Abdul Rehman **email id:- abdul@northern.edu.pk** **Whatsapp#03087792217**

Objectives:

The objective for this week will be understanding the designing of basic sequential circuits along with finite state machines

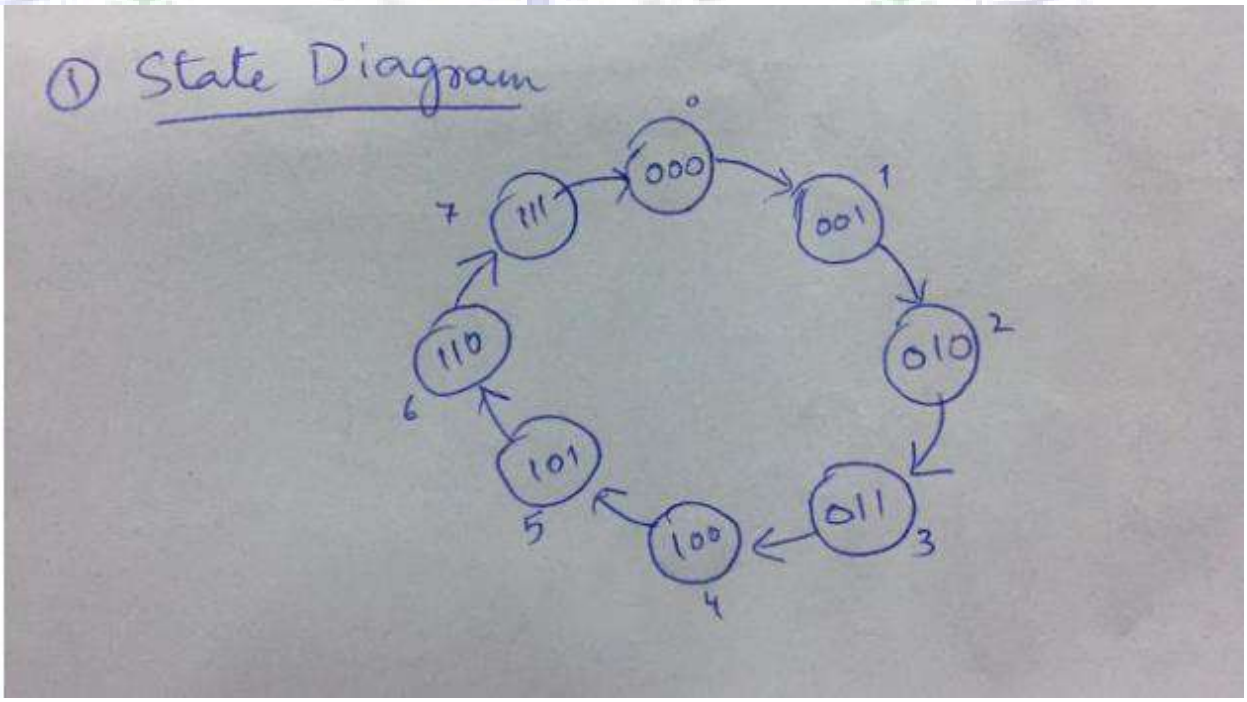
- Sequential circuit designing
- counters
- Finite state machines

LECTURE-27-28

Example#1:

Design a 3 bit counter that will count from 0 to 7 and then it will return back to 0.

State Diagram:



State Table:

