

CONTENTS IN DETAIL

PREFACE	xv
Who This Book Is For	xv
About This Book.	xvi
The Programming in the Book	xvi
Why Read This Book?	xvii
Chapter Organization	xvii
Efficient Use of This Book	xix
ACKNOWLEDGMENTS	xxi
CHAPTER 1: SETTING THE STAGE	1
Computer Subsystems	1
Program Execution	2
The Programming Environment.	4
What You've Learned	6
CHAPTER 2: DATA STORAGE FORMATS	9
Describing Switches and Groups of Switches.	10
Representing Switches with Bits.	10
Representing Groups of Bits	10
Using Hexadecimal Digits	12
The Mathematical Equivalence of Binary and Decimal	13
Getting to Know Positional Notation	13
Converting Binary to Unsigned Decimal.	15
Converting Unsigned Decimal to Binary.	16
Storing Data in Memory	18
Expressing Memory Addresses	19
Characters	20
Unsigned Integers	23
Exploring Data Formats with C	25
C and C++ I/O Libraries	25
Writing and Executing Your First C Program.	27
Examining Memory with a Debugger	30
Using Your Debugger	31
Understanding Byte Storage Order in Memory	35
What You've Learned	36
CHAPTER 3: COMPUTER ARITHMETIC	39
Adding and Subtracting Unsigned Integers	40
Adding in the Decimal Number System	40
Subtracting in the Decimal Number System	41
Adding and Subtracting Unsigned Integers in Binary	43

Adding and Subtracting Signed Integers	45
Two's Complement	45
Computing Two's Complement	48
Adding and Subtracting Signed Integers in Binary	50
Circular Nature of Integer Codes	53
What You've Learned	55

CHAPTER 4: BOOLEAN ALGEBRA 57

Basic Boolean Operators	58
Boolean Expressions	60
Boolean Algebra Rules	61
Boolean Algebra Rules That Are the Same as Elementary Algebra	61
Boolean Algebra Rules That Differ from Elementary Algebra	63
Boolean Functions	66
Canonical Sum or Sum of Minterms	67
Canonical Product or Product of Maxterms	69
Comparison of Canonical Boolean Forms	70
Boolean Expression Minimization	71
Minimal Expressions	71
Minimization Using Algebraic Manipulations	73
Minimization Using Karnaugh Maps	76
Combining Basic Boolean Operators	86
What You've Learned	88

CHAPTER 5: LOGIC GATES 89

Crash Course in Electronics	90
Power Supplies and Batteries	90
Passive Components	91
Transistors	100
MOSFET Switch	101
CMOS Switch	104
NAND and NOR Gates	107
NAND as a Universal Gate	108
What You've Learned	111

CHAPTER 6: COMBINATIONAL LOGIC CIRCUITS 113

The Two Classes of Logic Circuits	114
Adders	115
Half Adder	115
Full Adder	116
Full Adder from Two Half Adders	117
Ripple-Carry Addition and Subtraction Circuits	119
Decoders	121
Multiplexers	124
Tristate Buffer	125
Programmable Logic Devices	127
Programmable Logic Array	128
Read-Only Memory	130
Programmable Array Logic	131
What You've Learned	133

CHAPTER 7: SEQUENTIAL LOGIC CIRCUITS	135
Latches	136
SR Latch Using NOR Gates	136
SR Latch Using NAND Gates	139
SR Latch with Enable	141
The D Latch	142
Flip-Flops	144
Clocks	144
D Flip-Flop	145
T Flip-Flop	147
JK Flip-Flop	148
Designing Sequential Logic Circuits	151
Designing a Counter	152
Designing a Branch Predictor	156
What You've Learned	161
CHAPTER 8: MEMORY	163
The Memory Hierarchy	163
Mass Storage	164
Main Memory	165
Cache Memory	166
Registers	168
Implementing Memory in Hardware	168
Four-Bit Register	168
Shift Register	171
Register File	172
Read-Write Memory	172
Static Random-Access Memory	173
Dynamic Random-Access Memory	175
What You've Learned	176
CHAPTER 9: CENTRAL PROCESSING UNIT	177
CPU Overview	178
CPU Subsystems	178
Instruction Execution Cycle	180
x86-64 Registers	182
General-Purpose Registers	183
Status Register	186
C/C++ Integral Data Types and Register Sizes	187
Using gdb to View the CPU Registers	188
What You've Learned	193
CHAPTER 10: PROGRAMMING IN ASSEMBLY LANGUAGE	195
Compiling a Program Written in C	196
From C to Assembly Language	197
Assembler Directives That We Won't Use	199
Assembler Directives That We Will Use	202
Creating a Program in Assembly Language	203
Assembly Language in General	204
First Assembly Language Instructions	205

