CSE 370 Final Exam

Winter

1. True False (1 point each, 14 points total):

(a) You can implement any Boolean function using only de-multiplexers. F

(b) You can implement any Boolean function using only NAND gates. T

(c) A Moore state machine usually has fewer states than the equivalent Mealy machine. F

(d) A D flip-flop samples its input on one edge of the clock, and changes its output on the other edge. F

(e) A register is a group of storage elements (latches or flip-flops) that you read or write as a single unit. T

(f) The twos-complement number system can represent larger negative numbers (in absolute value) than positive numbers for a given bit-width. T

(g) For large adders, a ripple-carry approach will be faster (i.e. shorter combinational-logic delay) than a carry-lookahead approach. F

(h) The number b'0110011' is identical in twos-complement, ones-complement, and unsigned binary. T

(i) Taking a Boolean expression's dual is the same as applying DeMorgan's theorem. F

(j) An NAND gate is the same as a NOR gate with its inputs complemented. F

(k) f(ABCD) = $\Sigma m(4,5,10,11,12,13,14,15) = (A + C')(B + C)$ T

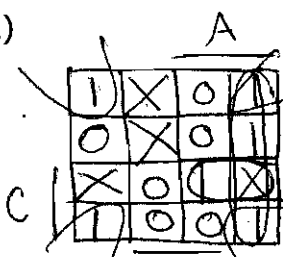
(l) A static 1-hazard causes an output that should stay at 0 to go high for a short time. F

(k) In general, an output encoding of an FSM uses fewer state bits than a binary encoding F

2. (8 points) The following questions refer to the function

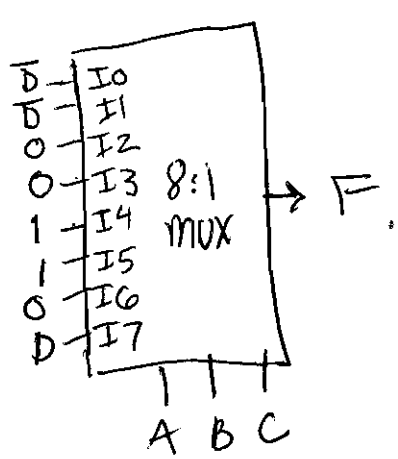
$$F(A,B,C,D)=\Sigma\, m(0,2,8,9,10,15)+ \Sigma\, d(3,4,5,11)$$

(a). (4 points) Use a K-map to write a minimized sum-of-products for F.

