1. (1pts) What is the Hamming distance between these two bit patterns: 111101 and 101010?

2. (2pts) Write the equation for the carry out of the 4th adder cell in an ALU using carry-lookahead, in terms of P's and G's.

3. (2 pts) What is a Karnaugh Map?

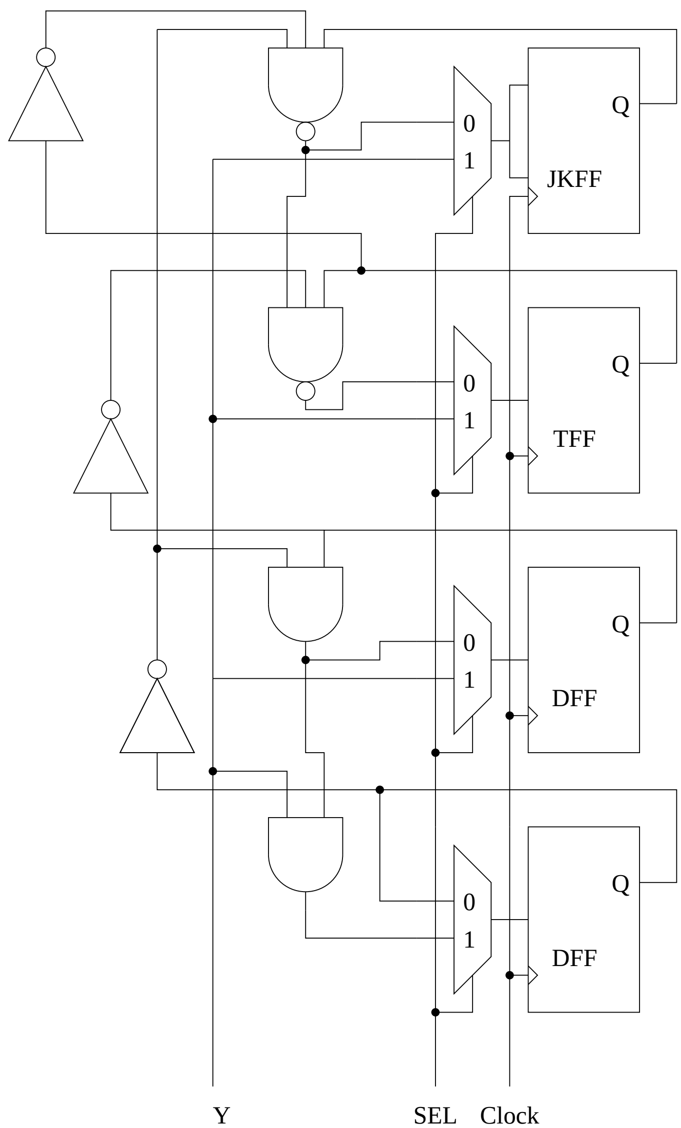
4. (2pts) What is the difference between a Flip-Flop and a latch?

5. (3pts) What is the difference between the Mealy and Moore models of sequential design? What is the advantage to the Moore approach?

6. (4pts) 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?

7. (11 pts) Assuming edge-triggered flipflops, what is the maximum clock frequency possible for the following circuit 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 from the rising edge of the clock until the output of the FF is valid.) **You must show the worst case path in order to receive full credit.**

AND: 3ns NAND: 4ns NOT: 1ns MUX: 5ns
Tprop (DFF): 7ns Tsetup (DFF): 5ns Thold (DFF): 3ns
Tprop (TFF): 8ns Tsetup (TFF): 3ns Thold (TFF): 2ns
Tprop (JKFF): 8ns Tsetup (JKFF): 2ns Thold (JKFF): 2ns



8. (3pts) How far apart must valid code words be to allow Double (2) Error Detection (DED)?
Triple Error Correction (TEC)?
Quadruple (4) Error Correction quintuple (5) Error Detection (QECQED)?

9. (6 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

1 1 0 1 1 0 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.)

10. (6pts) In the same machine as above, the following bit pattern is retrieved from memory:

0 1 1 0 0 0 1 0 0 1 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:

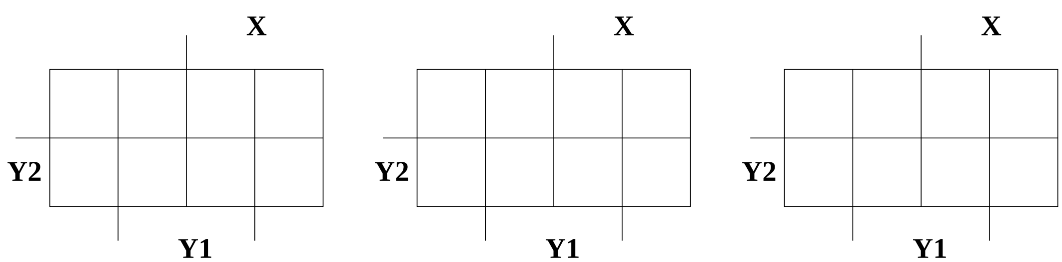
Present State		Next State
TE	ST	Z'
0	0	1
0	1	Z
1	0	0
1	1	Zbar

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. (3 pts) Show how you can make an AND, an OR, and a NOT gate using only NAND gates.

