

# Sequential circuits

## Unit-4

### State Diagram:

In this diagram, a state is represented by a circle, & the transition b/w states is indicated by direct lines (or arcs) connecting the circles.

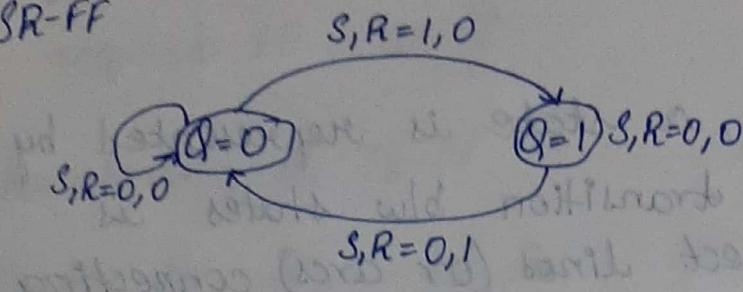
Note- In this model the effect of all previous inputs on the O/P's is represented by a state of the circuit. Thus, the O/P of the circuit at any time depends upon the its current state & the input. These also determine the next state of the circuit. The Relationship that exists among the inputs, outputs, present states & next states can be specified by either the state table or the state diagram.

### State Table:

The State Table representation of a sequential circuit consists of three sections labelled present state, next state & O/P. The present state designates the state of flip flop before the occurrence of a clock pulse. The next state shows the states of flipflops after the clock pulse, and the O/P section lists the value of the O/P variables during the present state.

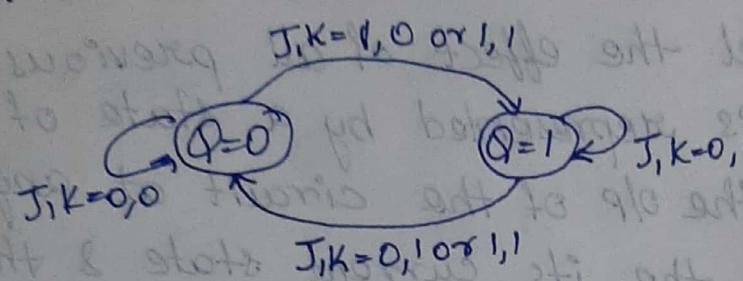
# State diagrams for various flipflops.

SR-FF



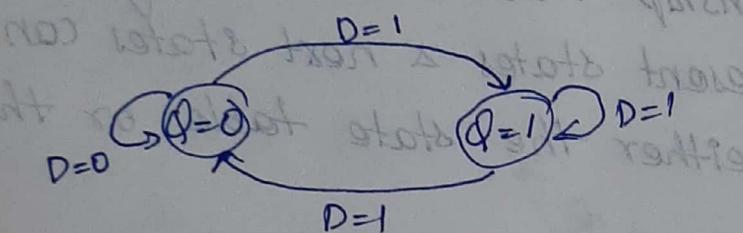
S	R	Q
0	0	0
0	1	1
1	0	1
1	1	Undetermined

