

## 2nd Edition Errata

Note: The number(s) in parentheses next to each item refers to the printing which contains the error. That is, 1 indicates an error in the second edition's first printing that has been corrected in subsequent printings, whereas (1,2) indicates the error is in both the first and second printings of the second edition.

### Page 6 (1,2,3)

Line 2: Replace AltaVec instruction set With AltiVec instruction set.

Line 29: Correct the chapter number for recursion to Chapter 17

Line 33: Correct the chapter number for pointer variables to Chapter 16

### Page 18, Problem 1.12 (1,2,3)

Replace: Lynne and Calvin are trying to decided . . .

With: Lynne and Calvin are trying to decide . . .

### Page 40, Example 2.14 (1,2,3)

For clarity, please replace: 110000 010 0010100000000000000000 with 1 10000010

0010100000000000000000

and 011111 110 1111111111111111111111 with 0 11111110 1111111111111111111111

### Page 41, Line 4. (1,2,3)

Replace: The entire set of eight-bit ASCII codes is listed in Figure E.3 of Appendix E.

With: The entire set of eight-bit ASCII codes is listed in Figure E.2 of Appendix E.

### Page 44, Problem 2.14. (1)

Replace: Add the following binary representaions.

With: Add the following bit patterns.

### Page 46, Problem 2.36, Part (e) (1,2,3)

Replace Develop a procedure ...

With Using the operations discussed in this chapter, develop a procedure ...

### Page 48, Problem 2.44. (1,2)

Replace ASCII representation for 3 (i.e., 0110 0011)

With ASCII representation for 3 (i.e., 0011 0011)

### Page 63, Section 3.3.4 (1,2)

First sentence of first paragraph should read:

Figure 3.17 illustrates...

### Page 81, Figure 3.33 (1,2)

The figure is missing a tristate driver to drive MDR onto the bus alongwith the control signal GateMDR.

Page 85, Problem 3.8. (1)

Replace  $Y = A \text{ AND } (B \text{ OR } C)$

With  $Y = \text{NOT } (A \text{ AND } (B \text{ OR } C))$

Page 87, Problem 3.18. (1,2,3)

Correct Part (d)'s last line to read as follows:

What information is lost?

Page 97, Line 3 under Section 4.1(1,2,3)

Replace: what the computer must do to complete

With: what the computer must do to complete

Page 121, Figure 5.4. (1,2,3)

The values in R3 and R5 should be swapped. R3 should be 1010111100001111 and R5 should be 0101000011110000. This makes the figure consistent with the text at the bottom of page 120.

Page 127, Figure 5.8. (1,2,3)

Replace the contents of the IR with 0110 001 010 011101

Page 138, Paragraph 5 (1,2,3)

Please correct EOT ( End of Text) to EOT ( End of Transmission).

Page 142, Figure 5.18 (1,2)

The figure is missing a tristate driver to drive MDR onto the bus alongwith the control signal GateMDR.

Page 148, Problem 5.22. (1,2)

Replace: x3010      1110 1110 0011 0001

With:      x3010      1110 0110 0011 1111

Page 149, Problem 5.26. (1,2,3)

Please correct the problem to read as follows:

Your task is to consider the successor to the LC-3. We will add ten additional opcodes to the ISA, and expand the register set from 8 to 16. We will change the memory to byte-addressable, with total address space of 64K bytes. Instructions will remain 16 bits wide. Also, we will encode all instructions, both old and new, with the same fields as the original 15 instructions, although we may need to change the size of some of the fields.

a. Is there any problem completing the detailed specification of the successor to the LC-3, as described above? Explain.

b. How many bits do we need in the PC to be able to address all of memory?

(c) and (d) remain the same.

Page 151, Problem 5.31. (1,2,3)

Replace: The figure at the top of the next page

With: The figure below

Page 151, Problem 5.32. (1,2,3)

Please correct the problem to read as follows:

If the condition codes have values  $N=0$ ,  $Z=0$ ,  $P=1$  at the beginning of the execution of the following sequence of LC-3 instructions, what will their values be at the end of the execution of the following sequence of LC-3 instructions?

x3050 0000 0010 0000 0010

x3051 0101 0000 0010 0000

x3052 0000 1110 0000 0010

x3053 0101 0000 0010 0000

x3054 0001 0000 0011 1111

Page 151, Problem 5.33. (1,2,3)

Please add the following instruction to the start of the program:

x2FFF 0101 0000 0010 0000

Page 162, Last line of second last paragraph of Section 6.1.4. (1,2,3)

Should read: .... essentially identical to Figure 5.16 of Chapter 5.

