1 Computer Structure*    *18.1.2 Processor Programming Model*
  - 18.1.3 Instruction Formats*
- 18.2 Assembly Language 654
- 18.3 Addressing 659
  - 18.3.1 Postbyte Addressing Modes*    *18.3.2 Register*    *18.3.3 Immediate*
  - 18.3.4 Absolute*    *18.3.5 Register Indirect*    *18.3.6 Based*
  - 18.3.7 Indexed*    *18.3.8 Based Indexed*    *18.3.9 Relative*
  - 18.3.10 Memory Mapping*
- 18.4 Operations 668
  - 18.4.1 Data Movement*    *18.4.2 Arithmetic*    *18.4.3 Logicals and Shifts*
  - 18.4.4 Program Control*    *18.4.5 String Manipulation*
  - 18.4.6 80186/188 Extended Instructions*
- 18.5 Sample Programs 679
- 18.6 Input/Output, Interrupts, and Traps 696
  - 18.6.1 Input/Output*    *18.6.2 Interrupts*    *18.6.3 Traps*
- 18.7 80286 Processor Organization 702
  - 18.7.1 Programming Model*    *18.7.2 Privilege Levels and Protection*
  - 18.7.3 Memory Mapping and Management*    *18.7.4 Task Management*
  - 18.7.5 Gates*    *18.7.6 Interrupts and Traps*
  - 18.7.7 Extended Instructions*
- 18.8 80386 Processor Organization 712
  - 18.8.1 Programming Model*    *18.8.2 Operating Modes*
  - 18.8.3 Memory Mapping, Protection, and Tasks*
  - 18.8.4 Descriptor Formats and Address and Data Sizes*    *18.8.5 Paging*
  - 18.8.6 Input/Output, Interrupts, and Traps*    *18.8.7 Virtual 8086 Mode*
  - 18.8.8 Extended Addressing Modes*    *18.8.9 Extended Instructions*
- References 721
- Exercises 721

## 19 A Hypothetical Stack Machine: The H11 723

- 19.1 Processor and Memory Organization 723
- 19.2 Instruction Set 724
- 19.3 Sample Programs 728
- 19.4 Strengths and Weaknesses of Stack Machines 729
- 19.5 The PDP-11 as a Stack Machine 732
- Exercises 733

## A The ASCII Character Code 734

- A.1 ASCII Encoding 734
- A.2 ASCII and Terminals 734

## B Serial Communication 737

- B.1 Serial Data Links 737
- B.2 Asynchronous Serial Communication Format 741

## Index 744