FINITE STATE MACHINE (FSM)

BY

JAYANTA DEVNATH (120123021)

SOURAV SARKAR (120123041)

DIPANJAN SARKAR (120123015)

DATE : 22/10/2013

DEFINITION OF FSM:

A finite state machine (FSM) or finite state automation is a mathematical model of computation used to design both "Computer Programs" and "Sequential Logic Circuits".It is conceived as an abstract machine that can be in one of a finite number of *states*.

(source : wiki)

- It is called a Finite State Machine because it can have, at most, a finite number of states.
- It is composed of a combinational logic unit and flip-flops placed in such a way as to maintain state information.

REPRESENTING A FINITE STATE MACHINE

It can be represented using a **state transition table** which shows the *current state*, *input*, *any outputs*, and the *next state*.

Input / Current State	Input_0	Input_1	….	Input_n
State_0	Next State / Output		….	Next State / Output
State_1	….		….	….
….	….		….	….
State_n	….		….	….

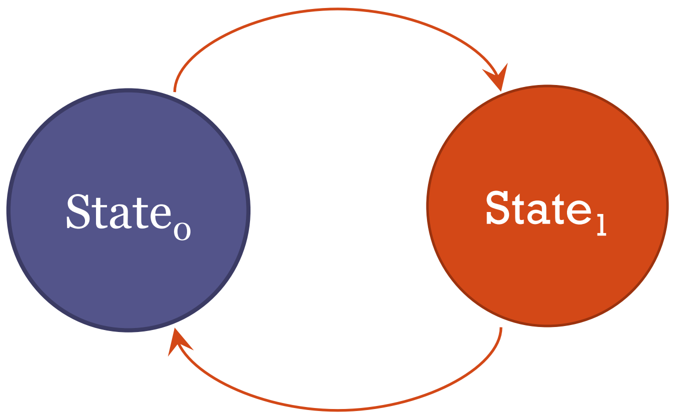
NOTE:

Sequential systems contain state stored in memory elements internal to the system. Their behavior depends both on the set of inputs supplied and on the contents of the internal memory, or state of the system. Thus, a sequential system cannot be described with a truth table. Instead, a sequential system is described as a **finite-state machine (or often just *state machine*).**

REPRESENTING A FINITE STATE MACHINE CONTD…

- Finite State Machine (FSM) can also be represented using a **state diagram** which has the same information as the state transition diagram.

.



TYPES OF FSM

There are two types of finite state machines :

