1. (1pts) What is the Hamming distance between these two bit patterns: 001101 and 101110?

2. (2pts) Write the equation for the carry out of the 5th adder cell in an ALU using carry-lookahead, in terms of P's and G's.

3. (2pts) Using a 4-1 mux, implement the following function:
   (!X) + (X * !Y * Z)

4. (2pts) What is the difference between the Mealy and Moore models of sequential design? What is the advantage to the Moore approach?

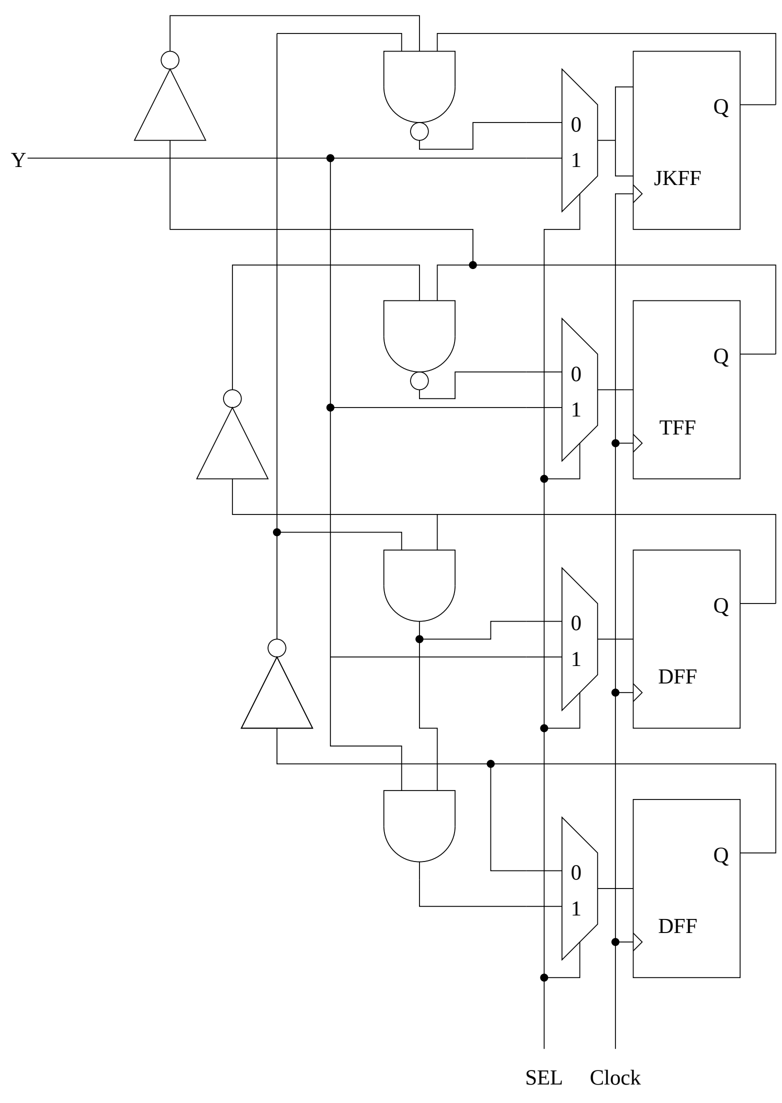
5. (2pts) What is the difference between a Flip-Flop and a latch?

6. (3 pts) Show how you can make an AND, an OR, and a NOT gate using only NOR gates.

7. (3pts) In the ALU you designed in the homework, how did you indicate that the values were to be subtracted instead of added?  Why did this work so well?

8. (10 pts) Assuming rising edge-triggered flipflops, what is the maximum clock frequency possible for the following circuit? (In other words, what is the maximum clock frequency that will still guarantee correct behavior?) Use the following delay values, and assume all input signals become valid at time 0 (including Y). (Tprop is the propagation time for the flipflop, the time it takes from the rising edge of the clock until the output of the FF is valid.)

