

Ian Zhang  
Professor Alvin Lebeck  
Assignment 3: Written

1. (10 pts) Given four inputs (X3 X2 X1 X0) design and implement a logic function that is true if and only if the input when interpreted as an unsigned binary number input is less than 5.

a. Show the truth table and sum of products for your circuit.

A	B	C	D	Output
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

Sum of products:

$$(\neg A \ \neg B \ \neg C \ \neg D) \vee (\neg A \ \neg B \ \neg C \ D) \vee (\neg A \ \neg B \ C \ \neg D) \vee (\neg A \ \neg B \ C \ D) \vee (\neg A \ B \ \neg C \ \neg D)$$

Simplification:

$$(\neg A \ \neg B \ \neg C \ \neg D) \vee (\neg A \ \neg B \ \neg C \ D) \vee (\neg A \ \neg B \ C \ \neg D) \vee (\neg A \ \neg B \ C \ D) = (\neg A \ \neg B)$$
$$(\neg A \ \neg B) \vee (\neg A \ B \ \neg C \ \neg D)$$

b. Implement this circuit using only AND, OR, and NOT gates in Logisim (make sure to test your circuit and check its validity).

Okay.

2. (15 pts) Seven Segment Display Driver A Liquid Crystal Display (LCD) clock works by turning on specific segments to display a number for each digit. Each digit is composed of 7 segments, and by turning on the correct segments any number 0-9 can be displayed. For this problem you will use the 7-Segment display provided in Logisim (see here for documentation, note there is one input for a decimal point, but we'll ignore that). Your job is to design the control circuit so that this 7-segment display can display any 3-bit binary unsigned number (0-7). Your circuit will take as input the 3-bit unsigned binary number and generate the 7 outputs that connect to the 7-segment display.

a. Show the truth table and sum of products representation for your circuit.

Input 1	Input 2	Input 3	Output 1	Output 2	Output 3	Output 4	Output 5	Output 6	Output 7
A	B	C	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1
0	1	0	1	0	1	1	1	1	0
0	1	1	1	0	1	1	0	1	1
1	0	0	1	1	0	1	0	0	1
1	0	1	1	1	1	0	0	1	1
1	1	0	1	1	0	0	1	1	1
1	1	1	0	0	1	1	0	0	1

Sum of products:

Output 1/a:

$$(\neg A \& B \& C) \mid (\neg A \& B \& \neg C) \mid (A \& \neg B \& C) \mid (A \& \neg B \& \neg C) \mid (A \& B \& C)$$

Simplification:

$$(A \& \neg B \& C) \mid (A \& \neg B \& \neg C) = (A \& \neg B)$$

$$(\neg A \& B \& C) \mid (\neg A \& B \& \neg C) = (\neg A \& B)$$

$$(A \& B \& C) \mid (A \& \neg B) \mid (\neg A \& B)$$

Output 2/b:

$$(\neg A \& \neg B \& C) \mid (A \& \neg B \& C) \mid (A \& \neg B \& \neg C) \mid (A \& B \& C)$$

Simplification:

$$(\neg A \& \neg B \& C) \mid (A \& \neg B \& C) = (\neg B \& C)$$

$$(\neg B \& C) \mid (A \& \neg B \& \neg C) \mid (A \& B \& C)$$

Output 3/c:

$$(\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge C) \vee (A \wedge \neg B \wedge C) \vee (A \wedge B \wedge C)$$

Simplification:

$$(\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge C) = (\neg A \wedge B)$$

$$(A \wedge B \wedge C) \vee (A \wedge \neg B \wedge C) = (A \wedge C)$$

$$\mathbf{(\neg A \wedge \neg B \wedge \neg C) \vee (A \wedge C) \vee (\neg A \wedge B)}$$

Output 4/d:

$$(\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge C) \vee (A \wedge \neg B \wedge \neg C) \vee (A \wedge B \wedge C)$$

Simplification:

$$(\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C) = (\neg A \wedge \neg B)$$

$$(\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge C) = (\neg A \wedge B)$$

$$(\neg A \wedge \neg B) \vee (\neg A \wedge B) = \neg A$$

$$\mathbf{\neg A \vee (A \wedge \neg B \wedge \neg C) \vee (A \wedge B \wedge C)}$$

Output 5/e:

$$(\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (A \wedge B \wedge \neg C)$$

