1. What is microarchitecture?

2. The diagram below shows many muxes. If $A = 0b10$, $B = 0b11$, and $C = 0b00$, then what is Output?

```
A[1]
\  
A 0
B 1
A[0]
\  
C 0
B 1
C[1]
\  
A 0
C 1
B[0]
\  
C 0
A 1
\  
Output
```

3. In the diagram below shows 2 registers (REG1 and REG2), 3 inputs (D0, REG1_we, and REG2_we), and 2 (intermediate) outputs (D1 and D2). Complete the table below.

<table>
<thead>
<tr>
<th>Time</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>D0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>8</td>
<td>13</td>
<td>21</td>
<td>34</td>
</tr>
<tr>
<td>REG1_we</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>REG2_we</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>D1</td>
<td>55</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>89</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. Encode the following from assembly to machine code.

```assembly
subi r28,1 ; decrement down counter
breq 1 ; go print 1s counter if done with 8 bits
rjmp -8 ; loop to check next bit
```

5. Decode the following from machine code to assembly.

```assembly
1110000011001000 (0xE0C8)
0010111111011110 (0x2FDE)
```
6. This question walks through the process of implementing a new instruction in our microarchitecture. That new instruction will be asr (arithmetic shift right). There is more than 1 correct answer.

a. Describe any/all new states we need to add to our state machine to implement asr. Draw/include them in the state machine diagram below.

b. If you have nothing better to do over this weekend, then think about (or repeat) part a for different instructions.
   i. and
   ii. andi
   iii. com
   iv. dec
7. Trace the following instructions (they are borrowed and modified from lecture26-27). Do not be concern with the cycle (i.e. it is okay to start from either 0 or 1. It is also okay with either reset it after each instruction or continue counting up for the entire program).

```assembly
ldi r16, 45 ; 0xE20D
ldi r17, 23 ; 0xE117
ldi r18, 11 ; 0xE02B
add r17, r18 ; 0xF12
cp r17, r16 ; 0x1710
breq 4 ; 0xF021
rjmp -4 ; 0xCFFC
```

<table>
<thead>
<tr>
<th>Cycle</th>
<th>State</th>
<th>Changed registers/values and control signals (if any):</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
