ROM

Types:

ROM

PROM

EPROM

EEPROM

HOW ROM works

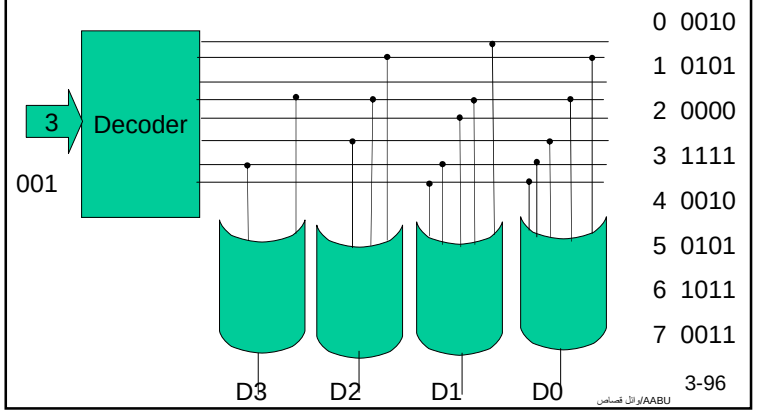
It consists of a decoder and OR gates.

In general Memory has address and Data lines.

We call a ROM with n address lines and m data lines 2^nxm memory

This memory has n words, each word is m bits

8 x 4 bit Memory



From the previous discussion its clear that each output of the ROM represents a SOP

Ex.

F_0= Σ(1,3,5,6,7)

F_1= Σ(0,3,4,6,7)

F_2=

F_3=

Another example

Assume we want to build the Following Ckt that has two inputs and three outputs

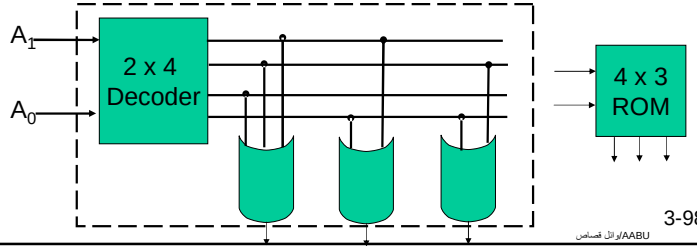
$F_0 = \Sigma(1,3)$

$F_1 = \Sigma(0,3)$

$F_2 = \Sigma(0,1,2)$

We can build this Ckt using a 4x3 ROM = 2^2 x 3 bit ROM

This ROM is built from a decoder, OR gates



Q. What is the Data stored in this ROM?

BCD to Ex.3 Converter

What is the size of the ROM needed?

What will be the data stored?

BCD to 7 segment Converter

What is the size of the ROM needed?

What will be the data stored?

Combinational vs. Sequential

Combinational Circuit

- **always gives the same output for a given set of inputs**
  - **ex: adder always generates sum and carry, regardless of previous inputs**

Sequential Circuit

- **stores information**
- **output depends on stored information (state) plus input**
  - **so a given input might produce different outputs, depending on the stored information**
- ***example:* ticket counter**
  - **advances when you push the button**
  - **output depends on previous state**
- **useful for building "memory" elements and "state machines"**

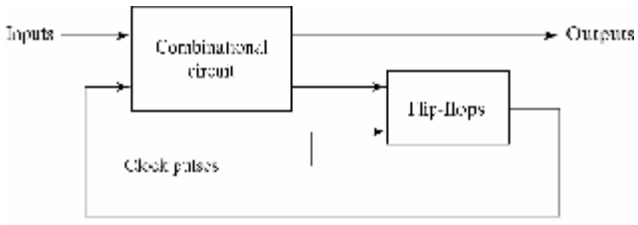
Sequential Logic

Synchronous sequential circuits:

A system whose behavior can be defined form the knowledge of its signals at discrete instants of time.

