VHDL - FSM

2015.07.22

강의자 **:23**기 백두현

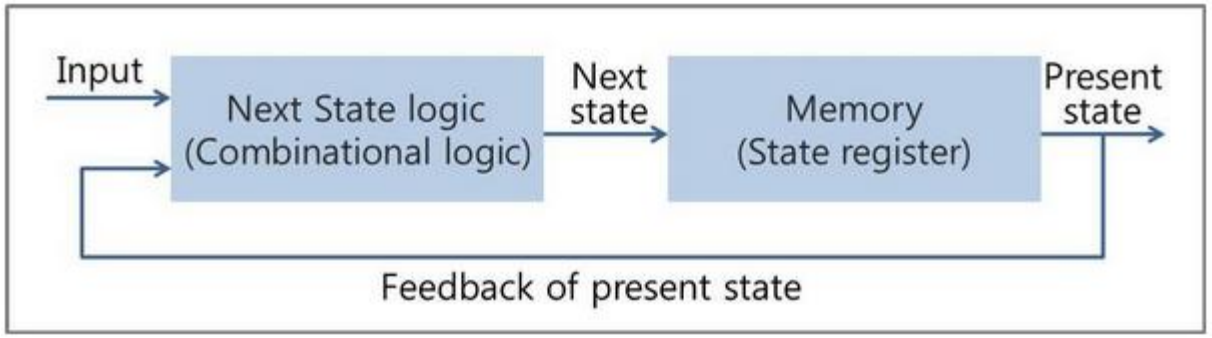
BARAM
KW univ
New generation of Robotics

목차

- FSM이란?
- Moore machine
- Mealy machine
- 과제 : 신호등

FSM 이란?

- **FSM(Finite State Machine)**



"Sequential하게 Finite한 개수의 State를 State Transition하는 Machine이다."
(순차적으로 State 변이하는 유한한 개수의 State로 이루어진 장치.)

FSM 이란?

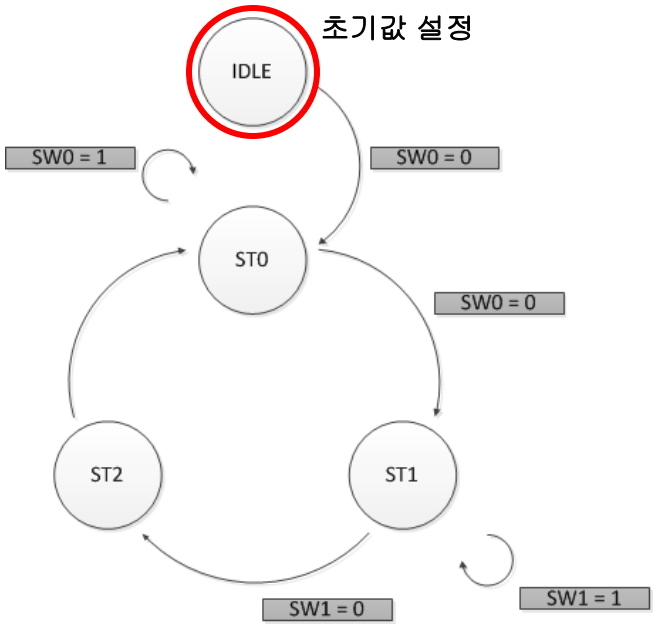
- **Sequential Logic과 Combinational Logic**

  - Combinational Logic(조합회로)
    - Present Output은 Present Input에 관하여만 영향을 미침.

  - Sequential Logic(순차회로)
    - Past Input이 계속 Present Output에 영향을 미침.
    - Sequential Logic = Combinational Logic + Memory

FSM 이란?