JK-FF



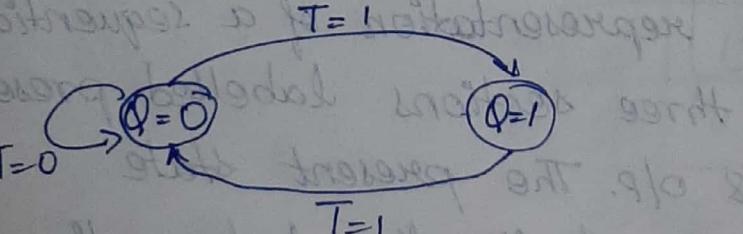
J	K	Q
0	0	0
0	1	1
1	0	1

D-FF



D	Q
0	0
1	1

T-FF

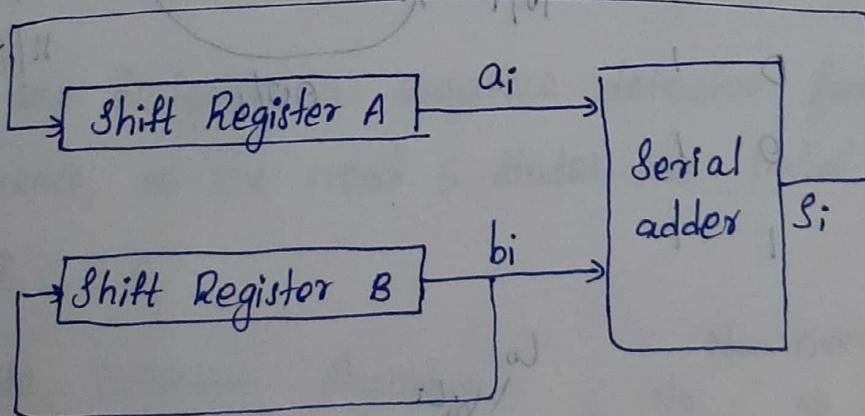


T	Q
0	0
1	1

## Serial Binary Adder (bit serial adder)

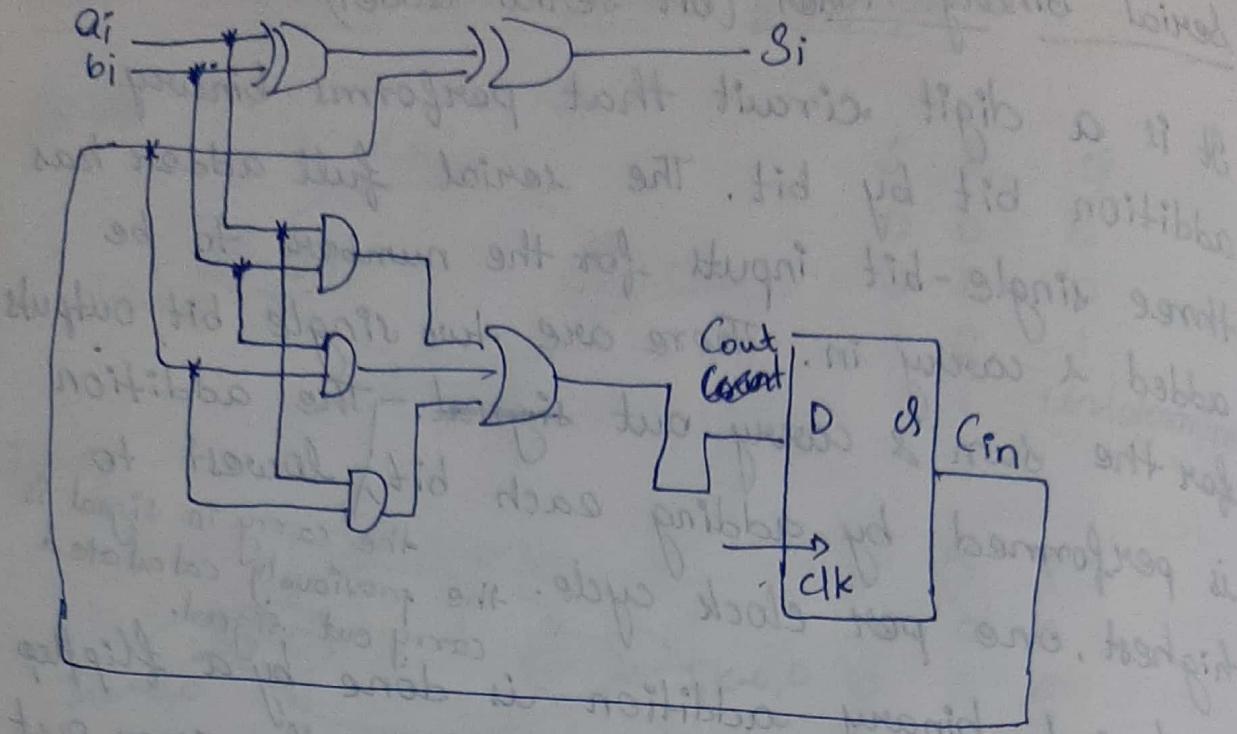
It is a digit circuit that performs binary addition bit by bit. The serial full adder has three single-bit inputs for the numbers to be added & carry in. There are two single bit, outputs for the sum & carry out signal. The addition is performed by adding each bit lowest to highest, one per clock cycle. the carry in signal is the previously calculated carry out signal.