Simplification:

$$(\neg A \wedge B \wedge \neg C) \vee (A \wedge B \wedge \neg C) = (B \wedge \neg C)$$

$$\mathbf{(\neg A \wedge \neg B \wedge \neg C) \vee (B \wedge \neg C)}$$

Output 6/f:

$$(\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge C) \vee (A \wedge \neg B \wedge C) \vee (A \wedge B \wedge \neg C)$$

Simplification:

$$(\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge C) = (\neg A \wedge B)$$

$$\mathbf{(A \wedge \neg B \wedge C) \vee (A \wedge B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B)}$$

Output 7/g:

$$(!A \& !B \& !C) \mid (!A \& !B \& C) \mid (!A \& B \& C) \mid (A \& !B \& !C) \mid (A \& !B \& C) \mid (A \& B \& !C) \mid (A \& B \& C)$$

Simplification:

$$(!A \& !B \& !C) \mid (!A \& !B \& C) = (!A \& !B)$$

$$(A \& B \& !C) \mid (A \& B \& C) = (A \& B)$$

$$(A \& !B \& !C) \mid (A \& !B \& C) = (A \& !B)$$

$$(A \& !B) \mid (!A \& B \& C) \mid (A \& B) \mid (!A \& !B)$$

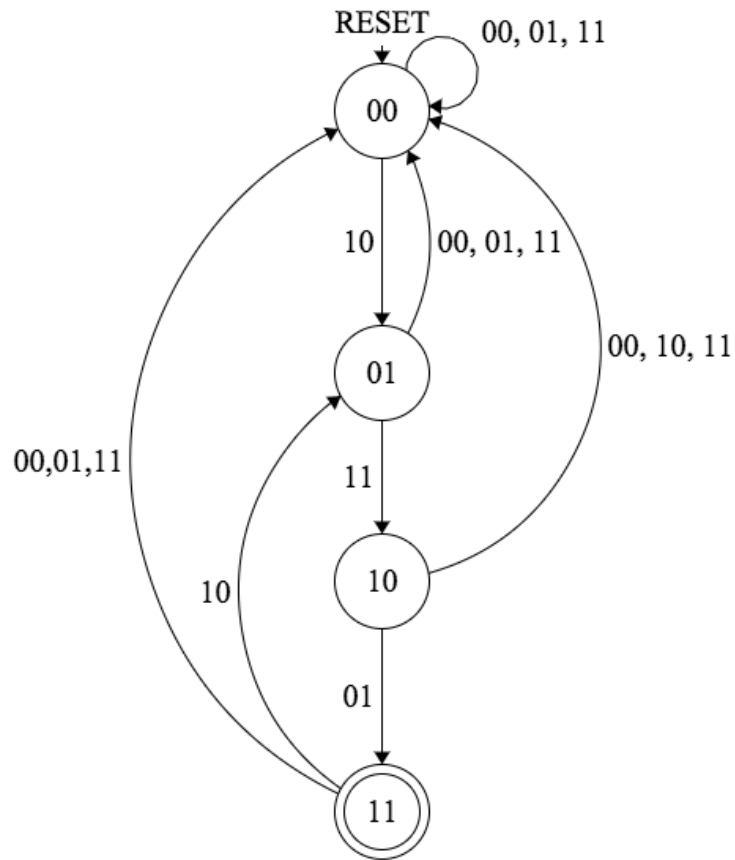
b. Use Logisim to implement your circuit using only AND, OR, and NOT gates.

Okay.

#### 4. (25 pts) Introduction to Finite State Machines

Design and implement a finite state machine that recognizes the input combination sequence 231. Your input is a sequence of two bit unsigned binary numbers. The output is an unlock signal, that remains valid for one clock only if the input sequence is detected. If the input is incorrect, the lock should reset, requiring the user to re-enter the full three digit combination sequence.

a. Show the truth table, boolean functions for the next state and output variables, and the state diagram for your finite state machine.



Current State		Inputs		Next State		Output
A	B	I <sub>1</sub>	I <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	F
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	1	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	0
1	0	0	1	1	1	0
1	0	1	0	0	0	0
1	0	1	1	0	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	1	1
1	1	1	1	0	0	1