Asynchronous sequential circuits: a system whose behavior depends on the order which its input signals changes at any instance of time

Fig. 5-1 Block Diagram of Sequential Circuit

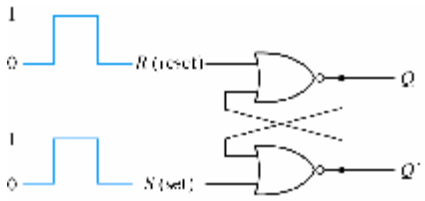


(a) Block diagram

(b) Timing diagram of clock pulses

Fig. 5-2 Synchronous Clocked Sequential Circuit

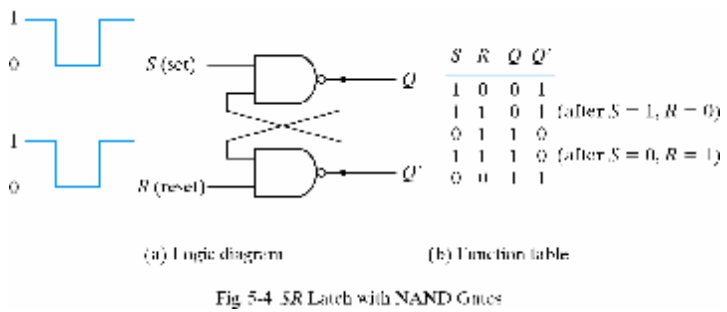
Flip Flops



S	R	Q	Q'	
1	0	1	0	
0	0	1	0	(after $S = 1, R = 0$)
0	1	0	1	
0	0	0	1	(after $S = 0, R = 1$)
1	1	0	0	

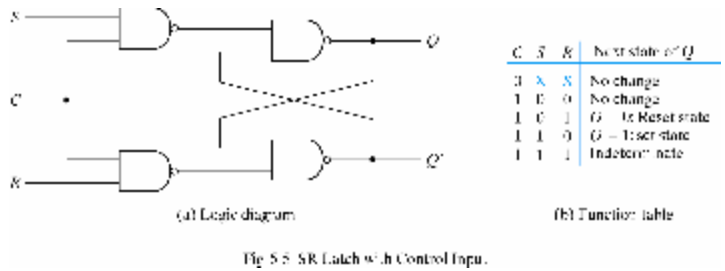
(a) Logic diagram (b) Function table

Fig. 5-3 *SR* Latch with NOR Gates



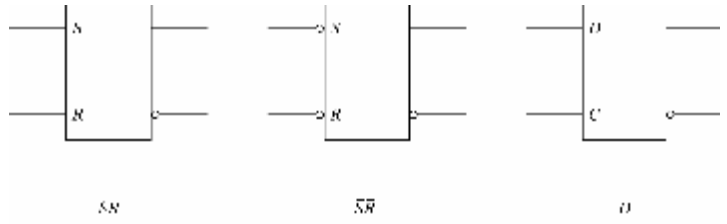
(a) Logic diagram (b) Function table

Fig. 5-4 *SR* Latch with NAND Gates



(a) Logic diagram (b) Function table

Fig. 5-5 SR Latch with Control Input.



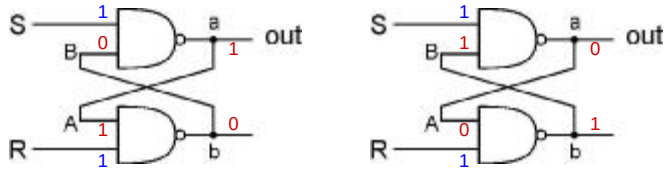
SR $\overline{SR}$ *Q*

Fig. 5-7 Graphic Symbols for Latches

R-S Latch: Simple Storage Element

R is used to "reset" or "clear" the element – set it to zero.

S is used to "set" the element – set it to one.

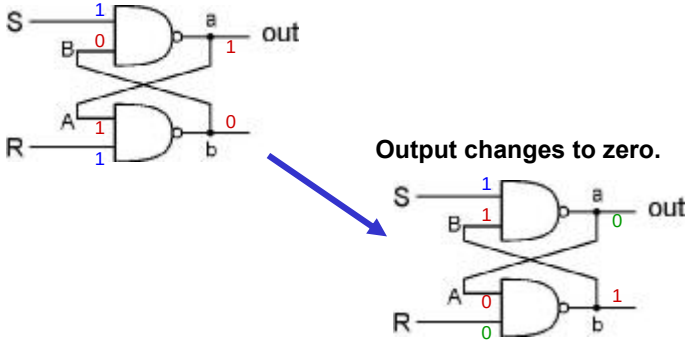


If both R and S are one, out could be <u>either</u> zero or one.