Page 172, Problem 6.6. (1,2,3)

Replace: Use your answers from Exercises 6.3 and 6.4

With: Use your answers from Exercises 6.4 and 6.5

Page 184, Figure 7.2 (1,2,3)

The figure is missing line 12 which is a comment:

12 ;

Page 187, line 4 under Note: (1)

Should read: greater than  $LC+1+255$  or less than  $LC+1-256$

Page 191, Problem 7.6. (1,2,3)

Please correct the problem to read as follows:

Our assembler has crashed and we need your help! Create a symbol table for the program shown below, and assemble the instructions at labels A, B and D.

```
.ORIG x3000
AND R0, R0, #0
A    LD R1, E
      AND R2, R1, #1
      BRp C
B    ADD R1, R1, #-1
C    ADD R0, R0, R1
```

```

        ADD R1, R1, #-2
D      BRp C
        ST R0, F
        TRAP x25
E      .BLKW 1
F      .BLKW 1
        .END

```

You may assume another module deposits a positive value into E before the module executes.  
 In fifteen words or fewer, what does the above program do?

Page 195, Problem 7.18 (1,2)

Please replace: If the strings are the same, the program terminates with the value 0 in R5

With: If the strings are the same, the program terminates with the value 1 in R5

Page 196, Problem 7.19 (1,2,3)

Please correct the problem to read as follows:

```

        .ORIG X3005
        LEA R2, DATA
        LDR R4, R2, #0
LOOP ADD R4, R4, #-3
        BRzp LOOP
        TRAP x25
DATA .FILL X000B
        .END

```

Page 196, Problem 7.20. (1)

Correct Part (a) to read:

```

        .ORIG x5000
        AND R0, R0, #0
        ADD R0, R0, #15
        ADD R0, R0, #6
        STI R0, PTR
        HALT
PTR .FILL x4000
        .END

```

Page 196, Problem 7.21. (1)

Correct the program to read:

```

        .ORIG x3000
        AND R0, R0, #0
        ADD R2, R0, #10
        LD R1, MASK
        LD R3, PTR1
LOOP LDR R4, R3, #0

```

```

        AND R4, R4, R1
        BRz NEXT
        ADD R0, R0, #1
NEXT ADD R3, R3, #1
        ADD R2, R2, #-1
        BRp LOOP
        STI R0, PTR2
        HALT
MASK .FILL x8000
PTR1 .FILL x4000
PTR2 .FILL x5000

```

What does the program do?

Page 197, Problem 7.26. New Problem (1,2,3)

Problem Statement: Recall the assembly language program of problem 7.6. Consider the following program:

```

        .ORIG x3000
        AND R0, R0, #0
D        LD R1, A
        AND R2, R1, #1
        BRp B
E        ADD R1, R1, #-1
B        ADD R0, R0, R1
        ADD R1, R1, #-2
F        BRp B
        ST R0, C
        TRAP x25
A        .BLKW 1
C        .BLKW 1
        .END

```

The assembler translates both assembly language programs into machine language programs. What can you say about the two resulting machine language programs?

Page 208, Figure 8.5, Lines 22, 23, and 24 (1,2,3)

Replace: .BKLW  
With: .BLKW

Page 217, Problem 8.15.(1,2)

Please correct the problem to read:

Interrupt-driven I/O:

(a) What does the following LC-3 program do?

```

        .ORIG    x3000
        LD       R3, A

```

```

        STI      R3, KBSR
AGAIN   LD       R0, B
        TRAP     x21
BRnzp   AGAIN
A       .FILL    x4000
B       .FILL    x0032
KBSR    .FILL    xFE00
        .END

```

(b) If someone strikes a key, the program will be interrupted and the keyboard interrupt service routine will be executed as shown below. What does the keyboard service routine do?

```

        .ORIG    x1000
        LDI      R0, KBDR
        TRAP     x21
        TRAP     x21
        RTI
KBDR    .FILL    xFE02
        .END

```

Note: RTI is an instruction that enables the computer to return to executing the program that was interrupted. It will be studied in Chapter 10. The only thing you need to know about it now is that it loads the PC with the address of the instruction that was about to be fetched when the interrupt occurred.

(c) Finally, suppose the program of part a started executing, and someone sitting at the keyboard struck a key. What would you see on the screen?

(d) In part c, how many times is the digit typed shown on the screen? Why is the correct answer: "I can not say for sure".

Page 235, Section 9.2.4 first paragraph line 9(1,2)

Replace : provide (in R0) a pointer to the starting address of the character string

With : provide (in R0) the starting address of the character string.