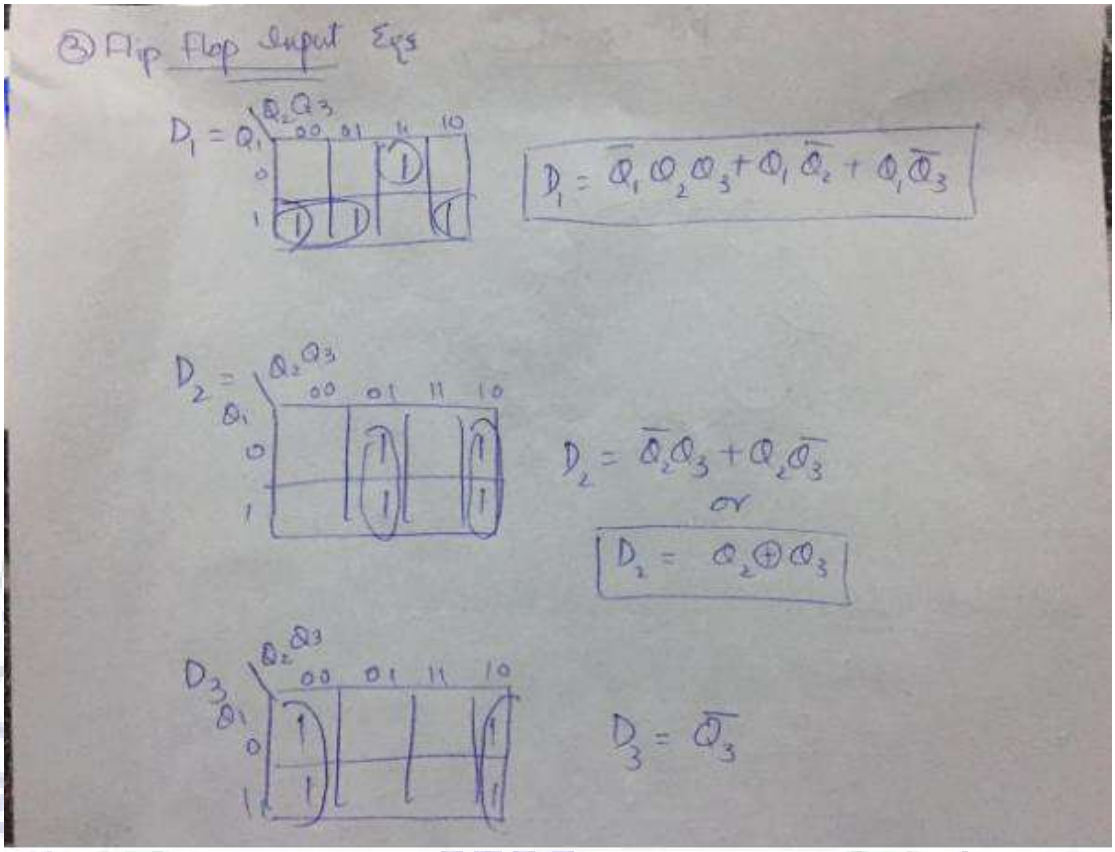
Abdul Rehman **email id:- abdul@northern.edu.pk** **Whatsapp#03087792217**

② State Table

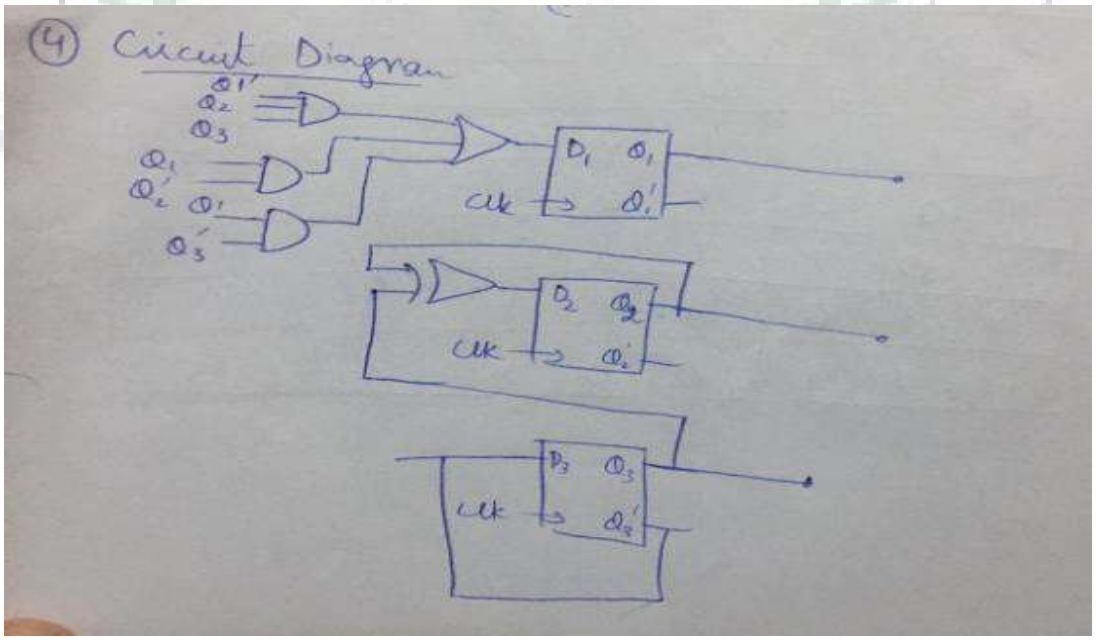
Current State			Next State			Flip-Flop Inputs		
Q_1	Q_2	Q_3	Q_1	Q_2	Q_3	D_1	D_2	D_3
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0

