1) Convert the following decimal numbers to 8-bit 2’s complement values, and then convert to two digits of hex. I’ll only check the hex answer. 1 point each.

33  -19  87  -44  -1

2) Convert the following 2-digit hex numbers (they are 8-bit 2’s complement) into decimal. 1 point each.

10  87  FF  AB  3D

3) Add the following pairs of 2-digit hex numbers together, and show the result in hex. IF THE RESULT IS OUT OF RANGE in 2 digits of hex, write "overflow" under you result. 2 points each

F3  13
F5  E6

4) Create a truth table for a full adder (it adds a single A and B bit, and a single Cin bit) 3 points.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Cin</th>
<th>Cout</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
5) Draw an inverter at the transistor level. 2 points.

6) Fill out a truth table for the following circuit. 4 points.

7) Draw a 2-input MUX using Boolean logic gates. 2 points.

8) Draw a 2-input decoder, using Boolean logic gates. 2 points.

9) Draw a gated D latch. 3 points.
(10: 5 points) Show the Karnaugh map for the following function, and use it to build a simple circuit that implements the function.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>f</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

(11: 3 points) You need to build a finite state machine for a machine that has three different lights, labeled A, B, and C. The machine has a single input switch. If the switch is on, the lights cycle through ABC, A-C, -BC, and then back to ABC (a dash means the light is off). If the switch is off, the lights move in the reverse order -BC, A-C, ABC. Draw the bubble diagram for this finite state machine (put the information on the lights inside the bubbles).
14) On an LC3 trap instruction, for example TRAP x22, how does the processor know where to branch to, in order to execute the trap routine? In other words, where does the new value for the PC come from? 2 points

15) Write LC3 code to push the ADDRESS of a variable X onto the stack, and then call a function DOUBLE. There are a pair of LC3 instructions you would use for push.... 6 points
   ; Main routine here -- push address of X on the stack, then call DOUBLE.
   ; And don't forget to do the appropriate thing to the stack after you return....
   ; You can assume that the register for the stack pointer is set up.

16) Write LC3 code for a function DOUBLE that will use the address provided on the stack to double the variable passed to the function. 7 points
   ; Your code for the function DOUBLE here. Your function should NOT make any permanent changes to any registers.
17) Convert the following hex values into LC3 assembly language. First convert to binary (using the boxes provided), and then write the assembly language equivalent directly above. The code starts at x3000. 2 points for each.

Instruction:  
xE006

Instruction:  
623F

Instruction:  
5401

Instruction:  
103D

Instruction:  
5440

Instruction:  
14AA

Instruction:  
HALT

18) After executing the code above, when you reach the HALT instruction, show the values in each of the registers (in hex). 4 points each.
R0          R1          R2

19) In lab, you had to convert LC3 assembly code into LC3 object code that could be run. What command would you use to convert foo.asm into something the simulator could use? 2 points.
20) Write LC3 assembly language subroutine called DOUBLE which will double the value in R0, and then return. 3 points.

21) Now, using your subroutine that you called DOUBLE, create a new subroutine called BYFOUR, which will multiply the value in R0 by calling DOUBLE twice. BYFOUR is also a subroutine. You will need to do something with one of the registers, to ensure that you can return to the code that calls BYFOUR. 3 points.

22) The following code modifies MESSAGE. What's MESSAGE look like after this runs? 2 points.

   LEA R0, MESSAGE
   LDR R2, R0, #2
   ADD R2, R2, #1
   STR R2, R0, #5
   HALT
   MESSAGE .STRINGZ "FUNPROBLEM"

23) Write LC3 code that is equivalent to this C language example. 2 points.

   if (R0 > 3)
     R2 = 5;
   else
     R2 = 6;
24) You have an LC3 stack. The values of A, B, C, and D are pushed onto the stack, in that order. Write the single LC3 instruction that you will need to get the value B into register R1. 2 points.

25) You can use a stack as temporary storage for computations. If you see a number, it is pushed on the stack. If you see math operation, it is performed immediately on the top two elements (with the result being pushed back). Suppose the following steps occur; what is on the top of the stack when you finish? 3 points.

```
3
7
2
add
3
multiply
add
```

26) You have a C program called foo.c. How would you compile this into an executable program called foo? 2 points.

27) What are the values of x and y at the end of this program? 4 points.

```c
int f(int a, int *b)
{
    a = 3;
    *b = *b + 2*a;
}
int main()
{
    int x, y;

    x = 10;
    y = 20;
    f(x, &y);
    // What are x and y now?
}
```

28) What's the first rule of Fight Club? (Here's that secret question again). 1 bonus point.

29) What's the name of the movie I've been trying to get you to go see? 1 bonus point.

30) What's the name of the main character, played by Benedict Cumberbatch? 1 bonus point.
### Figure A.2
Format of the entire LC-3 instruction set. **Note:** + indicates instructions that modify condition codes