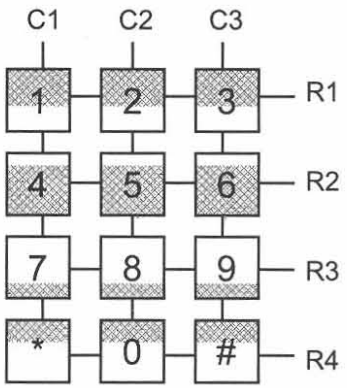


$F = \overline{B}\,\overline{D} + A\overline{B} + ACD$

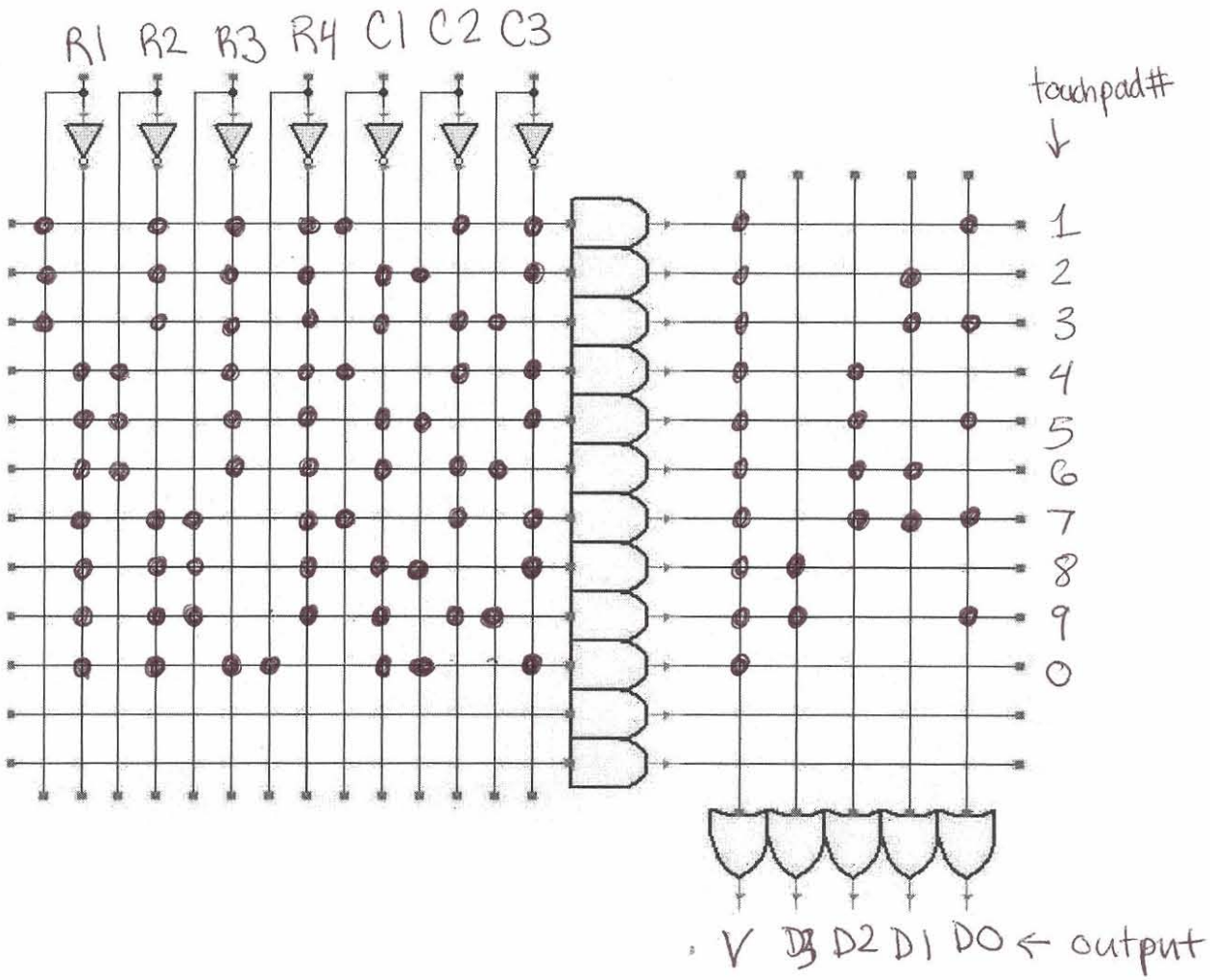
(b). (4 points) Use a ~~3:8~~ 8:1 multiplexer and a minimum of additional logic to write a circuit for F.



3. (8 points) A standard telephone touchpad is shown on the right. It consists of a 4 x 3 array of buttons. Which button is pressed is detected via 4 wires to indicate the Row of the button pressed (R1, R2, R3, R4) and 3 wires to indicate the Column of the button pressed (C1, C2, C3). The Row (Column) wires are High unless some button in that Row (Column) is pressed, in which case the wires become Low.