Flip Flop Input Equations:

Abdul Rehman **email id:- abdul@northern.edu.pk** **Whatsapp#03087792217**



Circuit Diagram:



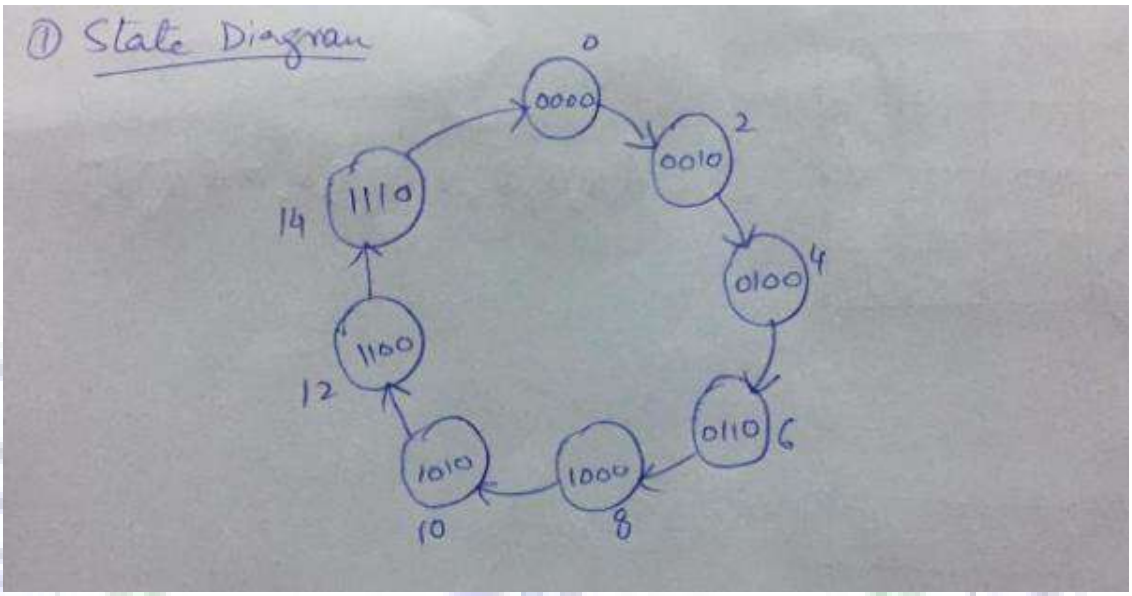
Example#2:

Digital Logic Design (ESC-151)

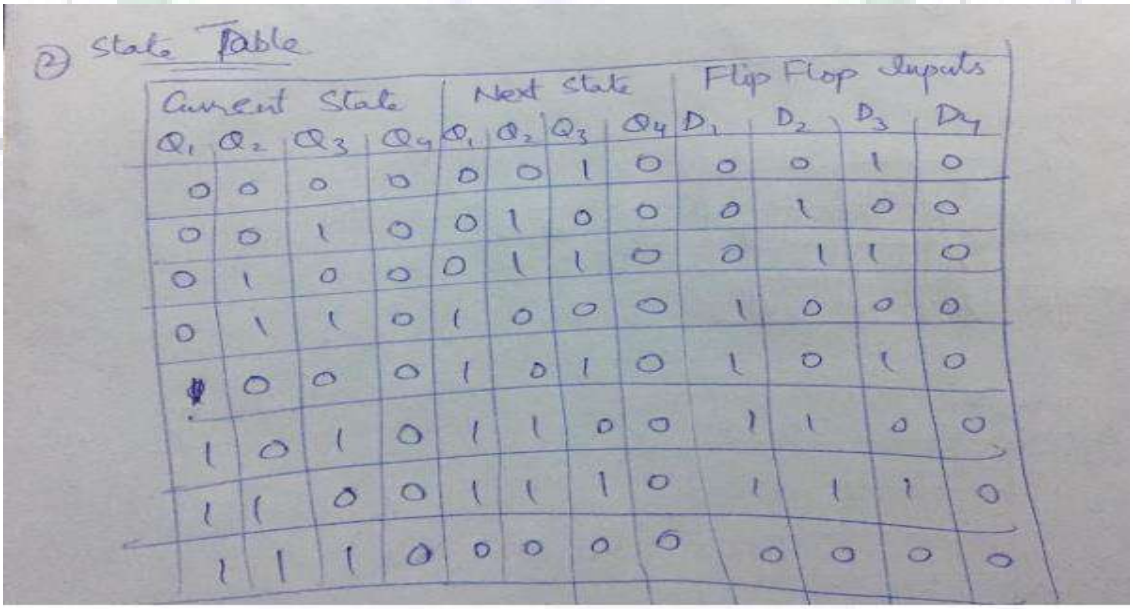
Abdul Rehman **email id:- abdul@northern.edu.pk** **Whatsapp#03087792217**

