17th December 2017

PRIFYSGOL BANGOR UNIVERSITY

VLSI Design

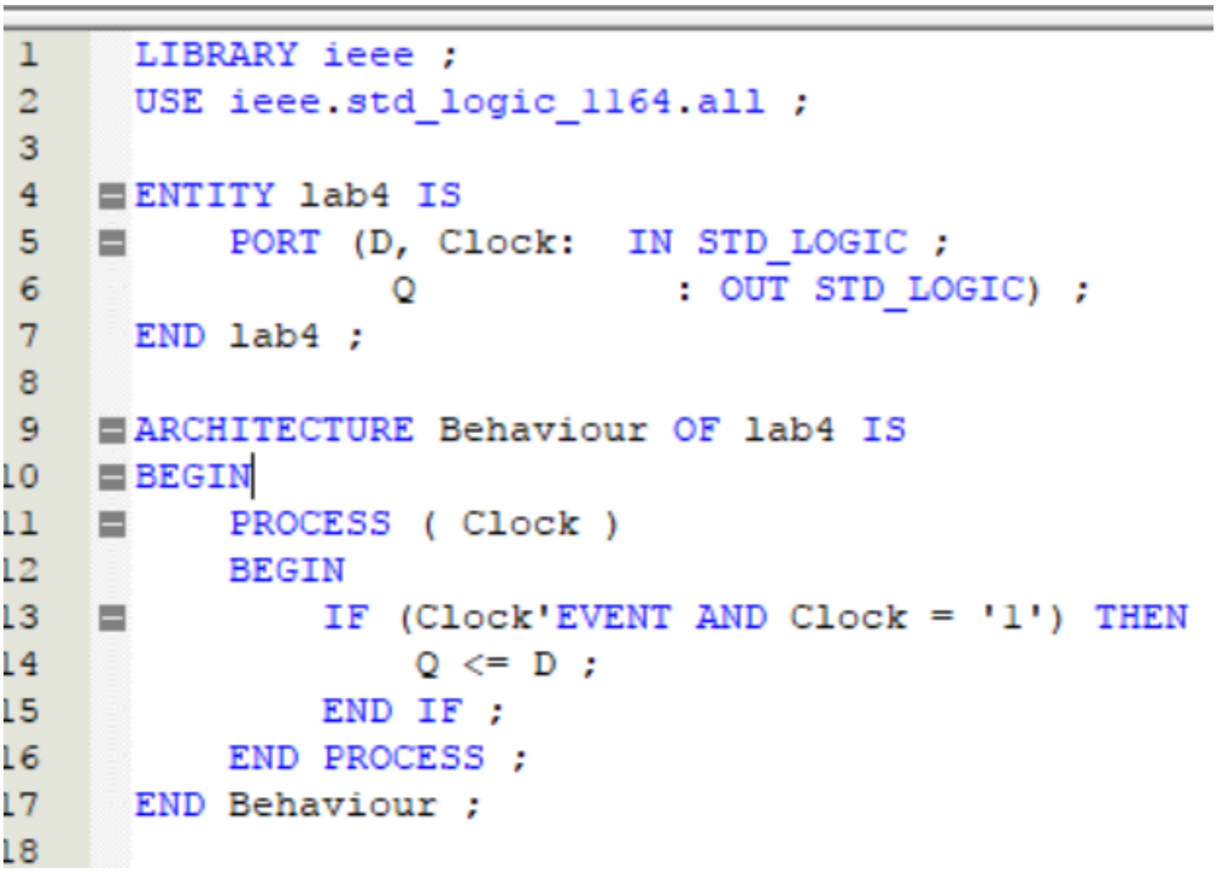
Lab Report 4

Martin Le Mintier

500494525

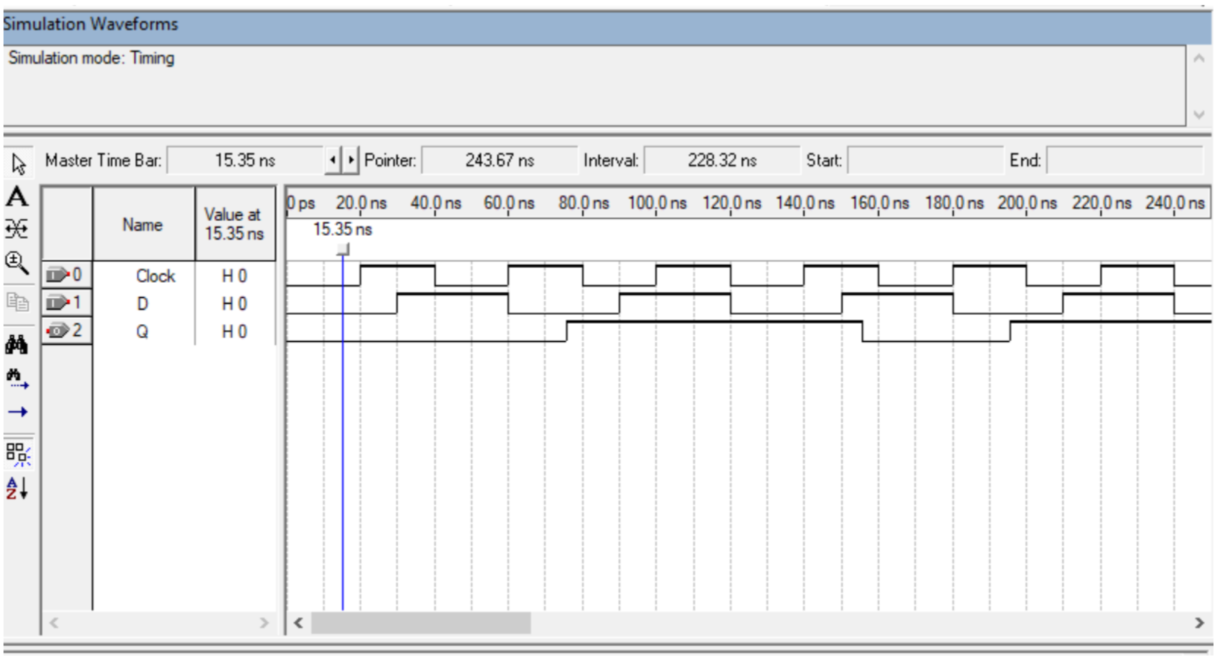
Flip Flop and Register

Here is the VHDL file with the cod from the subject.



```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY lab4 IS
    PORT (D, Clock:  IN STD_LOGIC ;
              Q            : OUT STD_LOGIC) ;
END lab4 ;

ARCHITECTURE Behaviour OF lab4 IS
BEGIN
    PROCESS ( Clock )
    BEGIN
        IF (Clock'EVENT AND Clock = '1') THEN
            Q <= D ;
        END IF ;
    END PROCESS ;
END Behaviour ;

```

And here is what we can see on the Simulation :



