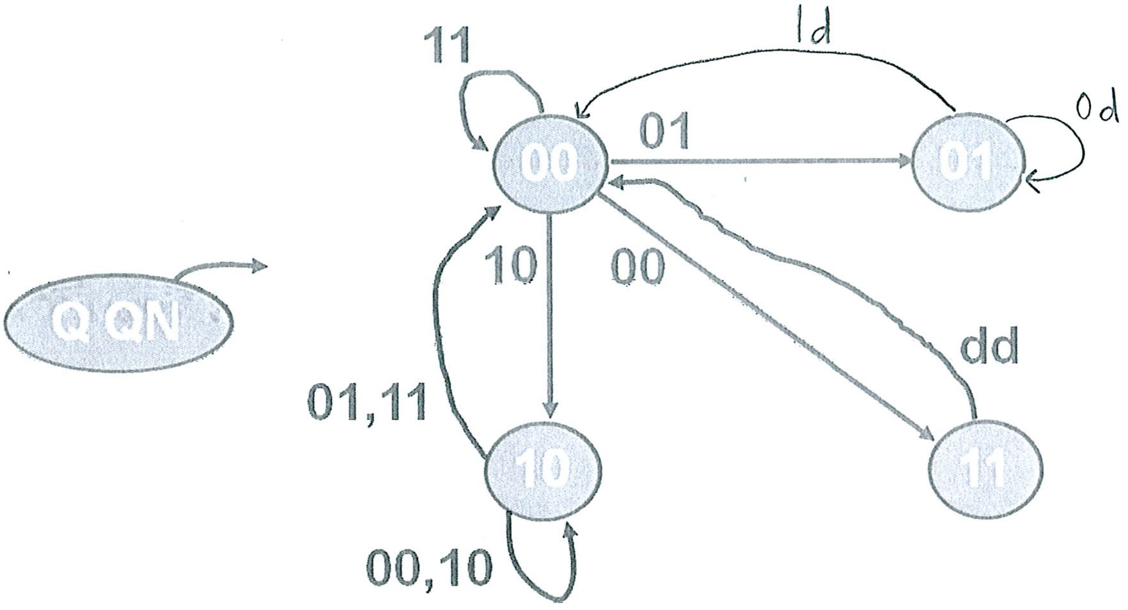
Huddle Board Exercise for Module 3 – No. 1
Monday, March 10, 2014

Use the following **PS-NS table** to construct a **state transition diagram**.

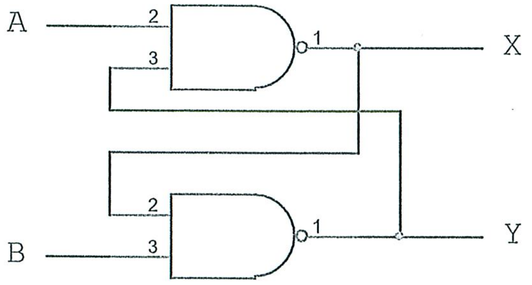
Present State Q(t) QN(t)	Present Inputs: S(t) R(t)			
	00	01	10	11
00	11	01	10	00
01	01	01	00	00
10	10	00	10	00
11	00	00	00	00

State Transition Diagram



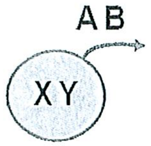
Huddle Board Exercise for Module 3 – No. 2
Wednesday, March 12, 2014

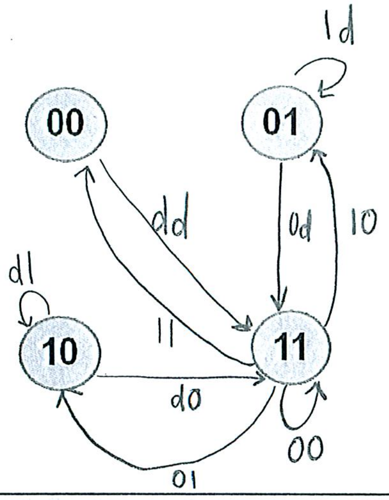
Use the following **circuit** below to determine the **next state equations**, complete the **state transition diagram**, and complete the **present state – next state table**.



