Hands-on

Lectures 1-13

Masoumeh (Azin) Ebrahimi

Lecture 1: Hands-on

Binary to Decimal:

$(1011010)_2 = (\quad\quad)_{10}$

Octal to Decimal:

$(567)_8 = (\quad\quad)_{10}$

Binary to Octal:

$(1011010)_2 = (\quad\quad)_8$

Hexadecimal to Decimal:

$(1AE)_{16} = (\quad\quad)_{10}$

Binary to Hexadecimal:

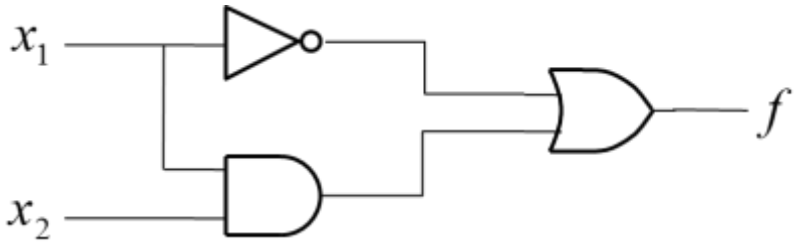
$(1011010)_2 = (\quad\quad)_{16}$

Decimal to Binary

$(35)_{10} = (\quad\quad)_2$

Lecture 2: Hands-on

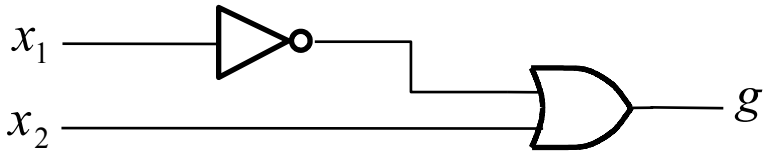
Fill the truth table based on the given circuit:



f =

x_1	x_2	$f(x_1,x_2)$
0	0	
0	1	
1	0	
1	1	

Fill the truth table based on the given circuit:



f =

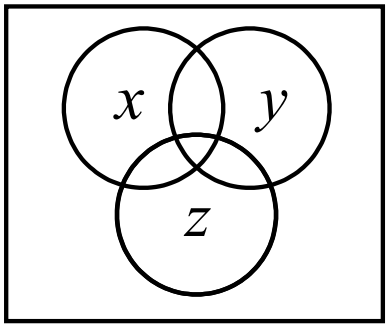
x_1	x_2	$f(x_1,x_2)$
0	0	
0	1	
1	0	
1	1	

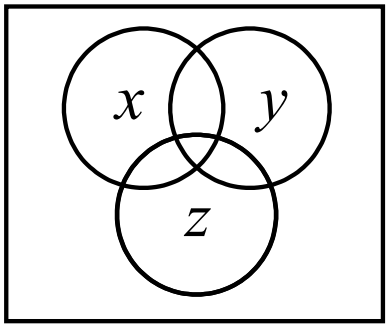
Draw the logic circuit for the following truth table:

x_1	x_2	$f(x_1,x_2)$
0	0	1
0	1	1
1	0	0
1	1	1

Show that consensus property holds using Venn diagram:

$$x \cdot y + y \cdot z + \overline{x} \cdot z = x \cdot y + \overline{x} \cdot z$$





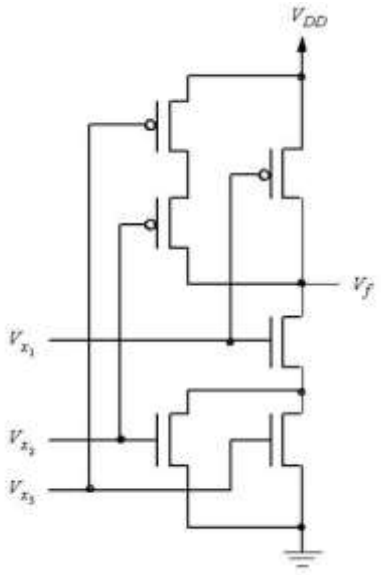
Write the function using minterms and maxterms.