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

Present State	Next State		Output	
	X=0	X=1	X=0	X=1
(Y1 Y2)	(Y1’ Y2’)	(Y1’ Y2’)		
00	00	10	0	1
01	01	01	0	0
10	00	10	1	1



14. (4pts) Using a 4-1 mux, implement the following function:
(!X * !Z) + (X * Y) + (X * Z)

15. (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'

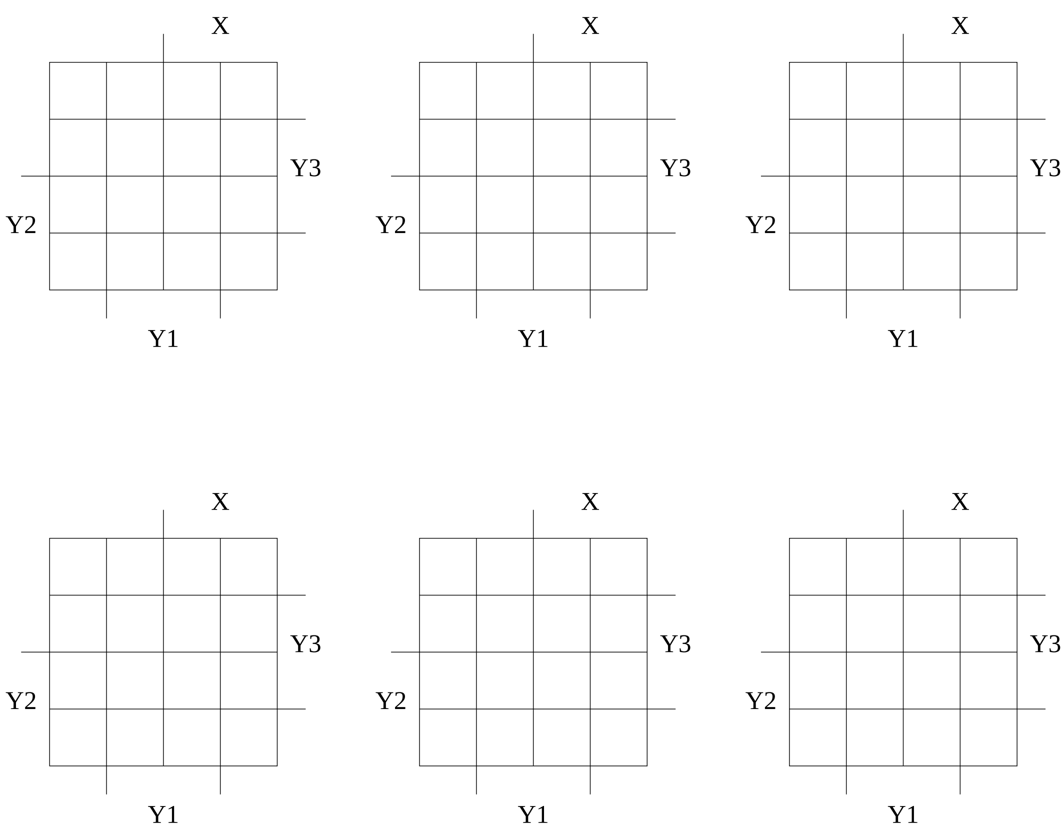
		Y1	Y1, X	X
	1	1	1	1
Y3				
Y2, Y3	1			1
Y2	1	1	1	d

Y2'

		Y1	Y1, X	X
	1		1	d
Y3				
Y2, Y3		1		1
Y2		1		d

Y3'

		Y1	Y1, X	X
			d	1
Y3		1	1	
Y2, Y3		1	d	
Y2				d



16. (7 pts) The Fleegles have built a coin-operated pet robot. Fleegle has two coins: the flanger (worth 6 twicks) and the flant (worth 18 twicks). Their robot requires an input of 40 twicks in order to sit up and beg. The robot must give change, but the maximum amount of change is 1 flanger. Let X1 be the flanger and X2 the flant, and assume both coins cannot be inserted simultaneously (Thus 10 = flanger inserted, 01 = a flant.)

Using a Mealy model, draw the State Transistion Diagram (the circles and the arcs) for this finite state machine. Label the transitions on the diagram using the format we used in class (inputs over outputs). Let state 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.

17. (8 pts) Now, repeat problem 16 using a Moore model. Assume the end states go back to state S0 whether or not any more money is input.