- **"quiescent" state -- holds its previous value**
- **note: if a is 1, b is 0, and vice versa**

Clearing the R-S latch

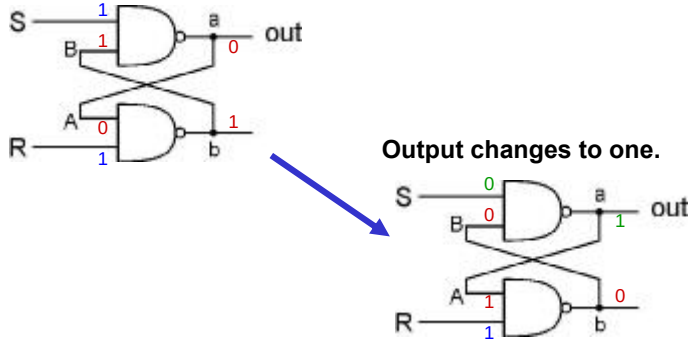
Suppose we start with output = 1, then change R to zero.



Then set R=1 to "store" value in quiescent state.

Setting the R-S Latch

Suppose we start with output = 0, then change S to zero.



Then set S=1 to "store" value in quiescent state.

R-S Latch Summary

R = S = 1
- **hold current value in latch**

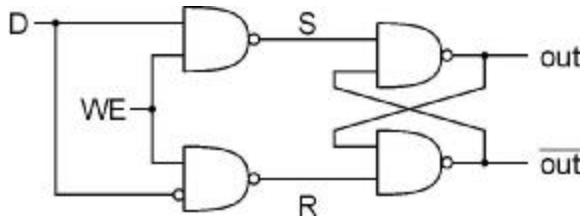
S = 0, R=1
- **set value to 1**

R = 0, S = 1
- **set value to 0**

R = S = 0
- **both outputs equal one**
- **final state determined by electrical properties of gates**
- *Don't do it!*

Gated D-Latch

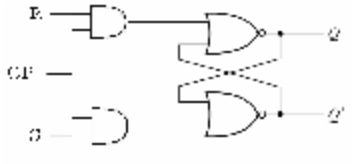
Two inputs: D (data) and WE (write enable)
- **when WE = 1, latch is set to value of D**
  - **S = NOT(D), R = D**
- **when WE = 0, latch holds previous value**
  - **S = R = 1**



Characteristic equations

Remember that the output depends on the inputs and the current state.

- **First find the characteristic table**
- **Then derive the characteristic equation for the RS flip flop.**



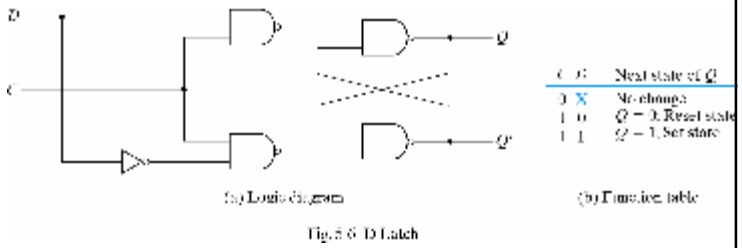
Q_t	S	R	Q_{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	X
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	X

SR

Q	0	0	X	1
	1	0	X	1

Q(t+1)=S+R'Q

Characteristic equation for D flip flop



Qt	D	Q_{t+1}
0	0	0
0	1	1
1	0	0
1	1	1

0	1
0	1

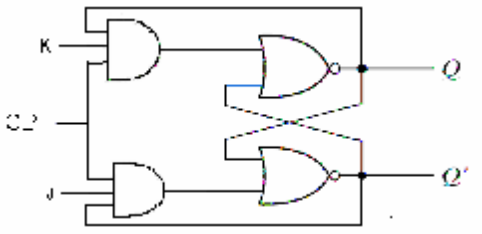
Q(t+1)=D

J K flip flop

It's a refinement of RS flip flop , that defined the indeterminate states in RS.

