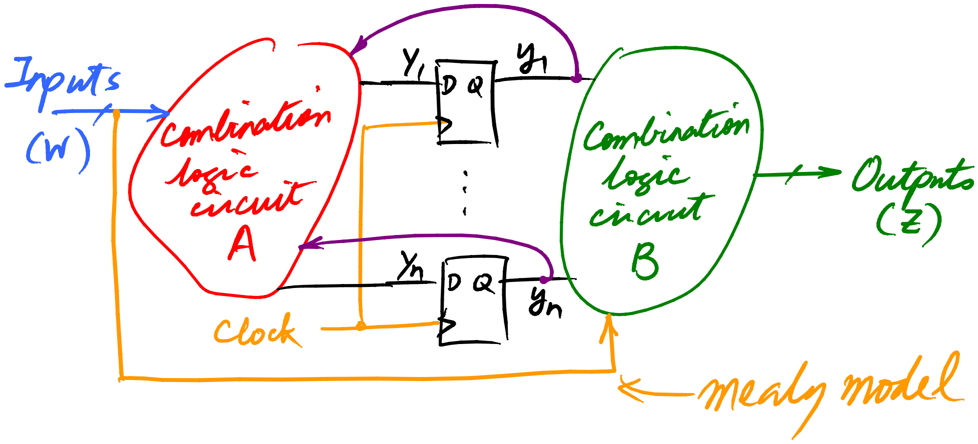
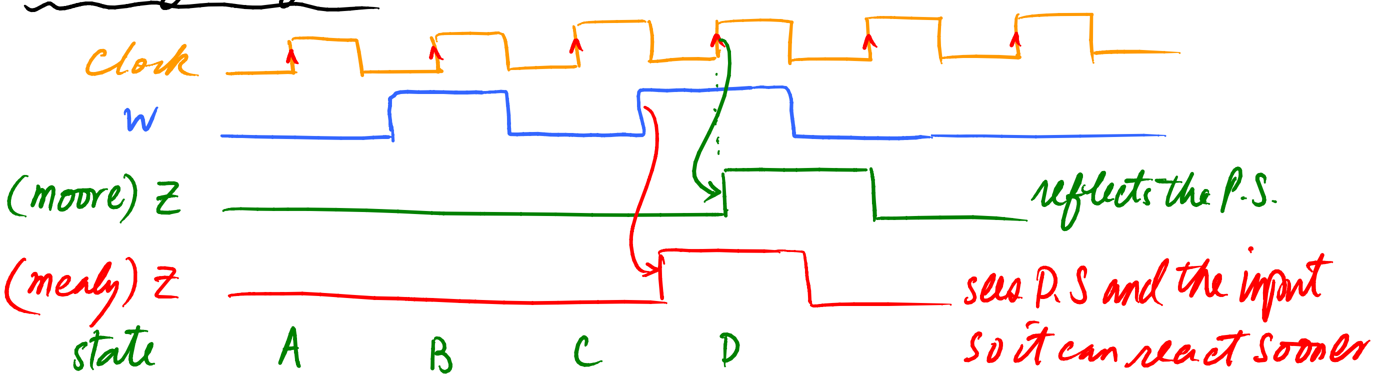
6.3 Mealy model of FSM

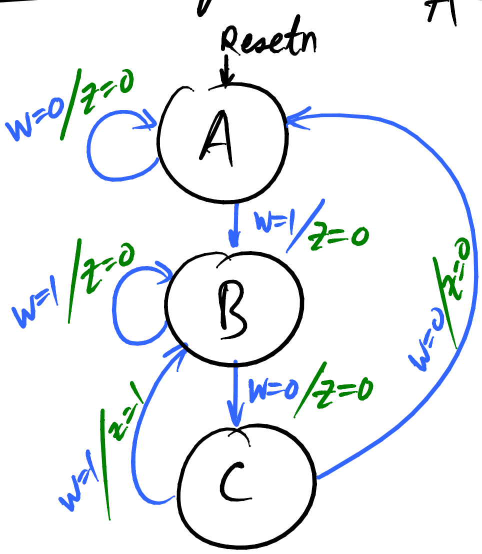


⇒ "101" pattern example

Timing Diagram



State-diagram



A – waiting for "1"

B – seeing the first "1"

C – seeing the "1" followed by "0"

