Thank you all for providing me feedbacks and corrections!

Chapter 1. See a Program Running

Chapter 2. Data Representation

Chapter 3. ARM Instruction Set Architecture
- Page 57, in Figure 3-3, “x DCW -1” should be “x DCW -2”.
- Page 63, in Figure 3-5, “BNE strcpy” is jumping to the beginning of the function, but should jump to the first LDRB instruction instead.
- Page 64, in the middle table, the explanation of REVSH should be “reverse byte order in the bottom halfword, and sign extend to 32 bits.”

Chapter 4. Arithmetic and Logic
- Page 87, “Note TEQ instruction cannot check the equivalence of two operands” Should be “Note TST instruction cannot check the equivalence of two operands”.
- Page 88, at the bottom of the page,
  “MRS CPSR, r0 ; copy CPSR to register r0
  MRS SPSR, r0 ; copy SPSR to register r0”
Cortex-M3 does not support this. Therefore, we change these examples to the following
  “MSR APSR, r0 ; Write flag state
  MSR BASEPRI, r0 ; Write base priority mask register; it prevents
  ; the activation of all exceptions with the same
  ; or lower priority level”

Chapter 5. Load and Store
- Page 98, “the difference between 0x08000144 and 0x0800012C is 20 in decimal. As a result, the memory address is written as [pc, #16].” should be “the difference between 0x08000144 and 0x0800012C is 24 in decimal. As a result, the memory address is written as [pc, #20].”
- Page 99, Example 5-5, “LDR r1, [pc,#20] ; @0x08000144” should be “LDR r1, [pc,#24] ; @0x08000144”
Chapter 6. Branch and Conditional Execution
• Page 105, Example 6-1

<table>
<thead>
<tr>
<th>C Program</th>
<th>Assembly Program</th>
</tr>
</thead>
</table>
| unsigned int x, y, z; x = 0x00000001; y = 0xFFFFFFFF; if (x > y) z = 1; else z = 0; | MOVS r5, #0x00000001 ; x
MOV r5, r6
CMP r5, r6
BLS then ; branch if ≤
MOVS r7, #1 ; z = 1
B endif ; skip the next instruction
then MOV r7,#0 ; z = 0
endif |

It should be changed to be the following.

<table>
<thead>
<tr>
<th>C Program</th>
<th>Assembly Program</th>
</tr>
</thead>
</table>
| unsigned int x, y, z; x = 0x00000001; y = 0xFFFFFFFF; if (x > y) z = 1; else z = 0; | MOVS r5, #0x00000001 ; x
MOV r5, r6
CMP r5, r6
BLS else ; branch if ≤
then MOV r7, #1 ; z = 1
B endif ; skip the next instruction
else MOV r7,#0 ; z = 0
endif |

Chapter 7. Structured Programming

Chapter 8. Subroutines

Chapter 9. 64-bit Data Processing

Chapter 10. Mixing C and Assembly

Chapter 11. Fixed-point and Floating-point Arithmetic

Chapter 12. Interrupt
• Page 256, Example 12-1,
  “LSR r2,r0,#3 ; Memory offset (in bytes): IRQn >> 3” should be
  “LSR r2,r0,#5 ; Memory offset (in bytes): IRQn >> 5”
• Page 257, “we only shift right IRQn by three bits instead of five bits This is because the register array index is based on words in the above in C code. However, in the assembly code, the STR instruction is based on the memory address, which is always in terms of bytes.

  \[ \text{Memory Address Offsets (in bytes)} = \frac{\text{Interrupt Number}}{8} \]

should be “we shift right IRQn by five bits. This is because the register array index is based on words in the above in C code and the STR instruction is based on the memory address, which is always in terms of bytes.

  \[ \text{Memory Address Offsets (in bytes)} = \frac{\text{Interrupt Number}}{32} \]
Chapter 13. Instruction Encoding and Decoding

Chapter 14. Generic-purpose I/O

Chapter 15. General-purpose Timers
• Page 336, Example 15-6,
  “NVIC_EnableIRQ(TIM4_IRQn); // Enable EXTI0_1 interrupt in NVIC” should be
  “NVIC_EnableIRQ(TIM4_IRQn); // Enable Timer 4 interrupt in NVIC”

Chapter 16. Stepper Motor Control

Chapter 17. Liquid-crystal Display (LCD)

Chapter 18. Real-time Clock (RTC)

Chapter 19. Direct Memory Access (DMA)

Chapter 20. Analog-to-Digital Converter
• Page 390, “For example, when the switch is closed for a time period of $2T_c$, the voltage across the capacitor $V_C$ is only 95.02% of the input voltage $V_{in}$ ” should read as “For example, when the switch is closed for a time period of $3T_c$, the voltage across the capacitor $V_C$ is only 95.02% of the input voltage $V_{in}$”

Chapter 21. Digital-to-Analog Converter

Chapter 22. Serial Communication Protocols

Chapter 23. Multitasking

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

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

Appendix C: HID Codes of a Keyboard

Bibliography

Index