Minimal Processing in a Function	208
Using gdb to Learn Assembly Language.	210
AT&T Syntax	217
What You've Learned	218

CHAPTER 11: INSIDE THE MAIN FUNCTION 221

The write and read System Call Functions	222
Passing Arguments in Registers	223
Position-Independent Code	225
The Call Stack	227
Stacks in General	227
Inside the Function Prologue and Epilogue	229
Local Variables in a Function	233
Variables on the Stack	234
Stack Corruption	237
Not Using the C Runtime Environment.	241
What You've Learned	243

CHAPTER 12: INSTRUCTION DETAILS 245

Looking at Machine Code	246
Instruction Bytes	247
Opcode Bytes.	247
ModR/M Byte	248
REX Prefix Byte	250
Immediate Addressing Mode	250
Memory Addressing Modes	252
Direct Memory Addressing.	252
Register Indirect with Offset	253
Register Indirect with Indexing	255
SIB Byte	255
Jump Instructions	256
Assemblers and Linkers.	258
The Assembler	258
The Linker.	260
What You've Learned	261

CHAPTER 13: CONTROL FLOW CONSTRUCTS 263

Jumps	264
Unconditional Jumps	264
Conditional Jumps.	265
Iteration	267
while Loop	267
for Loop	272
do-while Loop.	274
Selection.	276
if Conditional	276
if-then-else Conditional.	278
switch Conditional.	282
What You've Learned	286

CHAPTER 14: INSIDE SUBFUNCTIONS	287
Scope of Variable Names in C	288
Overview of Passing Arguments	289
Global Variables	290
Explicitly Passing Arguments	294
Passing Arguments in C	294
What's Going On in Assembly Language	296
Handling More Than Six Arguments	299
Pushing Arguments onto the Stack	299
Storing Arguments Directly on the Stack	303
Summary of Stack Frame Usage	306
Static Local Variables	309
What You've Learned	318
CHAPTER 15: SPECIAL USES OF SUBFUNCTIONS	319
Recursion	319
Accessing CPU Features in Assembly Language	326
A Separate Function Written in Assembly Language	326
Inline Assembly Language	332
What You've Learned	334
CHAPTER 16: COMPUTING WITH BITWISE LOGIC, MULTIPLICATION, AND DIVISION INSTRUCTIONS	335
Bit Masking	335
Bit Masking in C	336
Logic Instructions	339
Bit Masking in Assembly Language	340
Shifting Bits	343
Shifting Bits in C	343
Shift Instructions	347
Shifting Bits in Assembly Language	349
Multiplication	352
Multiplication in C	352
Multiply Instructions	355
Multiplication in Assembly Language	357
Division	360
Division in C	360
Division Instructions	364
Division in Assembly Language	366
What You've Learned	370
CHAPTER 17: DATA STRUCTURES	371
Arrays	372
Arrays in C	372
Arrays in Assembly Language	376
Records	380
Records in C	380
Records in Assembly Language	382

Passing Records to Other Functions in C	383
Passing Records to Other Functions in Assembly Language.	389
What You've Learned	394

CHAPTER 18: OBJECT-ORIENTED PROGRAMMING 395

Objects in C++	396
Using Objects in C++	398
Defining Class Member Functions	401
Letting the Compiler Write a Constructor and Destructor.	405
Objects in Assembly Language	407
What You've Learned	413

CHAPTER 19: FRACTIONAL NUMBERS 415

Fractional Values in Binary	416
Fixed-Point Numbers.	417
When the Fractional Part Is a Sum of Inverse Powers of Two.	417
When the Fractional Part Is in Decimal	421
Floating-Point Numbers.	425
Floating-Point Representation	425
IEEE 754 Floating-Point Standard	426
SSE2 Floating-Point Hardware	427
xmm Registers.	428
Programming with Floating-Point Numbers	430
Floating-Point Arithmetic Errors	433
Comments About Numerical Accuracy	440
What You've Learned	441

CHAPTER 20: INPUT/OUTPUT 443

Timing Considerations	444
Memory Timing.	444
I/O Device Timing	444
Bus Timing	445
Accessing I/O Devices	446
Port-Mapped I/O	447
Memory-Mapped I/O	447
I/O Programming	449
Polled I/O	449
Interrupt-Driven I/O	449
Direct Memory Access	450
Polled I/O Programming Algorithms.	450
UART Memory-Mapped I/O in C	452
UART Memory-Mapped I/O in Assembly Language	457
UART Port-Mapped I/O	460
What You've Learned	464

CHAPTER 21: INTERRUPTS AND EXCEPTIONS	465
Privilege Levels	466
CPU Response to an Interrupt or Exception	467
External Interrupts	468
Exceptions	469
Software Interrupts	469
System Calls	470
The int 0x80 Software Interrupt	470
The syscall Instruction	472
What You've Learned	474
 INDEX	 477