Table of Contents

Chapter 1. See a Program Running 1

1.1 Translate a C Program into a Machine Program 1
1.2 Load a Machine Program into Memory 4
1.2.1 Harvard Architecture and Von Neumann Architecture 4
1.2.2 Creating Runtime Memory Image 7
1.3 Registers 11
1.3.1 Reusing Registers to Improve Performance 12
1.3.2 Processor Registers 13
1.4 Executing a Machine Program 16
1.4.1 Loading a Program 17
1.4.2 Starting the Execution 18
1.4.3 Program Completion 22
1.5 Exercises 23

Chapter 2. Data Representation 25

2.1 Bit, Byte, Half-word, Word, and Double-word 25
2.2 Binary, Octal, Decimal, and Hexadecimal Numbers 27
2.3 Unsigned Integers 28
2.4 Signed Integers 29
2.4.1 Sign-and-Magnitude 31
2.4.2 One's Complement 32
2.4.3 Two's Complement 33
2.4.4 Overflow Flag for Signed Addition or Subtraction 37
2.5 Character String 47
2.6 Exercises 51

Chapter 3. ARM Instruction Set Architecture 53

3.1 ARM Assembly Instruction Sets 53
3.2 ARM Cortex-M3 Organization 55
3.3 Going from C to Assembly 57
3.4 Assembly Instruction Format 60
3.5 Anatomy of an Assembly Program 62
3.6 Assembly Directives 66
3.7 Exercises 70

Chapter 4. Arithmetic and Logic 71

4.1 Program Status Register 71
4.2 Updating Program Status Flags 72
4.3 Shift and Rotate 74
4.4 Arithmetic Instructions 76
4.5 Barrel Shifter 79
4.6 Bitwise Logic 80
4.7 Order of Bits and Bytes 84
4.8 Sign and Zero Extension 86
4.9 Data Comparison 87
4.10 Data Movement Between Registers 88
4.11 Exercises 89

Chapter 5. Load and Store 91

5.1 Load Constant into Registers 91
5.1.1 Data Movement Instruction MOV and MVN 91
5.1.2 Pseudo Instruction LDR and ADR 92
5.1.3 Comparison of LDR, ADR and MOV 93
5.2 Big Endian and Little Endian 94
5.3 Accessing Data in Memory 95
5.4 Memory Addressing 95
5.4.1 Pre-index, Post-index, and Pre-index with Update 95
5.4.2 Load and Store Instructions 97
5.4.3 PC-relative Addressing 98
5.4.4 Example of Accessing an Array 99
5.5 Exercises 100

Chapter 6. Branch and Conditional Execution 101

6.1 Condition Testing 101
6.2 Branch Instructions 104
6.3 Conditional Execution 107
6.4 If-then Statement 108
6.5 If-then-else Statement 111
6.6 For Loop 112
6.7 While Loop 113
6.8 Do While Loop 114
6.9 Continue Statement 115
6.10 Break Statement 116
6.11 Switch Statement 117
6.12 Exercises 119

Chapter 7. Structured Programming 123

7.1 Basic Control Structures 123
7.2 Register Reuse 128
7.3 Example of Factorial Numbers 131
7.4 Example of Counting Ones in a Word 132
7.5 Example of Finding the Maximum of an Array 134
7.6 Example of Counting Digits 136
7.7 Example of Parity Bit 137

Embedded Systems with ARM Cortex-M3 Microcontrollers in Assembly Language and C
Dr. Yifeng Zhu
542 pages, Copyright 2014
ISBN-10: 0982692625
ISBN-13: 978-0982692622

7.8 Example of Perfect Numbers 139
7.9 Example of Armstrong Numbers 141
7.10 Example of Palindrome String 142
7.11 Example of Converting String to Integer (atoi) 144
7.12 Example of Binary Search 145
7.13 Example of Bubble Sort 147
7.14 Exercises 149

Chapter 8. Subroutines 151

8.1 Calling a Subroutine 152
8.2 Stack 153
8.3 Preserving Runtime Environment via Stack 155
8.4 Passing Arguments to Subroutine via Registers 157
8.4.1 Pass a Variable by Value and by Reference 158
8.4.2 Example of Passing by Value 161
8.4.3 Write a Subroutine in Different Files 163
8.4.4 Example of Passing by Reference 164
8.4.5 Example of Greatest Common Divisor 165
8.4.6 Example of Concatenating Two Strings 167
8.4.7 Example of Comparing Two Strings 168
8.4.8 Example of Inserting an Integer into a Sorted Array 169
8.4.9 Example of Converting Integer to String (itoa) 170
8.4.10 Example of Matrix Transpose 172
8.4.11 Example of Removing a Character from a String 174
8.4.12 Example of Finding Unique Numbers in an Array 175
8.5 Passing Arguments Through Stack 178
8.6 Recursive Functions 180
8.6.1 Example of Factorial Numbers 182
8.6.2 Example of Reversing a String 184
8.6.3 Example of String Permutation 185
8.7 Exercises 187