x_1	x_2	x_3	f
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

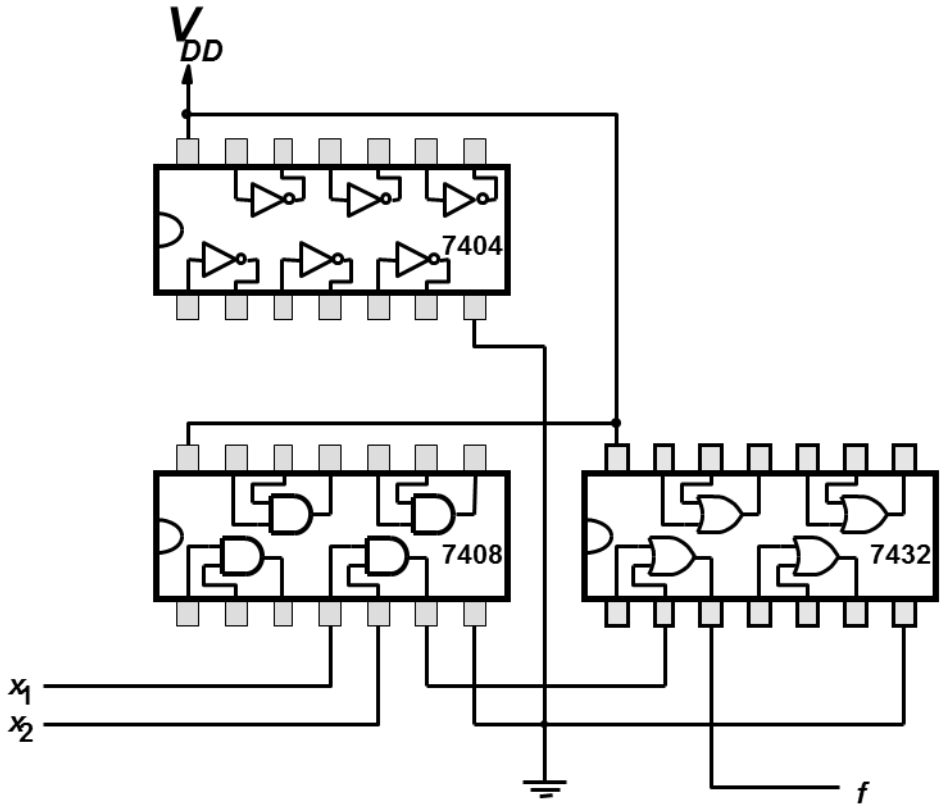
Draw the logic circuit.

Lecture 3: Hands-on

What kind of function the following CMOS circuit implements?

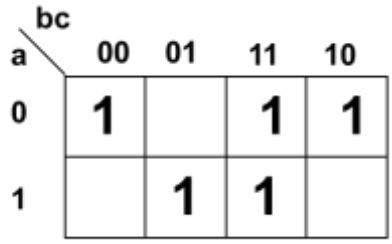


Implement the function f = x1x2 + $\overline{x}$2x3



Lecture 4: Hands-on

Find the minimum number of groups that cover all “1”. Groups should be in their biggest size.



a \ bc	00	01	11	10
0	1		1	1
1		1	1	

Find the minimum cost implementation for the following K-map:

cd \ ab	00	01	11	10
00	0	0	0	0
01	0	1	1	0
11	1	1	1	1
10	0	0	1	0

Write the logic function of f0 and f1:

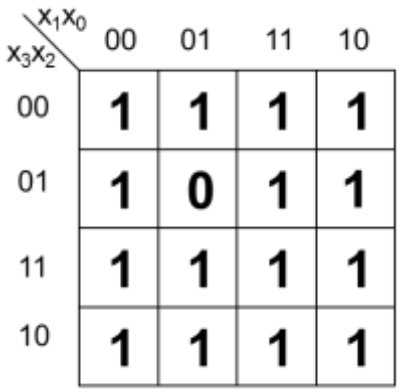
x_3x_2 \ x_1x_0	00	01	11	10
00	1	0	1	1
01	0	0	0	1
11	0	0	0	1
10	1	0	1	1

f_0

x_3x_2 \ x_1x_0	00	01	11	10
00	1	0	1	1
01	0	0	0	1
11	1	0	0	1
10	1	0	1	1

f_1

Find the minimized function:



x_3x_2 \ x_1x_0	00	01	11	10
00	1	1	1	1
01	1	0	1	1
11	1	1	1	1
10	1	1	1	1

Lecture 5: Hands-on

Represent -10 in 2's complement scheme when the number of bits is 5.

Represent -10 in 2's complement scheme when the number of bits is 8.

Find the 2's complement of 110010 in two ways:

1) Complement each bit, then add 1:

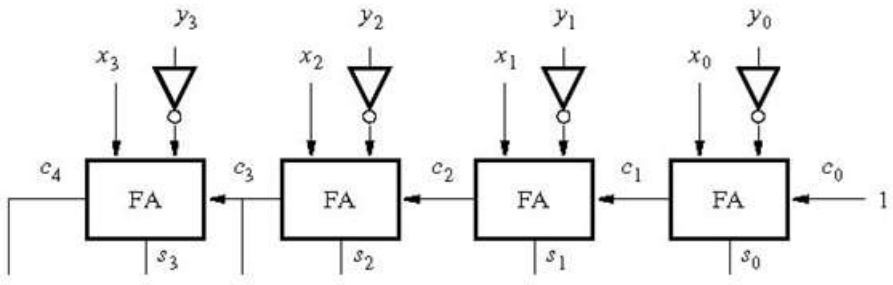
2) Start from the right-hand side, then copy all the bits that are 0 and the first bit that is 1. Finally complement all other bits

Perform subtractions in the 2's complement scheme:

```
   (+5)
 - (-2)
 ------
   (+7)
```

```
   (-5)
 - (-2)
 ------
   (-3)
```

How to recognize overflow, negative numbers and zero values in the following circuit? Make the connections.



