Mooooorrrrr:

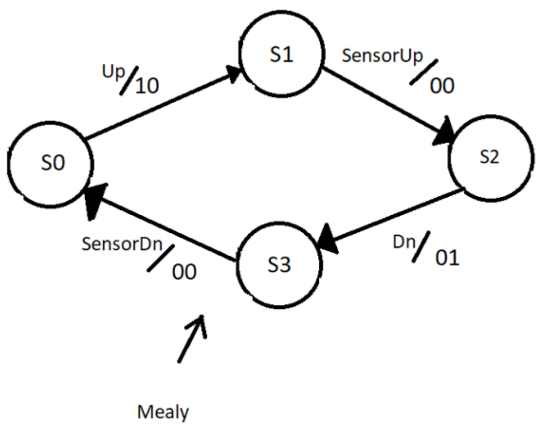
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity FSM is
        port(
                Clk             : in  STD_LOGIC;
                Reset      : in       STD_LOGIC;
                Up          : in       STD_LOGIC;
                Dn          : in       STD_LOGIC;
                SensorUp  : in   STD_LOGIC;
                SensorDn  : in   STD_LOGIC;
                State   : out        STD_LOGIC_VECTOR(3 downto 0);
                MoveUp      : out        STD_LOGIC;
                MoveDn      : out        STD_LOGIC
        );
end FSM;

architecture Behavioral of FSM is
type STATE_TYPE is (S0, S1, S2, S3, S4);
signal CurrentState, NextState : STATE_TYPE;

begin
MEM:
    -- Блок памет на състоянието
    process (Reset, Clk)
        begin
                if (Reset = '1') then
                        CurrentState <= S0;
                elsif (falling_edge(Clk)) then
                        CurrentState <= NextState;
                end if ;
        end process MEM;
```

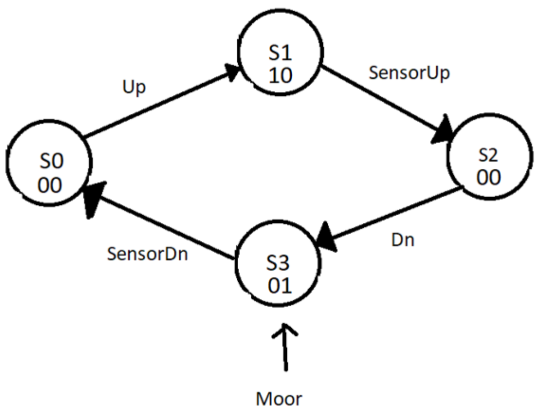
```
NEXT_STATE_LOGIC:
   -- Логика за определяне на следващото състояние
       process (CurrentState, Up, Dn, SensorUp, SensorDn)
       begin
      NextState <= CurrentState;
       MoveUp <=  '0';
      MoveDn <=  '0';

      case CurrentState is
         when S0=>
            if (Up = '1') then
               NextState <= S1;
            end if;
         when S1=>
            if (SensorUp = '1') then
               NextState <= S2;
            end if;
         when S2=>
            if (Dn = '1') then
               NextState <= S3;
            end if;
         when S3 =>
            if (SensorDn = '1') then
               NextState <= S0;
            end if;
         when others=>
            NextState <= S0;
      end case;

       end process NEXT_STATE_LOGIC;

       -- Логика (изходи)
```
MoveUp <= '1' when CurrentState = S1 else '0';
MoveDn <= '1' when CurrentState = S3 else '0';

```
        -- Извеждане на състоянието
    with CurrentState select
                State <= "0000" when S0,
                                "0001" when S1,
                                "0010" when S2,
                                "0011" when S3,
                                "0100" when S4,
                                "1111" when others;
end Behavioral;
```

Mealy:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity FSM is
        port(
                Clk             : in  STD_LOGIC;
                Reset      : in      STD_LOGIC;
                Up          : in      STD_LOGIC;
                Dn          : in      STD_LOGIC;
                SensorUp  : in   STD_LOGIC;
                SensorDn  : in   STD_LOGIC;
                State  : out       STD_LOGIC_VECTOR(3 downto 0);
                MoveUp      : out       STD_LOGIC;
                MoveDn      : out       STD_LOGIC
        );
end FSM;

architecture Behavioral of FSM is
type STATE_TYPE is (S0, S1, S2, S3, S4);
signal CurrentState, NextState : STATE_TYPE;
signal asdaf: LOGIC_VECTOR(3 downto 0);

begin
MEM:
   -- Блок памет на състоянието
   process (Reset, Clk)
       begin
               if (Reset = '1') then
                       CurrentState <= S0;
               elsif (falling_edge(Clk)) then
                       CurrentState <= NextState;
               end if ;
       end process MEM;

NEXT_STATE_LOGIC:
   -- Логика за определяне на следващото състояние
       process (CurrentState, Up, Dn, SensorUp, SensorDn)
       begin
      NextState <= CurrentState;
      MoveUp <=  '0';
      MoveDn <=  '0';
      case CurrentState is
         when S0=>
            if (Up = '1') then
               NextState <= S1;

               MoveUp <=  '1';
               MoveDn <=  '0';
            end if;
         when S1=>
            if (SensorUp = '1') then
               NextState <= S2;

               MoveUp <=  '0';
               MoveDn <=  '0';
            end if;
         when S2=>
            if (Dn = '1') then
               NextState <= S3;

               MoveUp <=  '0';
               MoveDn <=  '1';
            end if;
         when S3 =>
            if (SensorDn = '1') then
               NextState <= S0;

               MoveUp <=  '0';
               MoveDn <=  '0';
            end if;
         when others=>
            NextState <= S0;
      end case;

      end process NEXT_STATE_LOGIC;
      -- Извеждане на състоянието
   with CurrentState select
            State <= "0000" when S0,
                         "0001" when S1,
                         "0010" when S2,
                         "0011" when S3,
                         "0100" when S4,
                         "1111" when others;
end Behavioral;
```


S1
S0
S2
S3
Up/10
SensorUp/00
Dn/01
SensorDn/00
Mealy


S1
10
S0
00
S2
00
S3
01
Up
SensorUp
Dn
SensorDn
Moor