CHAPTER 1: INTRODUCTION

No known errata.

CHAPTER 2: BINARY NUMBER SYSTEMS

Page 23 (bottom of page) and Page 24 (top of page): Change “Unsigned 8-bit representation of 25_{10}” to “Unsigned 8-bit representation of 20_{10}”.

Page 27, Problem 2: Change "8-bit" to "12-bit".

CHAPTER 3: WRITING FUNCTIONS IN ASSEMBLY

Page 47, Section 3.7.5: Change section title to “Function GetClockCycleCount”

Page 47: Add the following two sections:

3.7.6 Function PushButtonPressed
Function prototype:

```
int PushButtonPressed(void);
```

Returns 1 if the blue pushbutton is pressed, 0 if not.

3.7.7 Function GetRandomNumber
Function prototype:

```
uint32_t GetRandomNumber(void);
```

Returns a 32-bit value from the internal random number generator of the Cortex-M4F MCU.

CHAPTER 4: COPYING DATA

Section 4.6 (Addressing Modes), page 59, paragraph titled “PC-Relative addressing”:

The use of PC-Relative addressing with the STR instruction and its variants (STRB, STRH, STRD) has been deprecated, similar to the use of PC-Relative addressing with the VSTR instruction as described in Section 9.4.4. Although there are several examples such as STR R0, x found in chapter 4, they would actually be rejected by the assembler. To store into a memory location using its label requires a two-instruction sequence such as ADR R1, x or LDR R1,=x followed by STR R0, [R1]. The ADR instruction is only able to reference locations that are within 4095 bytes of the ADR, which would usually mean that the referenced location resides in flash memory (and therefore cannot be modified during execution). To use a label to reference data in read/write memory thus requires the LDR R1,=x instruction.
Note that since the text uses assembly to write small functions called from a C main program, there is rarely (if ever) a need for PC-Relative addressing because the operands of such functions are typically only the function parameters that are passed to the function in registers and the result is left in a register.

Page 53: Add the following footnote: "The memory operand of LDRD must reside at a mod 4 (word aligned) address to avoid an address alignment fault".

Page 57: Add the following footnote: "The memory operand of STRD must reside at a mod 4 (word aligned) address to avoid an address alignment fault".

Page 68: Add the following footnote: "The memory operands of LDMIA, STMIA, LDMDB and STMDB must reside at a mod 4 (word aligned) address to avoid an address alignment fault".

Page 69, Listing 4-1: Add the following note just below the listing: "The dst and src parameters are assumed to hold word aligned addresses, or else an address fault will occur".

CHAPTER 5: INTEGER ARITHMETIC

Page 91, Table 5-5: Remove the entries for non-existent instructions UQADD and UQSUB.

CHAPTER 6: MAKING DECISIONS AND WRITING LOOPS

Page 101, Figure 6-4: Replace the BLE instruction by BNE.

Page 104, bullet point 2 in the middle of the page: Change “Only the last instructions...” to “Only the last instruction...” (singular form).

Page 112, Problem 1b: Change the initialization in the for loop to "y = 1;".

CHAPTER 7: MANIPULATING BITS

Page 119, Table 7-1, 2nd row, 6th column: Replace "00001101" by "00000101".

Page 127, Listing 7-3:

Replace the code with the following sequence:

PackTime:

<table>
<thead>
<tr>
<th>Instruction</th>
<th>R0, R1, R2</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSL</td>
<td>R0, R1, 11</td>
<td>move hour into position at far left</td>
</tr>
<tr>
<td>BFI</td>
<td>R0, R1, 5, 6</td>
<td>insert minutes into bits 5-10 of R0</td>
</tr>
<tr>
<td>LSR</td>
<td>R2, R1, 1</td>
<td>divide the seconds by two</td>
</tr>
<tr>
<td>BFI</td>
<td>R0, R2, 0, 5</td>
<td>insert seconds/2 into bits 0-4 of R0</td>
</tr>
<tr>
<td>BX</td>
<td>LR</td>
<td></td>
</tr>
</tbody>
</table>

Page 133, Problem 1c:
Change the function prototype to:  int64_t ASR64(int64_t s64) ;

Page 134, Problem 7: Replace the first two sentences in the problem description by “Write a function in ARM assembly language that returns the number of bits in the parameter not including leading and trailing 0’s. For example, if the parameter is 006203F016, the function should return the value 19.”

Page 135, Problem 9, 1st sentence:
Change "the bit" to "bit 0 of that byte".

CHAPTER 8: MULTIPLICATION AND DIVISION REVISITED

Page 138, Table 8-2, 2nd row: Change “00111110” to “00011110”.

CHAPTER 9: GETTING STARTED WITH FLOATING POINT

Page 161, Section 9.6: Change “ARTIHMETIC” to “ARITHMETIC”

Page 162, Table 9-7, last 2 rows: Remove the word "Fused" in both rows, change "VFMA" to "VMLA", and change "VFMS" to "VMLS".

Page 162, Table 9-7, last 2 rows: Remove the word "Fused" in both rows, change "VFMA" to "VMLA", and change "VFMS" to "VMLS".

Page 163, Listing 9-1: Change "VMLA" to "VLMA".

Page 164, Table 9-8, 2nd row: Change "S_d,0" to "S_d,#0.0"

Page 167, Problem 6: Change the parameter "radians" to "x".

Pages 167-168, Problems 4, 5, 6: The exponents in each of the Taylor series expansions should be raised, as in:

Problem 4:  \( x^{-1} = (x - 1)^0 - (x - 1)^1 + (x - 1)^2 - (x - 1)^3 + \cdots \)

Problem 5:  \( \sin(x) = \frac{x^1}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots \)

Problem 6:  \( e^x = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots \)

CHAPTER 10: WORKING WITH FIXED-POINT REAL NUMBERS

Page 179, top of the page: Change the last term of the equation from A\(_{LO}\)B\(_{HI}\) to A\(_{LO}\)B\(_{LO}\)

Page 180, bottom of the page: Inside the box, change the number on the second rule from "1" to "2".

Page 181, Figure 10-4: Change the comment that appears to the right of the B\(_{63}\times A\) term to "If B<0, subtract A from the most-significant half of unsigned product".
Page 188, Programming Problem 3: Change "problem 0" to "problem 1".

Page 188, Programming Problem 3: Insert the word "to" between the words "program" and "compute".

Page 188-189, Problems 4, 5, 6: Change “Figure 10-1 and Figure 10-4” to “Listing 10-1 and Listing 10-4”.

Page 189-190, Problems 7, 8, 9: Change the reference to “problem 3” to “problem 6”.

Page 190, Problem 9: Change the parameter "radians" to "x".

CHAPTER 11: INLINE CODE

No known errata.

CHAPTER 12: PROGRAMMING PERIPHERAL DEVICES

No known errata.

APPENDIX A: CREATING AN EMBITZ PROJECT FROM SCRATCH

No known errata.

APPENDIX B: GRAPHICS LIBRARY FUNCTIONS

Page 233: Change the name of function ClearDisplay to ClearScreen.

Page 233: Add the following two function prototypes just below the prototype for function SetColor:

```c
void SetForeground(uint32_t color) ; // Same as SetColor
void SetBackground(uint32_t color) ;
```

Page 234: Remove the last parameter ("int alignment") from the function prototype for function DisplayStringAt.

APPENDIX C: TOUCH SCREEN LIBRARY FUNCTIONS

Page 235, Listing C-1: Insert a call to TS_Init() immediately before the while statement.