✡ Finite State Machine (Pg 302)

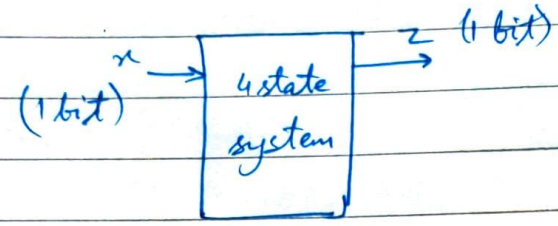
- FSM: * A machine/system that goes through a finite/fixed no. of states.
  * It has a fixed no. of i/p o/p combinations

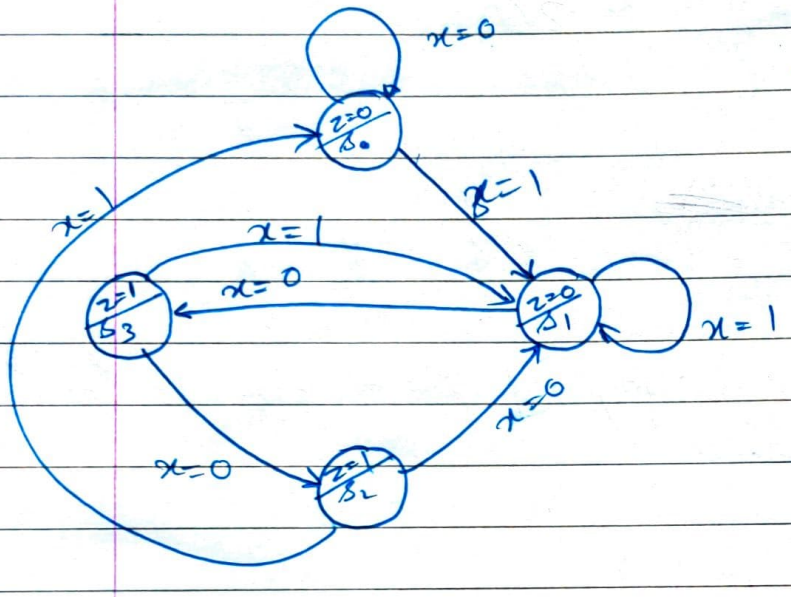
- Typical uses → control, monitor, calculate

- FSM is a <u>way of designing</u> circuits.

- Sequential <u>circuits</u> / system : Counters
  Traffic Lights
  digital lock

* Example:

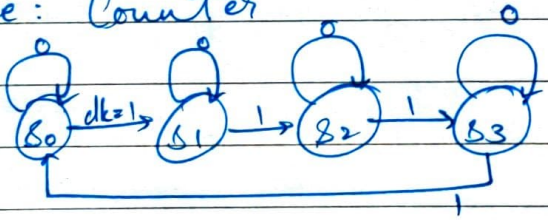




★ Next state $\propto$ f (i/p, current state)

	A	B	→ state variables
δ_0	0	0	
δ_1	0	1	
δ_2	1	0	
δ_3	1	1	

* Example: Counter

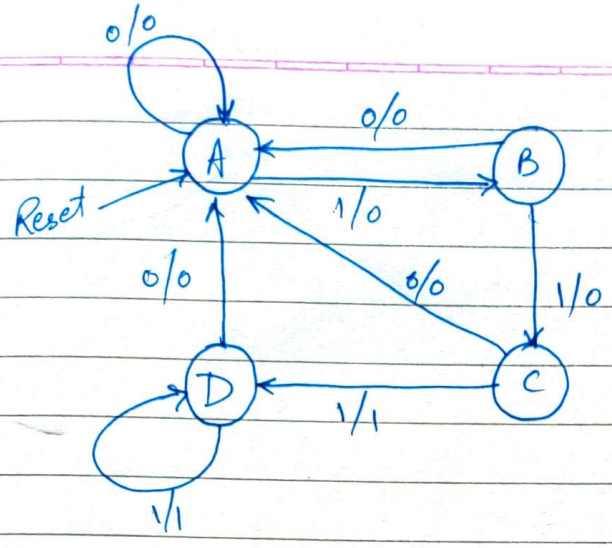


★ Next Step $\propto$ f (i/p, current state)

* FSM is just a way at looking at the design.