Chapter 9. 64-bit Data Processing 189

9.1 64-bit Addition 189
9.2 64-bit Subtraction 190
9.3 64-bit Counting Leading Zeroes 191
9.4 64-bit Sign Extension 191
9.5 64-bit Logic Shift Left 192
9.6 64-bit Logic Shift Right 193
9.7 64-bit Multiplication 194
9.8 64-bit Unsigned Division 194
9.9 64-bit Signed Division 197
9.10 Exercises 198

Chapter 10. Mixing C and Assembly 199

10.1 Data Types and Access 200
10.1.1 Signed or Unsigned Integers 200
10.1.2 Data Alignment 201
10.1.3 Data Structure Padding 203
10.2 Special Variables 206
10.2.1 Static Variables 206
10.2.2 Volatile Variables 210
10.3 Inline Assembly 212
10.3.1 Assembly Functions in a C Program 212
10.3.2 Inline Assembly Instructions in a C Program 213
10.4 Calling Assembly Subroutines from a C Program 214
10.4.1 Example of Calling an Assembly Subroutine 214
10.4.2 Example of Accessing C Variables in Assembly 215
10.5 Calling C Functions from Assembly Programs 216
10.5.1 Example of Calling a C Function 216
10.5.2 Example of Accessing Assembly Data in a C Program 217
10.6 Exercises 218

Chapter 11. Fixed-point and Floating-point Arithmetic 221

11.1 Fixed-point Arithmetic 222
11.1.1 Fixed-point Representation 223
11.1.2 Fixed-point Range and Resolution Tradeoff 224
11.1.3 Fixed-point Addition and Subtraction 225
11.1.4 Fixed-point Multiplication 226
11.1.5 Fixed-point Division 227
11.2 Floating-point Arithmetic 228
11.2.1 Floating-point Representation 229
11.2.2 Special Values 233
11.2.3 Overflow and Underflow 234
11.2.4 Subnormal Numbers 235
11.2.5 Tradeoff between Numeric Range and Resolution 236
11.2.6 Rounding Rules 238
11.2.7 Floating-point Addition 240
11.2.8 Floating-point Multiplication 244
11.3 Exercises 247

Chapter 12. Interrupt 249

12.1 Introduction to Interrupt 249
12.2 Interrupt Service Routine (ISR) 250
12.3 Nested Vectored Interrupt Controller (NVIC) 254
12.3.1 Enable and Disable Peripheral Interrupts 255
12.3.2 Interrupt Priority 257

Embedded Systems with ARM Cortex-M3 Microcontrollers in Assembly Language and C
Dr. Yifeng Zhu
542 pages, Copyright 2014
ISBN-10: 0982692625
ISBN-13: 978-0982692622