State Diagram

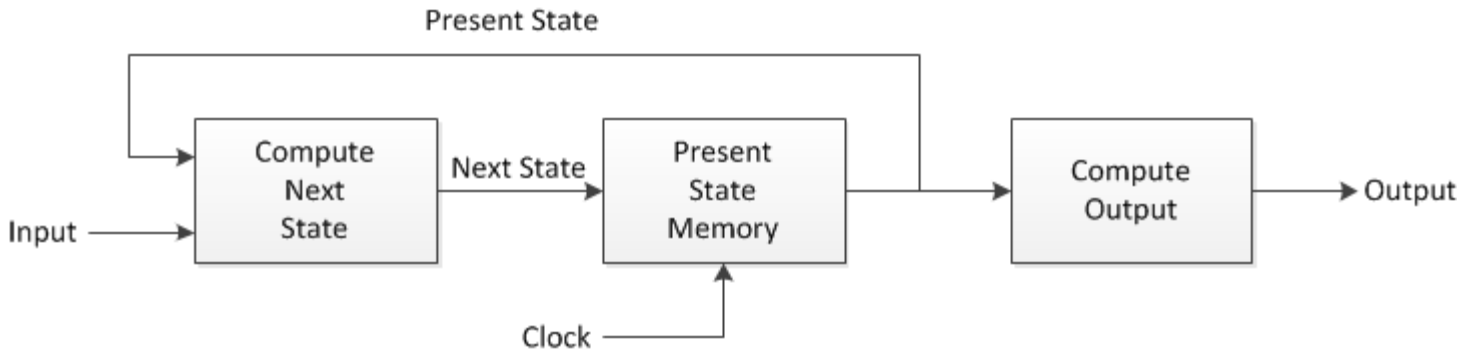


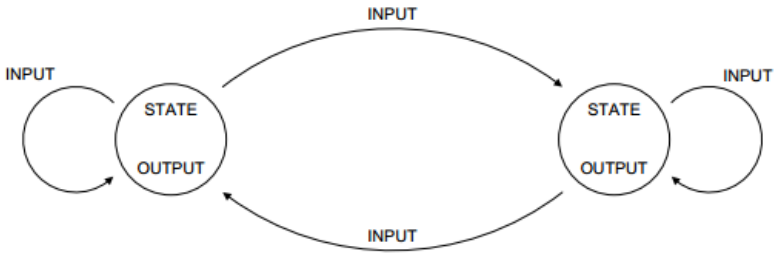
- State간의 이동을 나타낸다.
- 해당 State에서 취할 행동을 결정한다.

좋은 State Machine 일수록 State가 간결해 진다.
=> System의 소요시간이 줄어든다.

Moore Machine

- Moore Machine - Output only depends upon Present State.
  - Outputs are unconditional(not directly dependent on input signal)



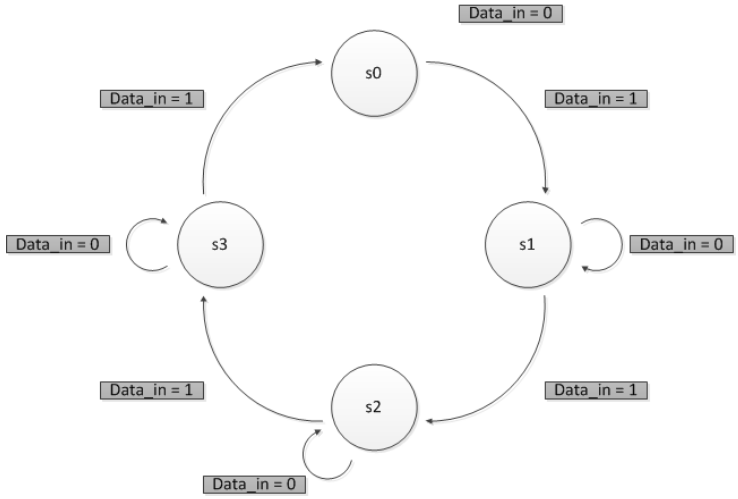


GENERIC MOORE STATE MACHINE

Moore Machine

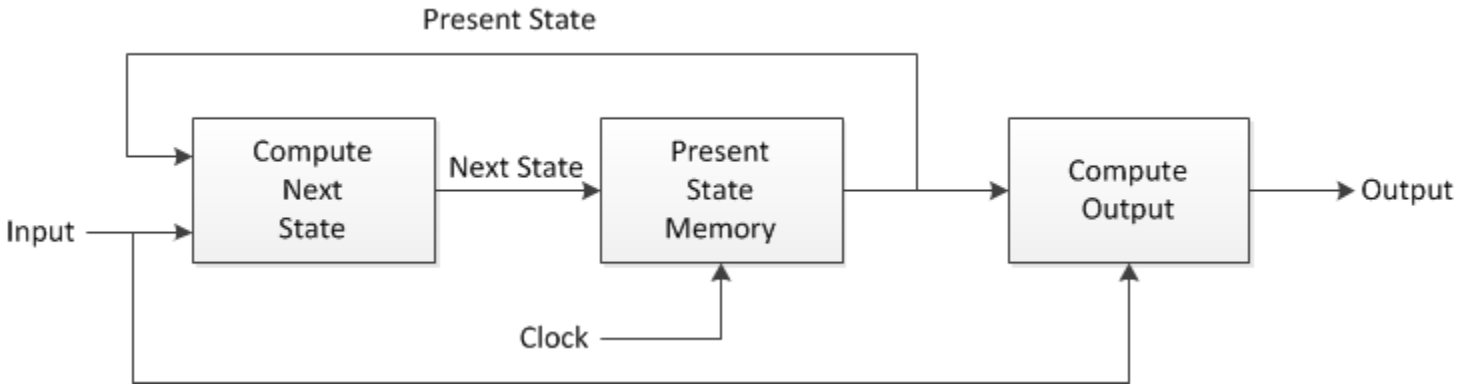
Moore Machine Source(From ALTERA)