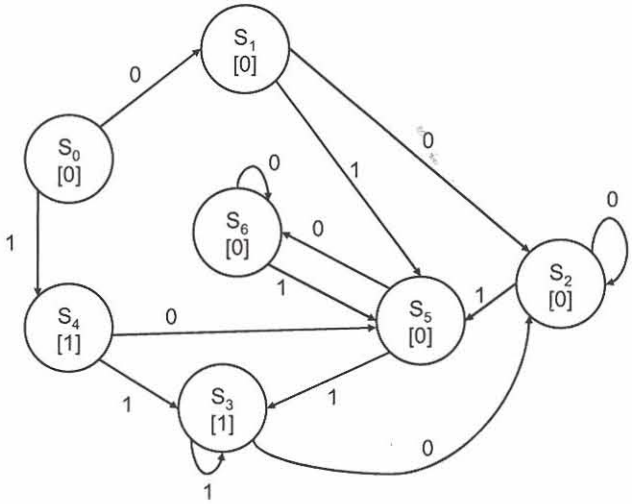


Design a decoder for this keypad that will convert the Row and Column signals to an encoding of the digit pressed in BCD format (D3, D2, D1, D0). Since some combinations of Row and Column signals do not represent valid inputs (non-numerical inputs # and * are also not valid) you need to produce a Valid bit (V) that will be 1 if some digit was entered. Design a circuit implementing this function on the PAL diagram below. Label all your inputs and outputs.

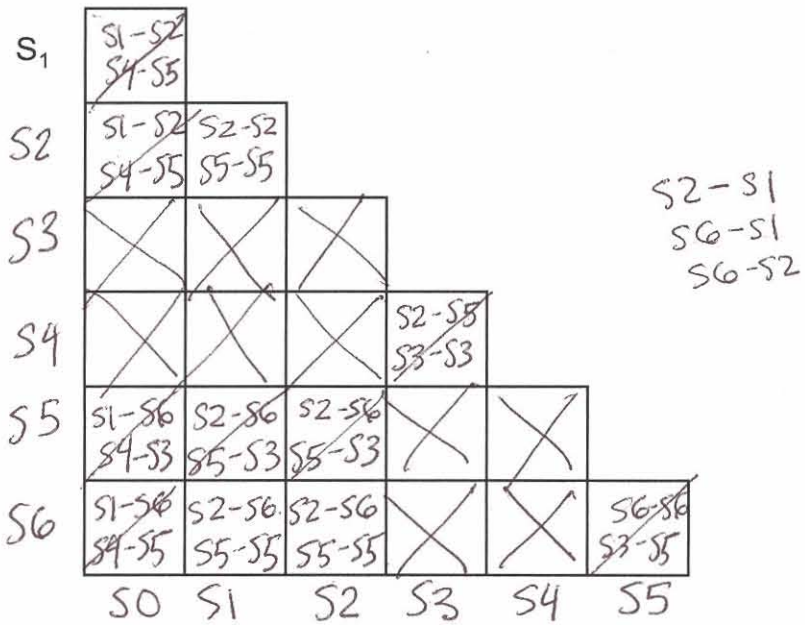


4. (14 points) Verilog question removed.

5. (15 points) Given the following state diagram.



5(a) (6 points) Label and fill out the entries of the implication table below, listing the order in which you cross off cells in part (b) as you do:

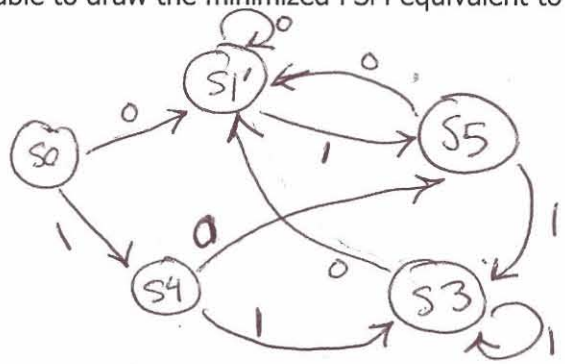


(b) (3 points) List the state pairs (as Sa-Sb) in the order in which you crossed them off in (a).

S1-S0 S2-S0 S6-S0 S5-S1 S5-S2 S6-S5 S4-S3

(c) (6 points) Use the above table to draw the minimized FSM equivalent to the above FSM.

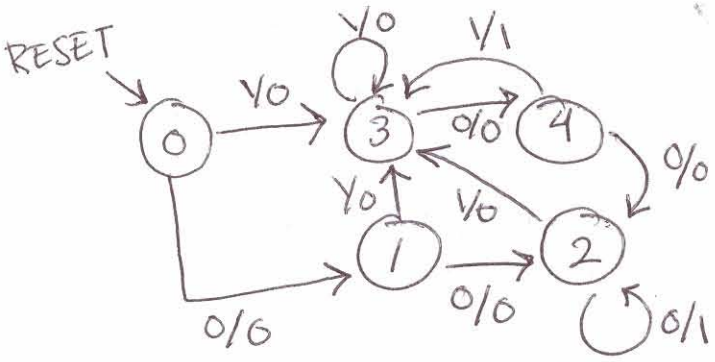
S1, S2, S6 → S1'



6.

