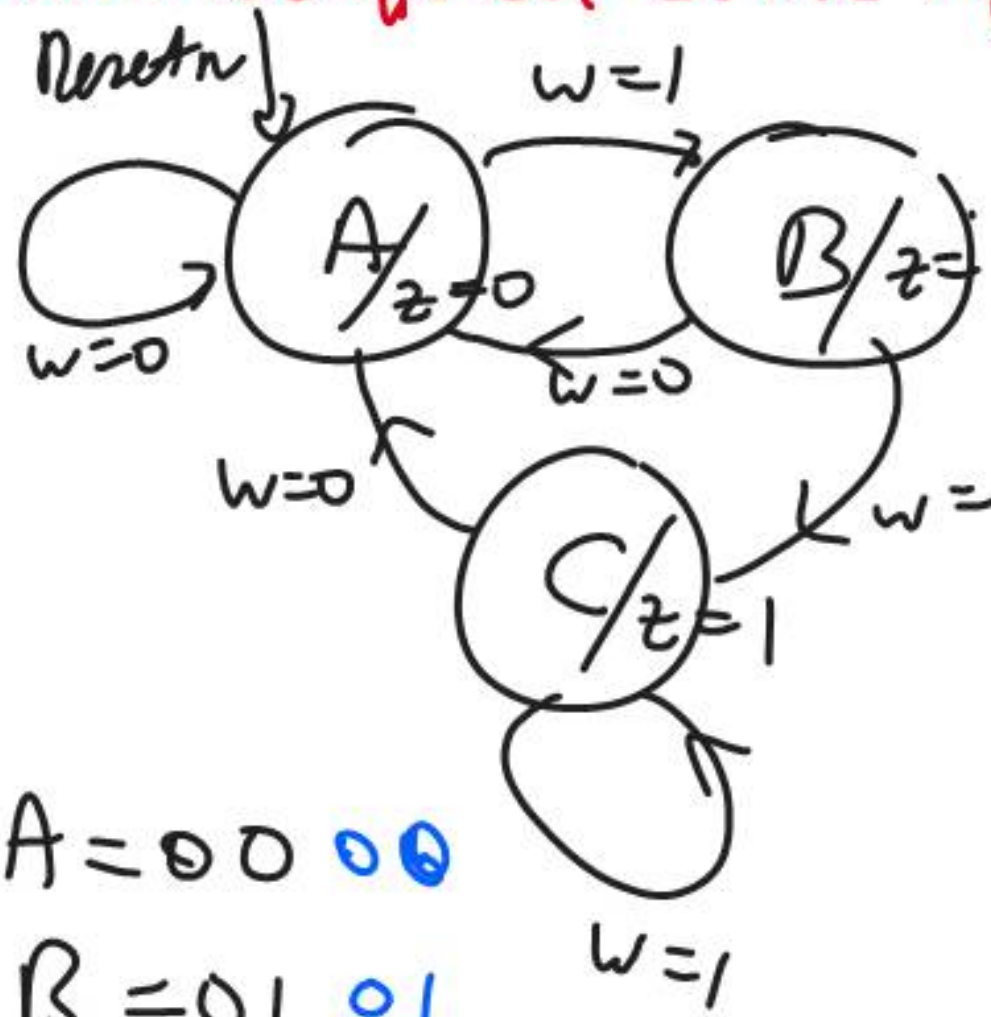
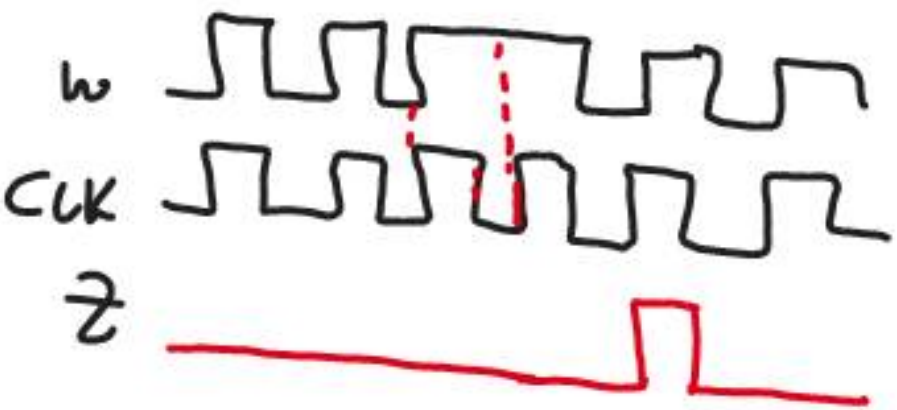
Sequence counter (detector) 12.05.2011

