

CONTENTS IN DETAIL

ACKNOWLEDGMENTS xix

INTRODUCTION xxi

Who This Book Is For	xxi
About This Book	xxii
The Programming in the Book	xxii
Why Read This Book?	xxiii
Chapter Organization	xxiii
Efficient Use of This Book	xxv

1 1 SETTING THE STAGE

Computer Subsystems	1
Creating and Executing a Program	2
The Programming Environment	4
What You've Learned	8

2 9 DATA STORAGE FORMATS

Switches and Groups of Switches	9
Representing Switches with Bits	10
Representing Groups of Bits	10
Using Hexadecimal Digits	11
The Mathematical Equivalence of Binary and Decimal	12
Getting to Know Positional Notation	12
Converting Binary to Unsigned Decimal	14
Converting Unsigned Decimal to Binary	15
Storing Data in Memory	17
Memory Addresses	17
Characters	19
Unsigned Integers	22
Exploring Data Formats with C	23
Using C and C++ I/O Libraries	23
Writing and Executing Your First C Program	25

Examining Memory with a Debugger	29
Using Your Debugger	30
Understanding Byte Storage Order in Memory	33
What You've Learned	35

3

COMPUTER ARITHMETIC

37

Unsigned Integers in the Decimal Number System	38
Addition	38
Subtraction	39
Unsigned Integers in the Binary System	40
Addition	41
Subtraction	42
Adding and Subtracting Signed Integers	43
Understanding Two's Complement	43
Computing Two's Complement	46
Adding and Subtracting Signed Integers in Binary	47
Two Numbers of the Opposite Sign	48
Two Positive Numbers	49
Two Negative Numbers	49
The Circular Nature of Integer Codes	51
What You've Learned	52

4

BOOLEAN ALGEBRA

53

Basic Boolean Operators	54
Boolean Expressions	56
Boolean Algebra Rules	57
Rules That Are the Same as Elementary Algebra	57
Rules That Differ from Elementary Algebra	59
Boolean Functions	61
Canonical Sum or Sum of Minterms	62
Canonical Product or Product of Maxterms	64
Comparison of Canonical Boolean Forms	65
Boolean Expression Minimization	66
Minimal Expressions	67
Minimization Using Algebraic Manipulations	68
Minimization Using Karnaugh Maps	71
Combining Basic Boolean Operators	81
What You've Learned	82

5	LOGIC GATES	85
Crash Course in Electronics		85
Power Supplies and Batteries		86
Passive Components		87
Power Consumption		95
Transistors		96
MOSFET Switch		97
CMOS Switch		100
NAND and NOR Gates		102
NAND as a Universal Gate		104
What You've Learned		106

6	COMBINATIONAL LOGIC CIRCUITS	107
The Two Types of Logic Circuits		107
Adders		108
Half Adder		108
Full Adder		109
Full Adder from Two Half Adders		111
Ripple-Carry Addition and Subtraction Circuits		112
Decoders		114
Multiplexers		117
Tristate Buffer		118
Programmable Logic Devices		119
Programmable Logic Array		120
Read-Only Memory		122
Programmable Array Logic		124
What You've Learned		125

7	SEQUENTIAL LOGIC CIRCUITS	127
Latches		128
SR Latch Using NOR Gates		128
SR Latch Using NAND Gates		131
SR Latch with Enable		133
The D Latch		134
Flip-Flops		136
Clocks		136
D Flip-Flop		137
Designing Sequential Logic Circuits		140
A Counter		140
A Branch Predictor		144
What You've Learned		149

8		
MEMORY		151
The Memory Hierarchy	151	
Mass Storage	152	
Main Memory	153	
Cache Memory	154	
Registers	156	
Implementing Memory in Hardware	156	
Four-Bit Registers	156	
Shift Registers	159	
The Register File	160	
Read/Write Memory	160	
Static Random-Access Memory	161	
Dynamic Random-Access Memory	162	
What You've Learned	164	

9		
CENTRAL PROCESSING UNIT		167
CPU Overview	168	
CPU Subsystems	168	
Instruction Execution Cycle	170	
A64 Registers	172	
General-Purpose Registers	173	
Special-Purpose Registers	174	
C/C++ Integral Data Types and Register Sizes	175	
Using gdb to View the CPU Registers	176	
What You've Learned	183	

