CS220: Computer Organization
Quiz#2 Solution Sketch

Name:
Roll No.:

1. Consider a 2 GB DIMM card with one channel. The DIMM card has 64 x4 chips. The interface between the DIMM card and the SRAM cache is 128 bytes. The channel width is such that the best performing burst length is eight. Each chip has several banks and the row buffer size in each bank in a chip is 2 KB. The number of rows in a bank is 2048. A computer has installed two such DIMM cards to have a total 4 GB of memory and each DIMM card has its own channel. Write down the length (in number of bits) of different portions of an address as decoded by the DRAM controller below. **(1x7 points)**

```
-----------,-----------,-----------,-----------,-----------,-----------,-----------.
Row number     Rank         Bank     Column-high   Channel    Column-low
```

Solution: Since the best performing burst length is eight, the channel width must be exactly one-eighth of the interface between the DIMM card and the SRAM cache. Therefore, the channel width is 128 bits or 16 bytes. This fixes the least significant four bits of the address to 0 because $\log_2(\text{channelwidth})$ is the least significant four bits of the address. Since the interface between the DIMM card and the SRAM cache is 128 bytes, the channel bit (to select one of the two channels) would be placed after seven bits from the least significant side. The width of column-low is same as $\log_2(BL)$ i.e., 3 because column-low is incremented internally during the bursts of a burst length. Since the row buffer size is 2 KB or 16 Kb and the width of each column is 4 bits (due to x4 chips), the number of columns would be 16 Kb/4 bits = 4K. Therefore, column-high is 9 bits long. The capacity of a bank is 2 KB$\times$2048 = 2^{25} bits. The capacity of a chip is 2 GB/64 = 2^{28} bits. So, the number of banks per chip = $2^{28}/2^{25}$ = 8. So, the number of bits needed for bank number is 3. Since the channel width is 128 bits and the chips are x4, 32 chips are needed in a rank. Therefore, there are two ranks in each channel and the rank number needs one bit. The row number needs 11 bits. The total number of address bits is 32 (covering 4 GB of installed memory) as shown below.

```
    11          1          3           9          1          3          4
-----------,-----------,-----------,-----------,-----------,-----------,-----------.
Row number     Rank         Bank     Column-high   Channel    Column-low
```

2. A finite state machine (FSM) has 2020 states and a single-bit input. The FSM state starting from zero increments by one to reach state 2019 irrespective of the input. At state 2019, on input 0, the state changes to 30. At state 2019, on input 1, the state changes to 41. The FSM is implemented using a state sequencer, a microcode ROM, and dispatch ROMs. What is the total size (in bits) of the microcode ROM? **(1 point)**

Solution: There is only one branch point in the FSM. Therefore, only one dispatch ROM would be needed and the state selection multiplexer needs to select between the incremented state and the dispatch ROM output. Therefore,

the microcode ROM contains 0 in all rows except the last which contains 1. So, the width of the microcode ROM is one bit and it has 2020 rows. So, the total size is 2020 bits.

3. A computer represents all integers in two's complement using eight bits. Consider the integer $x = 11001100$. Write the two's complement representation of the largest integer that can be added to x without causing an overflow? Write the two's complement representation of the smallest integer that can be added to x without causing an overflow? **(1+1 point)**

Solution: Suppose we add y to x. There won't be any overflow if and only if the carry into the most significant position is equal to the carry out of the most significant position. To find the largest possible value of y, we set the most significant bit of y to zero. This implies that the no overflow condition is met irrespective of what we place in the next most significant position of y. Therefore, the largest y is 01111111. To find the smallest value of y, we set the most significant bit of y to one. Next, we need to make sure that the remaining bits of y evaluate to the smallest possible value while meeting the no overflow condition. This gives us smallest $y = 10110100$.