V (overflow) N (negative) Z (zero)

Lecture 6: Hands-on

Multiply -5 by +2 in binary by first converting -5 to the positive number and then keeping track of the result's sign.

Multiply 13 * 8 in binary.

Divide 1011 by 10 (binary).

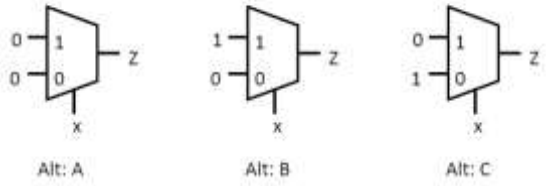
Multiply 010100 by 4:

Divide 010100 by 4:

Lecture 7: Hands-on

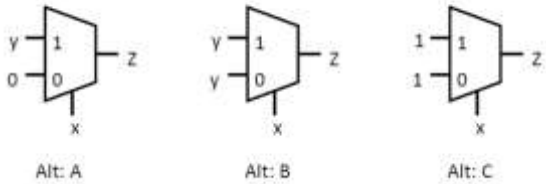
How to connect the inputs of the MUX in order to implement an inverter?

Desired function: $z = \overline{x}$



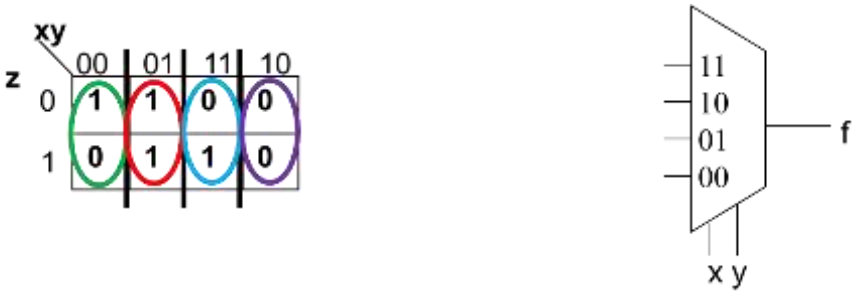
How to connect the inputs of the MUX in order to implement an AND gate?

Desired function: $z = xy$



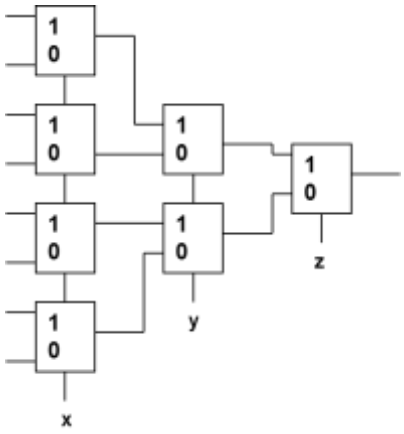
Connect the inputs of the MUX in order to implement OR and XOR gates.

Connect the inputs of the MUX in order to implement the following function: $f = \overline{z}\,\overline{x} + \overline{x}y + zy$

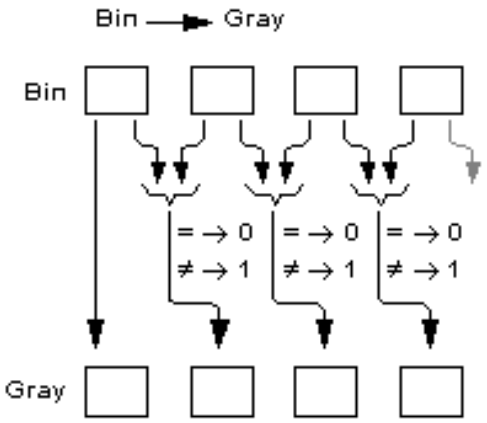


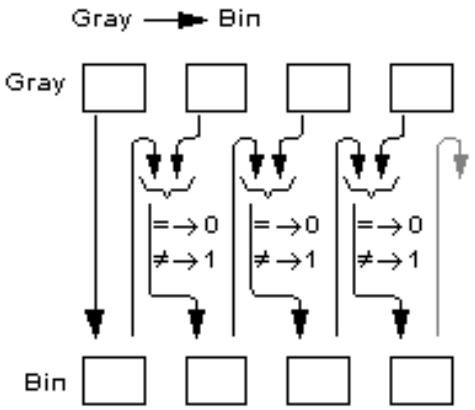
Connect the inputs of the MUX in order to implement the following function in the table:

x \ yz	00	01	11	10
0	**0**	**1**	**0**	**1**
1	**1**	**0**	**1**	**0**



Convert 0101 from binary to gray code. Also convert 0010 from gray code to binary code.





