

# Table of Contents

<b>Chapter 1.</b>	<b>See a Program Running .....</b>	<b>1</b>
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 .....	17
1.4.1	Loading a Program .....	18
1.4.2	Starting the Execution .....	19
1.4.3	Program Completion .....	23
1.5	Exercises .....	24
<b>Chapter 2.</b>	<b>Data Representation .....</b>	<b>27</b>
2.1	Bit, Byte, Halfword, Word, and Double-word .....	27
2.2	Binary, Octal, Decimal, and Hexadecimal Numbers .....	29
2.3	Unsigned Integers .....	30
2.4	Signed Integers .....	31
2.4.1	Sign-and-Magnitude .....	33
2.4.2	One's Complement .....	34
2.4.3	Two's Complement .....	35
2.4.4	Carry Flag for Unsigned Addition or Subtraction .....	36
2.4.5	Overflow Flag for Signed Addition or Subtraction .....	39
2.4.5.1	Interpreting the Carry and Overflow Flags .....	42
2.4.5.2	Two's Complement Simplifies Hardware Implementation .....	45
2.5	Character String .....	49
2.6	Exercises .....	53
<b>Chapter 3.</b>	<b>ARM Instruction Set Architecture .....</b>	<b>55</b>
3.1	ARM Assembly Instruction Sets .....	55
3.2	ARM Cortex-M Organization .....	58
3.3	Going from C to Assembly .....	60
3.4	Assembly Instruction Format .....	63
3.5	Anatomy of an Assembly Program .....	65
3.6	Assembly Directives .....	69
3.7	Exercises .....	73
<b>Chapter 4.</b>	<b>Arithmetic and Logic .....</b>	<b>75</b>
4.1	Program Status Register .....	75
4.2	Updating Program Status Flags .....	77
4.3	Shift and Rotate .....	78

4.4	Arithmetic Instructions .....	80
4.5	Barrel Shifter .....	83
4.6	Bitwise Logic Operations.....	84
4.7	Order of Bits and Bytes.....	89
4.8	Sign and Zero Extension.....	90
4.9	Data Comparison.....	91
4.10	Data Movement between Registers .....	92
4.11	Bit Field Extract .....	93
4.12	Exercises .....	94
<b>Chapter 5.</b>	<b>Load and Store .....</b>	<b>97</b>
5.1	Load Constant into Registers.....	97
5.1.1	Data Movement Instruction MOV and MVN.....	97
5.1.2	Pseudo Instruction LDR and ADR.....	98
5.1.3	Comparison of LDR, ADR, and MOV .....	99
5.2	Big Endian and Little Endian .....	100
5.3	Accessing Data in Memory .....	101
5.4	Memory Addressing .....	101
5.4.1	Pre-index, Post-index, and Pre-index with Update .....	101
5.4.2	Load and Store Instructions.....	103
5.4.3	PC-relative Addressing.....	104
5.4.4	Example of Accessing an Array .....	105
5.5	Loading and Storing Multiple Registers .....	106
5.6	Exercises .....	108
<b>Chapter 6.</b>	<b>Branch and Conditional Execution.....</b>	<b>111</b>
6.1	Condition Testing .....	111
6.2	Branch Instructions.....	114
6.3	Conditional Execution.....	117
6.4	If-then Statement .....	118
6.5	If-then-else Statement.....	121
6.6	For Loop.....	122
6.7	While Loop.....	123
6.8	Do While Loop.....	124
6.9	Continue Statement .....	125
6.10	Break Statement .....	126
6.11	Switch Statement .....	126
6.12	Exercises .....	129
<b>Chapter 7.</b>	<b>Structured Programming .....</b>	<b>133</b>
7.1	Basic Control Structures.....	133
7.2	Register Reuse.....	138
7.3	Example of Factorial Numbers .....	141
7.4	Example of Counting Ones in a Word .....	142
7.5	Example of Finding the Maximum of an Array .....	144