In RS flip flop the state 11 is not allowed ,

In JK flip flop the state 11 makes the flip flop changes (switches) its output.



Problem in JK flip flop:

- **When J=1 ,K=1 and the clock is 1 , $Q_{t+1}=Q'_t$**
- **Assume Q=1, it will flip to 0 then to 1 then to 0 and so on as long as the clock is 1.**
- **To avoid that the clock pulse (duration) must be less than the propagation delay of the Flip flop.**
- **But this is not a solution.**
- **The solution is to build a Master slave or edge triggered construction.**

Characteristic table and Equation

Q_t	J	K	Q_{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

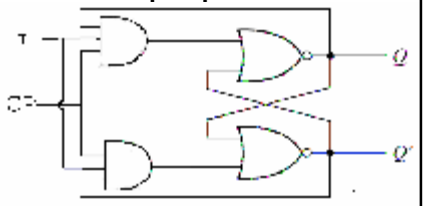
Q \ JK	0 0	0 1	1 1	1 0
0	0	0	1	1
1	1	0	0	1

$$Q(t+1)=JQ'+K'Q$$

T flip flop

It's a single input version of the JK Flip flop

T flip Flop has the same Problem of JK when T=1



Qt	T	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

0	1
1	0

$$Q(t+1)=TQ'+T'Q$$

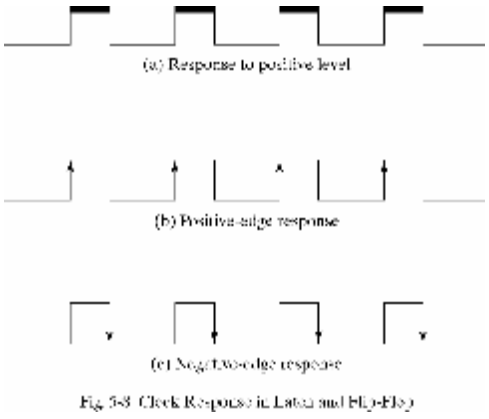


Fig. 5-8 Clock Response in Latch and Flip-Flop

Master Slave D flip flop

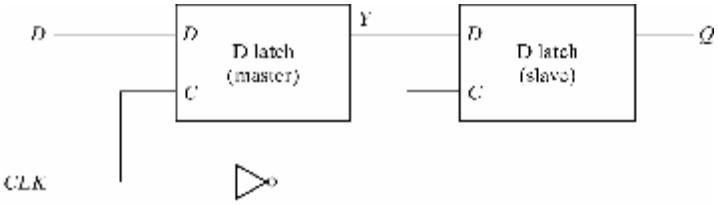
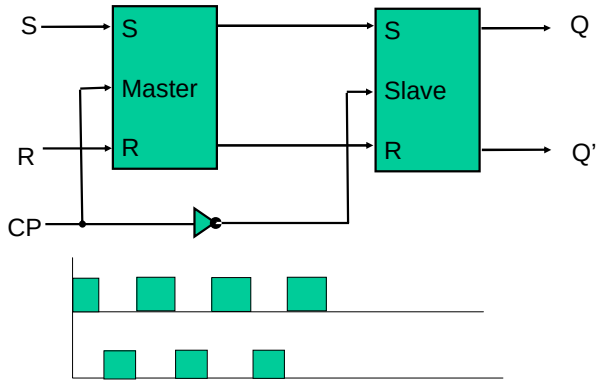


