Given the following truth table:

A	B	C	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

(1) (2 pts) Write down the Canonical SOP expression for F

(2) (3 pts) Write down the minimum SOP expression for F

(3) (4 pts) Now use a 4-1 mux to implement F

(4) (3 pts) 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.

(5) (4 pts) Given

$$\mathbf{f} = \bar{A}\bar{B}\bar{C} + ABC + \bar{B}C + AB\bar{C}$$
$$\mathbf{g} = \bar{A}\bar{B} + AC + \bar{B}C + AB$$

Does f = g? (show why or why not)

(6) (2pts) How far apart must valid code words be to allow Triple (3) Error Detection (TED)? Septuple (7) Error Correction (SEC)?

(7) (2 pts) What is a Karnaugh Map?

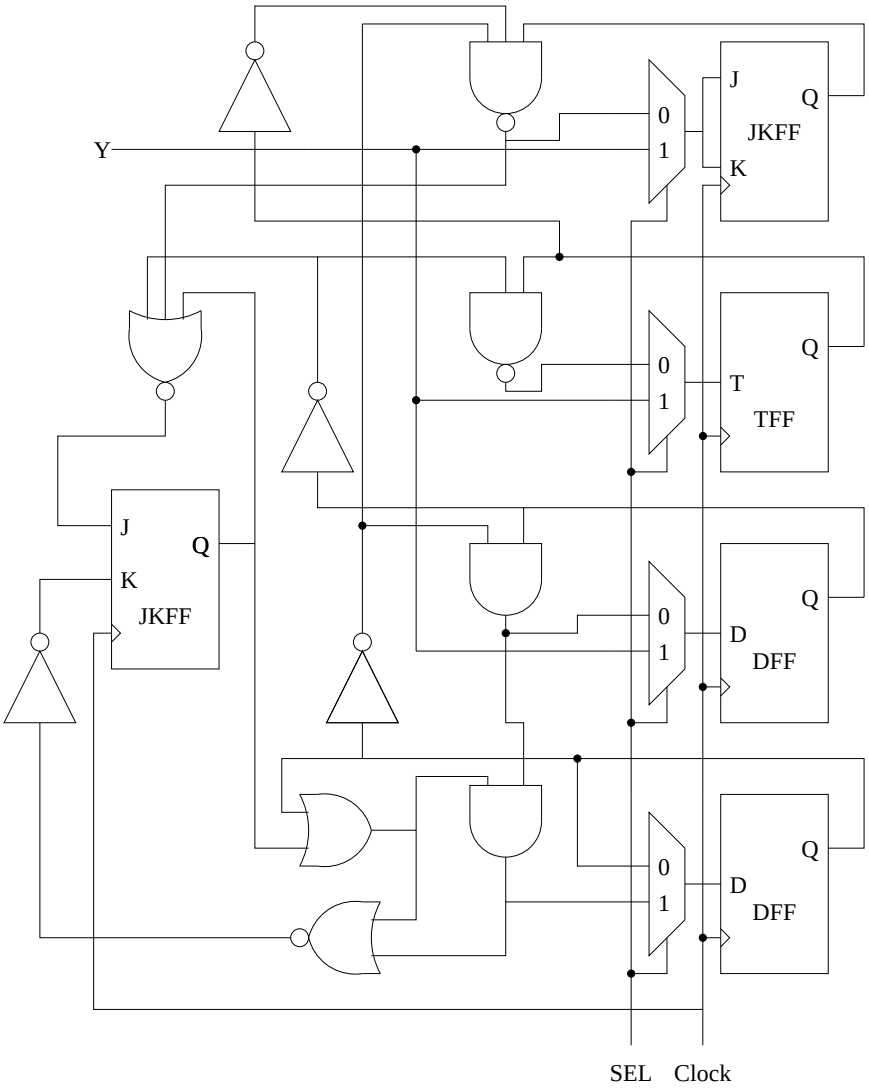
(8) (2pts) What is the difference between a Flip-Flop and a latch?

(9) (3pts) What is the difference between the Mealy and Moore models of sequential design? What is the advantage to the Moore approach?

(10) (9 pts) Assuming rising edge-triggered flipflops, what is the minimum cycle time possible for the following circuit? (In other words, what is the minimum cycle time that will still guarantee correct behavior?) Use the following delay values, and assume all input signals become valid at time 0. (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: 5ns OR: 3ns NAND: 3ns NOR: 4ns NOT: 1ns MUX: 5ns
Tprop (TFF): 8ns Tsetup (TFF): 4ns Thold (TFF): 1ns
Tprop (DFF): 9ns Tsetup (DFF): 2ns Thold (DFF): 1ns
Tprop (JKFF): 7ns Tsetup (JKFF): 3ns Thold (JKFF): 1ns

Note: You must show the path in order to get credit.



(11) (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

0 1 1 0 1 1 0 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.)

(12) (6pts) In the same machine as above, the following bit pattern is retrieved from memory:

0 1 1 0 1 0 1 0 1 1 1 0

Assuming the data was encoded using the same 12-bit Error Correction code format described in the previous question, identify and correct any errors that may have occurred during transmission or storage. (No credit will be given without work being shown.)

(13) (6 pts) You have been asked to create a new flipflop, which has two inputs - the "A" and the "B". All you have to work with is a TFF. The ABFF is to exhibit the following behavior:

Present State		Next State
A	B	Z’
0	0	Zbar
0	1	0
1	0	1
1	1	Z

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

(14) (6 pts) You have derived the following karnaugh maps for the inputs to a JK flip-flop. Unfortunately, the parts department just called and your company is completely out of JK flip-flops. All they have left in stock is SR flip-flops, which you will have to use instead. Show the resulting karnaugh map for the modified version of the circuit (the one that uses the SR instead of the JK flip-flop.), and minimize the equations.