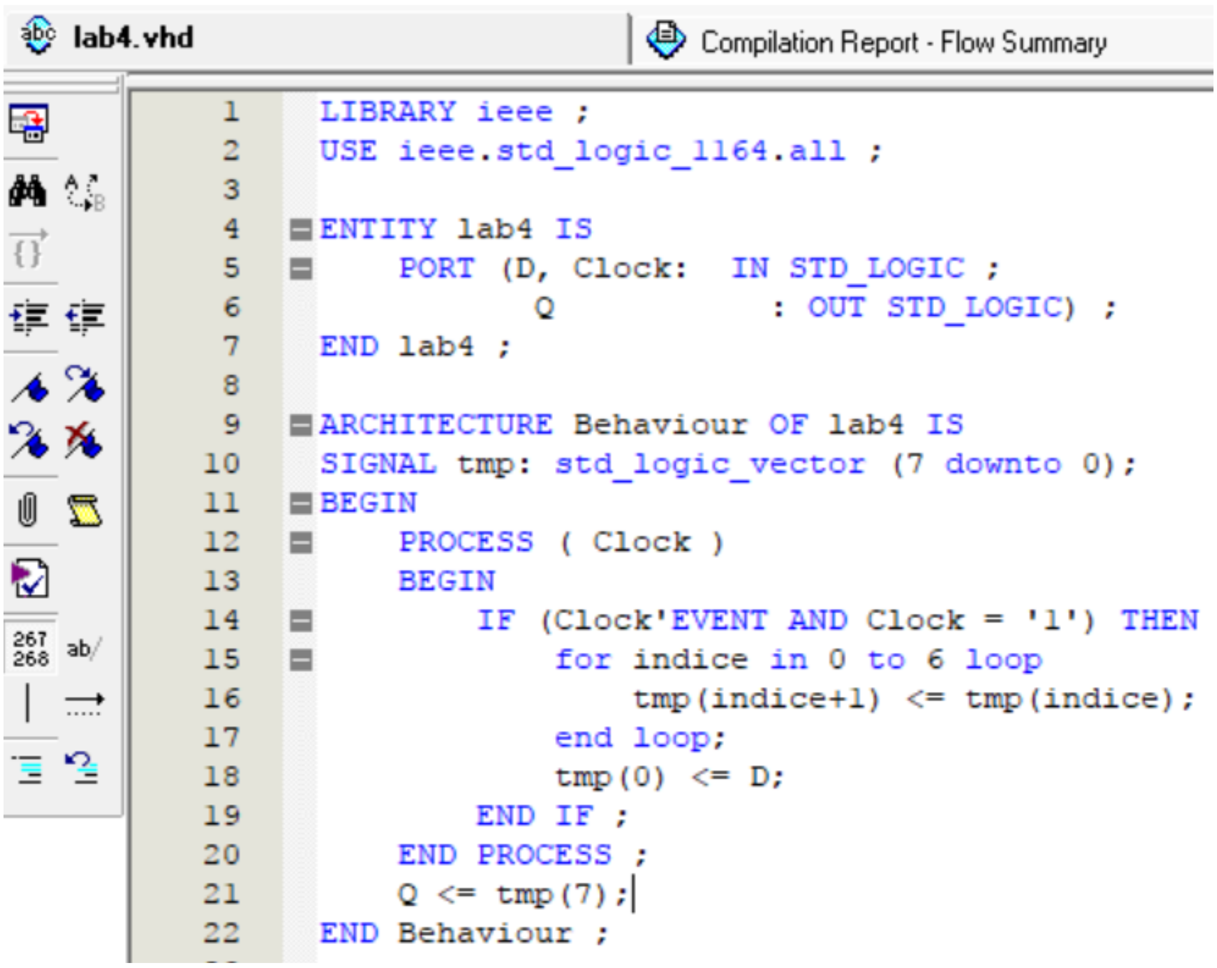
Simulation Waveforms

Simulation mode: Timing

Master Time Bar: 15.35 ns Pointer: 243.67 ns Interval: 228.32 ns Start: End:

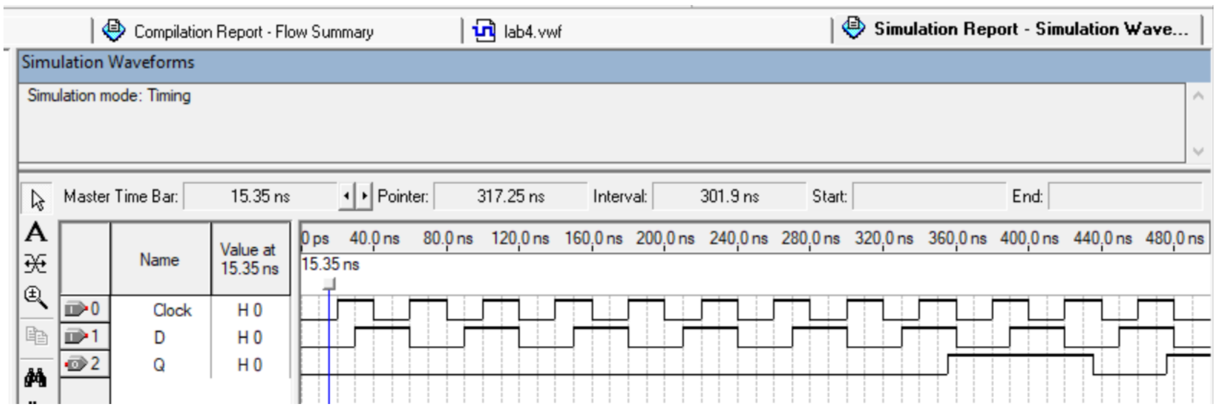
	Name	Value at 15.35 ns
0	Clock	H 0
1	D	H 0
2	Q	H 0