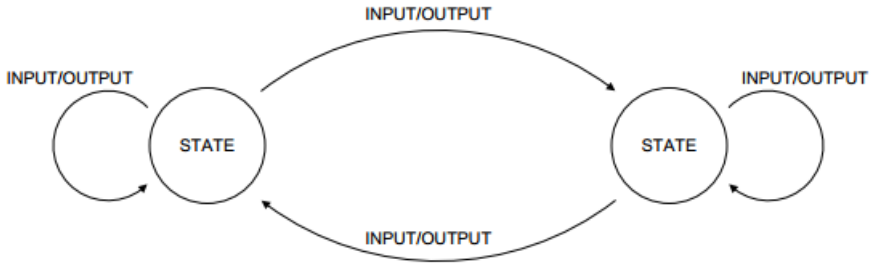
```
1  -- A Moore machine's outputs are dependent only on the current state.
2  -- The output is written only when the state changes.  (State
3  -- transitions are synchronous.)
4
5  library ieee;
6  use ieee.std_logic_1164.all;
7
8  entity moore_4s is
9
10   port(
11      clk       : in std_logic;
12      data_in   : in std_logic;
13      reset     : in std_logic;
14      data_out  : out   std_logic_vector(1 downto 0)
15   );
16
17 end entity;
18
19 architecture rtl of moore_4s is
20
21    -- Build an enumerated type for the state machine
22    type state_type is (s0, s1, s2, s3);
23
24    -- Register to hold the current state
25    signal state   : state_type;
26
27 begin
28    -- Logic to advance to the next state
29    process (clk, reset)
30    begin
31       if reset = '1' then
32          state <= s0;
33       elsif (rising_edge(clk)) then
34          case state is
35             when s0=>
36                if data_in = '1' then
37                   state <= s1;
38                else
39                   state <= s0;
40                end if;
41             when s1=>
42                if data_in = '1' then
43                   state <= s2;
44                else
45                   state <= s1;
46                end if;
47             when s2=>
48                if data_in = '1' then
49                   state <= s3;
50                else
51                   state <= s2;
52                end if;
53             when s3 =>
54                if data_in = '1' then
55                   state <= s0;
56                else
57                   state <= s3;
58                end if;
59          end case;
60       end if;
61    end process;
62
63    -- Output depends solely on the current state
64    process (state)
65    begin
66
67       case state is
68          when s0 =>
69             data_out <= "00";
70          when s1 =>
71             data_out <= "01";
72          when s2 =>
73             data_out <= "10";
74          when s3 =>
75             data_out <= "11";
76       end case;
77    end process;
78
79 end rtl;
80
```



Mealy Machine

- **Mealy Machine – Outputs determined by the current state and the current inputs.**
  - Outputs are conditional(directly dependent on input signals)





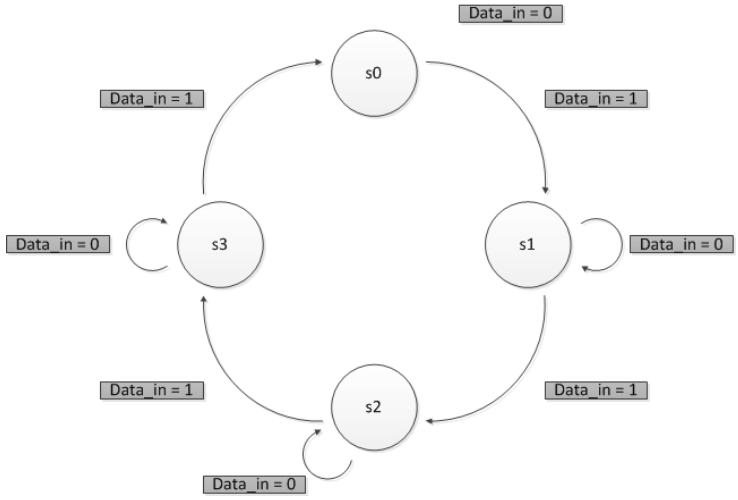
GENERIC MEALY STATE MACHINE

Mealy Machine

Mealy Machine Source(From ALTERA)