Page 237, Figure 9.11 (1,2)

Replace : 1\$, 2\$ with S1 and S2 respectively

Page 238, Figure 9.12 (1,2)

Replace : 1\$, 2\$ with S1 and S2 respectively

Page 241, Problem 9.5 (1,2,3)

Please correct the problem to read as follows:

```

        .ORIG    x3000
        LEA      R0, LABEL
        STR      R1, R0, #3
        TRAP     x22
        TRAP     x25
LABEL    .STRINGZ "FUNKY"

```

```
LABEL2 .STRINGZ "HELLO WORLD"  
      .END
```

Page 241, Problem 9.7 (1,2,3)

Please note that this problem belongs in chapter 10.

Page 243, Problem 9.11 (1,2,3)

This problem asks you to find the problem with the given code. However, there are multiple problems, only 1 of which was intended. Before solving the problem, replace line 8 of the program, currently "BRz G\_TEXT" with "BRnzp G\_TEXT". Also replace line 16 of the program, currently "LD R4. OADDR" with "LD R4, OADDR" (the only difference is that the period was replaced with a comma to avoid the syntax error).

Page 245, Problem 9.16 (1,2)

There is only one error in this problem.

Page 249, Problem 9.19 (1)

The devices should have been labeled (1)-(4) instead of (a)-(d).

Page 256, 3rd paragraph, line 3. (1,2,3)

Replace: . . . , the calling program can simply test Z to determine . . .

With: . . . , the calling program can simply test Z or P to determine . . .

Page 257, 3rd paragraph, line 3. (1,2,3)

Replace: . . . to determine whether the POP completed successfully . . .

With: . . . to determine whether the PUSH completed successfully . . .

Page 263, Figure 10.7 and the paragraph below (1,2,3)

This figure implies that first the PC of the interrupted program is pushed on the stack followed by the PSR. This is incorrect, the PSR should be pushed first, followed by the PC. The text in the paragraph beneath the figure should also be changed accordingly.

Page 268, 271, and 272, Figures 10.10, 10.14, and 10.15 (1,2,3)

Since the OpAdd algorithm is presented here as a subroutine, it should save and restore any registers that it uses. The same holds true for the OpMult and OpNeg algorithms. However, please note that later in the chapter (Figure 10.22 on page 280 and the text on page 281), we can see that when these algorithms are integrated with the calculator's main routine, they are not used as subroutines.

Page 273, Section 10.4.2, Line 4. (1,2,3)

Replace: ASCIIIBUFF.R1

With: ASCIIIBUFF. R1

Page 274, Figure 10.18 (1,2,3)

Replace : R1 ?= 0 with R1 = 0?

Page 276

At the bottom of page 276 are two exercises. The first of the two should be placed above the Section heading 10.4.3. It deals with ASCII to Binary conversion.

Page 279, Figure 10.21 (1,2,3)

The figure is missing a "-" in the conditional box which leads to OpNeg when true.

Page 280, Figure 10.22 (1,2,3)

Correct the following lines in the program:

05 ADD R6, R6, #1

23 BRz OpDisplay ; See Figure 10.26

Page 281, text beneath Figure 10.23, Line 2 (1,2,3)

Replace: . . . the main program of Figure 10.17.

With: . . . the main program of Figure 10.22.

Page 282, Figure 10.24 (1,2,3)

Lines 05 and 06 are incorrect and need to be replaced with 3 instructions as follows:

ADD R0, R0, #1 ; R0 = (addr. of StackBase + 1)

NOT R0, R0

ADD R0, R0, #1 ; R0 = -(addr. of StackBase + 1)

Page 283, Problem 10.3. (1,2,3)

In AFTER, R2 was misprinted as R0. Please correct.

Page 284, Problems 10.5 and 10.6. (1,2,3)

There are some awkward line breaks in these 2 problems, please ignore them.

Page 285, Problem 10.11. (1)

Replace : contents of locations x00F1 and x00F2

With : contents of locations x01F1 and x01F2

Page 285, Problem 10.14.(1)

Replace : If the buffer is full (i.e, if a character has been stored in location x40FF)

With : If the buffer is full (i.e, if a character has been stored in location x40FE)

Page 285, Problem 10.16.(1)

Replace : modified interrupt handler of Exercise 10.6

With : modified interrupt handler of Exercise 10.15

Page 287, Problem 10.23 (1,2,3)

First line of subroutine PUSH should be: PUSH ADD R6, R6, #-1

Second line of subroutine POP should be: ADD R6, R6, #1

Page 288, Problem 10.24. (1,2)

Please replace the problem with:



Suppose the keyboard interrupt vector is x34 and the keyboard interrupt service routine starts at location x1000. What can you infer about the contents of any memory location from the above statement?