J'

Y2 \ Y3	X			
	1	d	d	
	d	d	d	d
	d	d	d	d
	1	d		1

Y1

K'

Y2 \ Y3	X			
	d	d	d	d
		1	1	
		1	1	
	d	d	d	d

Y1

S'

Y2 \ Y3	X			

Y1

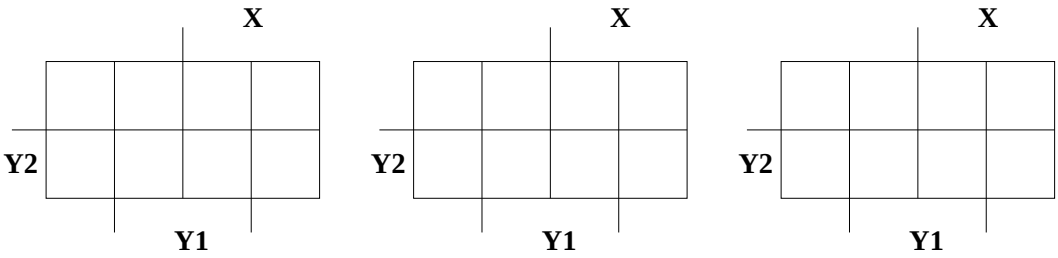
R'

Y2 \ Y3	X			

Y1

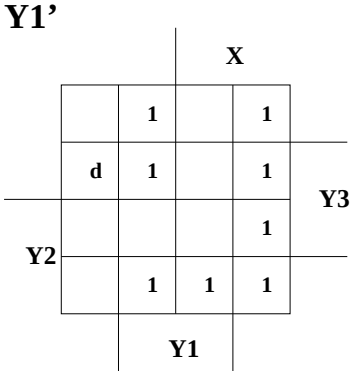
(15) (9) Given the following table, draw the Karnaugh maps for Y1', Y2' and Z in terms of X, Y1 and Y2, and then write **minimum** boolean equations for each.

Present State (Y1 Y2)	Next State X=0 (Y1' Y2')	Next State X=1 (Y1' Y2')	Output X=0	Output X=1
00	11	10	1	1
10	00	10	0	0
11	00	00	0	0

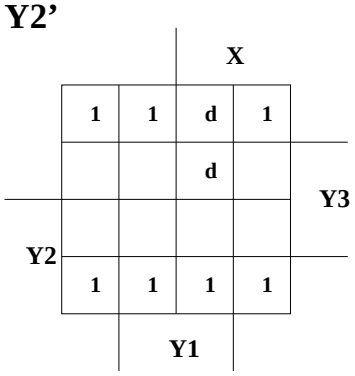


(16) (15 pts) Given the following Karnaugh maps, implement the sequential machine using an SR FF for Y1, a JK FF for Y2, and a T FF for Y3. You do not need to draw the gates, but you do need to write down the **minimized** input equations for each of the inputs of each of the Flip Flops in the circuit.

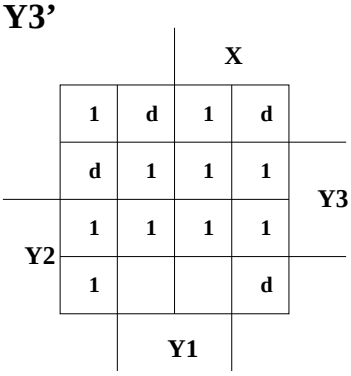
Y1'

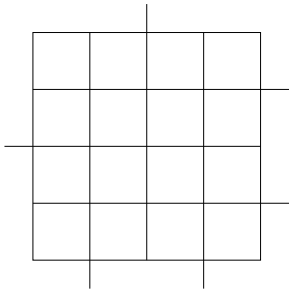


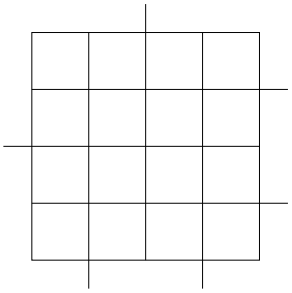
Y2'

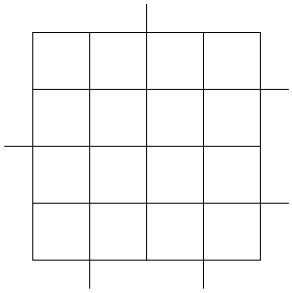


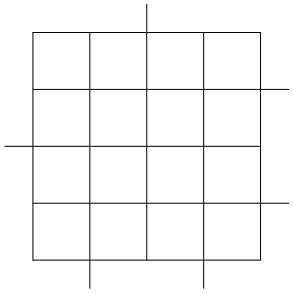
Y3'

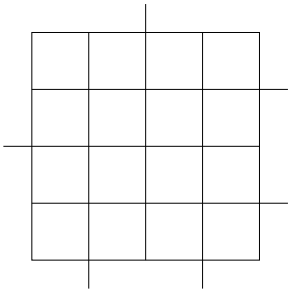


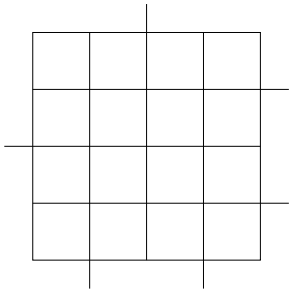












(17) (6 pts) A coin-operated machine takes dimes (X1) and nickels (X2), and dispenses merchandise (Z1=1) when the sum of the inputs is greater than or equal to 20. Only 1 coin can be input at a time. Naturally, the machine gives change.

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).

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

(18) (8 pts) Using this state transition diagram, minimize the number of states necessary and then assign bit patterns to each state. (Make a state transition table, in other words.)