Fig. 5-9 Master-Slave *D* Flip-Flop

Master Slave RS flip flop



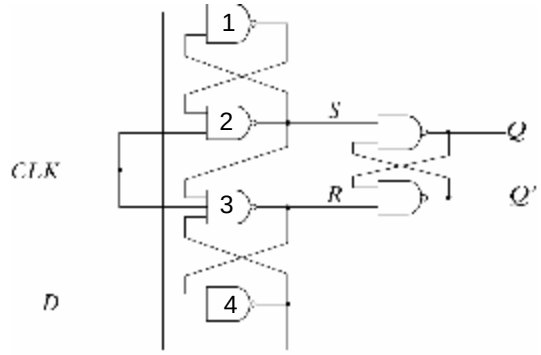


Fig. 5-10 D-Type Positive-Edge-Triggered Flip-Flop

When CLK = 0 , gates 2,3 are not working,gate 2=1. gate3=1 ➔ R=1,S=1

➔No change in the output.

If D=0; Gate 4=1, Gate1 =0.

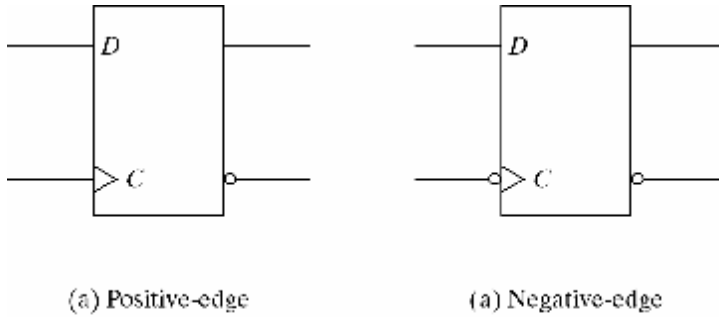
If D=1; Gate 4=0, Gate1 =1.

IF D=0➔ Gate1=0,Gate4=1,Gate2=1,Gate3=1;
and CLK goes to 1

Gate 4=1, Gate 3=0, Gate1=0;Gate 2=1;

After the CLK being 1, if D changed to 1, this will not affect on Gate 4 , nor any other gates.

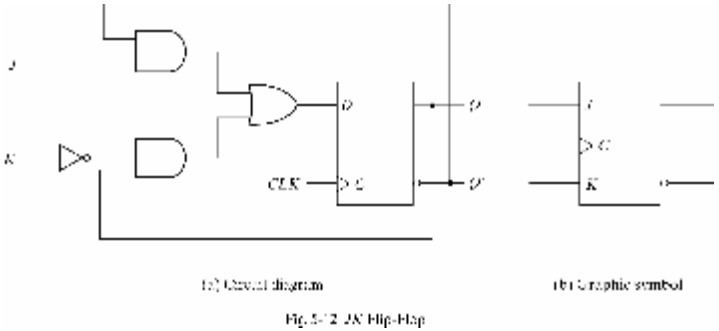
IF D=1➔ Gate1=1,Gate4=0,Gate2=1,Gate3=1;
and CLK goes to 1
Gate 4=0, Gate 3=1, Gate1=1;Gate 2=0;
After the CLK being 1, if D changed to 0, Gate 4 =1 , other gates will not affect by this change.



(a) Positive-edge (a) Negative-edge

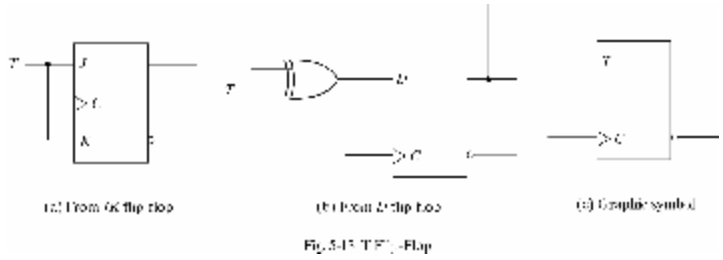
Fig. 5-11 Graphic Symbol for Edge-Triggered *D* Flip-Flop

