due at start of section on Friday 10 May 2019

Homework 11: LC-3 Machine Language

1. von Neumann Machines

Imagine that we change LC-3 memory to contain 1MB of byte-addressable memory. In other words, 2^{20} addresses, each holding 8 bits. Instructions remain as 16 bits (so now each instruction takes two consecutive memory locations).

a. How many bits are now needed for the PC?
b. How many bits are now needed for the IR?
c. How many bits are now needed for the MAR?
d. How many bits are now needed for the MDR?
e. Is instruction fetch faster, slower, or unaffected? Explain your answer.

2. Instruction Formats

The chief architect in charge of designing your company's new processor has drafted an ISA that includes many operations: ADD, SUBTRACT, XOR, XNOR, AND, NAND, OR, NOR, and NOT. The total number of opcodes (including operations, data movement, and control flow) is 19. The chief architect suggests having 15 registers in the register file. Instructions that require three register operands then need ceiling $[\log_2 (19 \times 15 \times 15 \times 15)] = 16$ bits.

In a few sentences, explain to the chief architect why eliminating a few of the opcodes and allowing 16 registers is a better choice in terms of the microarchitecture. For credit, your response must also allow for 16-bit instructions. Be specific about which opcodes should be eliminated.

3. Machines are Busy

Do problem 5.6 from Patt and Patel. Assume that the BUSYNESS vector contains bits for 16 machines instead of the 8 discussed in Section 2.7.1. Note that your instructions must be encoded into bits.

4. Reasoning About Offsets

Do problem 5.24 from Patt and Patel.

5. Executing XOR

Write a sequence of LC-3 instructions (in bits) to set R0 equal to R1 XOR R2. Assume that values have already been placed into R1 and R2 for you. You may not change the values of any other registers (only R0, R1, and R2). Include RTL or assembly comments explaining the action of each binary instruction. *Hints: You MAY change R1 and R2. Use at most eight instructions.*

6. Understanding Induction

The following LC-3 instructions execute starting from the point shown by the comment.

```
; start LC-3 execution here
0101 011 011 1 00000
0001 011 011 1 00001
0001 100 100 0 00 100
0000 101 111111101
; end LC-3 execution here
```

After the code reaches the end of the code (the last comment), what bits are held in R3? And in R4? And in R5? If you cannot know the bits held, explain why.

7. Understanding Memory Accesses

The following LC-3 instructions execute starting from the point shown by the comment.

```
1010 1011 1100 1101 ; this is location D
; start LC-3 execution here
0010 001 111111110
0001 001 001 1 00110
0101 011 001 1 01111
1011 011 111111011
; end LC-3 execution here
```

After the code reaches the end of the code (the last comment), what bits are held in R1? And in R3? And in memory location D? If you cannot know the bits held, explain why.

8. Understanding Loops

The following LC-3 instructions execute starting from the point shown by the comment.

```
0000 0000 0000 0110 ; this is location D
0000 0000 0000 0111 ; this is location D + 1
; start LC-3 execution here
0010 001 11111101
0010 010 11111101
0101 011 110 1 00000
0001 011 001 0 00 011
0001 010 010 1 11111
0000 001 111111101
1110 100 111110111
0111 011 100 000001
; end LC-3 execution here
```

After the code reaches the end of the code (the last comment), what bits are held in R1? And in R2? And in R3? And in R4? And in memory location D? And in memory location D + 1? If you cannot know the bits held, explain why.

9. Understanding Stability

The following LC-3 instructions execute starting from the point shown by the comment.

```
1111 0000 0000 0000 ; this is location D
; start LC-3 execution here
0101 011 011 1 00010
0010 101 111111101
1001 010 011 111111
0101 011 101 0 00 011
0101 101 101 1 11100
0000 100 111111100
; end LC-3 execution here
```

After the code reaches the end of the code (the last comment), what bits are held in R2? And in R3? And in R5? And in memory location D? If you cannot know the bits held, explain why.