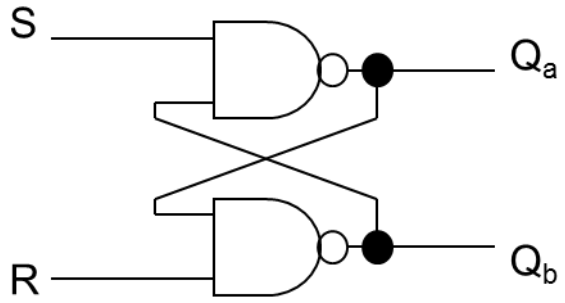
Lecture 8: Hands-on

Fill the characteristic table of the following SR-latch when

S=0 and R=1

S=1 and R=0

If you have enough time then try the case when S=1 and R=1.

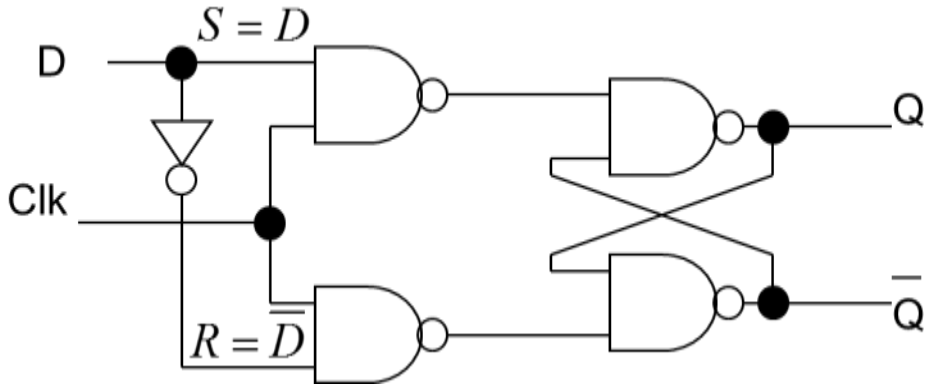


S	R	Q_a	Q_b
0	0	1	1
0	1		
1	0		
1	1		

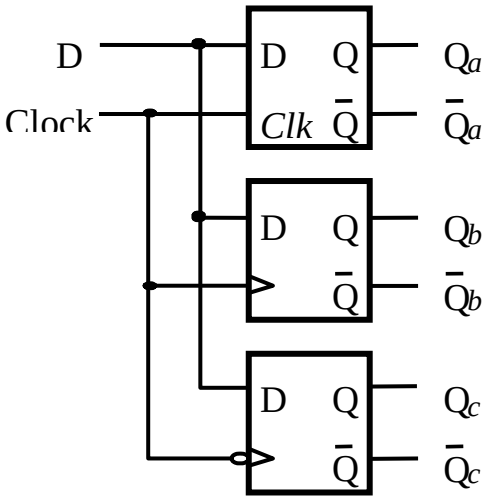
Prohibited input combination

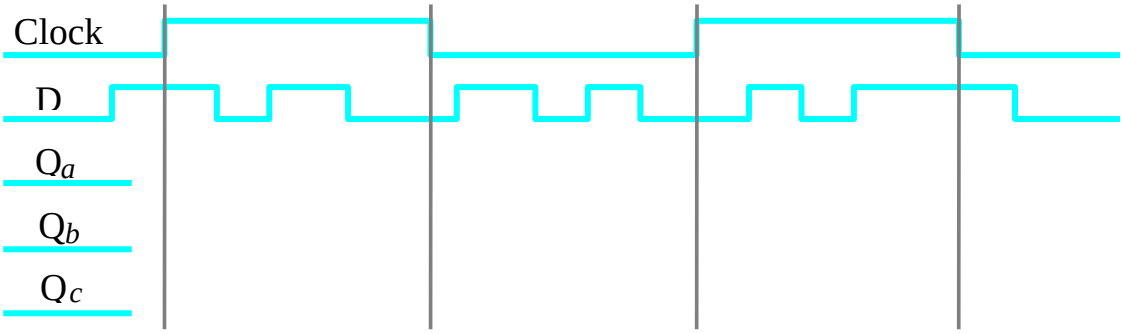
What is the value of Q(t+1) when clk=1 and D=1?

What is the value of Q(t+1) when clk=0?

