Homework04

1. What does the following program do (in 20 words or fewer):

   0101 100 100 1 00000
   1001 000 001 111111
   0001 000 000 1 00001
   0001 000 000 000 010
   0000 011 000000001
   0001 100 100 1 00001
   1111 0000 0010 0101

   R2<R1 R4=1

   R2>=R1 R4=0

2. What does the following program do (in 20 words or fewer):

   0101 000 000 1 00000
   0101 101 001 1 00001
   0000 010 000000001
   0001 000 000 1 00001
   1111 0000 0010 0101
   R5 is even R0=0
   R5 is odd R0=1

3. (Adapted from 5.31) The following diagram shows a snapshot of the 8 registers of the LC-3 before and after the instruction at location x1000 is executed. Fill in the bits of the instruction at location x1000.

Register	Before	After
R0	x0000	x0000
R1	x1111	x1111
R2	x2222	x2222
R3	x3333	x3333
R4	x4444	x4444

R5	x5555	xFFF8
R6	x6666	x6666
R7	x7777	x7777

Memory Location	Value
x1000	0001 _101 000 1 11000__

4. The memory locations x3000 to x3007 contain the values as shown in the table below. Assume the memory contents below are loaded into the simulator and the PC has been set to point to location x3000. Assume that a breakpoint has been placed to the left of the HALT instruction (i.e. at location x3006 which contains 1111 0000 0010 0101). Assume that before the program is run, each of the 8 registers has the value x0000 and the NZP bits are 010.

Memory Location	Value
x3000	0101000000100000
x3001	0001000000100101
x3002	0010001000000100
x3003	0001000000000000
x3004	0001001001111111
x3005	0000001111111101
x3006	1111000000100101
x3007	0000000000000100

a. In no more than 15 words, summarize what this program will do when the Run button is pushed in the simulator.

   Hint: What relationship is there between the value loaded from memory and the final value in R0 after the program has completed?

   5 is put in R0 and shifted left the value at location x3007 times

b. What are the contents of the PC, the 8 general purpose registers (R0-R7), and the N, Z, and P condition code registers after the program completes?

   PC x3006

   R0 x0050

   R1 x0000

   R2 x0000

   R3 x0000

   R4 x0000

   R5 x0000

   R6 x0000

   R7 x0000

   N 0

   Z 1

   P 0

c. What is the total number of CPU clock cycles that this program will take to execute until it reaches the breakpoint?

   Note: You should refer to the state machine (pg 702) to determine how many cycles an instruction takes. Assume each state that access memory takes 5 cycles to complete and every other state takes 1 cycle to execute. States that check for ACV also take 1 cycle to execute

Memory Location	Value	Instruction	Cycles takes to exectue once	number of times executed	Total Cycles for instruction
X3000	0101000000100000	AND	10	1	10
X3001	0001000000100101	ADD	10	1	10
X3002	0010001000000100	LD	17	1	17
X3003	0001000000000000	ADD	10	4	40
X3004	0001001001111111	ADD	10	4	40
X3005	0000001111111101	Branch	10 if not taken 11 if taken	3 times taken 1 time not taken	43

Total Cycles 10+10+17+40+40+43 = 160

5. What does the following program do (in 15 words or fewer)? The PC is initially at x3000. (Assume that before the program is run,R0 has the value x0000.)

Memory Location	Value
x3000	0001 000 000 1 10000
x3001	0010 001 011111110
x3002	0000 010 000000100
x3003	0000 011 000000001

x3004	0001 000 000 1 00001
x3005	0001 001 001 000 001
x3006	0000 111 111111011
X3007	1001 000 000 111111
X3008	0001 000 000 1 00001
x3009	1111 0000 0010 0101

Counts the number of bits that are set to 0 in the word at x3100

6. Prior to executing the following program, memory locations x3100 through x4000 are initialized to random values, exactly one of which is negative. The following program finds the address of the negative value, and stores that address into memory location x3050. Two instructions are missing. Fill in the missing instructions to complete the program. The PC is initially at x3000.

Memory Location	**Value**
x3000	1110 000 011111111
x3001	**0110 001 000 000000**
x3002	**0000 100 000000010**
x3003	0001 000 000 1 00001
x3004	0000 111 111111100
x3005	0011 000 001001010

x3006	1111 0000 0010 0101

7. The LC-3 has just finished executing a large program. A careful examination of each clock cycle reveals that the number of executed store instructions (ST, STR, and STI) is greater than the number of executed load instructions (LD, LDR, and LDI). However, the number of memory write accesses is less than the number of memory read accesses, *excluding instruction fetches*. How can that be? Be sure to specify which instructions may account for the discrepancy

   A large number of LDI instructions (two read accesses) and STI instructions (one read access and one write access) could account for this discrepancy.

8. We would like to have an instruction that does nothing. Many ISAs actually have an opcode devote to doing nothing. It's usually called NOP, for NO OPERATION. The instruction is fetched, decoded, and executed. The execution phase is to do nothing! Which of the following three instructions could be used for NOP and have the program still work correctly?
   a) 0001 001 001 1 00000
   b) 0000 111 000000001
   c) 0000 000 000000000

   What does the instruction(s) couldn't be used for NOP do that other do not do?

   a) Add R1, R1, #0 => differs from a NOP in that it sets the CC's.
   b) BRnzp #1 => Unconditionally branches to one after the next address in the PC. Therefore no, this instruction is not the same as NOP.
   c) Branch that is never taken. Yes same as NOP.

9. The LC-3 does not have an opcode for the logical function OR. The four instruction sequence below performs the OR of the contents of register 1 and register 2 and puts the result in register 3. Fill in the two missing instructions so that the four instruction sequence will do the job.
   1) 1001 100 001 111111
   2) 1001 101 010 111111
   3) 0101 110 100 000 101

4) 1001 011 110 111111