7.6	Example of Counting Digits.....	146
7.7	Example of Parity Bit.....	147
7.8	Example of Perfect Numbers .....	149
7.9	Example of Armstrong Numbers.....	151
7.10	Example of Palindrome String .....	152
7.11	Example of Converting String to Integer (atoi) .....	154
7.12	Example of Binary Search.....	155
7.13	Example of Bubble Sort.....	157
7.14	Exercises.....	159
<b>Chapter 8.</b>	<b>Subroutines .....</b>	<b>161</b>
8.1	Calling a Subroutine .....	162
8.2	Stack .....	164
8.3	Implementation of Stack via STM and LDM.....	165
8.4	Preserving Runtime Environment via Stack.....	166
8.5	Passing Arguments to Subroutine via Registers.....	169
8.5.1	Pass a Variable by Value and by Reference .....	170
8.5.2	Example of Passing by Value .....	173
8.5.3	Write a Subroutine in Different Files .....	175
8.5.4	Example of Passing by Reference.....	176
8.5.5	Example of Greatest Common Divisor .....	177
8.5.6	Example of Concatenating Two Strings.....	179
8.5.7	Example of Comparing Two Strings .....	180
8.5.8	Example of Inserting an Integer into a Sorted Array.....	181
8.5.9	Example of Converting Integer to String (itoa) .....	182
8.5.10	Example of Matrix Transpose.....	184
8.5.11	Example of Removing a Character from a String .....	186
8.5.12	Example of Finding Unique Numbers in an Array .....	187
8.6	Passing Arguments through Stack.....	190
8.7	Recursive Functions .....	192
8.7.1	Example of Factorial Numbers .....	194
8.7.2	Example of Reversing a String.....	196
8.7.3	Example of String Permutation .....	197
8.8	Exercises.....	199
<b>Chapter 9.</b>	<b>64-bit Data Processing.....</b>	<b>203</b>
9.1	64-bit Addition.....	203
9.2	64-bit Subtraction .....	204
9.3	64-bit Counting Leading Zeros .....	205
9.4	64-bit Sign Extension.....	205
9.5	64-bit Logic Shift Left .....	206
9.6	64-bit Logic Shift Right .....	207
9.7	64-bit Multiplication .....	208
9.8	64-bit Unsigned Division .....	209

9.9	64-bit Signed Division .....	211
9.10	Exercises .....	213
<b>Chapter 10.</b>	<b>Mixing C and Assembly .....</b>	<b>215</b>
10.1	Data Types and Access .....	216
10.1.1	Signed or Unsigned Integers .....	216
10.1.2	Data Alignment .....	217
10.1.3	Data Structure Padding .....	219
10.2	Special Variables .....	222
10.2.1	Static Variables .....	222
10.2.2	Volatile Variables .....	226
10.3	Inline Assembly .....	228
10.3.1	Assembly Functions in a C Program .....	228
10.3.2	Inline Assembly Instructions in a C Program .....	229
10.4	Calling Assembly Subroutines from a C Program .....	230
10.4.1	Example of Calling an Assembly Subroutine .....	230
10.4.2	Example of Accessing C Variables in Assembly .....	231
10.5	Calling C Functions from Assembly Programs .....	232
10.5.1	Example of Calling a C Function .....	232
10.5.2	Example of Accessing Assembly Data in a C Program .....	233
10.6	Exercises .....	234
<b>Chapter 11.</b>	<b>Interrupt .....</b>	<b>237</b>
11.1	Introduction to Interrupt .....	237
11.2	Interrupt Service Routine (ISR) .....	238
11.3	Nested Vectored Interrupt Controller (NVIC) .....	242
11.3.1	Enable and Disable Peripheral Interrupts .....	243
11.3.2	Interrupt Priority .....	245
11.3.3	Global Interrupt Enable and Disable .....	249
11.4	System Timer .....	250
11.5	External Interrupt .....	258
11.6	Software Interrupt .....	261
11.7	Exercises .....	262
<b>Chapter 12.</b>	<b>Fixed-point and Floating-point Arithmetic .....</b>	<b>265</b>
12.1	Fixed-point Arithmetic .....	266
12.1.1	Unsigned Fixed-point Representation .....	267
12.1.2	Signed Fixed-point Representation .....	268
12.1.3	Converting to Fixed-point Format .....	269
12.1.4	Fixed-point Range and Resolution Tradeoff .....	270
12.1.5	Fixed-point Addition and Subtraction .....	271
12.1.6	Fixed-point Multiplication .....	273
12.1.7	Fixed-point Division .....	274
12.2	Floating-point Arithmetic .....	275
12.2.1	Floating-point Representation .....	275