Design a 4 bit counter that will do a count of 2 each time.

State Diagram:



State Table:



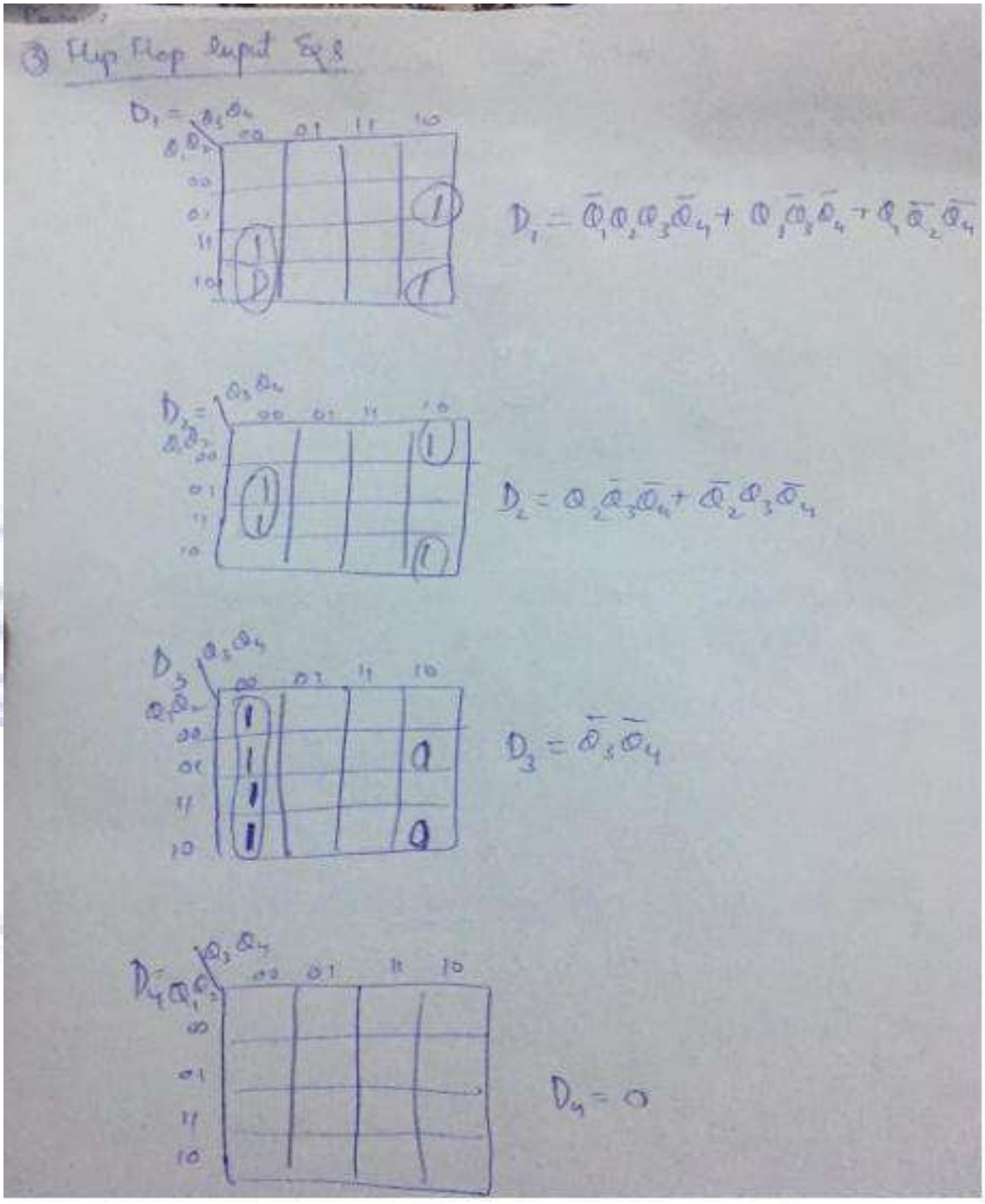
② State Table

Current State				Next State				Flip Flop Inputs			
Q_1	Q_2	Q_3	Q_4	Q_1	Q_2	Q_3	Q_4	D_1	D_2	D_3	D_4
0	0	0	0	0	0	1	0	0	0	1	0