States

0	no observations
1	one 0
2	two 0's
3	one 1
4	one 1, one 0



State encoding (almost hot)

0	000
1	001
2	010
3	011
4	100

State transition table

state	in	next state	out
000	0	001	0
000	1	011	0
001	0	010	0
001	1	011	0
010	0	010	1
010	1	011	0
011	0	100	0
011	1	011	0
100	0	010	0
100	1	011	1

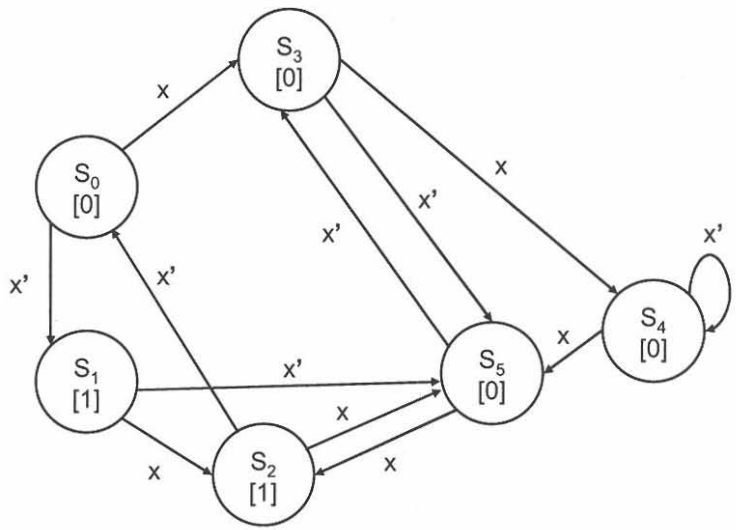
6. (15 points) A Mealy finite state machine has one input (X) and one output (Z). After each reset, an output Z=1 occurs every time an input sequence of 000 or 101 is observed and an output of Z=0 occurs whenever any other input sequence is observed. (Sequences may overlap.)

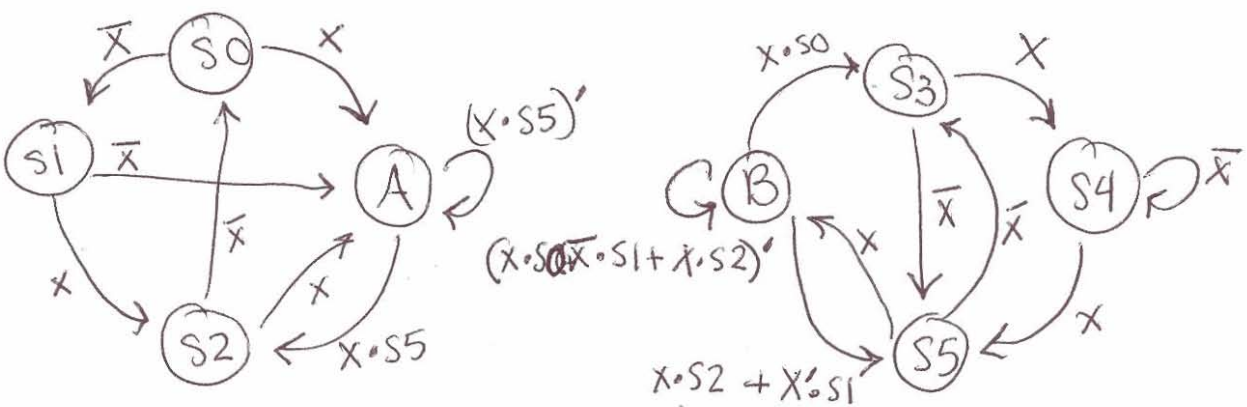
(a) (6 points) Draw a state diagram for this Mealy machine and give a name to each state (Hint: 5 states suffice)

(b) (3 points) Describe an encoding for each state of this Mealy machine using the "almost" one-hot encoding in which one state is encoded as all 0's. Name each bit of that encoding.