Now if we try to convert our code to an 8bit register



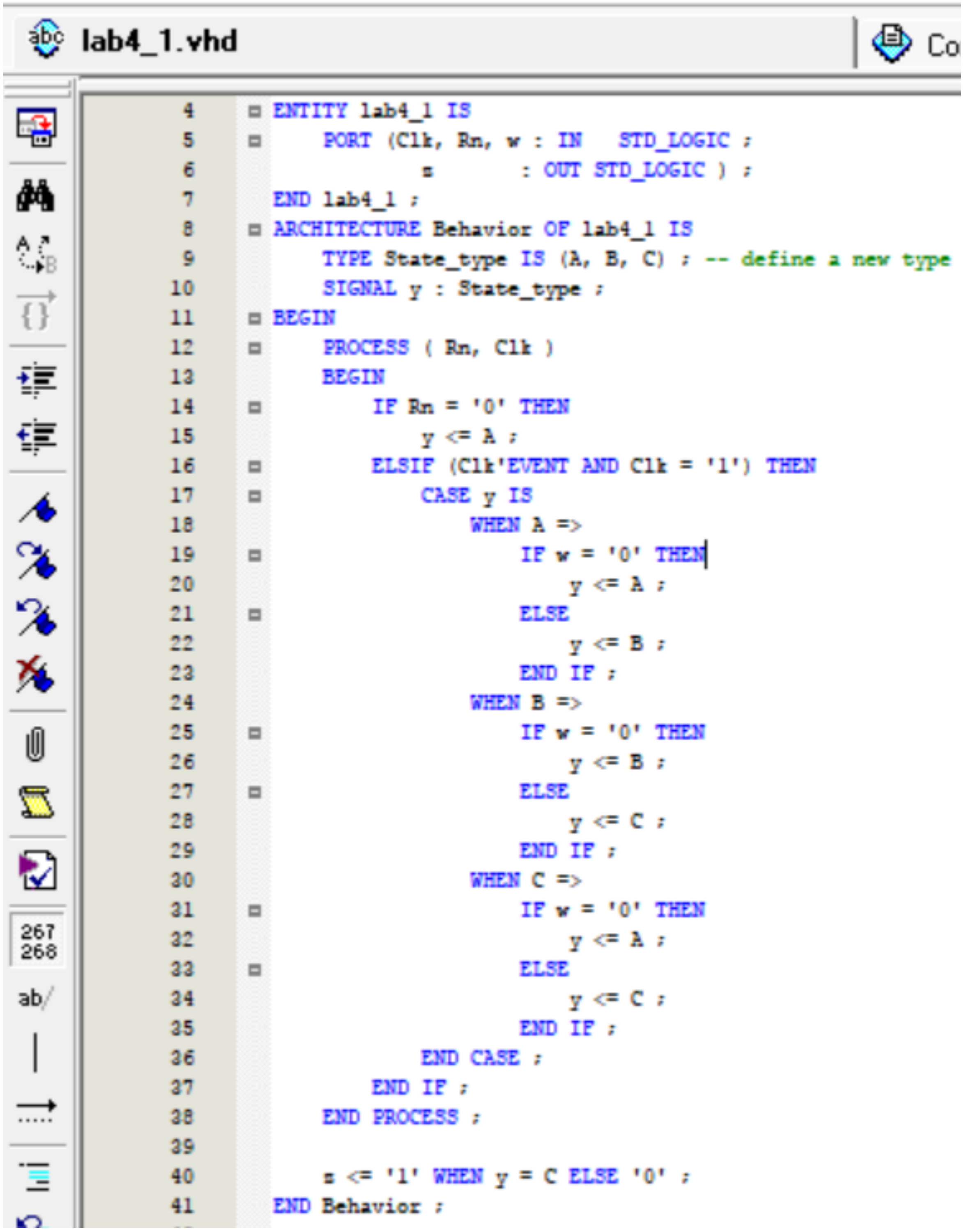
```vhdl
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY lab4 IS
    PORT (D, Clock:  IN STD_LOGIC ;
             Q           : OUT STD_LOGIC) ;
END lab4 ;

ARCHITECTURE Behaviour OF lab4 IS
SIGNAL tmp: std_logic_vector (7 downto 0);
BEGIN
    PROCESS ( Clock )
    BEGIN
        IF (Clock'EVENT AND Clock = '1') THEN
            for indice in 0 to 6 loop
                tmp(indice+1) <= tmp(indice);
            end loop;
            tmp(0) <= D;
        END IF ;
    END PROCESS ;
    Q <= tmp(7);
END Behaviour ;
```

