EEM16 Quiz #2

TOTAL POINTS

20 / 20

QUESTION 1

1 (a), (b) **10 / 10**

✓ **- 0 pts** Correct

- 1.5 pts (a) extra state (1)

- 1.5 pts (a) incorrect arc

- 3 pts (a) missing state

- 2 pts (b) incorrect Moore outputs on table

- 0.5 pts (b) mislabeled table entry

- 1 pts missing/incorrect entry on table

QUESTION 2

2 (c), (d) **10 / 10**

✓ **- 0 pts** Correct

- 1 pts (c) missing feedback arc in equation

- 1.5 pts (c) several arcs missing or incorrect

- 2.5 pts (c) incorrect nx equation sum of product

- 2 pts (c) incorrect f equation

- 5 pts (c) incorrect

- 0.5 pts (d) too many states (4 - unnecessary initial state)

- 1.5 pts (d) too many states (4+)

- 5 pts (d) incorrect

Quiz #2

Name (Last, First):

Student Id #:

Do not start working until instructed to do so.

1. *You must answer in the* ***space provided*** *for answers after every question. We will ignore answers written anywhere else in the booklet.* ***All pages in this booklet must be accounted*** *for otherwise it will not be graded.*
2. *This quiz is closed book/notes.*
3. *You may not use any electronic device.*

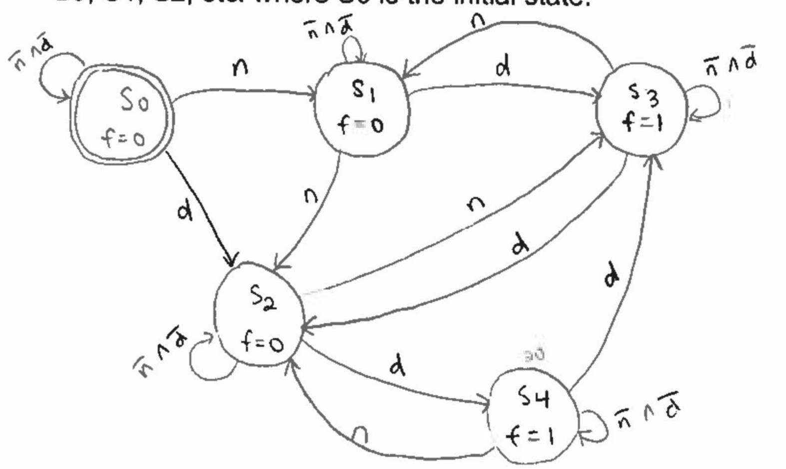
Following table to be filled by course staff only

	Maximum Score	Your Score
TOTAL	20	

This page is left intentionally blank and can be used for scratch work.

You are to design a state machine that accepts nickels (*n*) and dimes (*d*) as inputs (both single bit) and <u>*asserts an output when 15 cents have been deposited*</u> (single bit output, *f*). The machine initializes to a state where no money has been inserted with *f=1'b0.*

(a) Implement this function using a Moore state machine. You should design this so that the output, *f*, stays asserted until another coin is inserted. The machine operates continuously so if 3 dimes were deposited in succession, the output, *f*, should assert after the 2nd dime and on the 3rd dime (and stay asserted until another coin is inserted). Clearly show all arcs with their corresponding input condition (*n,d*), and the value of *f* within each state. Label your states, S0, S1, S2, etc. where S0 is the initial state.



(b) Fill in a state transition table for 2 of the states, S0, the initial state, and a state that produces an output *f*. If you have more than one state that produces an output, choose one (Sx where x can be whatever you chose in your numbering). You don't need to use all the lines in the table below.

sx = s3

state	n	d	f	nx_state
S0	0	0	0	S0
S0	0	1	0	S3
S0	1	0	0	S1
S0				
Sx	0	0	1	S3
Sx	0	1	1	S2
Sx	1	0	1	S1
Sx			1	

(c) Let us encode the states with one-hot encoding where *s0* correspond to the state bit that is asserted in state, S0, and *sx* corresponds to the state bit that is asserted in Sx. Write the Boolean function for *f*, and *nx_sx* (indicating *that the next_state will be Sx)* (corresponding to entering the state, Sx, from part (b) above. state [4:0]

nx_sx = nx_s3 = $(state[1] \wedge d) \vee (state[2] \wedge n) \vee (state[4] \wedge d) \vee (state[3] \wedge \neg n \wedge \neg d)$

f = $state[3] \vee state[4]$

(d) Repeat (a) but instead design a Mealy FSM. Note that unlike the Moore machine, the output, *f*, only needs to be asserted for 1 cycle when the correct input(s) is asserted.