with a different state representation: ©





A = 00 00

B = 01 01

C = 10 11

d = 11 10

Present State		Next State		Output
		w=0	w=1	
	y_2y_1	Y_2Y_1	Y_2Y_1	z
A	00	00	01	0
B	01	00	11	0
C	11	00	11	1
	10	d d	d d	d

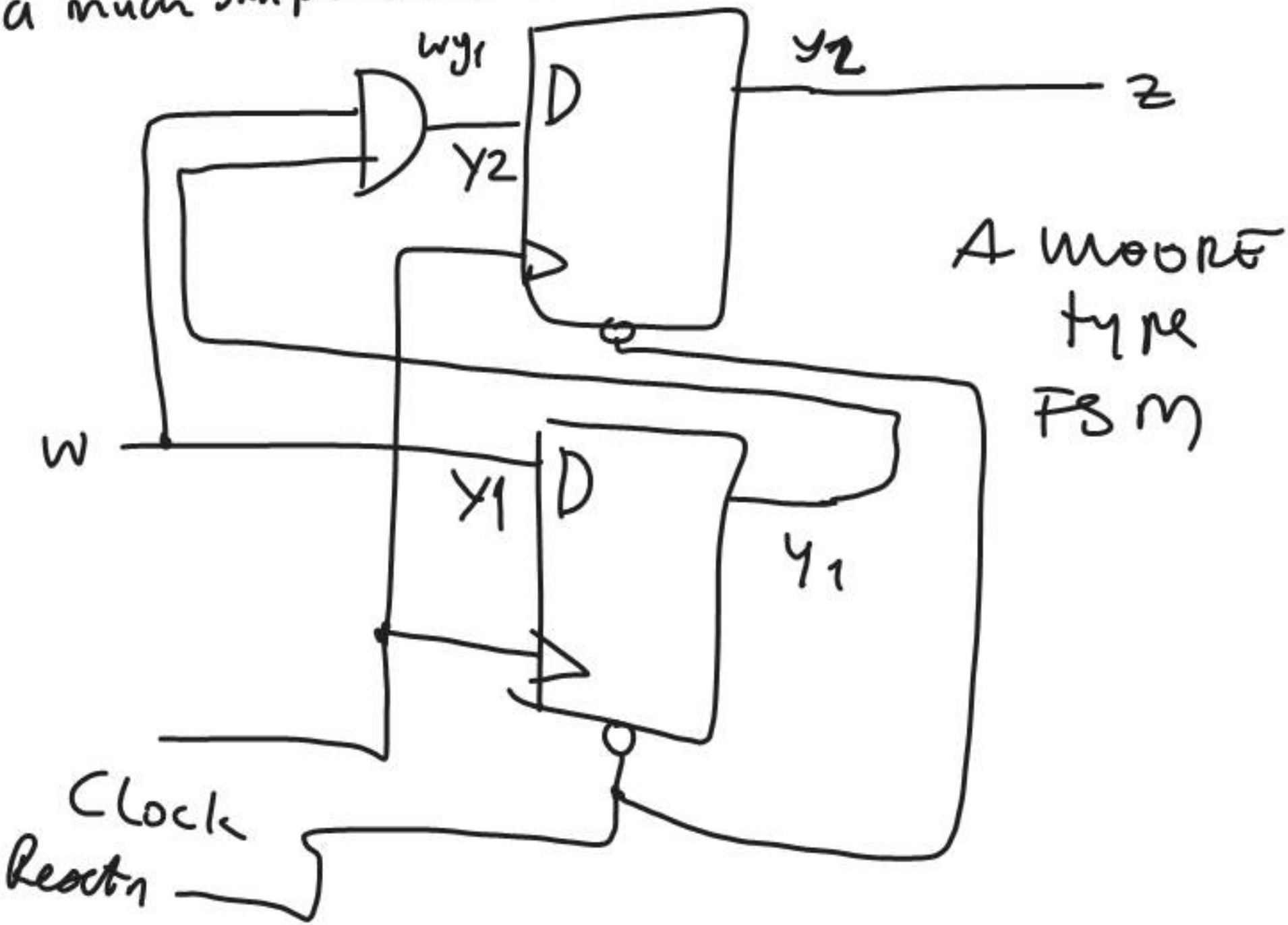
A Moore FSM

$Y_1 = w$

$Y_2 = w y_1$

$z = y_2$

We just changed the states assigned and obtain a much simpler circuit:



One-Hot encoding:

For each state all but one of the state variables are equal to 0.

	3 FFs $y_3 y_2 y_1$
A	0 0 1
B	0 1 0
C	1 0 0

↑ assigned to the states

$$Y_1 = \overline{w}$$

$$Y_2 = w y_1$$

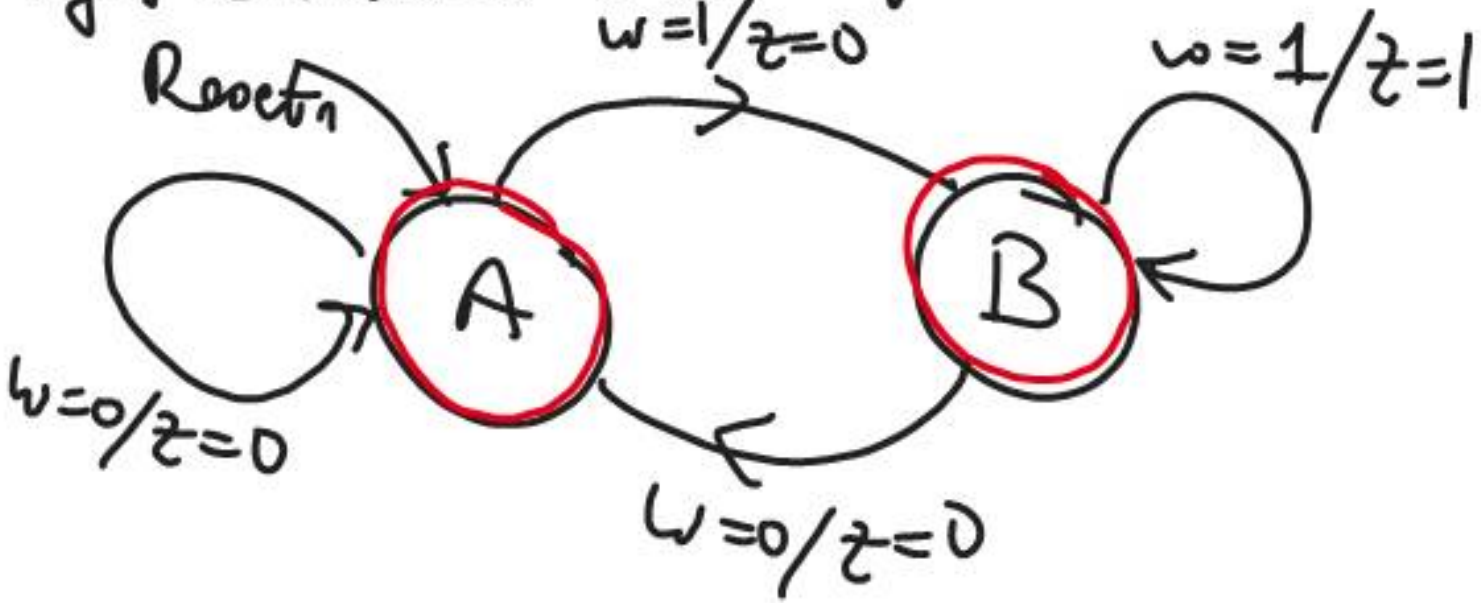
$$Y_3 = w \overline{y_1}$$

$$Z = y_3$$

MEALY STATE MODEL

Clock cycle:	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
w :	0	1	0	1	1	0	1	1	1	0	1
z :	0	0	0	0	1	0	0	1	1	0	0