$X(t+\tau) = A' + Y'$

$Y(t+\tau) = B' + X'$

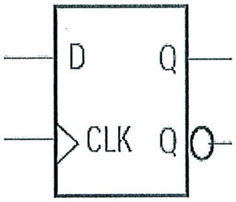


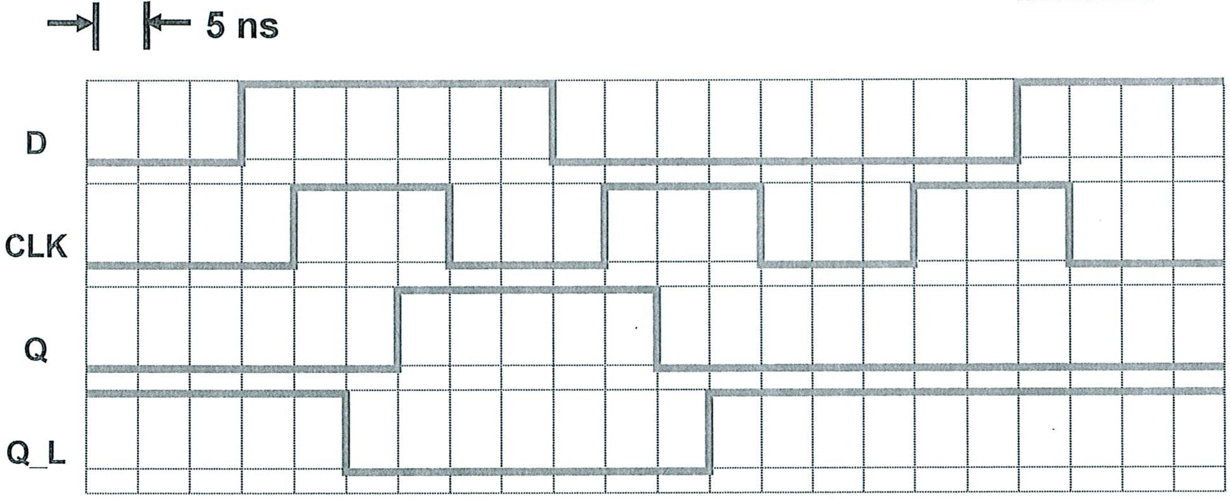


Present State	Present Input A(t) B(t)			
X(t) Y(t)	**0 0**	**0 1**	**1 0**	**1 1**
0 0	11	11	11	11
0 1	11	11	01	01
1 0	11	10	11	10
1 1	11	10	01	00

Huddle Board Exercise for Module 3 – No. 3
Monday, March 24, 2014

Complete the timing chart for the edge-triggered flip-flop, below, assuming its $t_{PLH(C \to Q)}$ is 10 ns and its $t_{PHL(C \to Q)}$ is 5 ns.





Determine the following:

(a) the **nominal setup time** provided for the D flip-flop, based on the excitation signals (D and CLK) depicted in the timing chart: 5 ns

(b) the **nominal hold time** provided for the D flip-flop, based on the excitation signals (D and CLK) depicted in the timing chart: 10 ns

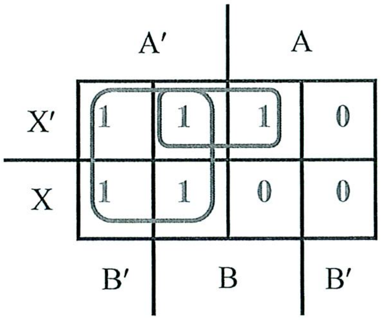
(c) the **nominal clock pulse width** provided for the D flip-flop, based on the excitation signals (D and CLK) depicted in the timing chart: 15 ns

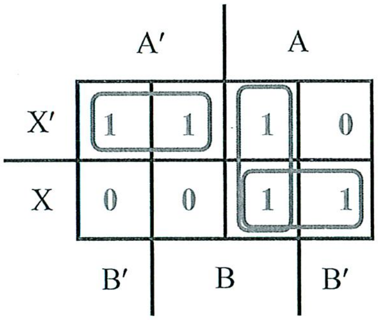
(d) the **duty cycle** of the clocking signal: 50%

Huddle Board Exercise for Module 3 – No. 4
Wednesday, March 26, 2014

Synthesize the state machine depicted by the following PS-NS table two different ways: using a D flip-flop and using a T flip-flop. Note that there are two inputs, A and B, along with a single state variable, X. Assume only *true* input variables are available, but you may use any types of gates deemed necessary. *Show all work.*