Page 315, Figure 12.2, instruction after the label LOOP. (1)

Replace: The result is in R2

With: The result is in R0

Page 319, First Paragraph(1,2,3)

Please correct the Statement  $x = a*b + c*d/2$ ; to  $x=a*b + c*d/4$ ;

Page 328, Figure 12.5 (1,2,3)

Correct the caption to read: The compiler's symbol table when it compiles the program from Figure 12.4

Page 329, Figure 12.6, Caption. (1,2,3)

Replace: This function has five local variables

With: This function has six local variables

Page 333, Figure 12.10, Lines 8 and 9. (1,2,3)

The comments for lines 8 and 9 should be swapped.

Page 333, Figure 12.10, Lines 28 and 32. (1,2,3)

Replace: NOT R3

With: NOT R3, R3

Replace: NOT R2

With: NOT R2, R2

Page 352, Figure 13.7, Line 7. (1,2,3)

Replace: BRpz DONE

With: BRzp DONE

Page 356, Figure 13.10, Lines 11 and 17. (1,2,3)

Replace: BRpz DONE

With: BRzp DONE

Replace: STR R0, R5, #-1

With: STR R1, R5, #-1

Page 358, Figure 13.12, Line 5. (1,2,3)

Replace: Initial

With: Initialize

Page 373, Problem 13.4, part e. (1,2,3)

Replace: for part 4

With: for part d

Page 374, Problem 13.9, part b. (1,2,3)

There should be a semicolon after "while (a > 0)"

Page 384, Figure 14.3 (1,2,3)

Correct line 8 to read : double inner; /\* Inner radius \*/

Correct line 9 to read : double outer; /\* Outer radius \*/

Correct line 13 to read : scanf("%lf", &inner);

Correct line 16 to read : scanf("%lf", &outer);

Page 462 (1,2,3)

Correct the last two lines to read as follows:

```
/* diskNumber 1; startPost 3; endPost 2; midPost 1 */
```

```
MoveDisk(1, 3, 2, 1)
```

Page 464 (1,2,3)

Replace the fourth MoveDisk call - MoveDisk(1,2,3,1)

With : MoveDisk(1,3,2,1)

Page 469, Figure 17.15 (1,2,3)

Correct lines 21 and 30 to read : ADD R6, R6, #1 ; pop return value

Page 504, Figure 19.2 (1,2,3)

Correct line 16 to read : int Collide(Flight \*planeA, Flight \*planeB);

Correct line 17 to read : void PotentialCollisions(Flight planes[]);

Page 510, line 22 (1,2,3)

Replace: CarNode

With : Car

Page 530, JSR/JSRR(1,2)

Please correct the operation of the JSR/JSRR instruction to read:

```
TEMP = PC+;
```

```
if (bit[11] == 0)
```

```
    PC = BaseR;
```

```
else
```

```
    PC = PC+ + SEXT(PCOffset11);
```

```
R7 = TEMP;
```

Page 565, first sentence of Section C.1 Overview (1,2,3)

Replace: Figure C.1 shows the two main components of an ISA:

With: Figure C.1 shows the two main components of a microarchitecture:

Page 566, paragraph that begins with "Figure C.1 identifies . . ." (1,2,3)

Replace: . . . that corresponds to these five items.

With: . . . that corresponds to these six items.

Page 568, Figure C.2 (A state machine for the LC-3).(1,2)

Please correct states 4,20 and 21 in the figure to read:

State 4:

[IR11]

State 20:

PC  $\leftarrow$  BaseR

R7  $\leftarrow$  PC

State 21:

PC  $\leftarrow$  PC+off11

R7  $\leftarrow$  PC

Page 572, Table C.1 (1,2,3)

The table is missing an entry for the LD.Priority control signal. Add the following entry:

LD.Priority/1: NO, LOAD

Page 573, Figure C.5 (1,2,3)

The signal labels COND2, COND1, and COND0 should be COND[2], COND[1], and COND[0].

Page 580, 7th paragraph, line 1. (1,2,3)

Replace: In states 47 and 48, the same sequence as in states 37 and 56 occurs, . . .

With: In states 47 and 48, the same sequence as in states 37 and 41 occurs, . . .

Page 584, Figure C.9 (1,2,3)

The control store is missing a column for the Ld.Priority signal.

Page 589, Section D.3.1, Subsection float, Line 5. (1,2,3)

Replace: See Section 2.7.1.

With: See Section 2.7.2.

Page 607, Section D.7.5. (1,2,3)

Replace: for (x = 0; x < 100; X++)

With: for (x = 0; x < 100; x++)

Page 608, Section D.7.5. (1,2,3)

The paragraph starting with: If the continue statement is executed, . . .

Should begin with: In the example below, if the continue statement is executed, . . .

Page 610, Line 1. (1,2,3)

Replace: COPY

With: REMAINDER

Page 626 (1,2,3)

In the "Instruction" section, the pages for the "BR" instruction are listed as 131, 525, 526. The 526 should be 528.