Boolean Functions:

$$D0 = A' B' I1 I2' \mid A B' I1' I2 \mid A B I1 I2$$

$$D1 = A' B I1 I2 \mid A B' I1' I2$$

$$F = A B$$

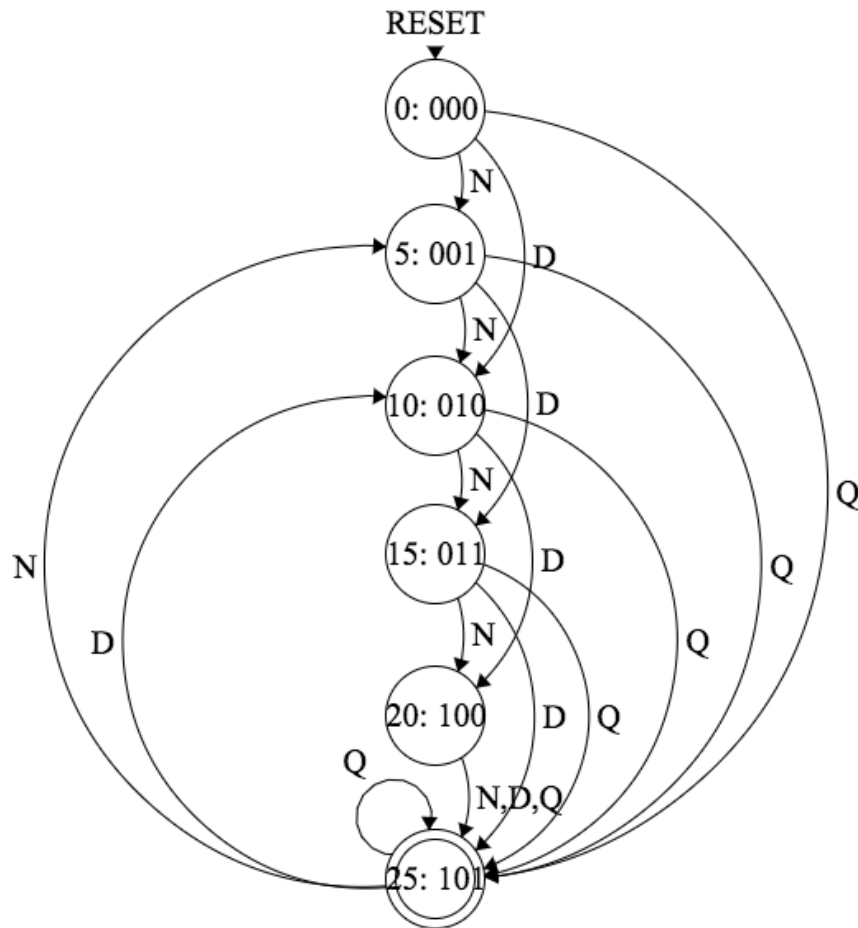
b. Implement your FSM in Logisim using D Flip Flops, the clock for your circuit should be a button that will be explicitly pressed after each new input value is entered.

Okay.

#### 5. (25 pts) FSM: Vending Machine

A vending machine has the ability to accept several different combinations of coins from a customer wishing to purchase an item. Everything in our machine costs 25 cents and the machine only accepts quarters (Q), dimes (D) and nickels (N). Design a finite state machine to control the dispense signal (F). More specifically (F) should be true when the amount of money deposited is equal or greater than 25 cents. The input to your circuit are the three signals, one for each coin, Q, D, N, that have the value one if a coin of that type was deposited, the output is the dispense signal (F). Our machine is greedy and doesn't return any change. After dispensing the machine resets and waits for another customer.

a. Show the truth table, boolean functions for the next state and output variables, and the state diagram for your finite state machine.



Current State			Inputs (Coins)		Next State			Output	
A	B	C	Q D N	\$Total	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	F	
0	0	0	0 0 0	0¢	x	x	x	x	
0	0	0	0 0 1	5¢	0	0	1	0	
0	0	0	0 1 0	10¢	0	1	0	0	
0	0	0	0 1 1	x	x	x	x	x	
0	0	0	1 0 0	25¢	1	0	1	0	
0	0	0	1 0 1	x	x	x	x	x	
0	0	0	1 1 0	x	x	x	x	x	
0	0	0	1 1 1	x	x	x	x	x	
0	0	1	0 0 0	5¢	x	x	x	x	
0	0	1	0 0 1	10¢	0	1	0	0	
0	0	1	0 1 0	15¢	0	1	1	0	
0	0	1	0 1 1	x	x	x	x	x	