A	B	X	X*	D	T
0	0	0	1	1	1
0	0	1	1	1	0
0	1	0	1	1	1
0	1	1	1	1	0
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	1	1	1
1	1	1	0	0	1



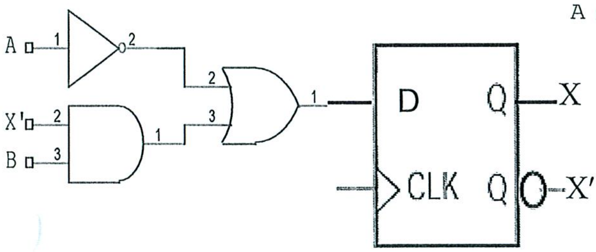


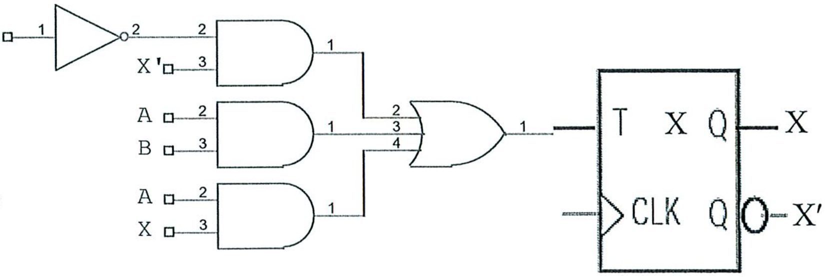
$D = A' + X' \cdot B$

$T = A' \cdot X' + A \cdot B + A \cdot X$

D flip-flop implementation:

T flip-flop implementation:

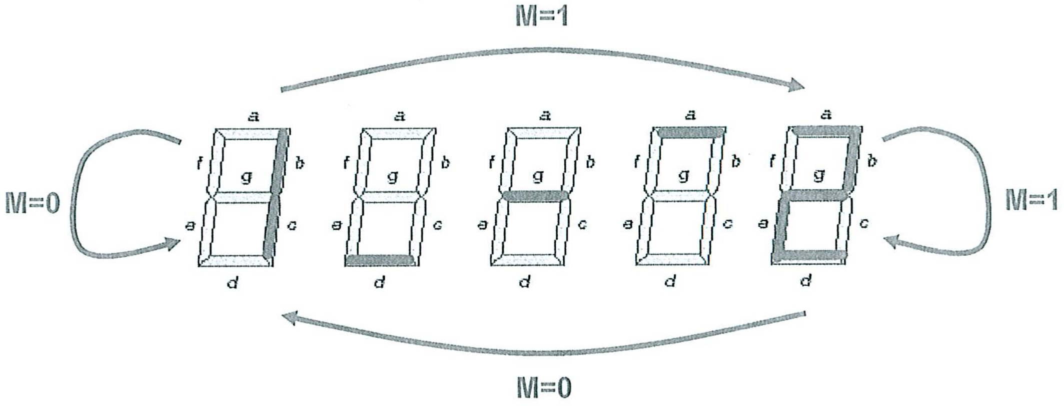




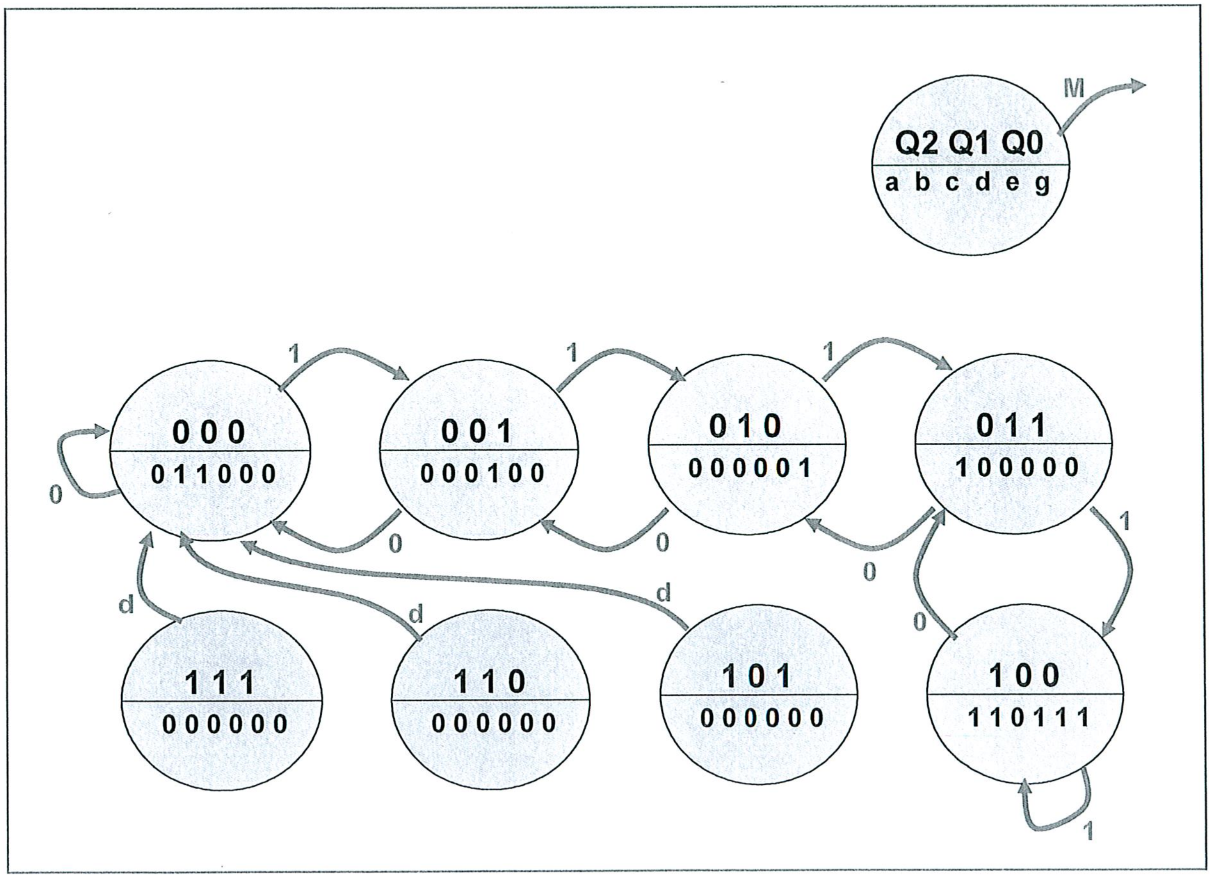
Huddle Board Exercise for Module 3 – No. 4a
Wednesday, March 26, 2014

Design a state machine that serves as a simple two-floor elevator controller. When the elevator is on floor one, the digit "1" should be output on a 7-segment display; when the elevator is on floor two, the digit "2" should be displayed. Input M should control the direction of the elevator: if M=0, the elevator should descend from floor two to floor one (and *stay* there once it reaches floor one); if M=1, the elevator should rise from floor one to floor two (and *stay* there once it reaches floor two). When the elevator is rising, the segments of the display should be sequenced in the order D → G → A (i.e., bottom → middle → top segment); when the elevator is descending, the segments of the display should be sequenced A → G → D (i.e., top → middle → bottom segment).

The following diagram illustrates the desired mode of operation:



Draw a Moore model state transition diagram:

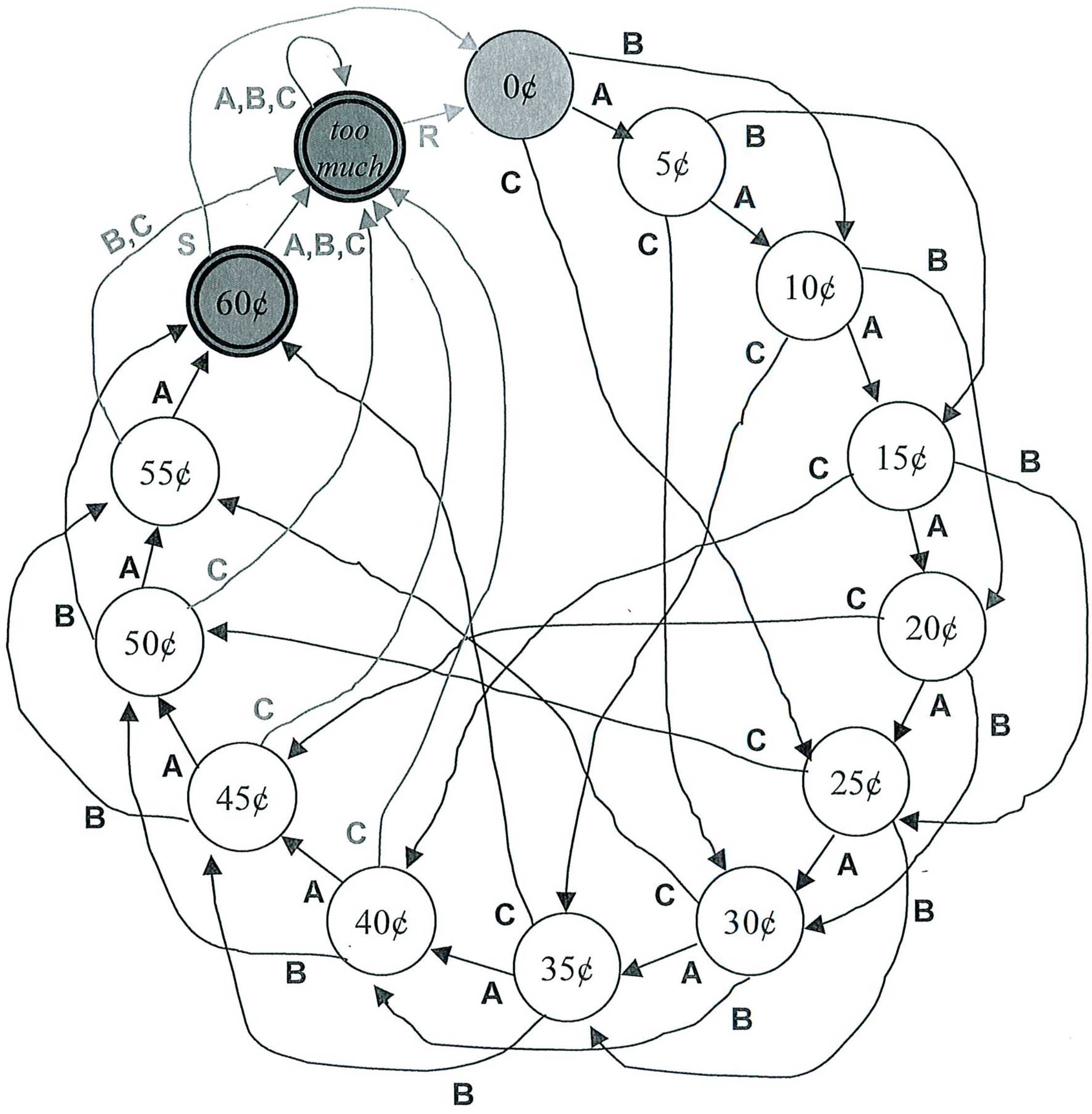


Huddle Board Exercise for Module 3 – No. 4b
Wednesday, March 26, 2014

Complete the state transition diagram, below, for a soda dispensing machine that accepts any combination of nickels, dimes, and quarters which totals $0.60 (exact change only).

Assume this state machine has the following input signals:

- **A – asserted when a nickel is inserted**
- **B – asserted when a dime is inserted**
- **C – asserted when a quarter is inserted**
- **S – asserted when a soda is selected, i.e., the money deposited is "consumed" by the machine, thus resetting the "running total" of the amount entered to zero (and hopefully releasing the desired soft drink)**
- **R – asserted when the "coin release" option is selected, i.e., the money deposited is released, allowing someone who has entered *too much* money to recover it, also resetting the "running total" of the amount entered to zero**

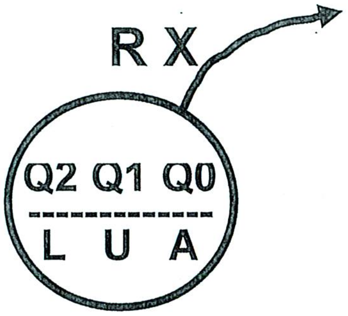


Huddle Board Exercise for Module 3 – No. 5
Wednesday, March 31, 2014

Design a digital combination lock that meets the following criteria:

- unlocks when a fixed combination (binary sequence) is entered: 101110
- has three inputs:
  - X – combination data
  - R – relock / reset
  - RESET – asynchronous reset
- has three output signals:
  - LOCKED
  - UNLOCKED
  - ALARM

Draw a *Moore Model* state transition diagram.



Solution on back.

Combination: 101110