12.2.2	Special Values.....	280
12.2.3	Overflow and Underflow.....	281
12.2.4	Subnormal Numbers .....	282
12.2.5	Tradeoff between Numeric Range and Resolution.....	283
12.2.6	Rounding Rules.....	285
<b>12.3</b>	<b>Software-based Floating-point Operations .....</b>	<b>288</b>
12.3.1	Floating-point Addition .....	289
12.3.2	Floating-point Multiplication .....	293
<b>12.4</b>	<b>Hardware-based Floating-point Operations .....</b>	<b>295</b>
12.4.1	FPU Registers .....	295
12.4.1.1	Floating-point General-purpose Registers (S0-S31, or D0-D15) .....	295
12.4.1.2	Coprocessor Access Control Register (CPACR) .....	297
12.4.1.3	Floating-point Status and Control Register (FPSCR) .....	298
12.4.1.4	Floating-point Context Address Register (FPCAR) .....	300
12.4.1.5	Floating-point Context Control Register (FPCCR) .....	301
12.4.2	Load and Store Floating-point Numbers.....	302
12.4.3	Copy Floating-point Numbers .....	303
12.4.4	Copy and Set the Status and Control Register .....	303
12.4.5	Single-precision Arithmetic Operations .....	304
12.4.6	Single-precision comparisons.....	305
12.4.7	Precision Conversion.....	306
12.4.8	FPU Exception and Exception handling.....	308
12.4.9	Example Assembly Programs .....	310
12.4.9.1	Look up a Float Array.....	310
12.4.9.2	Sine Function of Argument in Radians.....	311
<b>12.5</b>	<b>Exercises.....</b>	<b>313</b>
<b>Chapter 13.</b>	<b>Instruction Encoding and Decoding .....</b>	<b>317</b>
13.1	Tradeoff between Code Density and Performance .....	317
13.2	Dividing Bit Streams into 16- or 32-bit Instructions .....	318
13.3	Encoding 16-bit Thumb Instructions .....	320
13.4	Encoding 32-bit Instructions .....	321
13.5	Calculating Target Memory Address .....	322
13.6	Instruction Decoding Example 1 .....	324
13.7	Instruction Decoding Example 2 .....	328
13.8	Exercises.....	333
<b>Chapter 14.</b>	<b>General Purpose I/O (GPIO) .....</b>	<b>335</b>
14.1	Introduction to General Purpose I/O (GPIO).....	335
14.2	GPIO Input Modes: Pull Up and Pull Down.....	336
14.3	GPIO Input: Schmitt Trigger.....	337
14.4	GPIO Output Modes: Push-Pull and Open-Drain.....	340
14.4.1	GPIO Push-Pull Output .....	340
14.4.2	GPIO Open-Drain Output .....	341
14.5	GPIO Output Speed: Slew Rate .....	343