The Mealy machine is not only the function of the states but inputs as well

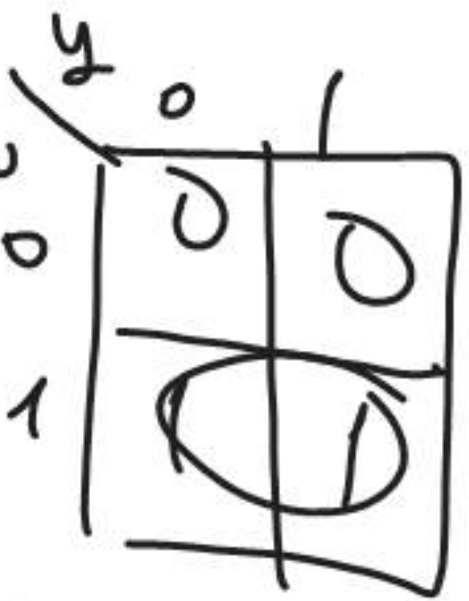


Present State	Next State		Output z	
	w=0	w=1	w=0	w=1
A	A	B	0	0
B	A	B	0	1

Only 1 FF is enough to represent 2 states.

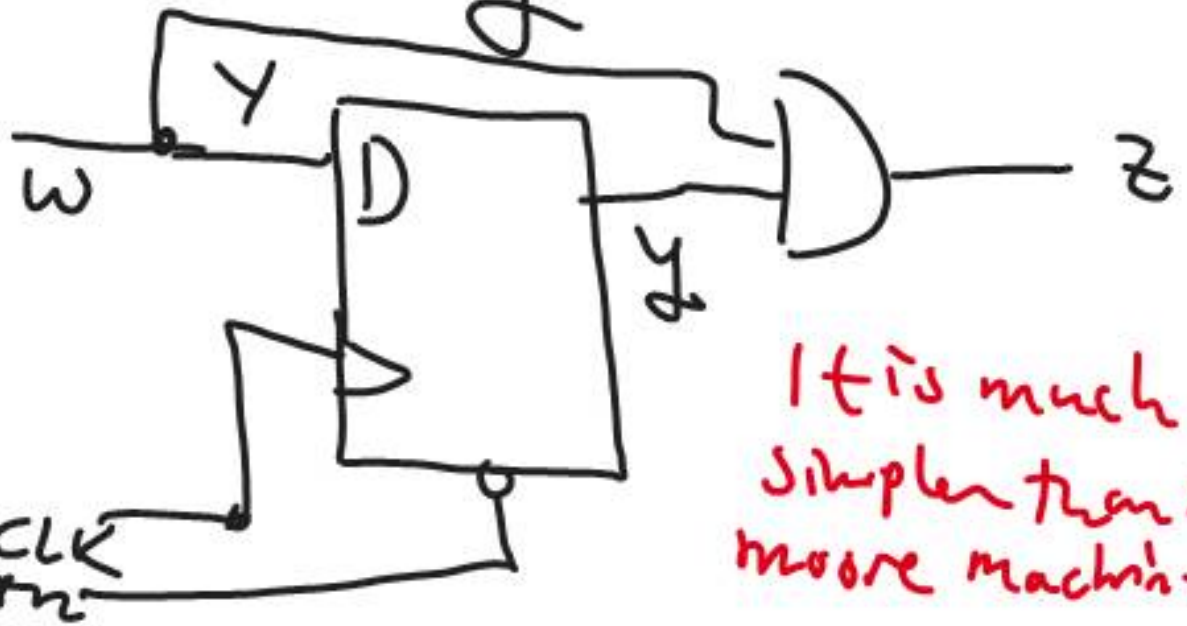
Present state y	Next state Y		Output z	
	$w=0$	$w=1$	$w=0$	$w=1$
A 0	0	1	0	0
B 1	0	1	0	1