JK flip Flop from D flip flop



(a) General diagram (b) Graphic symbol

Fig. 5-12 JK Flip-Flop



(a) From JK flip-flop (b) From D flip-flop (c) Graphic symbol

Fig. 5-13 T F/F

Analysis of sequential circuit

The behavior of a seq. ckt is determined form

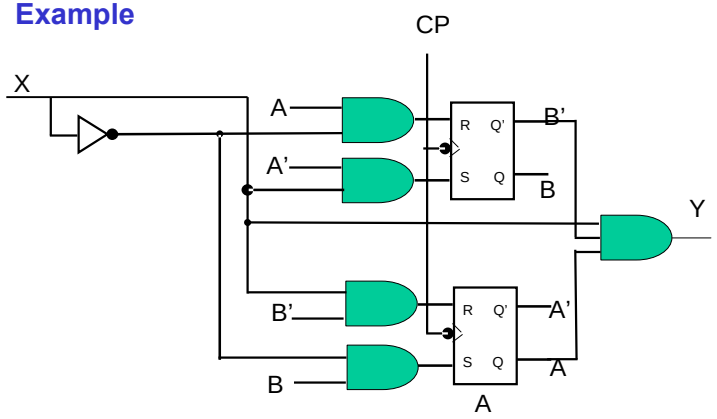
a) **Inputs**
b) **Outputs**
c) **States of its flip flop**

Both outputs and next state are function of input and present state.

From a seq. ckt diagram , we will find the

a) **state table**
b) **State diagram**

Example



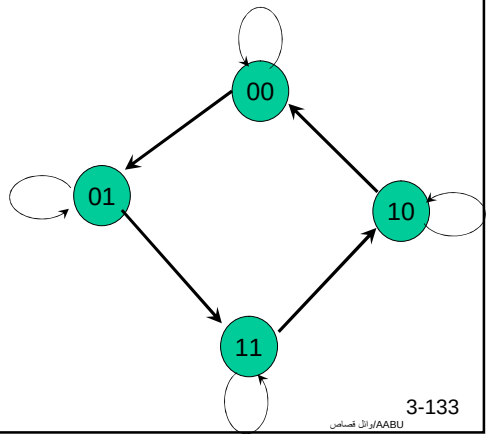
State Table

Present State	Next state		Output	
	X=0	X=1	X=0	X=1
AB	AB	AB	Y	Y
00	00	01	0	0
01	11	01	0	0
10	10	00	0	1
11	10	11	0	0

State diagram

The state table can be represented graphically using the state diagram.

Transition from a state to state is shown as arrow labeled with two values (Input/output)



State equation

State equation (or application equation) is an expression that shows the relation for the next state of each flip flop as a function of the present state and the inputs,

Method 1: using the characteristic equation of the flip flop

A(t+1) =S+R'Q

= X'.B+ (X.B')'A

= X'.B+(X'+B).A

= X'B+X'.A+A.B

B(t+1) = S+R'Q

= X.A'+(X'A)'.B

= X.A'+X.B+A'.B

State equation

Method 2:

From the state table.

A(t+1) = (A'B+AB'+AB).X' + ABX

=A'BX'+AB'X'+ABX' +ABX

=BX'+AX'+AB

6.7 Design procedure

1. Build the state diagram
2. Build the state table
3. Assign binary values for each state
4. Determine the number of flip flops needed and assign a symbol for each flip flop
5. Choose the type of flip flop to be used (we will use JK)
6. From the state table derive the excitation and output tables
7. Simplify the flip flop functions
8. Draw the logic diagram

The following formulas for JK flip flop inputs will help us

0 to 0 J=0, K=X (don't care)

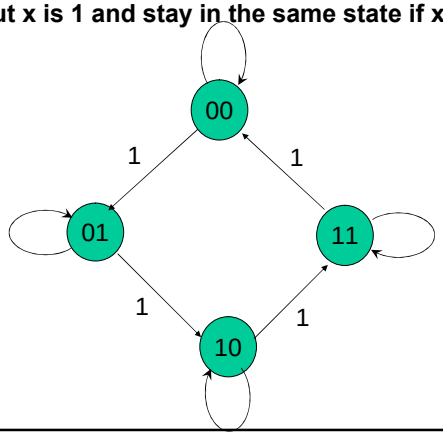
0 to 1 J=1, K=X

1 to 0 J=X, K=1

1 to 1 J=X, K=0

Example 1

Design a circuit that works as a counter from 0 to 3 if the input x is 1 and stay in the same state if x is 0