0	0	1	0	0	1	0	0	0	1	0	0
0	1	0	0	0	1	1	0	0	1	1	0
0	1	1	0	1	0	0	0	1	0	0	0
1	0	0	0	1	0	1	0	1	0	1	0
1	0	1	0	1	1	0	0	1	1	0	0
1	1	0	0	1	1	1	0	1	1	1	0
1	1	1	0	0	0	0	0	0	0	0	0

Flip Flop Input Equations:

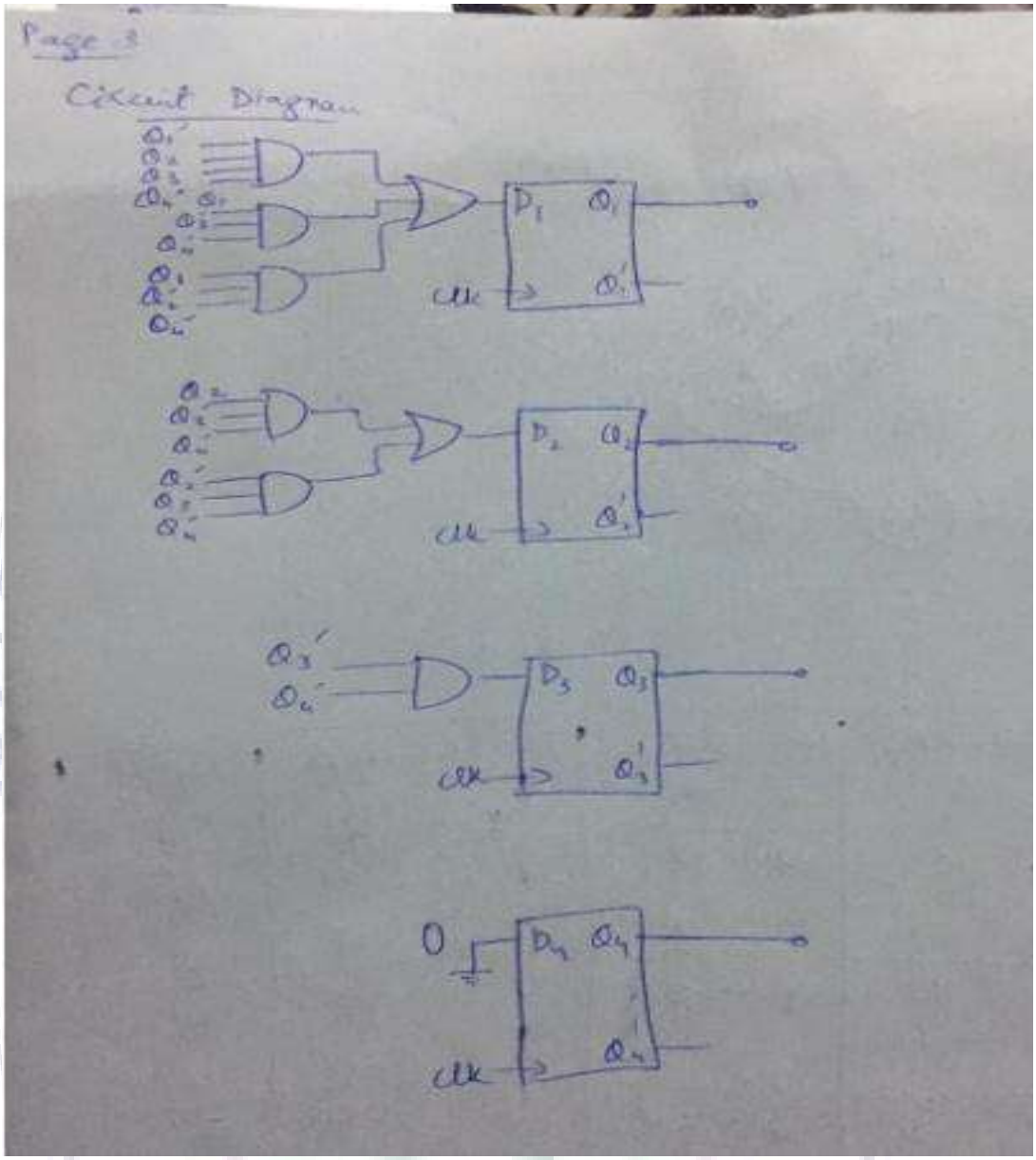
Digital Logic Design (ESC-151)