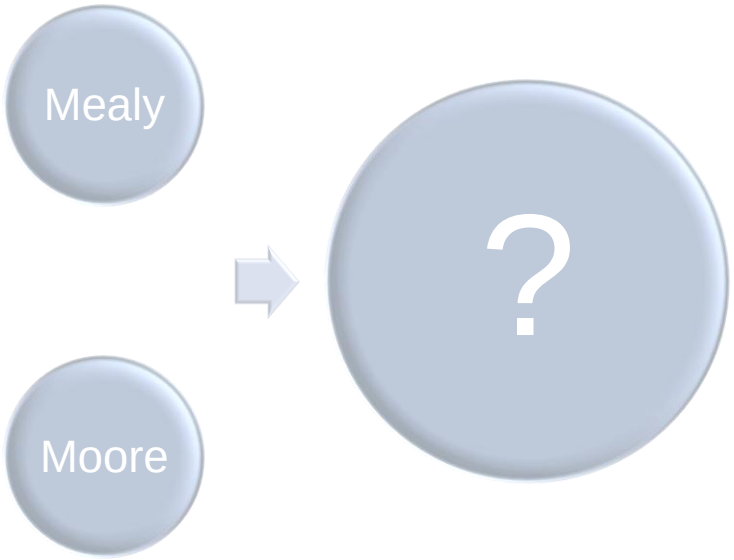
```
1  -- A Mealy machine has outputs that depend on both the state and
2  -- the inputs. When the inputs change, the outputs are updated
3  -- immediately, without waiting for a clock edge.  The outputs
4  -- can be written more than once per state or per clock cycle.
5
6  library ieee;
7  use ieee.std_logic_1164.all;
8
9  entity mealy_4s is
10
11    port
12    (
13       clk       : in std_logic;
14       data_in   : in std_logic;
15       reset  : in std_logic;
16       data_out : out std_logic_vector(1 downto 0)
17    );
18
19 end entity;
20
21 architecture rtl of mealy_4s is
22
23    -- Build an enumerated type for the state machine
24    type state_type is (s0, s1, s2, s3);
25
26    -- Register to hold the current state
27    signal state : state_type;
28
29 begin
30    process (clk, reset)
31    begin
32       if reset = '1' then
33          state <= s0;
34       elsif (rising_edge(clk)) then
35          -- Determine the next state synchronously, based on
36          -- the current state and the input
37          case state is
38             when s0=>
39                if data_in = '1' then
40                   state <= s1;
41                else
42                   state <= s0;
43                end if;
44             when s1=>
45                if data_in = '1' then
46                   state <= s2;
47                else
48                   state <= s1;
49                end if;
50             when s2=>
51                if data_in = '1' then
52                   state <= s3;
53                else
54                   state <= s2;
55                end if;
56             when s3=>
57                if data_in = '1' then
58                   state <= s3;
59                else
60                   state <= s1;
61                end if;
62          end case;
63
64       end if;
65    end process;
66
```

```
67    -- Determine the output based only on the current state
68    -- and the input (do not wait for a clock edge).
69    process (state, data_in)
70    begin
71       case state is
72          when s0=>
73             if data_in = '1' then
74                data_out <= "00";
75             else
76                data_out <= "01";
77             end if;
78          when s1=>
79             if data_in = '1' then
80                data_out <= "01";
81             else
82                data_out <= "11";
83             end if;
84          when s2=>
85             if data_in = '1' then
86                data_out <= "10";
87             else
88                data_out <= "10";
89             end if;
90          when s3=>
91             if data_in = '1' then
92                data_out <= "11";
93             else
94                data_out <= "10";
95             end if;
96       end case;
97    end process;
98
99 end rtl;
100
```



BARAM KW univ
New generation of Robotics

Mealy Machine ?? Moore Machine??



- Mealy 또는 Moore Machine의 선택은 시스템에 맞추어 간다.
- 시스템에 따라 Mealy 와 Moore Machine 모두 쓰일 수 있다.
- State Machine의 Source는 규정 되어 있지 않다.