14.6	Lighting up an LED .....	344
14.7	Push Button .....	349
14.8	Keypad Scan .....	353
14.9	Exercises .....	357
<b>Chapter 15.</b>	<b>General-purpose Timers .....</b>	<b>359</b>
15.1	Timer Organization and Counting Modes .....	359
15.2	Compare Output .....	362
15.2.1	Setting Output Mode .....	363
15.2.2	Example of Toggling LED .....	365
15.3	PWM Output .....	367
15.3.1	PWM Output Events .....	370
15.3.2	PWM Programming Flowchart .....	371
15.4	Input Capture .....	375
15.4.1	Input Capture Timer Diagram .....	377
15.4.2	Configuring Input Capture .....	378
15.4.3	Interfacing to Ultrasonic Distance Sensor .....	382
15.5	Exercises .....	387
<b>Chapter 16.</b>	<b>Stepper Motor Control .....</b>	<b>389</b>
16.1	Bipolar and Unipolar Stepper Motor .....	389
16.2	Step Angle .....	391
16.3	Wave Stepping .....	392
16.4	Full Stepping .....	393
16.5	Half Stepping .....	395
16.6	Micro-stepping .....	397
16.7	Driving Stepper Motor .....	399
16.8	Exercises .....	400
<b>Chapter 17.</b>	<b>Liquid-crystal Display (LCD) .....</b>	<b>401</b>
17.1	Static Drive .....	402
17.2	Multiplexed Drive .....	403
17.3	STM32L Internal LCD Driver .....	407
17.3.1	Basic Introduction .....	407
17.3.2	Generic LCD Driver to Display Strings .....	411
17.4	Interfacing with External Character LCD Controllers .....	416
17.4.1	External Connection Diagram .....	416
17.4.2	Internal Font Encoding .....	419
17.4.3	Sending Commands and Data to LCD .....	420
17.4.4	Programming Fonts .....	423
17.5	Exercises .....	424
<b>Chapter 18.</b>	<b>Real-time Clock (RTC) .....</b>	<b>425</b>
18.1	UNIX Epoch Time .....	425
18.2	RTC Frequency Setting .....	426

18.3	Binary Coded Decimal (BCD) Encoding .....	427
18.4	RTC Initialization .....	428
18.5	RTC Alarm.....	431
18.6	Exercises.....	434
Chapter 19.	Direct Memory Access (DMA).....	435
19.1	DMA Bus Matrix .....	435
19.2	Programming DMA .....	438
19.3	Exercises.....	442
Chapter 20.	Analog-to-Digital Converter (ADC).....	443
20.1	ADC Architecture .....	443
20.1.1	Digital Quantization .....	444
20.1.2	Sampling and Hold .....	446
20.2	ADC Sampling Error.....	447
20.3	ADC Conversion Modes.....	449
20.4	ADC Data Alignment .....	451
20.5	ADC Triggers .....	452
20.6	Measuring the Input Voltage.....	455
20.7	ADC Configuration Flowchart .....	455
20.8	ADC with DMA.....	460
20.9	Exercises.....	462
Chapter 21.	Digital-to-Analog Converter (DAC).....	463
21.1	DAC Architecture .....	463
21.2	DAC on STM32L Processors .....	464
21.3	Conversion Trigger .....	466
21.4	Buffered Output .....	467
21.5	Generating a Sinusoidal Wave via Table Lookup .....	468
21.6	Using Timer as a Trigger to DAC.....	472
21.7	Musical Synthesizing.....	476
21.7.1	Musical Pitch .....	476
21.7.2	Musical Duration .....	477
21.7.3	Amplitude Modulation of Tones .....	477
21.8	Exercises.....	482
Chapter 22.	Serial Communication Protocols.....	483
22.1	Universal Asynchronous Receiver and Transmitter.....	483
22.1.1	Communication Frame.....	484
22.1.2	Baud Rate .....	485
22.1.3	Example Program Code in C.....	488
22.1.4	Serial Communication to Bluetooth Module .....	492
22.2	Inter-Integrated Circuit (I <sup>2</sup> C) .....	495
22.2.1	Interfacing Serial Digital Thermal Sensors via I <sup>2</sup> C .....	499
22.2.2	I <sup>2</sup> C Clock Control.....	501