AND: 3ns NAND: 4ns NOT: 2ns MUX: 5ns
Tprop (DFF): 10ns Tsetup (DFF): 5ns Thold (DFF): 3ns
Tprop (TFF): 8ns Tsetup (TFF): 3ns Thold (TFF): 2ns
Tprop (JKFF): 9ns Tsetup (JKFF): 2ns Thold (JKFF): 2ns



9. (3pts) How far apart must valid code words be to allow Triple (3) Error Detection (TED)?
Triple Error Correction (TEC)?
Quintuple (5) Error Correction sextuple (6) Error Detection (QECSED)?

10. (10 pts) Here is a 12-bit Error Correction code format (same one used in class):

$$d_8 \ \ d_7 \ \ d_6 \ \ d_5 \ \ C_4 \ \ d_4 \ \ d_3 \ \ d_2 \ \ C_3 \ \ d_1 \ \ C_2 \ \ C_1$$

a. Given the *data* bit pattern

**0 0 1 0 1 1 1 0**

in a machine using the above ECC code, what bit pattern gets sent to memory? (No credit will be given without work being shown.)

b. In this same machine, the following bit pattern is retrieved from memory:

**0 1 1 0 0 1 1 0 0 0 1 1**

Assuming the above Error Correction code format, identify and correct any errors that may have occurred during transmission or storage. (No credit will be given without work being shown.)

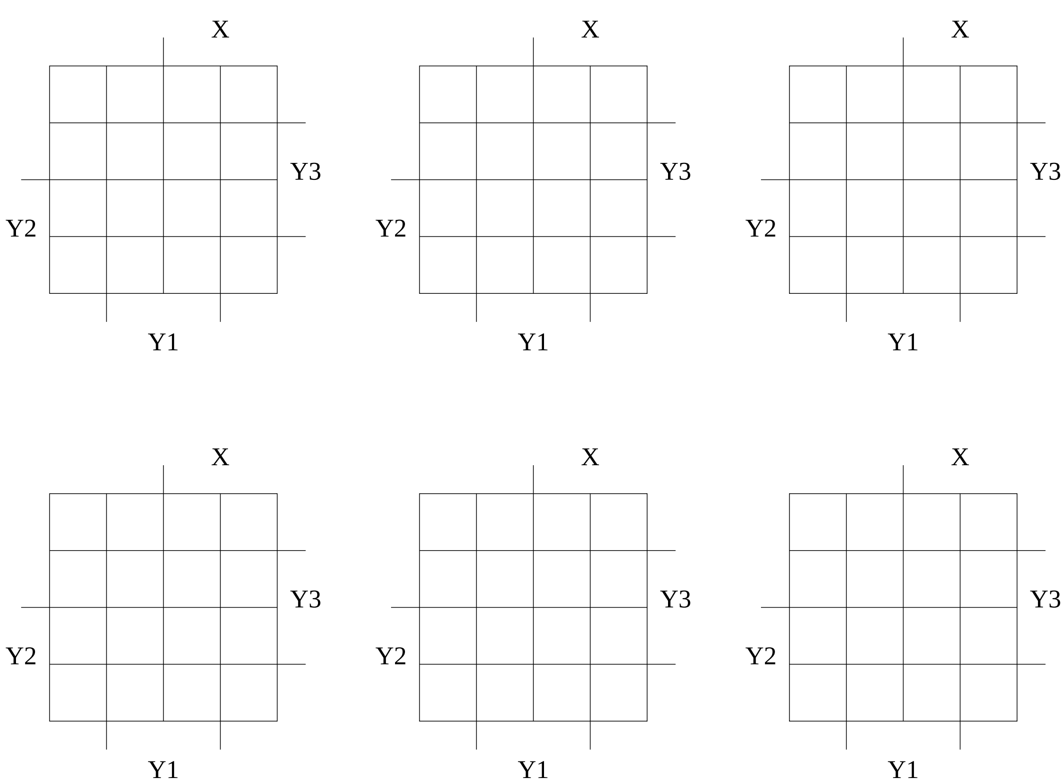
11. (11 pts) You have been asked to create a new flipflop, which has two inputs - the "TE" and the "ST". All you have to work with is a TFF. The TESTFF is to exhibit the following behavior:

| Present State | | Next State |
|---|---|---|
| TE | ST | Z' |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | Z |
| 1 | 1 | Zbar |

Write down what the T input must be (in terms of TE, ST, and Z) in order to provide the desired functionality. Be sure to minimize the equations.