MOORE MACHINE

and

MEALY MACHINE

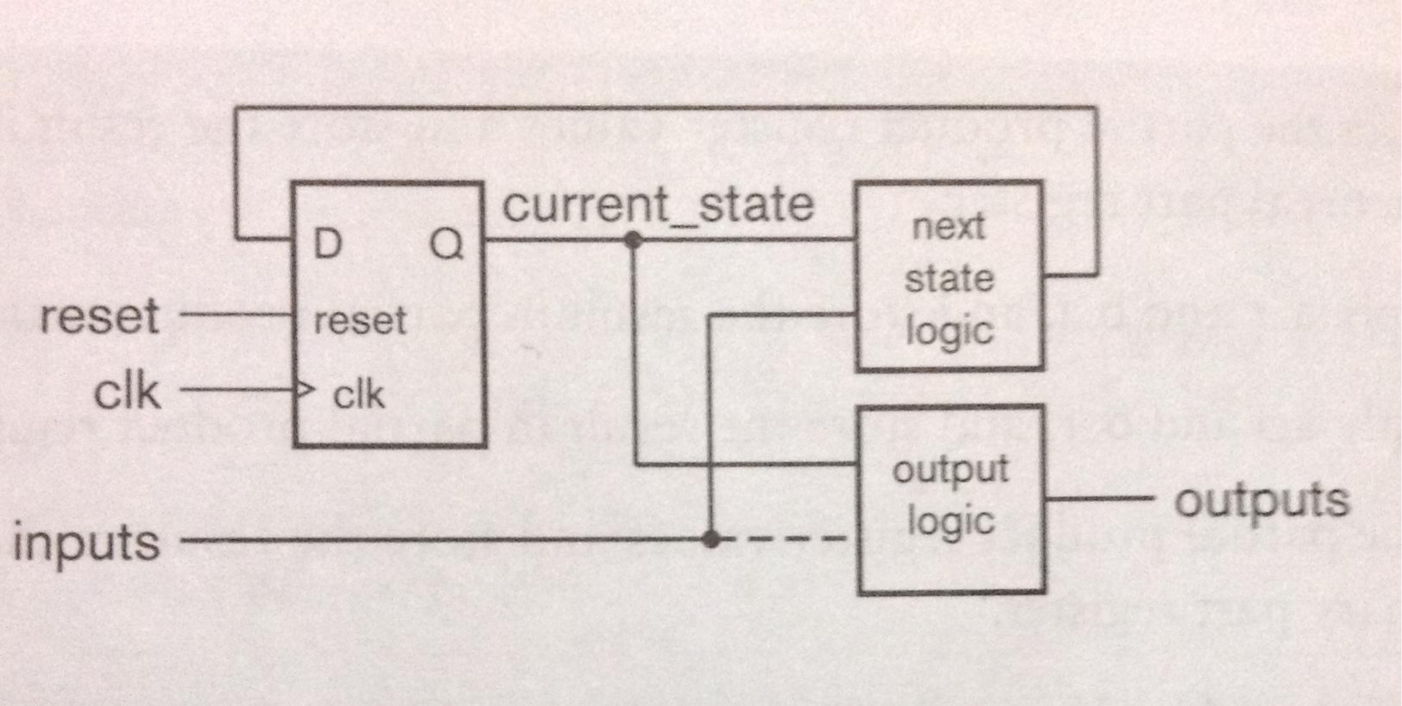
MOORE MACHINE

- It is a Finite State Machine(FSM) whose output values are determined solely by its current state and not by input values.
- In this type of machine if the input values change during clock cycle the outputs remain unchanged.

MEALY MACHINE

- It is a Finite State Machine(FSM) whose output values are determined both by the current state and the input values.
- In this type of machine if the input values change during a clock cycle the output values may change as a consequence.

CIRCUIT DIAGRAM REPRESENTATION FOR A FINITE STATE MACHINE



The above diagram without the dashed line is a representation for a MOORE MACHINE and with the dashed line it is a representation for a MEALY MACHINE.

NOTE:

The behavior of state machines can be observed in many devices in modern society which perform a predetermined sequence of actions depending on a sequence of events with which they are presented. Simple examples are vending machines which dispense products when the proper combination of coins is deposited, elevators which drop riders off at upper floors before going down, traffic lights which change sequence when cars are waiting, and combination locks which require the input of combination numbers in the proper order.

RELATIONSHIP OF MOORE MACHINES WITH MEALY MACHINES

- The difference between Moore machines and Mealy machines is that in the latter, the output of a transition is determined by the combination of current state and current input.
- In other words in a **state diagram …**
  - for a Moore machine, each node (state) is labeled with an output value
  - for a Mealy machine, each arc (transition) is labeled with an output value.

EXAMPLES OF FINITE STATE MACHINE

1. ODD PARITY CHECKER

For an ODD PARITY CHECKER…

$$\text{output} = \begin{cases} 1 & \text{If odd \# of 1s in input} \\ 0 & \text{If even \# of 1s in input} \end{cases}$$

Lets analyse this for both MOORE and MEALY Machines……

SOME NOTATIONS FOR THE EXAMPLES:

I/P → INPUT

O/P → OUTPUT

PS → PRESENT STATE / CURRENT STATE

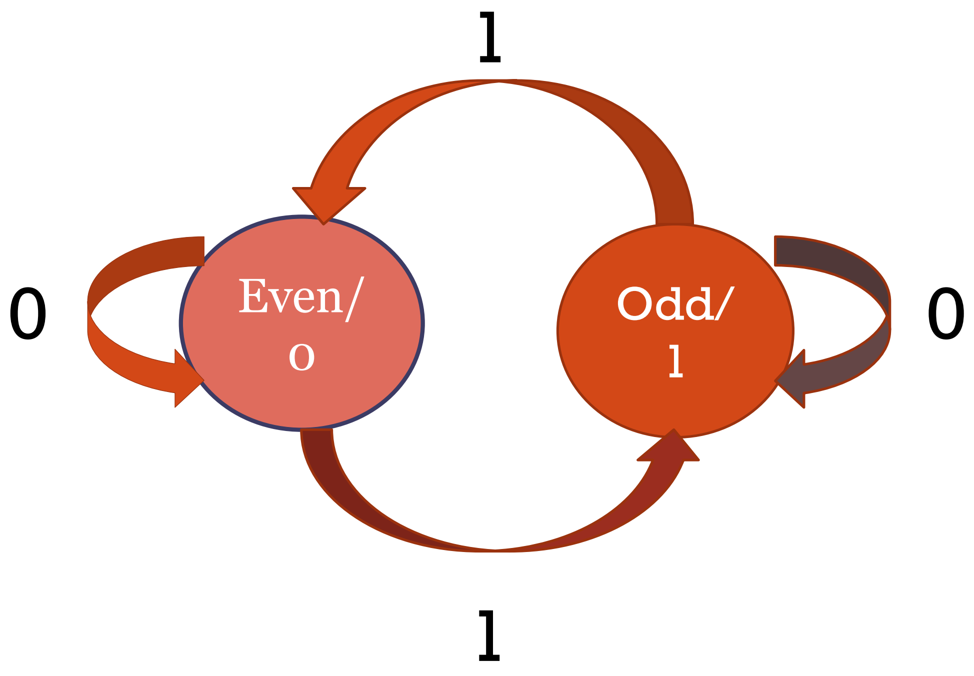
NS → NEXT STATE

$\oplus$ → XOR GATE

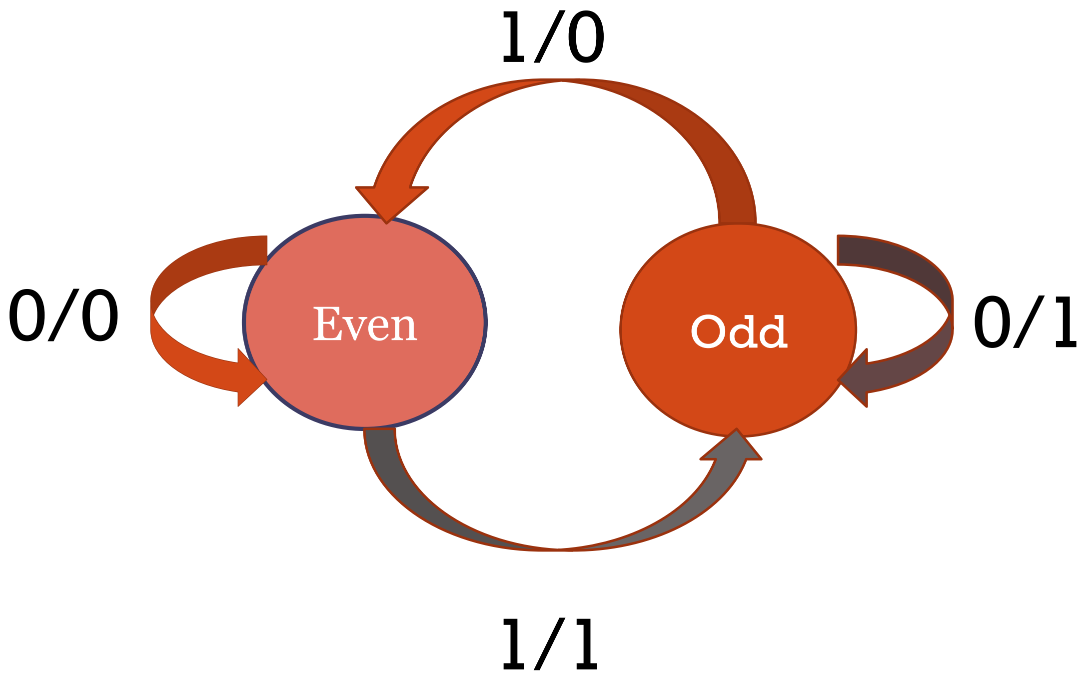
→ NUMBER

STATE DIAGRAMS

For MOORE Machine



For MEALY Machine



STATE TRANSITION TABLES

For MOORE Machine

PS	I/P	NS	O/P
EVEN	0	EVEN	0
EVEN	1	ODD	0
ODD	0	ODD	1
ODD	1	EVEN	1

For MEALY Machine