And here is what we can see on the Simulation :



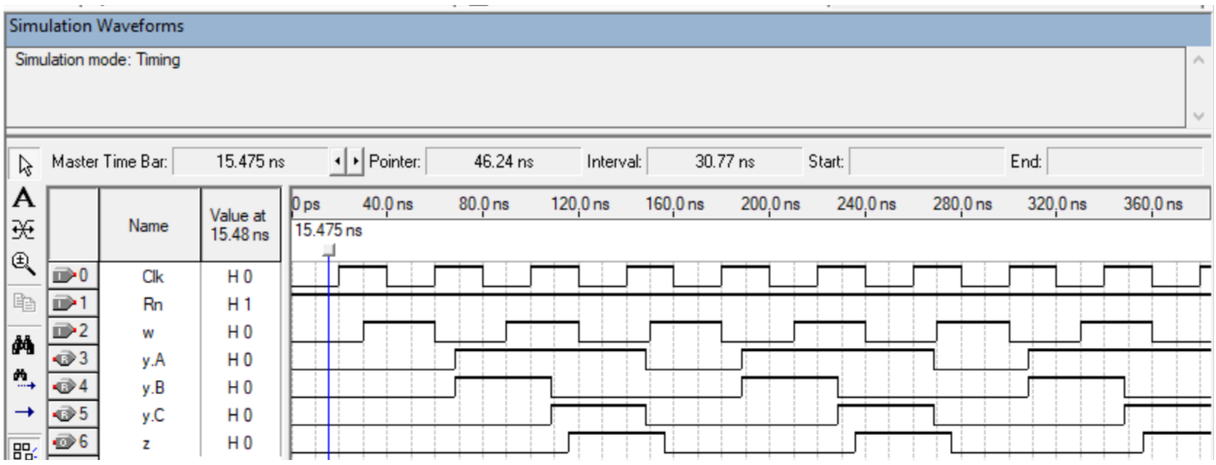
One Process FSM

Here is the VHDL file with the cod from the subject.



```
ENTITY lab4_1 IS
    PORT (Clk, Rn, w : IN   STD_LOGIC ;
                z       : OUT STD_LOGIC ) ;
END lab4_1 ;
ARCHITECTURE Behavior OF lab4_1 IS
    TYPE State_type IS (A, B, C) ; -- define a new type
    SIGNAL y : State_type ;
BEGIN
    PROCESS ( Rn, Clk )
    BEGIN
        IF Rn = '0' THEN
            y <= A ;
        ELSIF (Clk'EVENT AND Clk = '1') THEN
            CASE y IS
                WHEN A =>
                    IF w = '0' THEN
                        y <= A ;
                    ELSE
                        y <= B ;
                    END IF ;
                WHEN B =>
                    IF w = '0' THEN
                        y <= B ;
                    ELSE
                        y <= C ;
                    END IF ;
                WHEN C =>
                    IF w = '0' THEN
                        y <= A ;
                    ELSE
                        y <= C ;
                    END IF ;
            END CASE ;
        END IF ;
    END PROCESS ;

    z <= '1' WHEN y = C ELSE '0' ;
END Behavior ;
```

And here is what we can see on the Simulation :


Simulation Waveforms

Simulation mode: Timing

Master Time Bar: 15.475 ns   Pointer: 46.24 ns   Interval: 30.77 ns   Start: End:

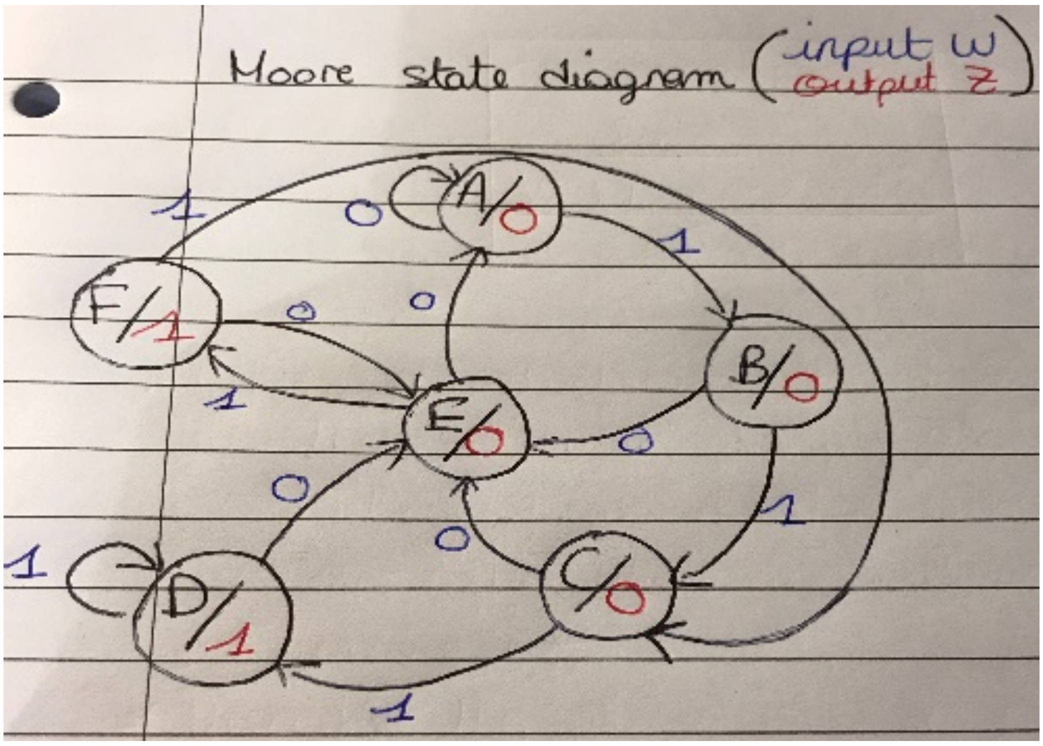
	Name	Value at 15.48 ns
0	Clk	H 0
1	Rn	H 1
2	w	H 0
3	y.A	H 0
4	y.B	H 0
5	y.C	H 0
6	z	H 0

Sequence detector

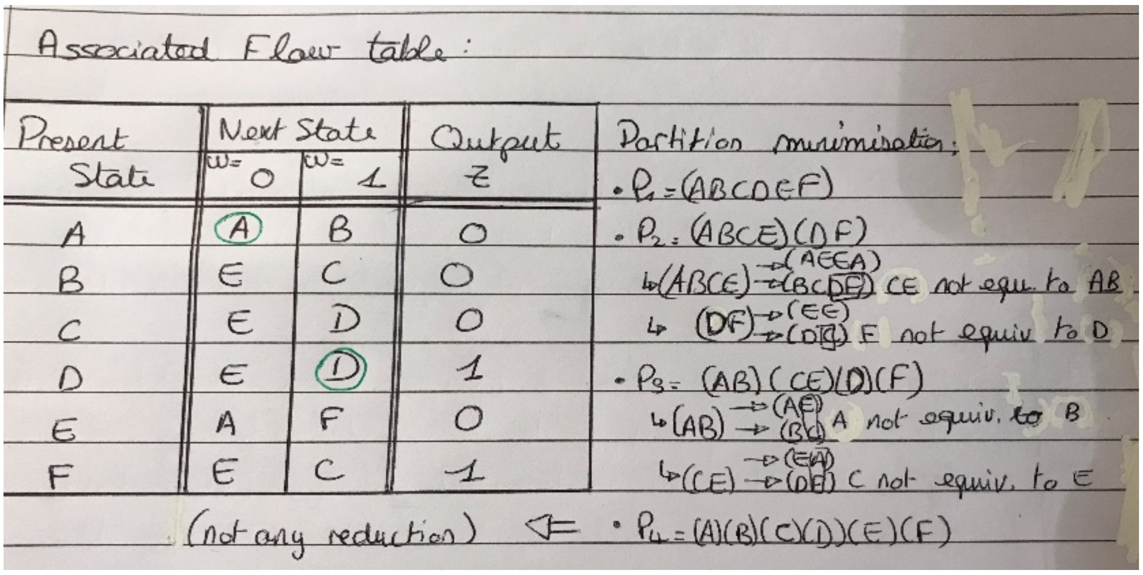
Design a VHDL entity, implemented in a third new project, which detects the input sequences “101” or “111” on its input.