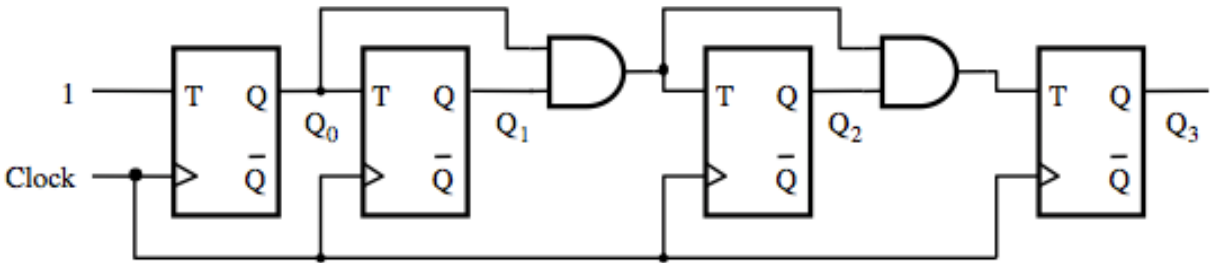


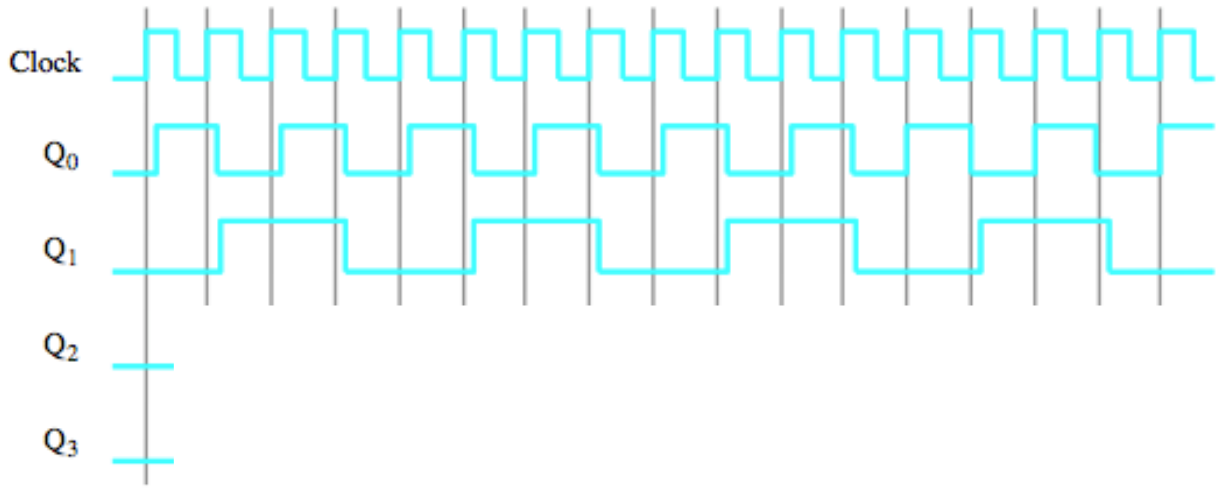
Complete the timing diagram for the following Latch, Positive edge triggered flipflop, and Negative edge triggered flipflop





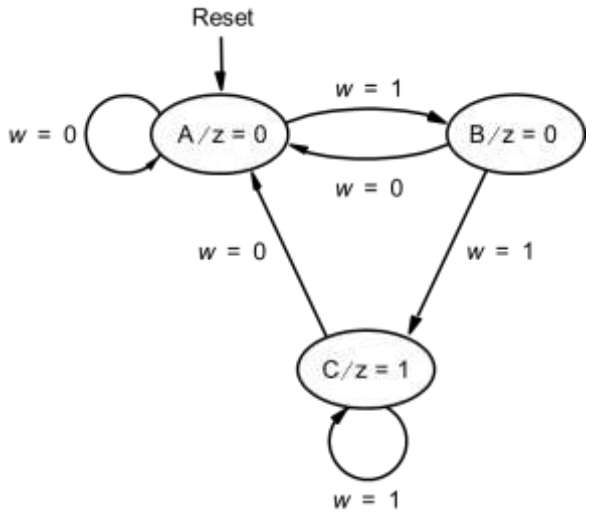
Complete the timing diagram:





Lecture 9: Hands-on

Assuming the following state diagram to detect two or more consecutive ones, we want to synthesis the circuit. Let's assume that A, B and C are coded as: A=00; B=01, and C=11.

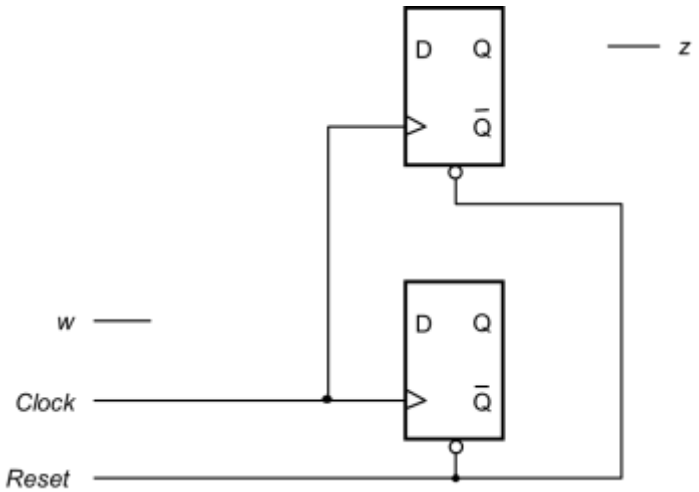


Given the above state diagram, fill the state table and state-assigned table:

Present state	Next state		Output
	$w = 0$	$w = 1$	z
A			0
B			0
C			1

	Present state	Next state		Output
		$w = 0$	$w = 1$	z
	y_2y_1	Y_2Y_1	Y_2Y_1	
A	00			
B	01			
C	11			
	10			

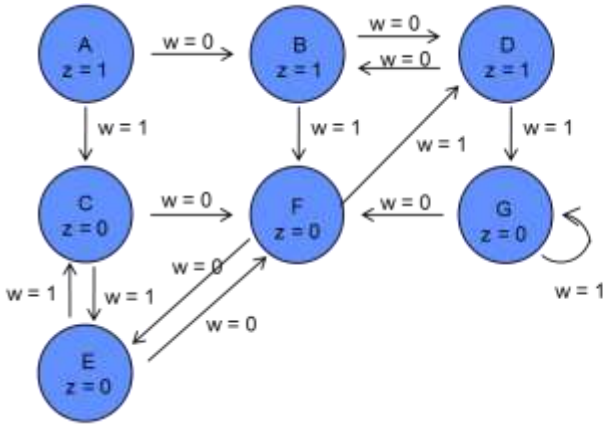
Extract the function of Y2, Y1 and z and based on these functions, synthesis the circuit.



Lecture 10: Hands-on

Synthesizing the circuit and state minimization.

From the state diagram fill the state table.



Present state	Next state		Output z
	w = 0	w = 1	
A	B	C	1

Partition the states with different outputs and find the 0- and 1- successor of each state.

P1= (ABCDEFG)

P2=( )( )

First block

0-successor:

1-successor:

Second block

0-successor:

1-successor:

P3 = (ABD) (CEG) (F)

P4 = (AD) (B) (CEG) (F)

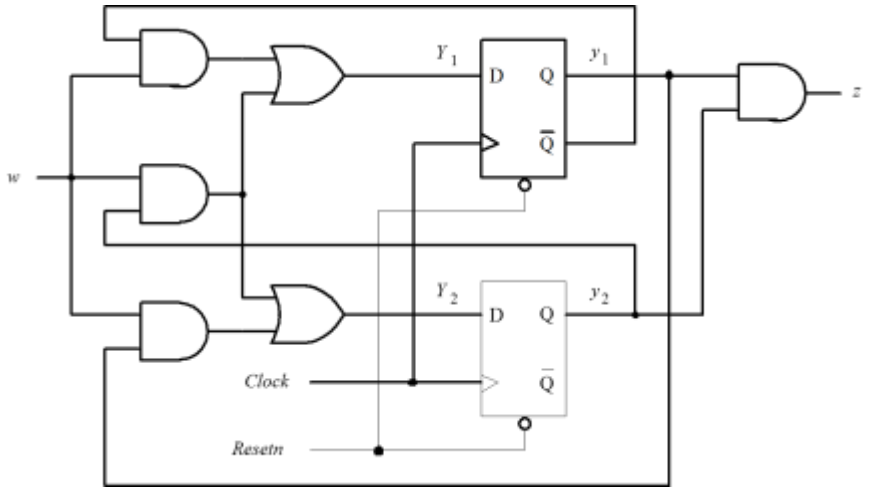
P5 = P4 = (AD) (B) (CEG) (F)

Fill the final state table and draw the new state diagram.

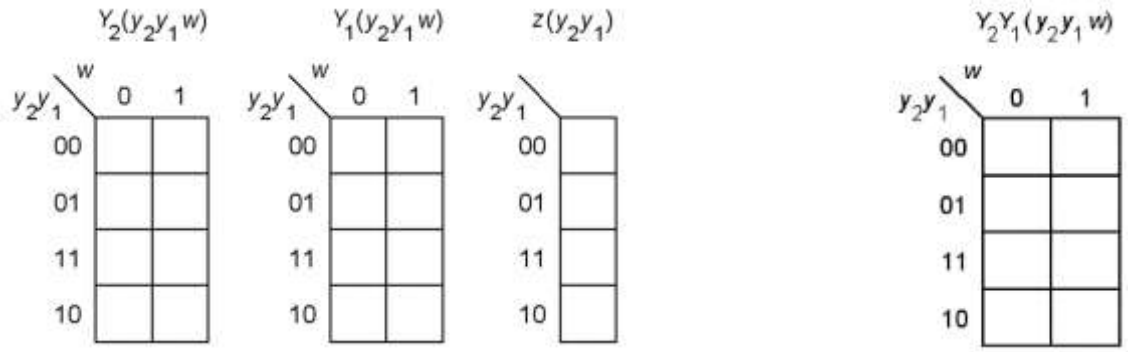
Present state	Next State		Output z
	w = 0	w = 1	
A	B	C	1

Analysis of synchronous sequential circuits:

Get expressions for next state decoder (Y1 and Y2) and output decoder (z).



