HCMUT ECE 120 Homework 9

Solve the following problems from Patt & Patel: 4.4, 4.8, 4.16, 5.6, 5.10, 5.13

And do these problems:

1. **Memory and Registers**
   Imagine that we change LC-3 memory to contain 16MB of byte-addressable memory. In other words, 2^{24} 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. **Implementing XOR**
   Write a sequence of LC-3 instructions (in bits) to set R4 equal to R2 XOR R5. Assume that values have already been placed into R2 and R5 for you. You may not change the values of any other registers (only R2, R4, and R5). Include RTL or assembly comments explaining the action of each binary instruction. *For credit, use no more than ten instructions. Hint: You MAY change R2 and R5.*

3. **Mystery Code #1**
   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 R2? If you cannot know the bits held, explain why.

4. **Mystery Code #2**

   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.

5. **Mystery Code #3**

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

```
; start LC-3 execution here
0101 001 001 1 00000
0001 001 001 1 00001
0001 011 001 1 00000
0001 011 011 0 00 011
0001 010 010 1 11111
0000 101 111111101
0001 011 011 1 00000
0000 010 000000110
0101 110 100 0 00 011
0000 010 000000001
0001 010 010 0 00 001
0001 011 011 0 00 011
0001 001 001 0 00 001
0000 111 111111000
; end LC-3 execution here
```

   Assume that the initial value of R4 is xDCBA, and the initial value of R2 is x0008. After the code reaches the end of the code (the last comment), what bits are held in each of the eight registers? If you cannot know the bits held, explain why.

   How do the answers change if the initial values of R4 and R2 are changed? In other words, what does the code calculate, and where does it leave the answer?