State table

	X=0	X=1
AB	AB	AB
00	00	01
01	01	10
10	10	11
11	11	00

We have 4 states so we need 2 flip flops

Excitation table

	Next state	flip flop inputs			
A B X	A B	JA	KA	JB	KB
0 0 0	0 0	0	X	0	X
0 0 1	0 1	0	X	1	X
0 1 0	0 1	0	X	X	0
0 1 1	1 0	1	X	X	1
1 0 0	1 0	X	0	0	X
1 0 1	1 1	X	0	1	X
1 1 0	1 1	X	0	X	0
1 1 1	0 0	X	1	X	1

Now we need to simplify the equation of each flip flop input

0	0	1	0
x	x	x	X

JA=Bx

x	x	x	X
0	0	1	0

KA=Bx

0	1	x	X
0	1	x	x

JB=x

x	x	1	0
x	x	1	0

KB=x

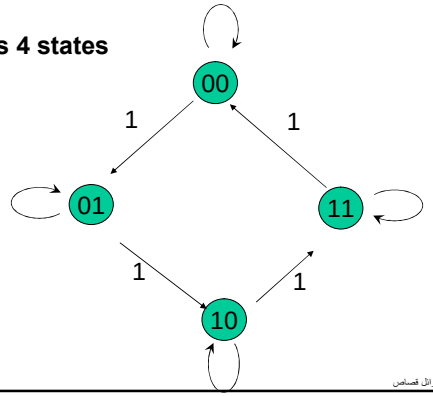
Example 2

Design a circuit that works as a down counter from 3 down to 0 if the input x is 1 and stay in the same state if x is 0

Example 3

Design a sequential Ckt that generate the following sequence 000, 001, 010,100 , using JK flip flops

This circuit has 4 states



Example 4

Repeat example 1 using RS flip flop

1) How RS Flip flop works

0 to 0 S=0, R=X (don't care)

0 to 1 S=1, R=0

1 to 0 S=0, R=1

1 to 1 S=X, R=0

2) Build the Excitation table, then simplify the equations that represents R and S for each flip flop.

Excitation table

	Next state	flip flop inputs			
A B X	A B	SA	RA	SB	RB
0 0 0	0 0	0	X	0	X
0 0 1	0 1	0	X	1	0
0 1 0	0 1	0	X	X	0
0 1 1	1 0	1	0	0	1
1 0 0	1 0	X	0	0	X
1 0 1	1 1	X	0	1	0
1 1 0	1 1	X	0	X	0
1 1 1	0 0	0	1	0	1

Now we need to simplify the equation of each flip flop input

0	0	1	0
X	X	0	X

SA=

X	X	0	X
0	0	1	0

RA=

0	1	0	X
0	1	0	X

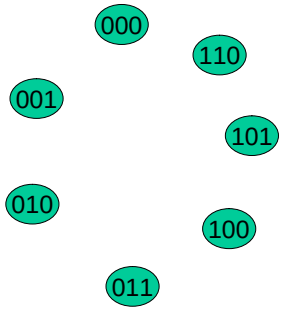
SB=

x	0	1	0
x	0	1	0

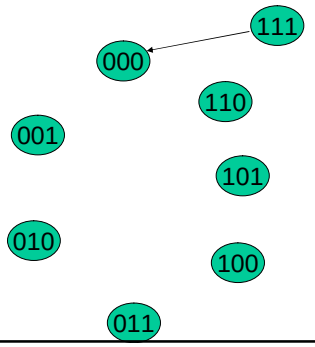
RB=

Example 5

Design an up_down counter that counts from 0 to 6, depending on the input value, if x=0 it counts down, if 1 it counts up