Serial binary addition is done by a flip flop & a full adder. The flip flop takes the carry out signal on each clock cycle & provides its values as the carry in signal on the next clock cycle. After all of the bits of the input operands have arrived, all of the bits of the sum have come out of the sum O/P.



$$\begin{array}{r} 1+1+0=0 \\ \hline 11|0 \end{array}$$
$$\begin{array}{r} 0+0+0=00 \\ \hline 00|0 \end{array}$$
$$0+1+0=1001/1$$
$$1+0+0=1010/1$$
$$1+1+1=1001/1$$
$$0+0+1=1\quad 0$$

$$\begin{array}{r} 0+1+1=0\quad 1 \\ \hline 10|1 \end{array}$$
$$\begin{array}{r} 1+0+1=0\quad 1 \\ \hline 11|1 \end{array}$$
$$1+1+1=1\quad 1$$



$a_i$   $b_i$   $c_i$   $s_o$   $c_o$

0 0 0 0 0  
0 0 1 1 0

0 1 0 1 0  
0 1 1 0 1

0 1 1 0 1  
1 0 0 1 0

1 0 1 0 1  
1 1 0 1 0

1 1 1 1 1  
1 1 1 1 1

$s_o$   $bci$   
 $a$

$s_o$   $bci$   
 $a$

$$s_o = a \oplus b \oplus c_i$$

$$c_o = b_i c_i + a_i c_i + a_i b_i$$

## Sequence detector

A sequence detector accepts as input a string of bits : either 0 or 1. Its output goes to 1 when a target sequence has been detected.

There are two basic types: overlap & non-overlap.

In sequence detector that allows overlap, the final bits of one sequence can be the start of another sequence.

Sequence to be detected is 11011

i/p string is 11011011011

11011 detector with overlap  $\times$  11011011011

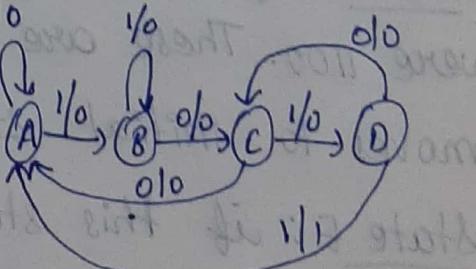
$\Sigma$  00001001001

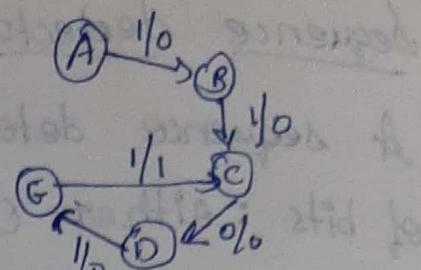
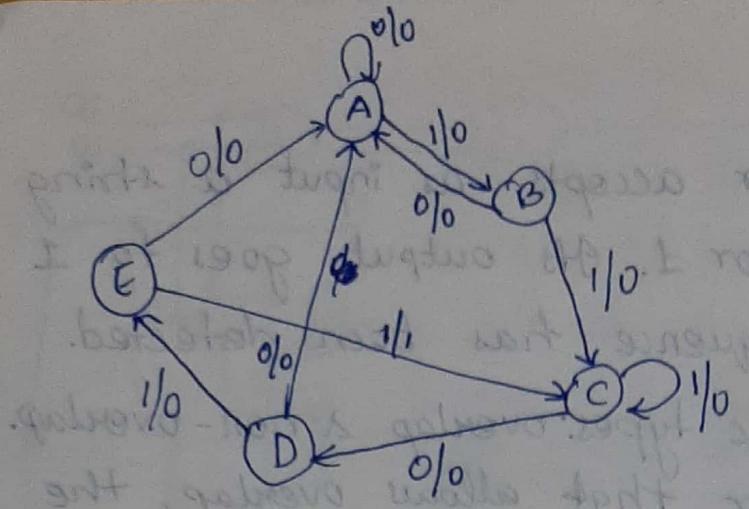
11011 detector without overlap  $\times$  11011011011

$\Sigma$  00001000001

We are designing a sequence detector for a 5 bit sequence, so we need 5 states. We label these states

A, B, C, D, E.

