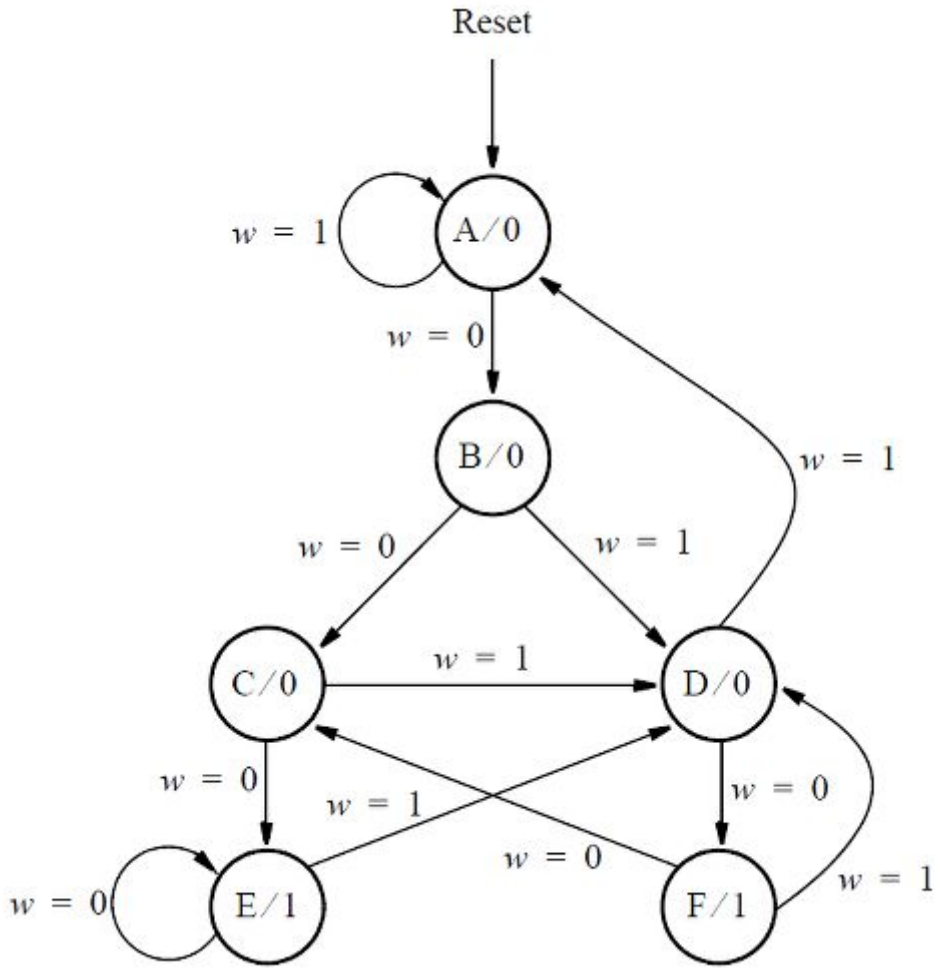
Exams/m2014 q6b

Consider the state machine shown below, which has one input *w* and one output *z*.



Assume that you wish to implement the FSM using three flip-flops and state codes *y[3:1]* = 000, 001, ... , 101 for states A, B, ... , F, respectively. Show a state-assigned table for this FSM. Derive a next-state expression for the flip-flop *y[2]*.

Implement just the next-state logic for *y[2]*. (This is much more a FSM question than a Verilog coding question. Oh well.)

這次把curr_state和next_state的表寫出來
找到所有Y2為1的組合 用力刻下來。

```
module top_module (
    input [3:1] y,
    input w,
    output Y2);

//       next_state
//state  w=0   w=1   output
//-------------------------
//A 000  001   000     0
```

```
//B 001  010   011     0
//C 010  100   011     0
//D 011  101   000     0
//E 100  100   011     1
//F 101  010   011     1
   assign Y2 = w &((y==3'b001)|(y==3'b010)|(y==3'b100)|(y==3'b101))|
      ~w &((y==3'b001)|(y==3'b101));

endmodule
```