Y

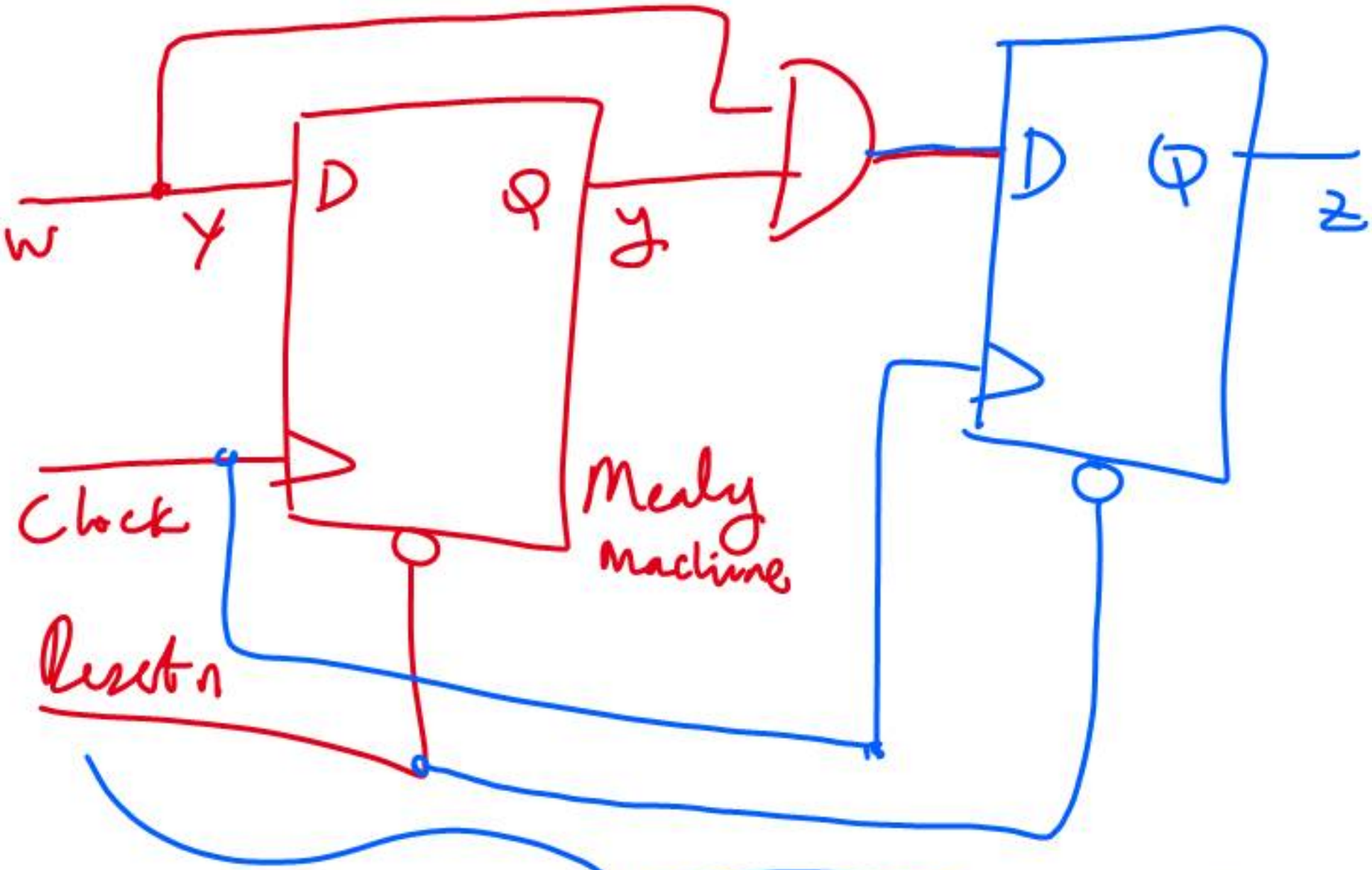


$Y = w$

$z = wy$



It is much simpler than the moore machine



Moore Machine (by delaying the output 1 Clock cycle (using a FF)