	State	Detection	Awaiting	Non-overlapping
A	-	11011	11011	
B	1	1011		
C	11	011		
D	110	11		
E	1101	1		



State A: it is a initial state, it is waiting on a 1. if it gets a '0', the machine remains to be in state A & continues to remain there while 0's are input.

State B: if state B get a 0, the last two bits input were '10'. This does not begin the sequence, so the machine goes back to state-A & waits on the next 1.

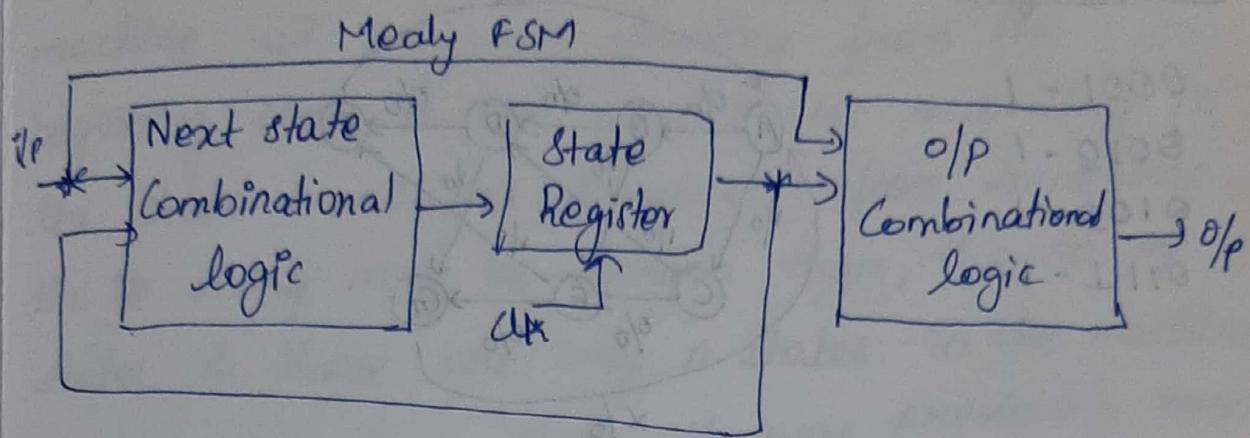
State C: if state c gets a 1, the last three bit inputs were 111 it can use the last two of these 1's to be the first two ones of the sequence 11011, so the machine stays in state c awaiting for 0.

State D: if state D goes a 0, then last four bits i/p were 1100. These are not part of sequence, so it moves to initial state.

State E: if this state gets a 0, the last five i/p were 11010 which is also not a part of sequence. hence moves to initial state.