22.2.3	I <sup>2</sup> C Maximum Rising Time .....	502
22.2.4	Sending Data to I <sup>2</sup> C Slave .....	503
22.2.5	Receiving Data from I <sup>2</sup> C Slave .....	504
22.2.6	Example Program Code in C .....	507
<b>22.3</b>	<b>Serial Peripheral Interface Bus (SPI) .....</b>	<b>513</b>
22.3.1	Data Exchange .....	514
22.3.2	Clock Configuration .....	516
22.3.3	Example Program Code in C .....	517
<b>22.4</b>	<b>Universal Serial Bus (USB) .....</b>	<b>521</b>
22.4.1	USB Bus Layer .....	522
22.4.2	USB Device Layer .....	524
22.4.3	USB Function Layer .....	527
22.4.3.1	USB Descriptors .....	527
22.4.3.2	Endpoint-Oriented Communication .....	530
22.4.3.3	USB Enumeration .....	531
22.4.4	USB Class Layer .....	535
22.4.5	Human Interface Device (HID) .....	536
<b>22.5</b>	<b>Exercises .....</b>	<b>541</b>
<b>Chapter 23.</b>	<b>Multitasking .....</b>	<b>543</b>
23.1	Cortex-M Processor Mode and Privilege .....	543
23.2	Supervisor Call (SVC) .....	545
23.3	CPU Scheduling .....	548
23.4	Exercises .....	558
<b>Chapter 24.</b>	<b>Digital Signal Processing (DSP) .....</b>	<b>559</b>
24.1	Fixed-point and Floating-point DSP .....	559
24.2	Fixed-point Data Types in DSP .....	560
24.3	Saturation .....	561
24.4	Arithmetic Instructions .....	563
24.4.1	Parallel 8-bit Add and Subtract .....	565
24.4.2	Parallel 16-bit Add and Subtract .....	569
24.4.3	32-bit Add and Subtract .....	571
24.4.4	Sum of Absolute Difference .....	572
24.4.5	Extension and Add .....	573
24.4.6	Add and Subtract Halfwords with Exchange .....	575
24.4.7	16-bit and 32-bit Multiplication .....	577
24.4.8	16-bit Multiply and Accumulate with 64-bit Result .....	580
24.4.9	16-bit Multiply and Accumulate with 32-bit Result .....	583
24.4.10	16×32 Multiply and Accumulate with 32-bit Result .....	585
24.4.11	32×32 Multiply and Accumulate with 32-bit Result .....	585
24.4.12	Unsigned Long Multiply with Accumulate Accumulate .....	586
24.5	Packing Halfwords into a Word .....	588
24.6	Signed and Unsigned Extension .....	589
24.7	GE Flags .....	590



<b>24.8</b>	<b>Byte Selection Instruction .....</b>	<b>591</b>
<b>24.9</b>	<b>Basic DSP Functions .....</b>	<b>592</b>
24.9.1	Vector Negate .....	592
24.9.2	Vector Absolution Value .....	593
24.9.3	Vector Offset with Saturation .....	595
24.9.4	Vector Shift Left with Saturation.....	597
24.9.5	Vector Mean.....	598
24.9.6	Vector Pairwise Multiplication.....	600
24.9.7	Vector Dot Product .....	602
24.9.8	Vector Min and Max.....	603
<b>24.10</b>	<b>Exercises .....</b>	<b>604</b>
<b>Appendix A: Cortex-M3/M4 Instructions .....</b>		<b>607</b>
<b>Appendix B: Floating-point Instructions (Optional on Cortex-M4 and Cortex-M7) ...</b>		<b>609</b>
<b>Appendix C: DSP Instructions on Cortex-M4 and Cortex-M7 .....</b>		<b>611</b>
<b>Appendix D: Cortex-M0/M0+/M1 Instructions .....</b>		<b>615</b>
<b>Appendix E: Cortex-M3 16-bit Thumb-2 Instruction Encoding .....</b>		<b>617</b>
<b>Appendix F: Cortex-M3 32-bit Thumb-2 Instruction Encoding .....</b>		<b>619</b>
<b>Appendix G: HID Codes of a Keyboard .....</b>		<b>626</b>
<b>Bibliography</b>		<b>628</b>
<b>Index</b>		<b>633</b>

[This page intentionally left blank]