PS	I/P	NS	O/P
EVEN	0	EVEN	0
EVEN	1	ODD	1
ODD	0	ODD	1
ODD	1	EVEN	0

STATE ENCODING AND LOGIC MINIMIZATION

For MOORE Machine

PS	I/P	NS	O/P
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

$O/P = PS$

$NS = (PS) \oplus (I/P)$

For MEALY Machine

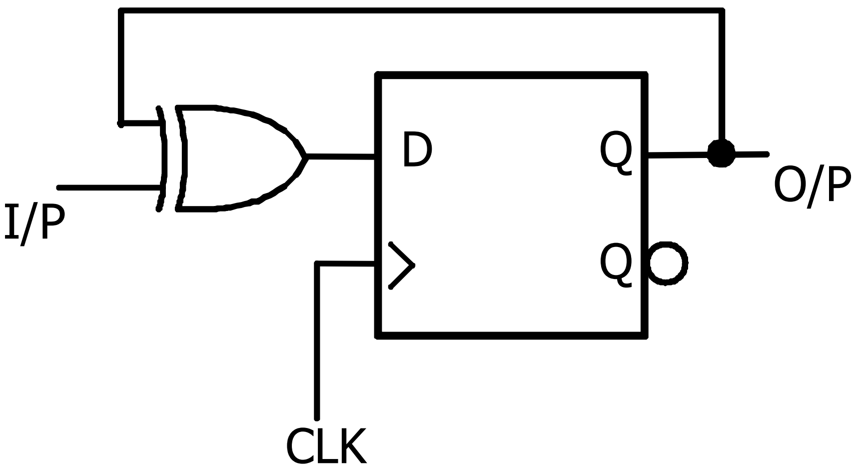
PS	I/P	NS	O/P
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	0