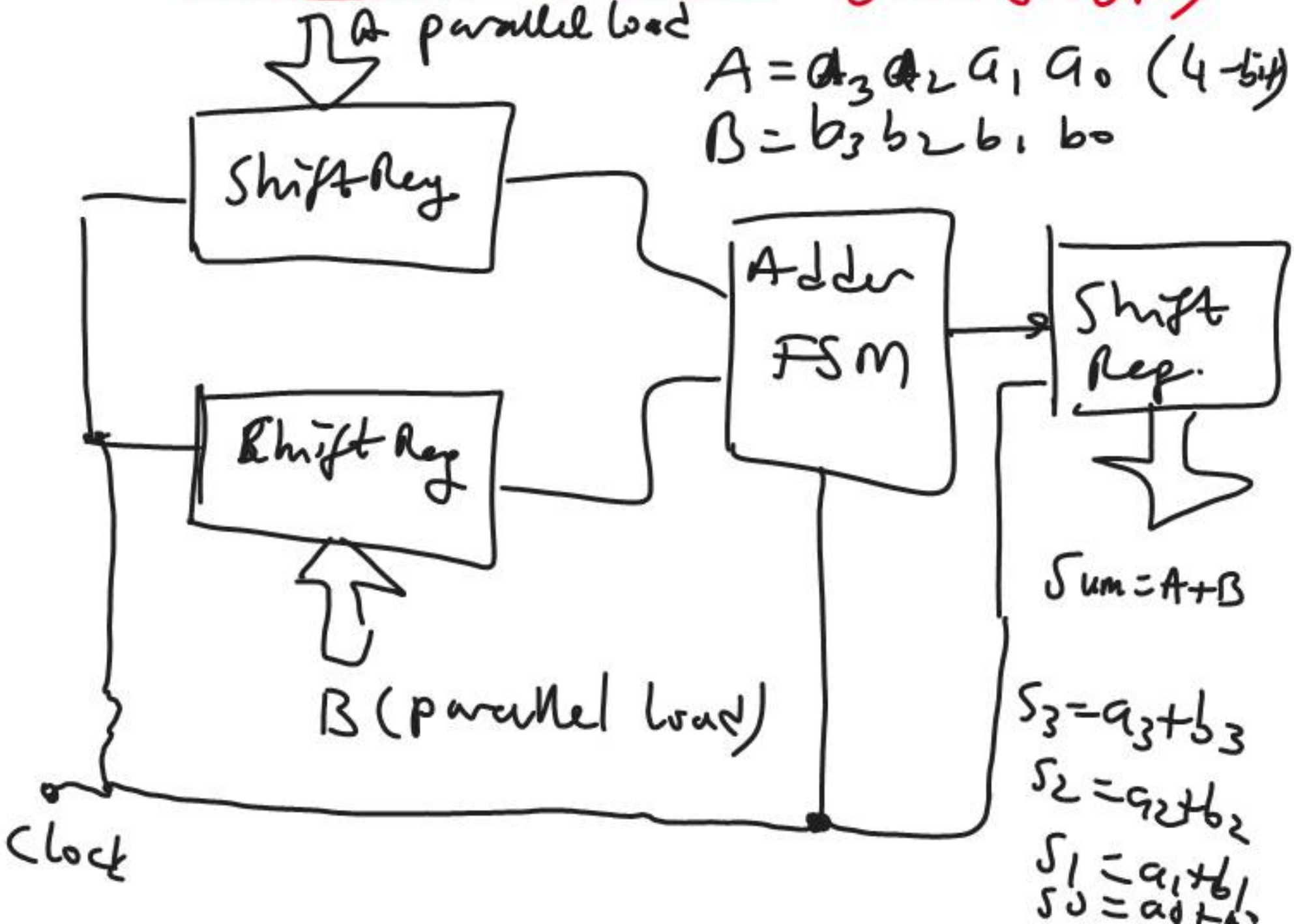
VHDL code for the Mealy machine:

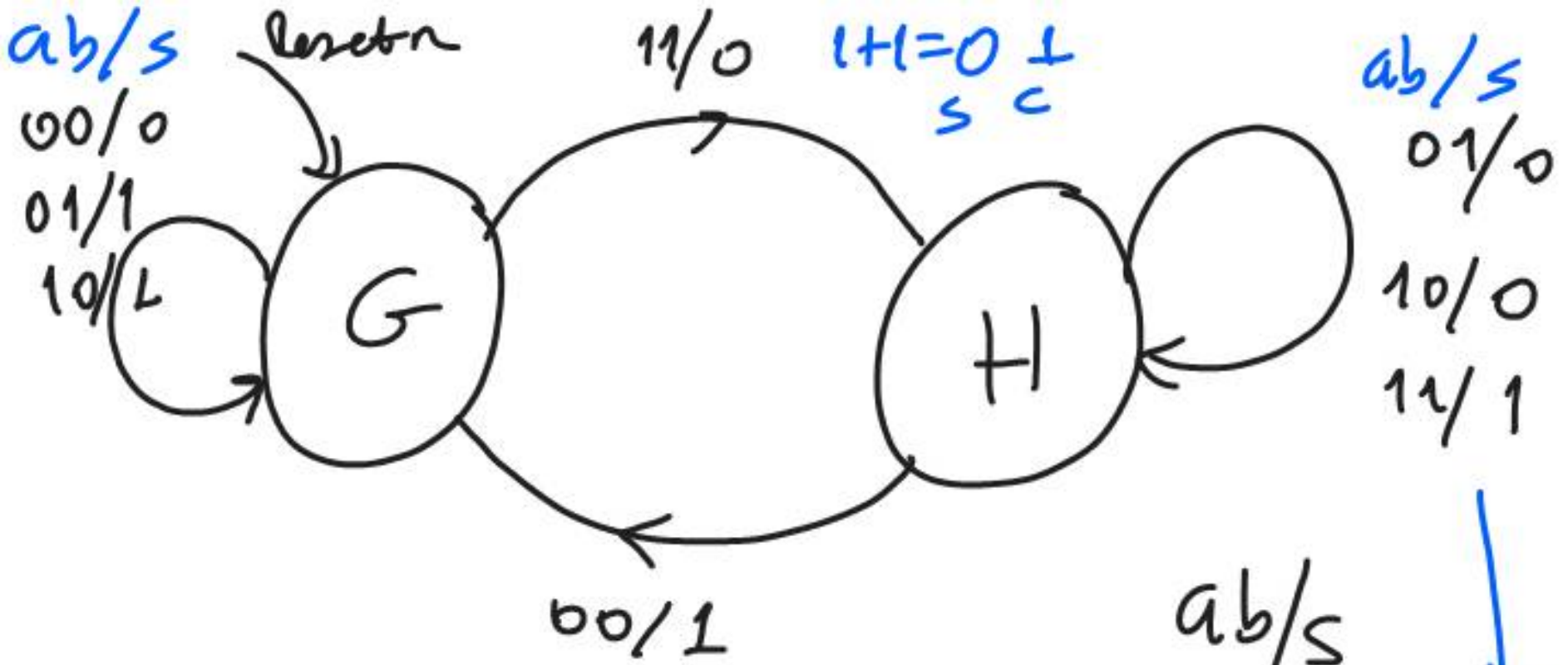
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY mealy IS
PORT(Clock, Reset, w: IN STD_LOGIC;
     z                : OUT STD_LOGIC);
END mealy;

ARCHITECTURE Behavior of mealy IS;
 TYPE State_type IS (A, B);
 SIGNAL y : State_type;
```

```
BEGIN
PROCESS(Reseton, Clock)
BEGIN
IF Reseton='0' THEN
    y <= A;
ELSIF (Clock'EVENT AND clock='1') THEN
CASE y IS
    WHEN A =>
    IF w='0' THEN y <= A;
    ELSE y <= B;
ENDIF;
```