Abdul Rehman **email id:- abdul@northern.edu.pk** **Whatsapp#03087792217**



Circuit Diagram:

Abdul Rehman **email id:- abdul@northern.edu.pk** **Whatsapp#03087792217**



:Finite State Machines:

A Finite State Machine is a model of computation based on a hypothetical machine made of one or more states. Only one single state of this machine can be active at the

Abdul Rehman **email id:- abdul@northern.edu.pk** **Whatsapp#03087792217**

same time. It means the machine has to transition from one state to another in to perform different actions.

A Finite State Machine is any device storing the state of something at a given time. The state will change based on inputs, providing the resulting output for the implemented changes.

The important points here are the following:

- We have a fixed set of states that the machine can be in
- The machine can only be in one state at a time
- A sequence of inputs is sent to the machine
- Every state has a set of transitions and every transition is associated with an input and pointing to a state

Real world examples

Let's see what could be a Finite State Machine in the real world:

Coin-operated turnstile

- **States:** locked, unlocked
- **Transitions:** pointing a coin in the slot will unlock the turnstile, pushing the arm of the unlocked turnstile will let the costumer pass and lock the turnstile again

Abdul Rehman **email id:- abdul@northern.edu.pk** **Whatsapp#03087792217**

Traffic Light

- **States:** Red, Yellow, Green
- **Transitions:** After a given time, Red will change to Green, Green to Yellow, and Yellow to Red

A Safe

- **States:** Multiple "locked" states, one "unlocked" state
- **Transitions:** Correct combinations move us from initial locked states to locked states closer to the unlocked state, until we finally get to the unlocked state. Incorrect combinations land us back in the initial locked state

Types of FSM:

1. Mealy state machine
2. Moore state machine

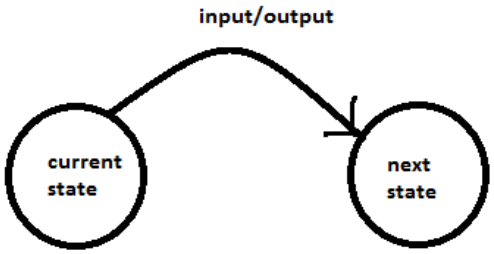
Difference between mealy and moore state machine:

Digital Logic Design (ESC-151)

Abdul Rehman **email id:- abdul@northern.edu.pk** **Whatsapp#03087792217**

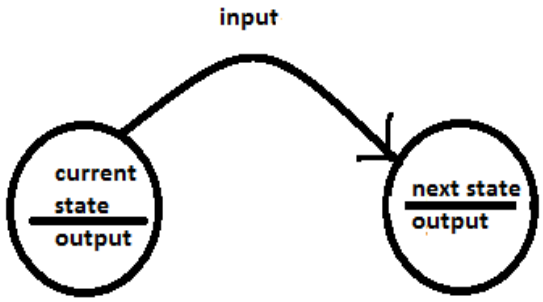
Mealy state machine

Output depends both upon the present state and the present input
Generally, it has fewer states than Moore Machine.
The value of the output function is a function of the transitions and the changes, when the input logic on the present state is done.
Mealy machines react faster to inputs. They generally react in the same clock cycle.



Moore state machine

Output depends only upon the present state.
Generally, it has more states than Mealy Machine.
The value of the output function is a function of the current state and the changes at the clock edges, whenever state changes occur.
In Moore machines, more logic is required to decode the outputs resulting in more circuit delays. They generally react one clock cycle later.



Digital Logic Design (ESC-151)

Abdul Rehman **email id:- abdul@northern.edu.pk** **Whatsapp#03087792217**

Reference Material:

Lesson plans

Text book: Digital design by Moris Mano 4th edition

Also see digital fundamentals 8th ed by Floyd

LAB Work:

Design the circuit of the mealy state machine example on proteus and send screenshots to Mr Hamza Sakhi via email at **hamzasakhic97@gmail.com**