10		
PROGRAMMING IN ASSEMBLY LANGUAGE		185
Starting in C	186	
From C to Assembly Language	188	
Unused Assembler Directives	190	
Required Assembler Directives	191	
Starting in Assembly Language	193	
Symbolic Names in Assembly Language	194	
Basic Format of Assembly Language Instructions	195	
Symbols Used for Instruction Descriptions	196	
First Instructions	196	
Using gdb to Learn Assembly Language	198	
What You've Learned	206	

11	INSIDE THE MAIN FUNCTION	207
Using the Call Stack		207
Stacks in General		208
The Stack Frame		210
A64 Memory Addressing		213
Local Variables on the Call Stack		215
Processing Data in a Function		216
Passing Arguments in Registers		220
Writing main in Assembly Language		222
What You've Learned		224

12	INSTRUCTION DETAILS	227
Looking at Machine Code		228
Encoding Load and Store Instructions		231
Encoding Data Processing Instructions		232
Moving Data from Register to Register		232
Moving a Constant to a Register		233
Performing Arithmetic		234
Computing Addresses		234
Looking at the Details of an Executable File		235
Encoding Program Flow Control Instructions		236
Translating Assembly into Machine Code		237
The Assembler		237
The Linker		239
What You've Learned		240

13	CONTROL FLOW CONSTRUCTS	243
Branches		244
Unconditional		244
Conditional		244
Iteration		246
write and read System Call Functions		247
while Loop		247
for Loop		252
do-while Loop		253
Conditionals		255
if		256
if-then-else		258
switch		262
What You've Learned		268

14		
INSIDE SUBFUNCTIONS		269
Scope of Variable Names in C	270	
Overview of Passing Arguments	271	
Global Variables	271	
Explicitly Passing Arguments	276	
In C	277	
In Assembly Language	278	
With More Than Eight Arguments	281	
Stack Frame Discipline	288	
Static Local Variables	293	
In C	294	
In Assembly Language	296	
Program Memory Characteristics	302	
What You've Learned	304	

15		
SPECIAL USES OF SUBFUNCTIONS		305
Recursion	305	
Accessing CPU Features in Assembly Language	314	
Writing a Separate Function	314	
Using Inline Assembly Language	318	
What You've Learned	322	

16		
BITWISE LOGIC, MULTIPLICATION, AND DIVISION INSTRUCTIONS		323
Bitmasking	323	
Bitmasking in C	324	
Basic Logic Instructions	327	
Bitmasking in Assembly Language	330	
Shifting Bits	333	
In C	333	
In Assembly Language	337	
Multiplication	340	
In C	340	
In Assembly Language	344	
Division	349	
In C	349	
In Assembly Language	354	
What You've Learned	358	

17	DATA STRUCTURES	361
Arrays		361
In C		362
In Assembly Language		367
Records		371
In C		372
In Assembly Language		380
What You've Learned		385

18	OBJECT-ORIENTED PROGRAMMING	387
Objects in C++		388
Creating an Object		390
Defining Class Member Functions		392
Writing a Constructor and Destructor via the Compiler		399
Overloading the Default Constructor in C++		403
Calling Assembly Language Functions in C++		406
What You've Learned		408

19	FRACTIONAL NUMBERS	409
Fractional Values in Binary		410
Fixed-Point Numbers		411
When the Fractional Part Is a Sum of Inverse Powers of Two		411
When the Fractional Part Is in Decimal		415
Floating-Point Numbers		419
Floating-Point Representation		420
IEEE 754 Floating-Point Standard		420
Floating-Point Hardware		422
Floating-Point Arithmetic Errors		425
Rounding Error		426
Absorption		427
Cancellation		428
Associativity		429
Comments About Numerical Accuracy		434
What You've Learned		435

20	INPUT/OUTPUT	437
Timing Considerations		437
Memory Timing		438
I/O Device Timing		438

Accessing I/O Devices	439
I/O Programming	442
Programming a General-Purpose I/O Device	443
Connecting the Blinking LED Circuit	445
Blinking an LED in C, All Models	445
Blinking an LED in Assembly Language, Models 3 and 4	448
Blinking an LED in Assembly Language, Model 5	456
Polled I/O Programming Algorithms	460
What You've Learned	465

21

EXCEPTIONS AND INTERRUPTS 467

Application vs. System Software	468
Privilege and Exception Levels	468
Exception Events	470
CPU Response to an Exception	471
Supervisor Calls	474
What You've Learned	477

INDEX 479