State Table

P.S	N.S w=0	N.S w=1	Output w=0	Output w=1
A	A	B	0	0
B	C	B	0	0
C	A	B	0	1

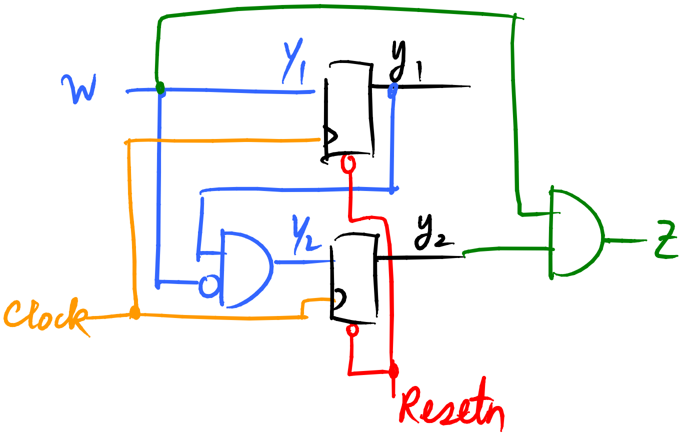
State-assignment

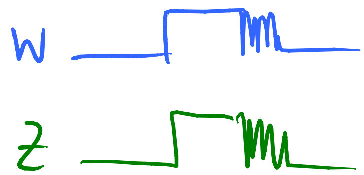
$y_2 y_1$	w=0 $Y_2 Y_1$	w=1 $Y_2 Y_1$	w=0 z	w=1 z
0 0	0 0	0 1	0	0
0 1	1 0	0 1	0	0
1 0	0 0	0 1	0	1
1 1				

use as don't care!

Next-state expressions $Y_2 = \bar{w} y_1$, $Y_1 = w$, output $z = y_2 w$

FSM. cct. diagram

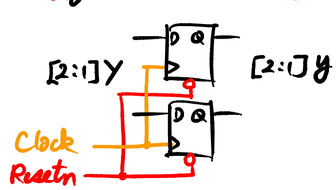




this could happen.

reaction of Mealy model is slightly faster than the Moore model (by approx. 1 clock cycle)

```
module seq101_Mealy(input clock, W, Resetn, output z);
  reg [2:1] y, Y;    // y2, y1 are the present states, Y2, Y1 are the next states
  parameter A=2'b00, B=2'b01, C=2'b10,    // state assignment
  // combinational circuit A, describing state transitions + output
  always@(w, y)
    case (y)
      A: if (!w)        B: if (!w)        C: if (!w)
         begin             begin             begin
           Y=A;              Y=B;              Y=A,
           z=0;              z=0;              z=0;
         end               end               end
         else              else              else
         begin             begin             begin
           Y=B;              Y=C;              Y=B;
           z=0;              z=0;              z=1;
         end               end               end
    endcase
  // FFs describing D-ff's with synchronous reset
  always@(posedge clock)
    if (Resetn==0)
      y<=A;
    else
      y<=Y;
endmodule
```



6.6 State minimization (Partitioning)

We'll use the state table for the "101" pattern recognition example as a starting point. For simplicity we will use A, B, C, D, E, F, G and H instead

	P.S	N.S W=0	N.S W=1	Output Z
A	S_{000}	S_{000}	S_{100}	0
B	S_{001}	S_{000}	S_{100}	0
C	S_{010}	S_{001}	S_{101}	0
D	S_{011}	S_{001}	S_{101}	0
E	S_{100}	S_{010}	S_{110}	0
F	S_{101}	S_{010}	S_{110}	1
G	S_{110}	S_{011}	S_{111}	0
H	S_{111}	S_{011}	S_{111}	0

P.S.	N.S W=0	N.S W=1	Output Z
A	A	E	0
B	A	E	0
C	B	F	0
D	B	F	0
E	C	G	0
F	C	G	1
G	D	H	0
H	D	H	0

0-successors (W=0 column), 1-successors (W=1 column)

Step 1 We start with every state in the partition

$$P_1 = (ABCDEFGH)$$

Step 2 We will divide P_1 into two sub partitions according to z.

$$P_2 = (\underbrace{ABCDEGH}_{z=0})_{①} \; (F)_{②} \quad \to z=1$$

Step 3 We will look at the 0-successors

$$ABCDEGH \xrightarrow[\text{0-succ.}]{} \underbrace{AABBCDD}_{\text{belongs to partition ①}}$$

Step 4 looking at 1-successor

$$ABCDEGH \xrightarrow[\text{1-succ}]{} EEFFGHH$$

partition ①, partition ②

P.S.	N.S W=0	N.S W=1	Output Z
A	A	E	0
B	A	E	0
C	B	F	0
D	B	F	0
E	C	G	0
F	C	G	1
G	D	H	0
H	D	H	0

0-successors (W=0 column), 1-successors (W=1 column)

$$(ABEGH)_{①} \; (F)_{②} \; (CD)_{③}$$

Repeat Step 3 & 4 until no further partion can be found.

$(ABEGH)_{①} \xrightarrow[\text{0-succ}]{} AACDD$

partition ①

partition ③

$(AB)_{①} (F)_{②} (CD)_{③} (EGH)_{④}$

$AB \xrightarrow[\text{0-succ}]{} AA$, $AB \xrightarrow[\text{1-succ}]{} EE$ (no need to divide)

$CD \xrightarrow[\text{0-succ}]{} BB$, $CD \xrightarrow[\text{1-succ}]{} FF$ (no need to divide)

$EGH \xrightarrow[\text{0-succ}]{} CDD$, $EGH \xrightarrow[\text{1-succ}]{} GHH$ (no need to divide)

minimum states = 4.

$AB \rightarrow A$

$CD \rightarrow C$

$EGH \rightarrow E$

$F \rightarrow F$

P.S	N.S $w=0$	N.S $w=1$	Output z
A	A	E	0
C	A	F	0
E	C	E	0
F	C	E	1