★ FSM State Diagram
* Eg: 3 state FSM: Sa, Sb, Sc

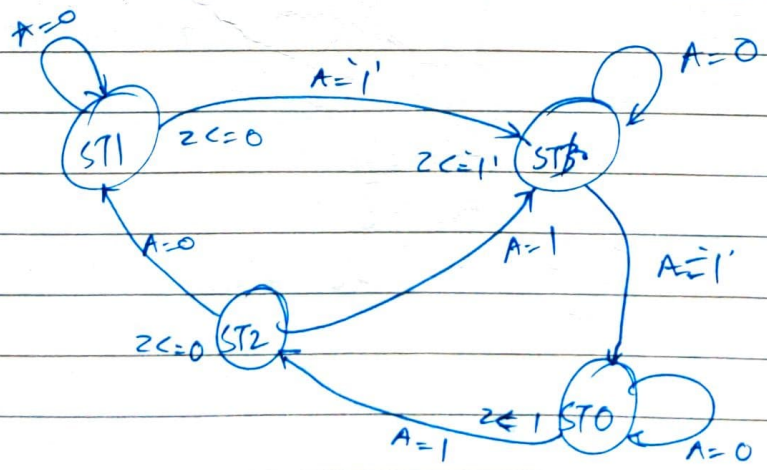
★ Eg: Design a FSM to detect 3 or more 1's in a serial bit stream the bits are applied serially in synchronism with a clock. The o/p will become '1' whenever it detects 3 or more consecutive 1's in the stream.



Reset	PS	i/p	NS	o/p
1	–	–	A	0
0	A	0	A	0
0	A	1	B	0
0	B	0	A	0
0	B	1	C	0
0	C	0	A	0
0	C	1	D	1
0	D	0	A	0
0	D	1	D	1

☆ Moore FSM (Pg 308)

* o/p $\propto$ { (present state only)

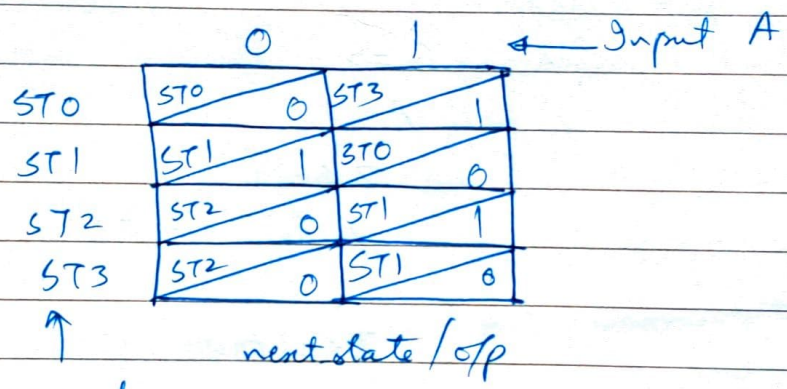


```
library
use

entity MOORE_FSM is
    port (A, clock : in bit;
          Z : out std_logic);
end MOOREFSM;

architecture FSM_Example of MOORE_FSM is
    type state_type is (ST0, ST1, ST2, ST3);
    signal Moore_state : state_type;
begin
    process (clock)
    begin
        if clock = '0' then  -- falling edge
            case Moore_state is
                when ST0 =>
                    Z <= '1';
                    if A = '1' then
                        Moore_State <= ST2;
                    end if;
                when ST1 =>
                    Z <= '0'
                    if A = '1' then
                        Moore_state <= ST3;
                    end if;
                when ST2 =>
                    Z <= '0';
                    if A = '0' then
                        Moore_state <= ST1;
                    else
                        ---- <= ST3;
                    end if;
```

```
        when ST3 =>
            z <= '1';
            if A = '1' then
                Moore_state <= ST0;
            end if;
        end case;
    end process;
end FSM_Example;
```

☆ Mealy FSM (Pg 310)

* o/p is f(i/p, present state)



Q) Draw the state transition diagram for above.