We tried Moore and Mealy Machine and then chose Moore because it was simpler to build.

Here is our state diagram:



And here is the flow table :



Associated Flow table :

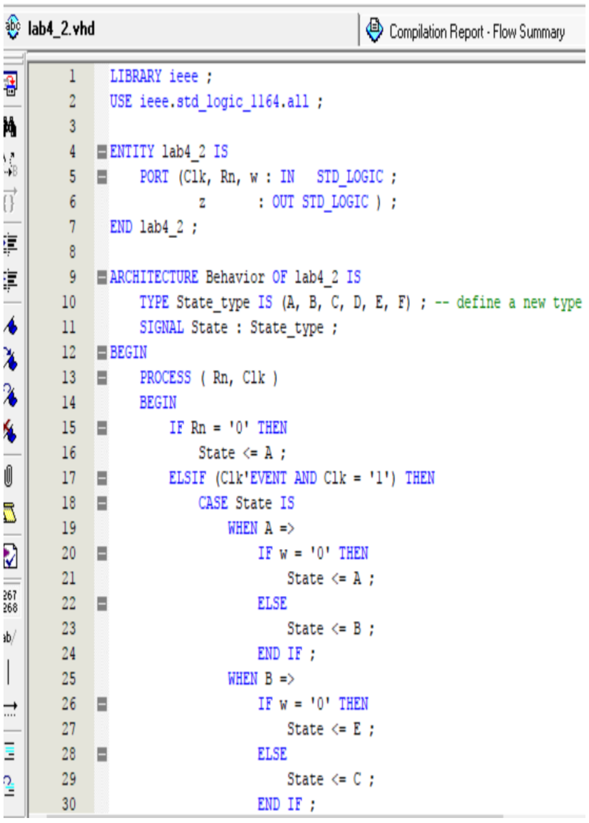
Present State	Next State $w=0$	Next State $w=1$	Output z
A	A	B	0
B	E	C	0
C	E	D	0
D	E	D	1
E	A	F	0
F	E	C	1

Partition minimisation:

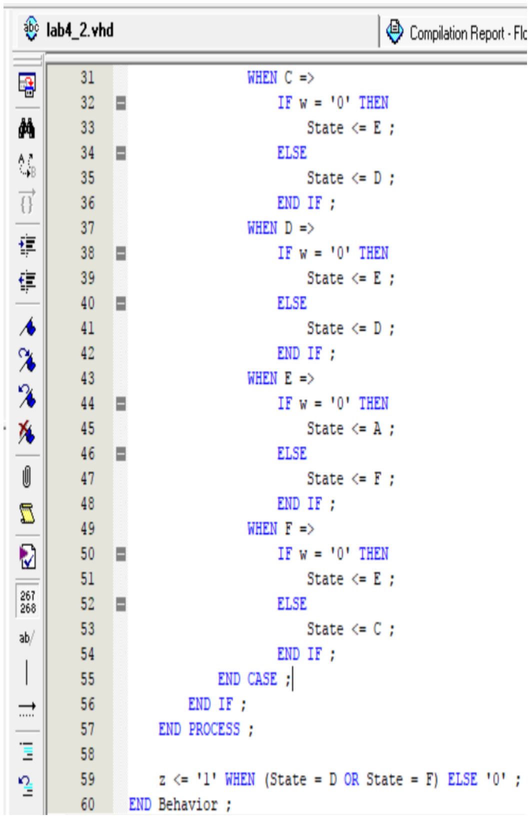
- $P_1 = (ABCDEF)$
- $P_2 = (ABCE)(DF)$
  - ↳ $(ABCE) \begin{matrix} \to (AEEA) \\ \to (BC\boxed{DF}) \end{matrix}$ CE not equ. to AB
  - ↳ $(DF) \begin{matrix} \to (EE) \\ \to (D\boxed{C}) \end{matrix}$ F not equiv to D
- $P_3 = (AB)(CE)(D)(F)$
  - ↳ $(AB) \begin{matrix} \to (A\boxed{E}) \\ \to (B\boxed{C}) \end{matrix}$ A not equiv. to B
  - ↳ $(CE) \begin{matrix} \to (E\boxed{A}) \\ \to (D\boxed{F}) \end{matrix}$ C not equiv. to E

