

# Contents

## Preface

### **1 Introduction**

- 1.1 [Computer Subsystems](#)
- 1.2 [How the Subsystems Interact](#)

### **2 Data Storage Formats**

- 2.1 [Bits and Groups of Bits](#)
- 2.2 [Mathematical Equivalence of Binary and Decimal](#)
- 2.3 [Unsigned Decimal to Binary Conversion](#)
- 2.4 [Memory — A Place to Store Data \(and Other Things\)](#)
- 2.5 [Using C Programs to Explore Data Formats](#)
- 2.6 [Examining Memory With gdb](#)
- 2.7 [ASCII Character Code](#)
- 2.8 [write and read Functions](#)
- 2.9 [Exercises](#)

### **3 Computer Arithmetic**

- 3.1 [Addition and Subtraction](#)
- 3.2 [Arithmetic Errors — Unsigned Integers](#)
- 3.3 [Arithmetic Errors — Signed Integers](#)
- 3.4 [Overflow and Signed Decimal Integers](#)
  - 3.4.1 [The Meaning of CF and OF](#)
- 3.5 [C/C++ Basic Data Types](#)
  - 3.5.1 [C/C++ Shift Operations](#)
  - 3.5.2 [C/C++ Bit Operations](#)
  - 3.5.3 [C/C++ Data Type Conversions](#)
- 3.6 [Other Codes](#)
  - 3.6.1 [BCD Code](#)
  - 3.6.2 [Gray Code](#)
- 3.7 [Exercises](#)

### **4 Logic Gates**

- 4.1 [Boolean Algebra](#)
- 4.2 [Canonical \(Standard\) Forms](#)
- 4.3 [Boolean Function Minimization](#)
  - 4.3.1 [Minimization Using Algebraic Manipulations](#)
  - 4.3.2 [Minimization Using Graphic Tools](#)
- 4.4 [Crash Course in Electronics](#)
  - 4.4.1 [Power Supplies and Batteries](#)
  - 4.4.2 [Resistors, Capacitors, and Inductors](#)
  - 4.4.3 [CMOS Transistors](#)
- 4.5 [NAND and NOR Gates](#)
- 4.6 [Exercises](#)

### **5 Logic Circuits**

- 5.1 [Combinational Logic Circuits](#)
  - 5.1.1 [Adder Circuits](#)
  - 5.1.2 [Ripple-Carry Addition/Subtraction Circuits](#)

- 5.1.3 [Decoders](#)
- 5.1.4 [Multiplexers](#)
- 5.2 [Programmable Logic Devices](#)
  - 5.2.1 [Programmable Logic Array \(PLA\)](#)
  - 5.2.2 [Read Only Memory \(ROM\)](#)
  - 5.2.3 [Programmable Array Logic \(PAL\)](#)
- 5.3 [Sequential Logic Circuits](#)
  - 5.3.1 [Clock Pulses](#)
  - 5.3.2 [Latches](#)
  - 5.3.3 [Flip-Flops](#)
- 5.4 [Designing Sequential Circuits](#)
- 5.5 [Memory Organization](#)
  - 5.5.1 [Registers](#)
  - 5.5.2 [Shift Registers](#)
  - 5.5.3 [Static Random Access Memory \(SRAM\)](#)
  - 5.5.4 [Dynamic Random Access Memory \(DRAM\)](#)
- 5.6 [Exercises](#)

## **[6 Central Processing Unit](#)**

- 6.1 [CPU Overview](#)
- 6.2 [CPU Registers](#)
- 6.3 [CPU Interaction with Memory and I/O](#)
- 6.4 [Program Execution in the CPU](#)
- 6.5 [Using gdb to View the CPU Registers](#)
- 6.6 [Exercises](#)

## **[7 Programming in Assembly Language](#)**

- 7.1 [Creating a New Program](#)
- 7.2 [Program Organization](#)
  - 7.2.1 [First instructions](#)
  - 7.2.2 [A Note About Syntax](#)
  - 7.2.3 [The Additional Assembly Language Generated by the Compiler](#)
  - 7.2.4 [Viewing Both the Assembly Language and C Source Code](#)
  - 7.2.5 [Minimum Program in 32-bit Mode](#)
- 7.3 [Assemblers and Linkers](#)
  - 7.3.1 [Assemblers](#)
  - 7.3.2 [Linkers](#)
- 7.4 [Creating a Program in Assembly Language](#)
- 7.5 [Instructions Introduced Thus Far](#)
  - 7.5.1 [Instructions](#)
- 7.6 [Exercises](#)

## **[8 Program Data – Input, Store, Output](#)**

- 8.1 [Calling write in 64-bit Mode](#)
- 8.2 [Introduction to the Call Stack](#)
- 8.3 [Viewing the Call Stack](#)
- 8.4 [Local Variables on the Call Stack](#)
  - 8.4.1 [Calling printf and scanf in 64-bit Mode](#)
- 8.5 [Designing the Local Variable Portion of the Call Stack](#)
- 8.6 [Using syscall to Perform I/O](#)
- 8.7 [Calling Functions, 32-Bit Mode](#)
- 8.8 [Instructions Introduced Thus Far](#)
  - 8.8.1 [Instructions](#)
  - 8.8.2 [Addressing Modes](#)
- 8.9 [Exercises](#)