$O/P = (PS) \oplus (I/P)$

$NS = (PS) \oplus (I/P)$

IMPLIMENTATION OF THE DESIGN

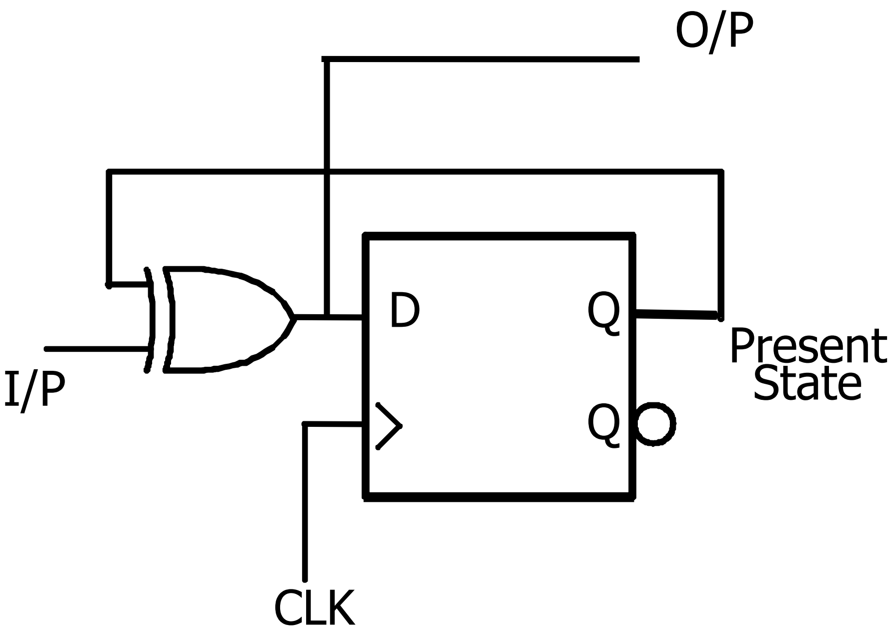
For MOORE Machine



$O/P = (PS)$

$NS = (PS) \oplus (I/P)$

For MEALY Machine



$O/P = (PS) \oplus (I/P)$

$NS = (PS) \oplus (I/P)$

2. 2-BIT BINARY GRAY CODE UP-COUNTER

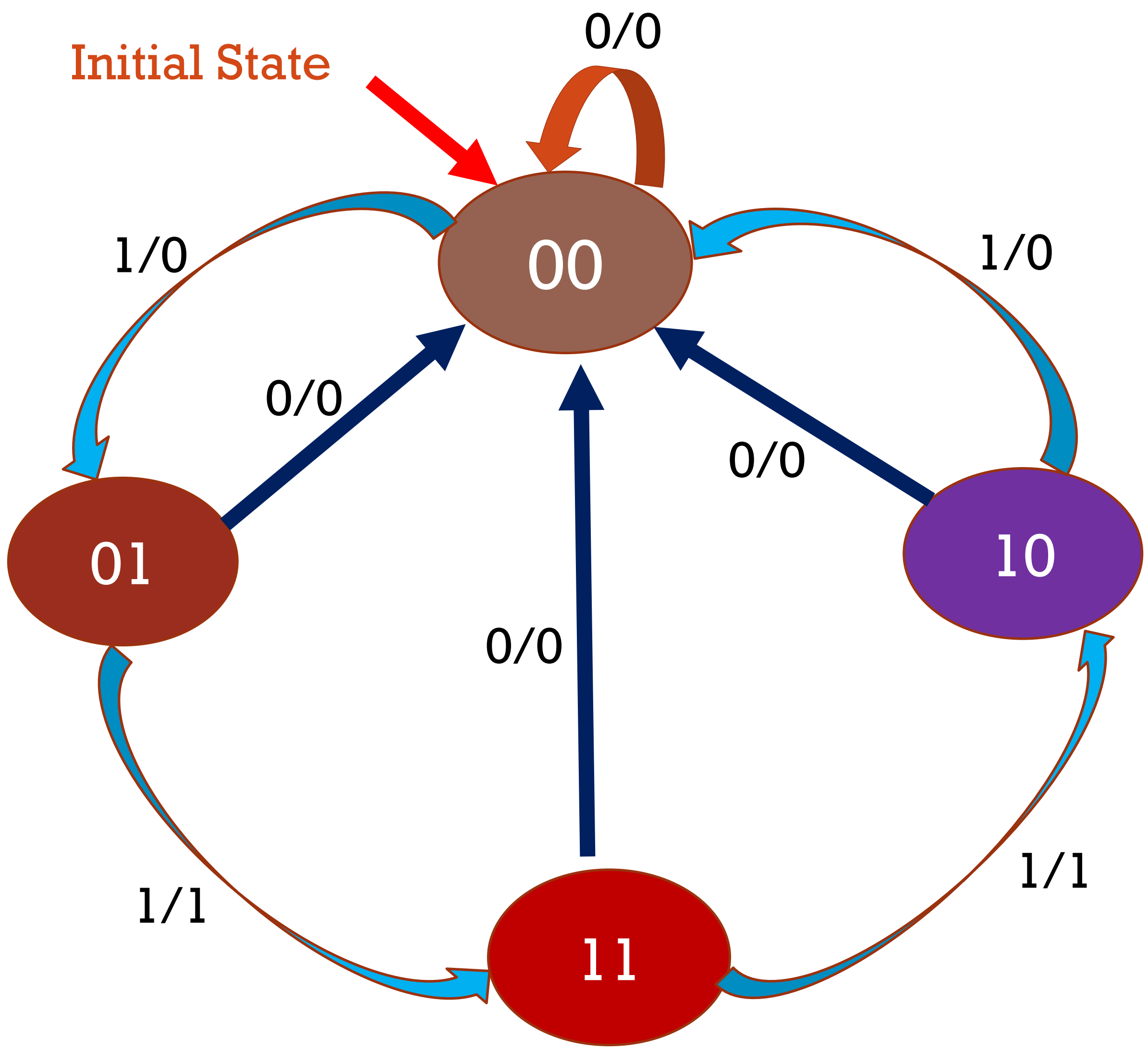
- INPUT CASE :

$$I/P = \begin{cases} 1 & \text{Then Keep Counting} \\ 0 & \text{Then Stop Counting And Return To The Initial State} \end{cases}$$

- OUTPUT CASE :

$$O/P = \begin{cases} 0 & \text{In Case Of Entering And Exiting The Initial State} \\ 1 & \text{Otherwise} \end{cases}$$