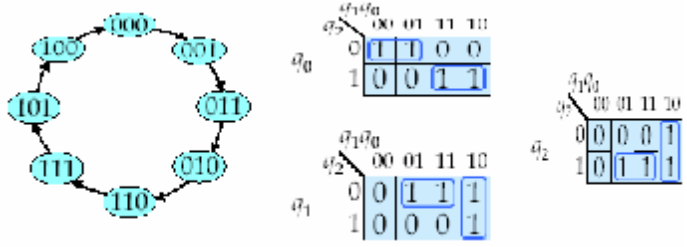


We have a small problem here.
What if the counter started with 111 ?
We need to move it to one of the valid states,
To 000 for example



Non-Standard Counters

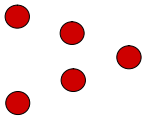
- Counters are sometimes defined that count in an order other than standard numerical order.
- The state machine below is for a *gray code* counter in which one bit changes at a time.

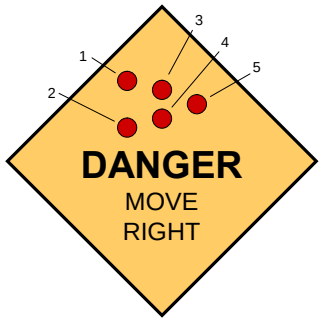


Complete Example

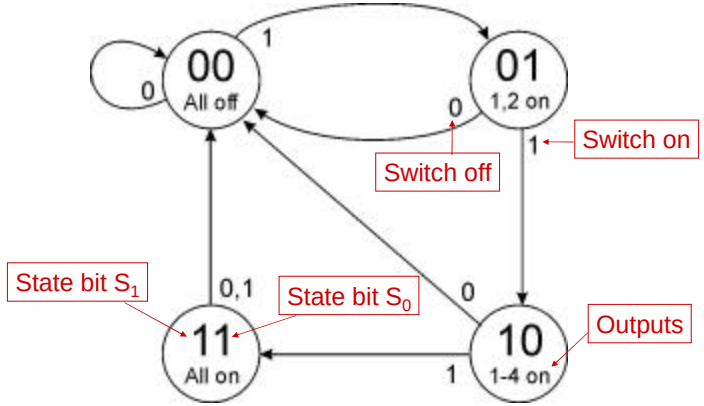
A blinking traffic sign

- **No lights on**
- **1 & 2 on**
- **1, 2, 3, & 4 on**
- **1, 2, 3, 4, & 5 on**
- **(repeat as long as switch is turned on)**
- **When the input is 0 , No lights on**





Traffic Sign State Diagram



Transition on each clock cycle.

Traffic Sign Truth Tables

Outputs
(depend only on state: S_1S_0)

Lights 1 and 2
Lights 3 and 4
Light 5

S_1	S_0	Z	Y	X
0	0	0	0	0
0	1	1	0	0
1	0	1	1	0
1	1	1	1	1

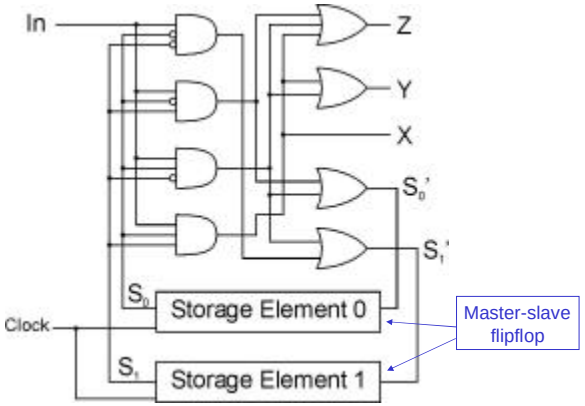
Next State: $S_1'S_0'$
(depend on state and input)

Switch

In	S_1	S_0	S_1'	S_0'
0	X	X	0	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0

Whenever In=0, next state is 00.

Traffic Sign Logic



From Logic to Data Path

The data path of a computer is all the logic used to process information.

- **See the data path of the LC-2 on next slide.**

Combinational Logic

- **Decoders -- convert instructions into control signals**
- **Multiplexers -- select inputs and outputs**
- **ALU (Arithmetic and Logic Unit) -- operations on data**

Sequential Logic

- **State machine -- coordinate control signals and data movement**
- **Registers and latches -- storage elements**