## **9 Computer Operations**

- 9.1 [The Assignment Operator](#)
- 9.2 [Addition and Subtraction Operators](#)
- 9.3 [Introduction to Machine Code](#)
  - 9.3.1 [Assembler Listings](#)
  - 9.3.2 [General Format of Instructions](#)
  - 9.3.3 [REX Prefix Byte](#)
  - 9.3.4 [ModRM Byte](#)
  - 9.3.5 [SIB Byte](#)
  - 9.3.6 [The `mov` Instruction](#)
  - 9.3.7 [The `add` Instruction](#)
- 9.4 [Instructions Introduced Thus Far](#)
  - 9.4.1 [Instructions](#)
  - 9.4.2 [Addressing Modes](#)
- 9.5 [Exercises](#)

## **10 Program Flow Constructs**

- 10.1 [Repetition](#)
  - 10.1.1 [Comparison Instructions](#)
  - 10.1.2 [Conditional Jumps](#)
  - 10.1.3 [Unconditional Jump](#)
  - 10.1.4 [while Loop](#)
- 10.2 [Binary Decisions](#)
  - 10.2.1 [Short-Circuit Evaluation](#)
  - 10.2.2 [Conditional Move](#)
- 10.3 [Instructions Introduced Thus Far](#)
  - 10.3.1 [Instructions](#)
  - 10.3.2 [Addressing Modes](#)
- 10.4 [Exercises](#)

## **11 Writing Your Own Functions**

- 11.1 [Overview of Passing Arguments](#)
- 11.2 [More Than Six Arguments, 64-Bit Mode](#)
- 11.3 [Interface Between Functions, 32-Bit Mode](#)
- 11.4 [Instructions Introduced Thus Far](#)
  - 11.4.1 [Instructions](#)
  - 11.4.2 [Addressing Modes](#)
- 11.5 [Exercises](#)

## **12 Bit Operations; Multiplication and Division**

- 12.1 [Logical Operators](#)
- 12.2 [Shifting Bits](#)
- 12.3 [Multiplication](#)
- 12.4 [Division](#)
- 12.5 [Negating Signed ints](#)
- 12.6 [Instructions Introduced Thus Far](#)
  - 12.6.1 [Instructions](#)
  - 12.6.2 [Addressing Modes](#)
- 12.7 [Exercises](#)

## **13 Data Structures**

- 13.1 [Arrays](#)
- 13.2 [structs \(Records\)](#)
- 13.3 [structs as Function Arguments](#)

- 13.4 [Structs as C++ Objects](#)
- 13.5 [Instructions Introduced Thus Far](#)
  - 13.5.1 [Instructions](#)
  - 13.5.2 [Addressing Modes](#)
- 13.6 [Exercises](#)

## **14 [Fractional Numbers](#)**

- 14.1 [Fractions in Binary](#)
- 14.2 [Fixed Point ints](#)
- 14.3 [Floating Point Format](#)
- 14.4 [IEEE 754](#)
- 14.5 [Floating Point Hardware](#)
  - 14.5.1 [SSE2 Floating Point](#)
  - 14.5.2 [x87 Floating Point Unit](#)
  - 14.5.3 [3DNow! Floating Point](#)
- 14.6 [Comments About Numerical Accuracy](#)
- 14.7 [Instructions Introduced Thus Far](#)
  - 14.7.1 [Instructions](#)
  - 14.7.2 [Addressing Modes](#)
- 14.8 [Exercises](#)

## **15 [Interrupts and Exceptions](#)**

- 15.1 [Hardware Interrupts](#)
- 15.2 [Exceptions](#)
- 15.3 [Software Interrupts](#)
- 15.4 [CPU Response to an Interrupt or Exception](#)
- 15.5 [Return from Interrupt/Exception](#)
- 15.6 [The `syscall` and `sysret` Instructions](#)
- 15.7 [Summary](#)
- 15.8 [Instructions Introduced Thus Far](#)
  - 15.8.1 [Instructions](#)
  - 15.8.2 [Addressing Modes](#)
- 15.9 [Exercises](#)

## **16 [Input/Output](#)**

- 16.1 [Memory Timing](#)
- 16.2 [I/O Device Timing](#)
- 16.3 [Bus Timing](#)
- 16.4 [I/O Interfacing](#)
- 16.5 [I/O Ports](#)
- 16.6 [Programming Issues](#)
- 16.7 [Interrupt-Driven I/O](#)
- 16.8 [I/O Instructions](#)
- 16.9 [Exercises](#)

## **A [Reference Material](#)**

- A.1 [Basic Logic Gates](#)
- A.2 [Register Names](#)
- A.3 [Argument Order in Registers](#)
- A.4 [Register Usage](#)
- A.5 [Assembly Language Instructions Used in This Book](#)
- A.6 [Addressing Modes](#)

## **B [Using GNU `make` to Build Programs](#)**

**C Using the gdb Debugger for Assembly Language**

**D Embedding Assembly Code in a C Function**

**E Exercise Solutions**

- E.2 [Data Storage Formats](#)
- E.3 [Computer Arithmetic](#)
- E.4 [Logic Gates](#)
- E.5 [Logic Circuits](#)
- E.6 [Central Processing Unit](#)
- E.7 [Programming in Assembly Language](#)
- E.8 [Program Data – Input, Store, Output](#)
- E.9 [Computer Operations](#)
- E.10 [Program Flow Constructs](#)
- E.11 [Writing Your Own Functions](#)
- E.12 [Bit Operations; Multiplication and Division](#)
- E.13 [Data Structures](#)
- E.14 [Fractional Numbers](#)
- E.15 [Interrupts and Exceptions](#)

**Bibliography**

**Index**