* HDL Code:

```
library
use

entity Mealy_FSM is
    port(a, clock : in bit;
         z : out std_logic);
end Mealy_FSM;
```

```
architecture FSM_ex of Mealy_FSM is
  type Mealy_Type is (ST0, ST1, ST2, ST3);
  signal P_state, N_state : Mealy_Type;
begin
SEQ_Part: process (clock)
 begin
    if clock = '0' then        -- falling edge
      P_state <= N_state;
    end if;
end process SEQ_Part;

COMBPart: Process (P_state, A)
 begin
     case P_state is
        when ST0 =>
            if A = '1' then
               Z <= '1';
               N_state <= ST3;
            else
               Z <= '0';
            end if;
        when ST1 =>
            if A = '1' then
               Z <= '0';
               N_state <= ST0;
            else
               Z <= '1';
            end if;
        when ST2 =>
            if A = '0' then
               Z <= '0';
            else
               Z <= '1'; N_State <= ST1;
            end if;
```

```
when ST3 =>
        z <= '0';
        if A = '0' then
            z <= '0';
            N_state <= ST2;
        else
            N_state <= ST1;
        end if;
    end case;
  end process COMBPart;
end FSM_ex;
```

★ Coding ~~Sy~~ Styles for ~~VHDL~~ FSM

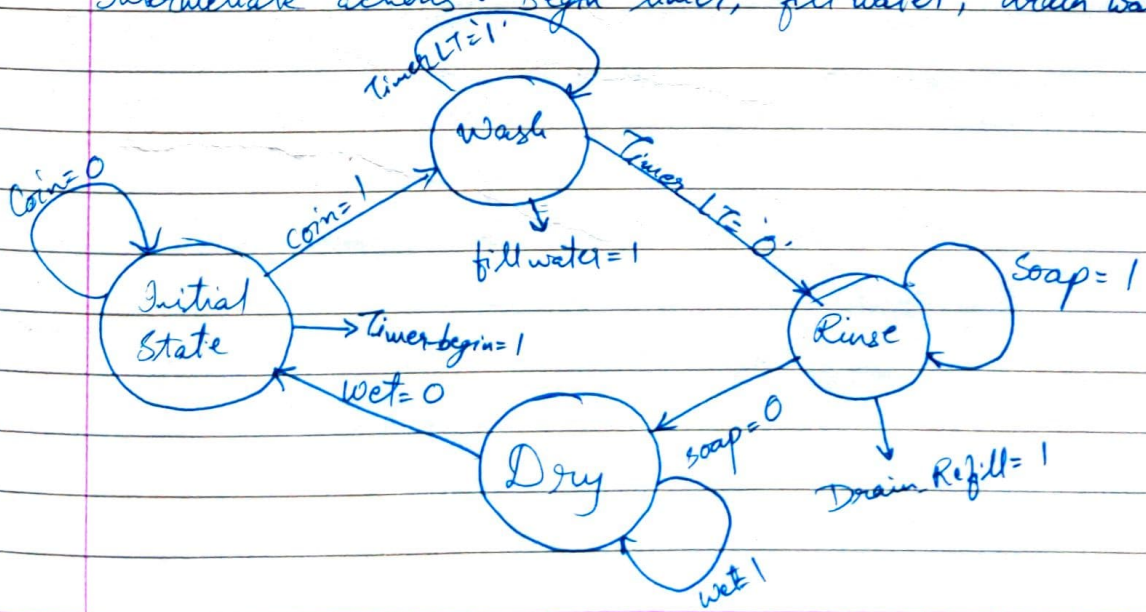
1) Single process → Normal VHDL code
2) Double process → Seperate processes for sequential & combinational logic.
3) Triple process → 1 Sequential & 2 combinational processes.

★ Example: Washing Machine

States: Wash, Rinse, Dry

I/p: Coin

Intermediate actions: Begin timer, fill water, drain water,



Date:

Washer- Dryer

I/ps: coin i/p
Soap present
Timer of 30 min
Wetness Indicator

O/p: Begin timer
fill water
Drain then refill
Drain completely

Q] Write VHDL code for the above system.

☆ Example: Turnstile

CS	I/P	NS	O/P
Locked	Coin	Unlocked	Unlocks the turnstile
Locked	Push	Locked	Nothing, You are locked
Unlocked	Coin	Unlocked	Nothing, You have wasted a coin
Unlocked	Push	Locked	When you have pushed through, locks the turnstile.