Fill in the Karnaugh maps and merge the Karnaugh maps into a coded state table:



Fill the state-assigned and state tables:

State-assigned table

Present state y_2y_1	Next State w = 0 Y_2Y_1	Next State w = 1 Y_2Y_1	Output z
0 0			
0 1			
1 0			
1 1			

A:"00" B:"01" C:"10" D:"11"

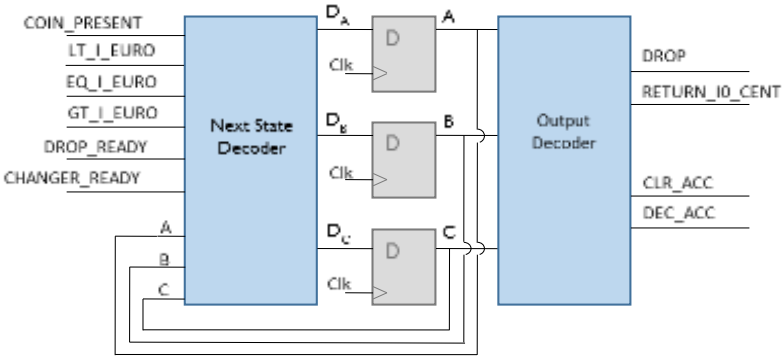
State table

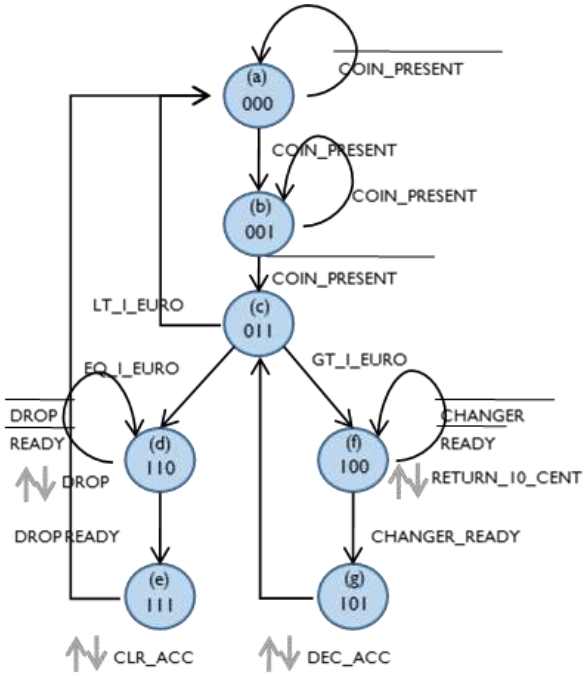
Present state	Next state w = 0	Next state w = 1	Output z

Draw the state diagram and realize the function.

Lecture 11: Hands-on

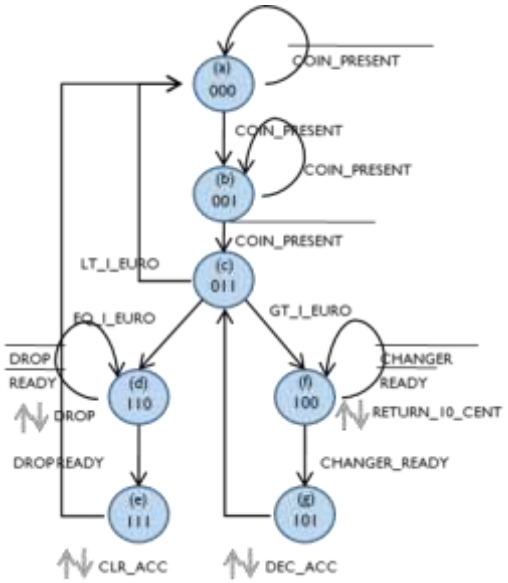
D_B		AB			
		00	01	11	10
C	0				
	1				






```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY Vending_Machine IS
   PORT (
      -- Inputs
      coin_present  : IN std_logic;









      -- Outputs
      dec_acc        : OUT std_logic;



       );
END Vending_Machine;

-------------------------------------------------

ARCHITECTURE Moore_FSM OF Vending_Machine IS

   TYPE   state_type IS (a, b, c, d, e, f, g);
   SIGNAL current_state, next_state: state_type;

BEGIN  -- Moore_FSM
```

```
NEXTSTATEDECODER: PROCESS (current_state, coin_present, gt_1_euro, eq_1_euro,
lt_1_euro, drop_ready, changer_ready)

   BEGIN  -- PROCESS NEXT_STATE
      CASE current_state IS
         WHEN a => IF coin_present = '1' THEN
                      next_state <= b;
                   ELSE
                      next_state <= a;
                   END IF;
         WHEN b => IF coin_present = '0' THEN
                      next_state <= c;
                   ELSE
                      next_state <= b;
                   END IF;
```




```
         WHEN OTHERS => next_state <= a;
      END CASE;
   END PROCESS NEXTSTATE;

-------------------------------------------------

OUTPUTDECODER: PROCESS (current_state)
   BEGIN  -- PROCESS OUTPUT
      drop           <= '0';



      CASE current_state IS
         WHEN d      => drop <= '1';



         WHEN OTHERS => NULL;
      END CASE;
   END PROCESS OUTPUT;

-------------------------------------------------

CLOCK: PROCESS (clk, reset_n)
   BEGIN  -- PROCESS CLOCK
      IF reset_n = '0' THEN               -- asynchronous reset (active low)
         current_state <= a;
      ELSIF clk'event AND clk = '1' THEN  -- rising clock edge
         current_state <= next_state;
      END IF;
   END PROCESS CLOCK;


END Moore_FSM;
```