(not any reduction) $\Leftarrow$ • $P_4 = (A)(B)(C)(D)(E)(F)$

Here is the VHDL code that we wrote as we didi in class :

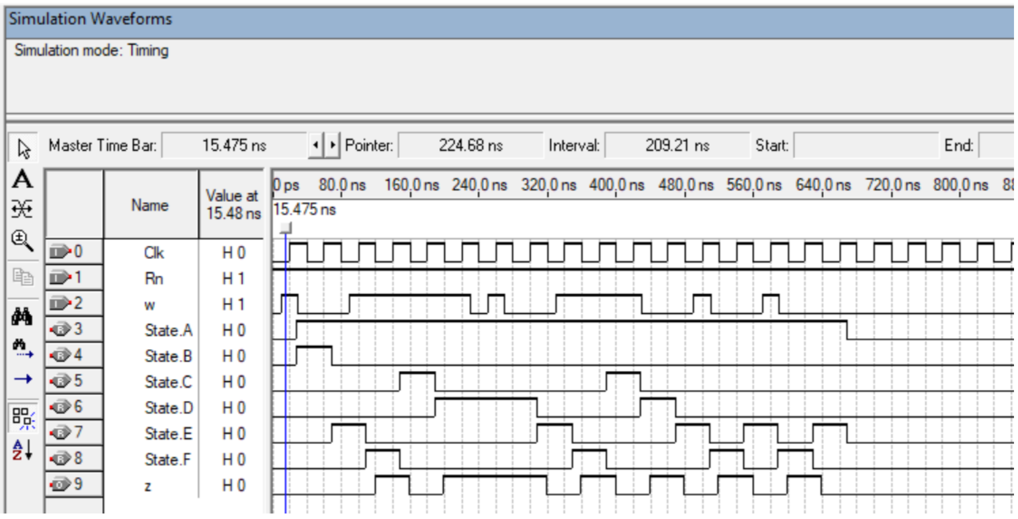


```vhdl
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY lab4_2 IS
    PORT (Clk, Rn, w : IN   STD_LOGIC ;
            z       : OUT STD_LOGIC ) ;
END lab4_2 ;

ARCHITECTURE Behavior OF lab4_2 IS
    TYPE State_type IS (A, B, C, D, E, F) ; -- define a new type
    SIGNAL State : State_type ;
BEGIN
    PROCESS ( Rn, Clk )
    BEGIN
        IF Rn = '0' THEN
            State <= A ;
        ELSIF (Clk'EVENT AND Clk = '1') THEN
            CASE State IS
                WHEN A =>
                    IF w = '0' THEN
                        State <= A ;
                    ELSE
                        State <= B ;
                    END IF ;
                WHEN B =>
                    IF w = '0' THEN
                        State <= E ;
                    ELSE
                        State <= C ;
                    END IF ;
```



```vhdl
                WHEN C =>
                    IF w = '0' THEN
                        State <= E ;
                    ELSE
                        State <= D ;
                    END IF ;
                WHEN D =>
                    IF w = '0' THEN
                        State <= E ;
                    ELSE
                        State <= D ;
                    END IF ;
                WHEN E =>
                    IF w = '0' THEN
                        State <= A ;
                    ELSE
                        State <= F ;
                    END IF ;
                WHEN F =>
                    IF w = '0' THEN
                        State <= E ;
                    ELSE
                        State <= C ;
                    END IF ;
            END CASE ;
        END IF ;
    END PROCESS ;

    z <= '1' WHEN (State = D OR State = F) ELSE '0' ;
END Behavior ;
```

And here is what we can see on the Simulation :


Simulation Waveforms

Simulation mode: Timing

Master Time Bar: 15.475 ns Pointer: 224.68 ns Interval: 209.21 ns Start: End:

	Name	Value at 15.48 ns
0	Clk	H 0
1	Rn	H 1
2	w	H 1
3	State.A	H 0
4	State.B	H 0
5	State.C	H 0
6	State.D	H 0
7	State.E	H 0
8	State.F	H 0
9	z	H 0