12. (16) Given the following table, draw the Karnaugh maps for Y1’, Y2’, and Y3’ and Z in terms of X, Y1, Y2 and Y3, and then write **minimum** boolean equations for each.

| Present State (Y1 Y2 Y3) | Next State | | Output | |
|---|---|---|---|---|
| | X=0 (Y1’ Y2’ Y3’) | X=1 (Y1’ Y2’ Y3’) | X=0 | X=1 |
| 000 | 100 | 100 | 0 | 0 |
| 001 | 000 | 100 | 1 | 0 |
| 010 | 100 | 000 | 0 | 0 |
| 011 | 010 | 010 | 0 | 1 |
| 100 | 110 | 011 | 0 | 0 |
| 110 | 010 | 011 | 0 | 0 |



13. (15 pts) Given the following Karnaugh maps, implement the sequential machine using an SR FF for Y1, a JK FF for Y2, and a T FF for Y3. You do not need to draw the gates, but you do need to write down the **minimized** input equations for each of the inputs of each of the Flip Flops in the circuit.

**Y1’**

| | | Y1 | Y1, X | X | |
|---|---|---|---|---|---|
| | 1 | | | 1 | |
| Y3 | | | | | |
| Y2, Y3 | 1 | 1 | 1 | | |
| Y2 | 1 | 1 | 1 | d | |

**Y2’**

| | | Y1 | Y1, X | X | |
|---|---|---|---|---|---|
| | 1 | 1 | 1 | d | |
| Y3 | | | | 1 | |
| Y2, Y3 | 1 | | 1 | 1 | |
| Y2 | | | | d | |

**Y3’**

| | | Y1 | Y1, X | X | |
|---|---|---|---|---|---|
| | | | d | 1 | |
| Y3 | 1 | 1 | 1 | | |
| Y2, Y3 | | 1 | d | | |
| Y2 | | | 1 | d | |

| | | Y1 | Y1, X | X |
|---|---|---|---|---|
| | | | | |
| Y3 | | | | |
| Y2, Y3 | | | | |
| Y2 | | | | |

| | | Y1 | Y1, X | X |
|---|---|---|---|---|
| | | | | |
| Y3 | | | | |
| Y2, Y3 | | | | |
| Y2 | | | | |

| | | Y1 | Y1, X | X |
|---|---|---|---|---|
| | | | | |
| Y3 | | | | |
| Y2, Y3 | | | | |
| Y2 | | | | |

| | | Y1 | Y1, X | X |
|---|---|---|---|---|
| | | | | |
| Y3 | | | | |
| Y2, Y3 | | | | |
| Y2 | | | | |

| | | Y1 | Y1, X | X |
|---|---|---|---|---|
| | | | | |
| Y3 | | | | |
| Y2, Y3 | | | | |
| Y2 | | | | |

| | | Y1 | Y1, X | X |
|---|---|---|---|---|
| | | | | |
| Y3 | | | | |
| Y2, Y3 | | | | |
| Y2 | | | | |

14. (20 pts) Planet, Far Far Away ... Zargons build coin-operated robot to "take care" of Professor who they don’t like (his tests are too long.) Zargon has two coins: the bargo (worth 8 neeblicks) and the flant (worth 24 neeblicks). Robot requires input of 40 neebliks to carry out seek-and-destroy mission. The robot must give change (it would be immoral to do otherwise!), but maximum amount of change is 1 bargo. Let X1 be the bargo and X2 the flant, and assume both coins cannot be inserted simultaneously (Thus 10 = bargo inserted, 01 = a flant.)

Draw the State Transistion Diagram (the circles and the arcs) for this finite state machine. Let S0=no money input (the Start state). Once you have a state transition diagram, minimize the number of states necessary and then assign bit patterns to each state and write down the corresponding state transition table. Assume you are using a Mealy model. Label the transitions on the diagram using the format we used in class (inputs over outputs).