12.3.3 Global Interrupt Enable and Disable 260

12.4 System Timer 261

12.5 External Interrupt 269

12.6 Software Interrupt 271

12.7 Exercises 273

Chapter 13. Instruction Encoding and Decoding 275

13.1 Tradeoff between Code Density and Performance 275

13.2 Dividing Bit Streams into 16- or 32-bit Instructions 276

13.3 Encoding 16-bit Thumb Instructions 278

13.4 Encoding 32-bit Instructions 279

13.5 Calculating Target Memory Address 280

13.6 Instruction Decoding Example 1 282

13.7 Instruction Decoding Example 2 286

13.8 Exercises 291

Chapter 14. Generic-purpose I/O 293

14.1 Introduction to Generic Purpose I/O (GPIO) 293

14.2 GPIO Input Modes: Pull Up and Pull Down 294

14.3 GPIO Output Modes: Push-Pull and Open-Drain 295

14.3.1 GPIO Push-Pull Output 295

14.3.2 GPIO Open-Drain Output 296

14.4 GPIO Output Speed: Slew Rate 298

14.5 Lighting up an LED 299

14.6 Push Button 304

14.7 Keypad Scan 308

14.8 Exercises 312

Chapter 15. General-purpose Timers 313

15.1 Timer Organization and Counting Modes 313

15.2 Compare Output 316

15.2.1 Setting Output Mode 317

15.2.2 Example of Toggling LED 319

15.3 PWM Output 321

15.3.1 PWM Output Events 324

15.3.2 PWM Programming Flowchart 325

15.4 Input Capture 329

15.4.1 Input Capture Timer Diagram 331

15.4.2 Configuring Input Capture 332

15.4.3 Interfacing to Ultrasonic Distance Sensor 336

15.5 Exercises 341

Chapter 16. Stepper Motor Control 343

16.1 Bipolar and Unipolar Stepper Motor 343

16.2 Step Angle 345

16.3 Wave Stepping 346

16.4 Full Stepping 347

16.5 Half Stepping 349

16.6 Micro-stepping 351

16.7 Driving Stepper Motor 353

16.8 Exercises 354

Chapter 17. Liquid-crystal Display (LCD) 355

17.1 Static Drive 356

17.2 Multiplexed Drive 357

17.3 LCD Software Driver 361

17.4 Generic LCD Driver to Display Strings 366

17.5 Exercises 370

Chapter 18. Real-time Clock (RTC) 371

18.1 UNIX Epoch Time 371

18.2 RTC Frequency Setting 372

18.3 Binary Coded Decimal (BCD) Encoding 373

18.4 RTC Initialization 374

18.5 RTC Alarm 377

18.6 Exercises 380

Chapter 19. Direct Memory Access (DMA) 381

19.1 DMA Bus Matrix 381

19.2 Programming DMA 383

19.3 Exercises 385

Chapter 20. Analog-to-Digital Converter 387

20.1 ADC Architecture 387

20.1.1 Digital Quantization 388

20.1.2 Sampling and Hold 390

20.2 ADC Sampling Error 391

20.3 ADC Conversion Modes 393

20.4 ADC Data Alignment 395

20.5 ADC Triggers 396

20.6 Measuring the Input Voltage 399

20.7 ADC Configuration Flowchart 399

20.8 ADC with DMA 403

20.9 Exercises 405

Chapter 21. Digital-to-Analog Converter 407

21.1 DAC Architecture 407

21.2 DAC on STM32L Processors 408

Embedded Systems with ARM Cortex-M3 Microcontrollers in Assembly Language and C
Dr. Yifeng Zhu
542 pages, Copyright 2014
ISBN-10: 0982692625
ISBN-13: 978-0982692622

21.3 Conversion Trigger 410
21.4 Buffered Output 411
21.5 Generating a Sinusoidal Wave via Table Lookup 412
21.6 Using Timer as a Trigger to DAC 416
21.7 Musical Synthesizing 420
21.7.1 Musical Pitch 420
21.7.2 Musical Duration 421
21.7.3 Amplitude Modulation of Tones 421
21.8 Exercises 426

Chapter 22. Serial Communication Protocols 427

22.1 Universal Asynchronous Receiver and Transmitter 427
22.1.1 Communication Frame 428
22.1.2 Baud Rate 429
22.1.3 Example Program Code in C 432
22.1.4 Serial Communication to Bluetooth Module 436
22.2 Inter-Integrated Circuit (I^{2}C) 439
22.2.1 Interfacing Serial Digital Thermal Sensors via I^{2}C 443
22.2.2 I^{2}C Clock Control 445
22.2.3 I^{2}C Maximum Rising Time 446
22.2.4 Sending Data to I^{2}C Slave 447
22.2.5 Receiving Data from I^{2}C Slave 448
22.2.6 Example Program Code in C 451
22.3 Serial Peripheral Interface Bus (SPI) 457
22.3.1 Data Exchange 458
22.3.2 Clock Configuration 460
22.3.3 Example Program Code in C 461
22.4 Universal Serial Bus (USB) 465
22.4.1 USB Bus Layer 466
22.4.2 USB Device Layer 468
22.4.3 USB Function Layer 471
22.4.4 USB Class Layer 479
22.4.5 Human Interface Device (HID) 480
22.5 Exercises 485

Chapter 23. Multitasking 487

23.1 Cortex-M3 Processor Mode and Privilege 487
23.2 Supervisor Call (SVC) 489
23.3 CPU Scheduling 492
23.4 Exercises 502

Appendix A: Cortex-M3 16-bit Thumb-2 Instruction Encoding 503

Appendix B: Cortex-M3 32-bit Thumb-2 Instruction Encoding 505

Appendix C: HID Codes of a Keyboard 512

Bibliography 514

Index 517

Embedded Systems with ARM Cortex-M3 Microcontrollers in Assembly Language and C
Dr. Yifeng Zhu
542 pages, Copyright 2014
ISBN-10: 0982692625
ISBN-13: 978-0982692622