0	0	1	1 0 0	25¢	1	0	1	0
0	0	1	1 0 1	x	x	x	x	x
0	0	1	1 1 0	x	x	x	x	x
0	0	1	1 1 1	x	x	x	x	x
0	1	0	0 0 0	10¢	x	x	x	x
0	1	0	0 0 1	15¢	0	1	1	0
0	1	0	0 1 0	20¢	1	0	0	0
0	1	0	0 1 1	x	x	x	x	x
0	1	0	1 0 0	25¢	1	0	1	0
0	1	0	1 0 1	x	x	x	x	x
0	1	0	1 1 0	x	x	x	x	x
0	1	0	1 1 1	x	x	x	x	x
0	1	1	0 0 0	15¢	x	x	x	x
0	1	1	0 0 1	20¢	1	0	0	0
0	1	1	0 1 0	25¢	1	0	1	0
0	1	1	0 1 1	x	x	x	x	x
0	1	1	1 0 0	25¢	1	0	1	0
0	1	1	1 0 1	x	x	x	x	x
0	1	1	1 1 0	x	x	x	x	x
0	1	1	1 1 1	x	x	x	x	x
1	0	0	0 0 0	20¢	x	x	x	x
1	0	0	0 0 1	25¢	1	0	1	0
1	0	0	0 1 0	25¢	1	0	1	0
1	0	0	0 1 1	x	x	x	x	x
1	0	0	1 0 0	25¢	1	0	1	0
1	0	0	1 0 1	x	x	x	x	x
1	0	0	1 1 0	x	x	x	x	x
1	0	0	1 1 1	x	x	x	x	x
1	0	1	0 0 0	25¢	x	x	x	x
1	0	1	0 0 1	5¢	0	0	1	1
1	0	1	0 1 0	10¢	0	1	0	1
1	0	1	1 0 0	25¢	1	0	1	1

Boolean Functions:

D0 =

$A'B'CQD'N' | A'B'C'QD'N' | A'B'C'Q'D'N | A'B'CQ'D'N | A'BCQ'DN' | A'B'CQ'DN' |$   
 $A'BC'QD'N' | A'BCQD'N' | A'BC'Q'D'N | A'B'C'Q'D'N | A'B'C'QD'N' |$   
 $A'B'CQD'N' | A'B'C'Q'DN'$

Simplification:



$$\begin{aligned}
&A' B' C' Q D' N' | A' B' C Q D' N' = A' B' Q D' N' \\
&A' B' C' Q' D' N | A' B' C Q' D' N = A' B' Q' D' N \\
&A' B C' Q D' N' | A' B C Q D' N' = A' B Q D' N' \\
&A' B' C' Q D' N' | A' B' C Q D' N' = A' B' Q D' N' \\
&A' B C' Q' D' N | A' B' C' Q' D' N = A' C' Q' D' N \\
&A' B C Q' D N' | A' B' C Q' D N' = A' C Q' D N'
\end{aligned}$$

$$A' B Q D' N' | A' B' Q D' N' = A' Q D' N'$$

$$A' Q D' N' | A B' Q D' N' | A B' Q' D' N | A' C' Q' D' N | A' C Q' D N' | A B' C' Q' D N'$$

D1 =

$$A' B' C Q' D N' | A' B' C' Q' D N' | A' B' C Q' D' N | A' B C' Q' D' N | A B' C Q' D N'$$

Simplification:

$$A' B' C Q' D N' | A' B' C' Q' D N' = A' B' Q' D N'$$

$$A' B' Q' D N' | A' B' C Q' D' N | A' B C' Q' D' N | A B' C Q' D N'$$

D2 =

$$\begin{aligned}
&A' B' C' Q D' N' | A' B' C Q D' N' | A' B C Q' D N' | A' B C' Q' D N' | A' B C Q D' N' | A' B C' Q D' N' | \\
&A B' C' Q D' N' | A B' C Q D' N' | A' B C Q' D' N | A B' C' Q' D' N | A B' C' Q' D N' |
\end{aligned}$$

Simplification:

$$A' B' C' Q D' N' | A' B' C Q D' N' = A' B' Q D' N'$$

$$A' B C Q' D N' | A' B C' Q' D N' = A' B Q' D N'$$

$$A' B C Q D' N' | A' B C' Q D' N' = A' B Q D' N'$$

$$A B' C' Q D' N' | A B' C Q D' N' = A B' Q D' N'$$

$$A B' Q D' N' | A' B' Q D' N' = B' Q D' N'$$

$$A' B C Q' D' N | A B' C' Q' D' N | A B' C' Q' D N' | B' Q D' N' | A' B Q D' N' | A' B Q' D N'$$

$$A' Q D' N' | A' B Q' D N' | A B' Q D' N' | A' B C Q' D' N | A B' C' Q' D' N | A B' C' Q' D N'$$

$$F = A B' C$$

b. Implement your FSM in Logisim using D Flip Flops, the clock for your circuit should be a button that will be explicitly pressed after each coin is deposited.

Okay.