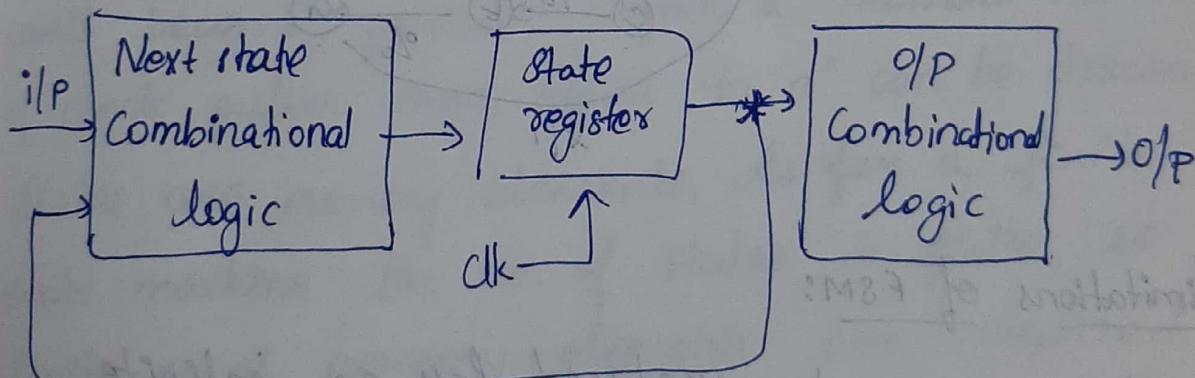
## Finite State Machine

### Mealy Machine



\* o/p is a function of both present state & input.

### Moore Machine



\* o/p is a function of only present state.

Parity bit generator

Odd  
Even

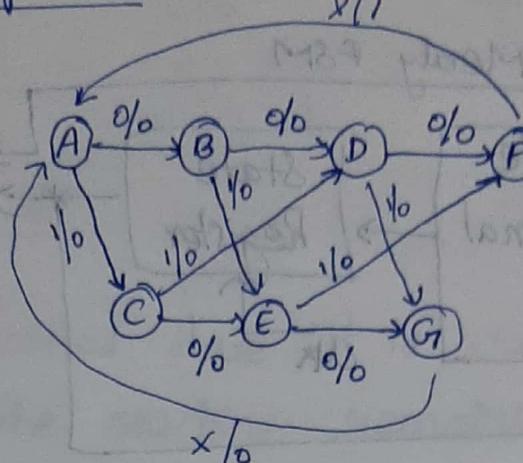
Odd parity bit generator (3 bit)

0001 - 1

0010 - 1

0100 - 1 (not allowed)

0111 - 1 (legal)



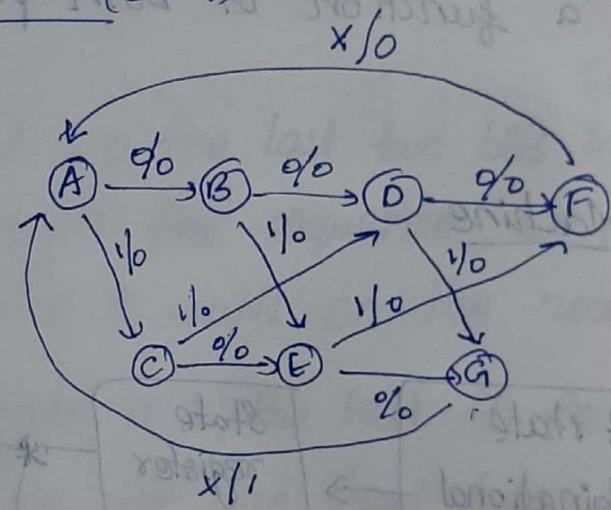
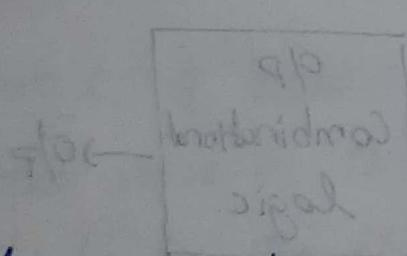
Even parity bit generator (3 bit)

0000 - 1

0011 - 1

0101 - 1

0110 - 1



Limitations of FSM:

- No FSM can be produced for an infinite sequence.
- No FSM can multiply two arbitrary large binary No.

### Capabilities of FSM:

Let a FSM have  $n$  states. Let a long sequence of I/P be given to the machine. The machine will progress starting from its beginning state to the next states according to the state transitions. However, after sometime the I/P string may be longer than, the no. of states. As there are only  $n$  states in the machine, it must come to a state it was previously been in & form this phase if the I/P remains the same the machine will function in a periodically repeating fashion from here a conclusion that "for an state machine the O/P will become periodic after a ~~moment~~ number of clock pulses than equal to  $n$ " can be drawn. States are memory elements. As for a finite state machine the no. of states is finite, so finite no. of memory elements are required to design a finite state machine.