Lecture 12: Hands-on

State reduction

Equivalence classes

- Outputs must have the same value
- Stable states must be at the same positions
- Don't cares for next state must be in the same positions

Merging equivalence groups

- Successors must be in the same classes

Primitive flow table

Pres state	Next State X=00	01	10	11	Q
A	(A)	F	C	-	0
B	A	(B)	-	H	1
C	G	-	(C)	-	0
D	-	F	-	(D)	1
E	G	-	(E)	D	1
F	-	(F)	-	K	0
G	(G)	B	J	-	0
H	-	L	E	(H)	1
J	G	-	(J)	-	0
K	-	B	E	(K)	1
L	A	(L)	-	K	1

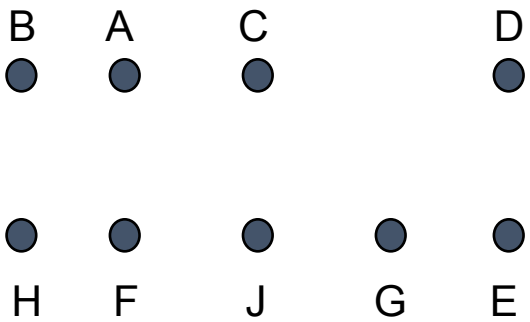
Reduced flow table

Pres state	Next State X=00	01	10	11	Q

Finding compatible states and merging them

- both Si and Sj have the same successor, or
- both Si and Sj are stable, or
- the successor of Si or Sj, or both, is unspecified

Moreover, both S_i and S_j must have the same output whenever specified



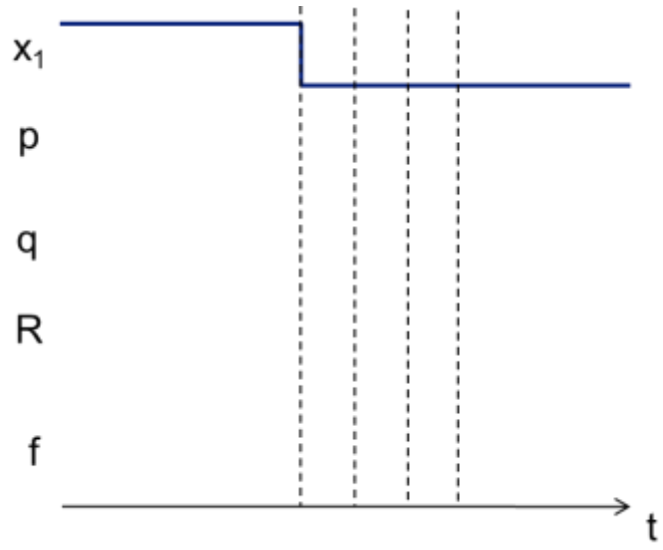
Pres state	Next State X=00	01	10	11	Q

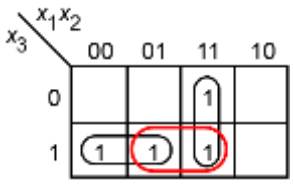
Lecture 13: Hands-on

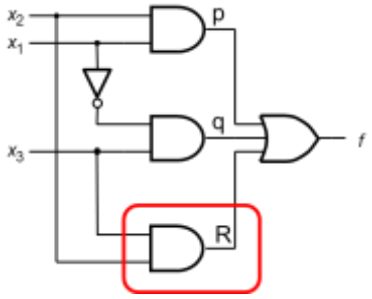
Static Hazard

$f = x_1x_2 + \bar{x}_1x_3$

What is the value of f under the input of $x_3x_2x_1$=111 and $x_3x_2x_1$=110. Complete the timing diagram for the following circuit by considering a delay for each gate.







Dynamic Hazard

$f = x_1\bar{x}_2 + \bar{x}_3x_4 + x_1x_4$

What is the value of f under the input of $x_4x_3x_2x_1$=1110 and $x_4x_3x_2x_1$=1111. Complete the timing diagram for the following circuit by considering a delay for each gate.