(c) (5 points) Write out the sum-of-products expressions for the state transitions and the output of your FSM design. Use a + to denote the updated version of each state bit.

7. (10 points) Show how to partition the following FSM into two communicating FSMs, one covering states S_0,S_1,S_2 and the other covering states S_3,S_4,S_5. Show the labels of all transitions in your new machines.

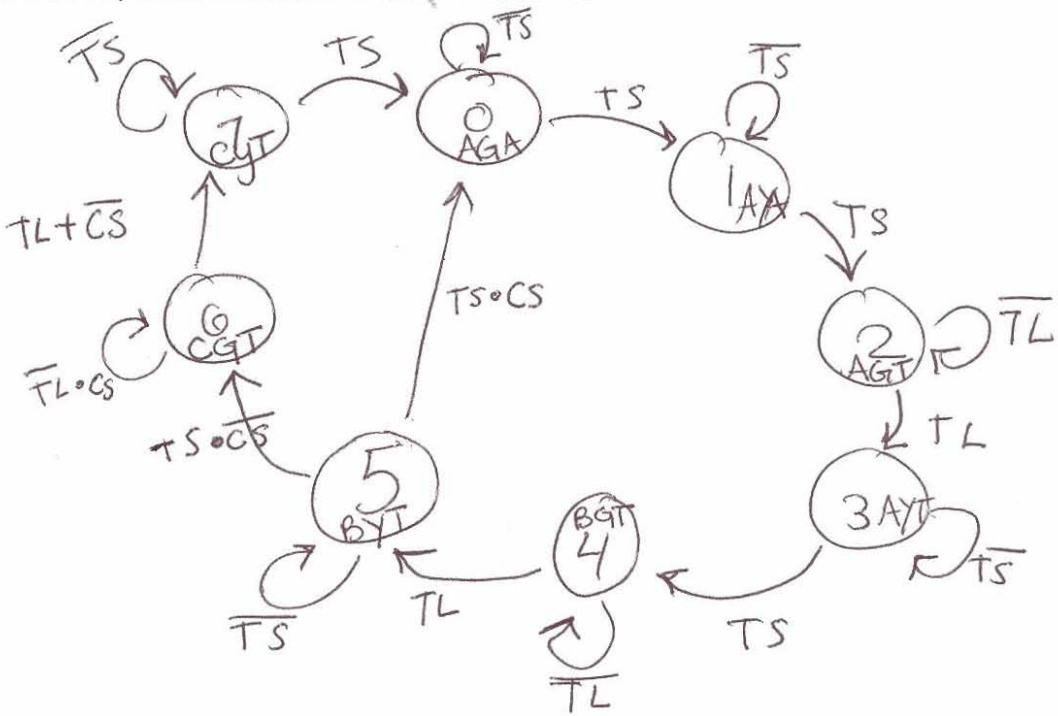




(a) (10 points) Draw a state diagram for the traffic light controller. Indicate the logical conditions for remaining in the current state and for moving to the next state. Please number your states 0 to n and label outputs with meaningful names such as "B Green", "B Yellow", "A South Green Arrow", "A North Red" etc.

(b) (6 points) Create a table that indicates precisely which lights are illuminated for each of your states. Use the following abbreviations: R = Red, Y = Yellow, G = Green, GA = Green left turn arrow, YA = Yellow left turn arrow, and indicate the direction and street.

States:
0 AGA
1 AYA
2 AGT
3 AYT
4 BGT
5 BYT
6 CGT
7 CYT



State	AN	AS	BE	BW	C
0	G, GA	R	R	R	R
1	G, YA	R	R	R	R
2	G	G	R	R	R
3	Y	Y	R	R	R
4	R	R	G	G	R
5	R	R	Y	Y	R
6	R	R	R	R	G
7	R	R	R	R	Y