```
WHEN B =>
IF w='0' THEN y <= A;
ELSE y <= B;
ENDIF;
END CASE;
END IF;
END PROCESS;
PROCESS (y,w)
BEGIN
CASE y IS
WHEN A =>
z <= '0';
WHEN B =>
z <= w;
END CASE;
END PROCESS;
END BEHAVIOR;
```

SERIAL ADDER (Mealy Type)





ab/s

G: State when Carry = 0

H: State when Carry = 1 $\Rightarrow$

q	0	1	1	0
b	1	0	1	0
c	1	1	1	1
s	0	0	1	1

State Table

Present State	Next State ab 00	ab 01	ab 10	ab 11	Output ab 00	ab 01	ab 10	ab 11
y	Y				S			
G	G	G	G	H	0	1	1	0
H	G	H	H	H	1	0	0	1

State assigned table

Present y	next Y 00	01	10	11	S 00	01	10	11
0	0	0	0	1	0	1	1	0
1	0	1	1	1	1	0	0	1

Y

y \ ab	00	01	11	10
0	0	0	1	0
1	0	1	1	1

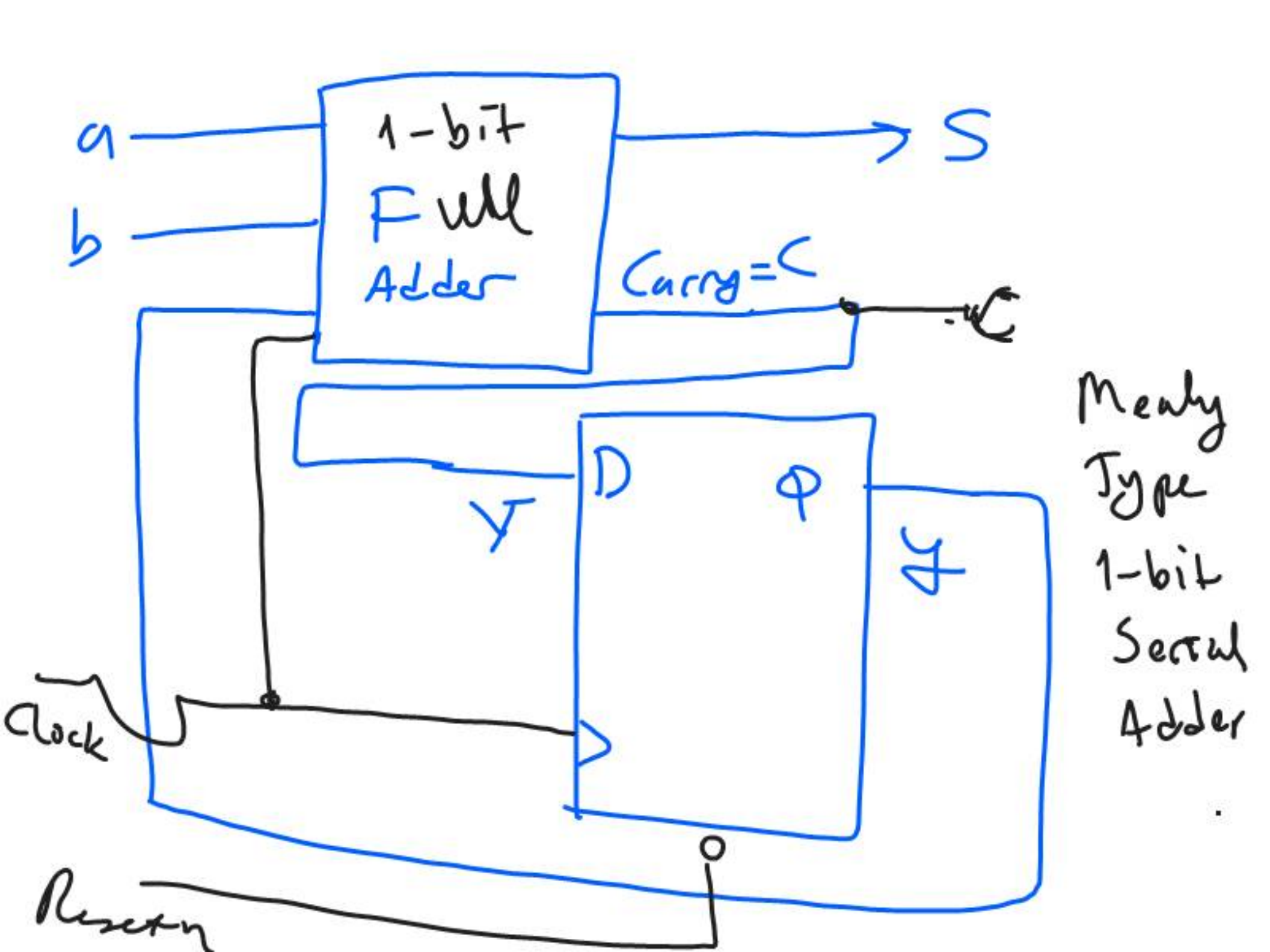
$Y = ab + ay + by$

(majority function - carry function)

S

y \ ab	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$S = a \oplus b \oplus y$


a
b
1-bit
Full
Adder
S
Carry=C
C
D
Q
Y
y
Clock
Reset
Mealy
Type
1-bit
Serial
Adder