STATE DIAGRAM



GRAY CODE
00
01
11
10

STATE TRANSITION TABLE AND LOGIC MINIMIZATION

PS Q_A Q_B	I/P	NS Q_A Q_B D_A D_B	O/P
00	0	00	0
00	1	01	0
01	0	00	0
01	1	11	1
10	0	00	0
10	1	00	0
11	0	00	0
11	1	10	1

Q_AQ_B

I/P

00	00	00	00
01	11	10	00

D_A Map

			Q_A	Q_A
	0	0	0	0
I/P	0	1	1	0
		Q_B	Q_B	

$$D_A = (Q_B) \cdot (I/P)$$

D_B Map

			Q_A	Q_A
	0	0	0	0
I/P	1	1	0	0
		Q_B	Q_B	

$$D_B = (\overline{Q_A}) \cdot (I/P)$$

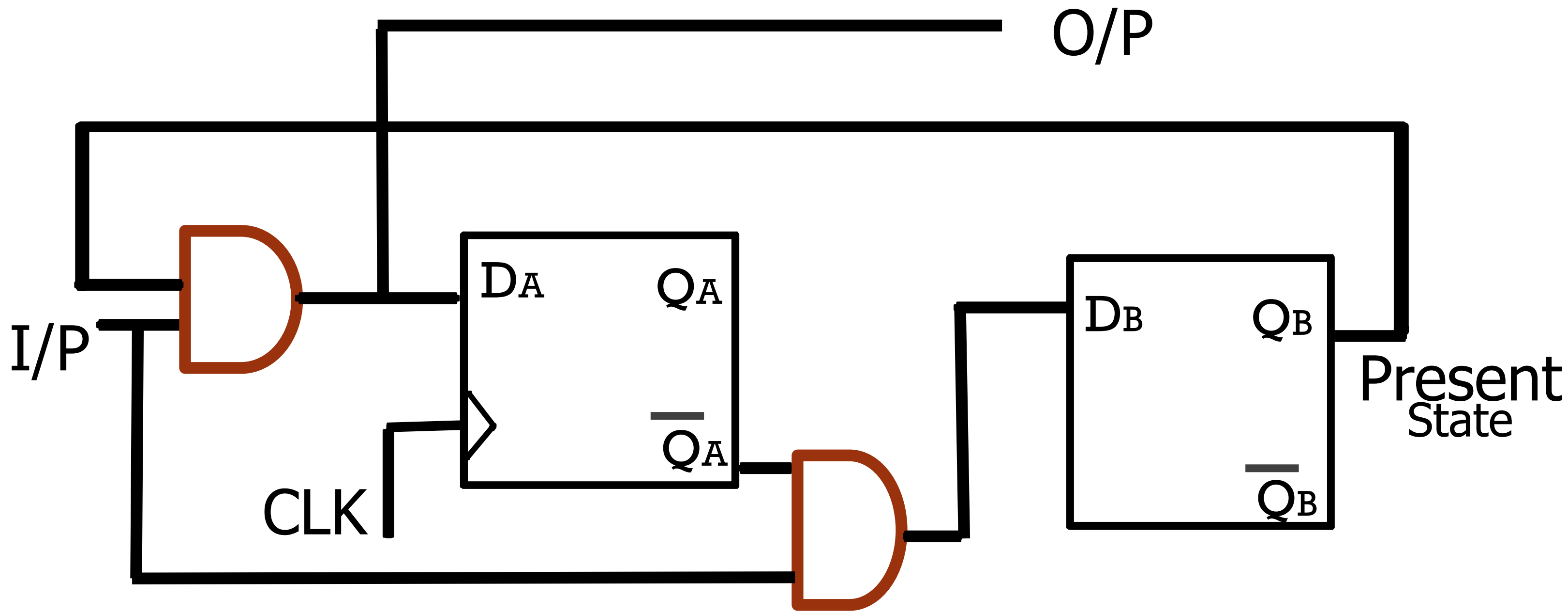
OUTPUT map is same as D_A map here

Therefore,

$$O/P = (Q_B) \cdot (I/P)$$

IMPLIMENTATION OF THE DESIGN

$D_A = (Q_B).(I/P)$
$D_B = (\overline{Q}_A).(I/P)$
$O/P = (Q_B).(I/P)$




Thank
You