Lab 10
Finite State Machine

ECE 380-002
University of Alabama

Yichen Huang
Thomas Dillman

2019/11/18

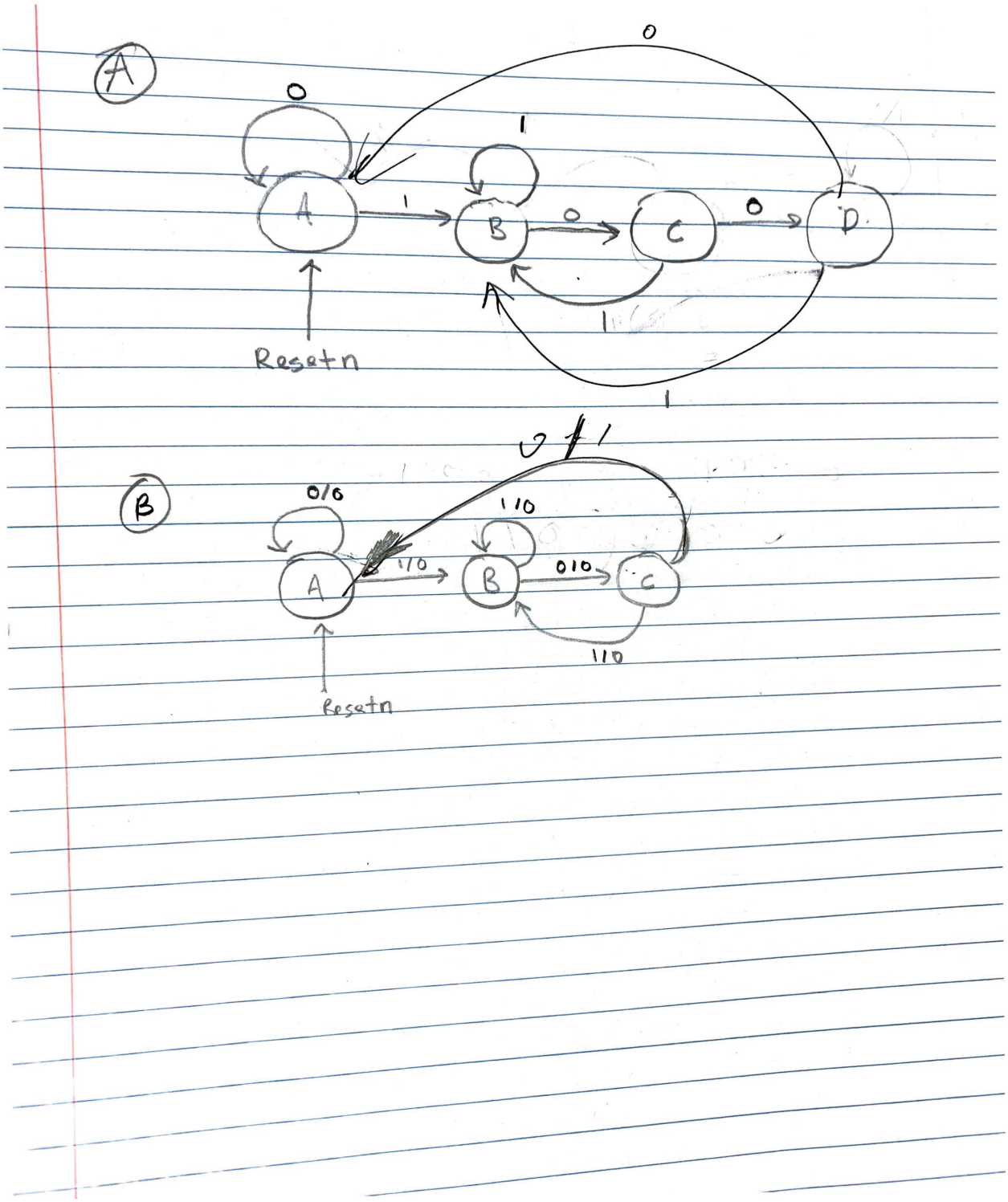
Introduction

In this lab, we mainly focus on the designing of two type of finite state machine (FSM) which are Moore-type and Mealy-type. These two machines will detect the input, when the inputs contain sequence of 100, the output will be 1 else the output will be 0.

Procedure

A. Prelab

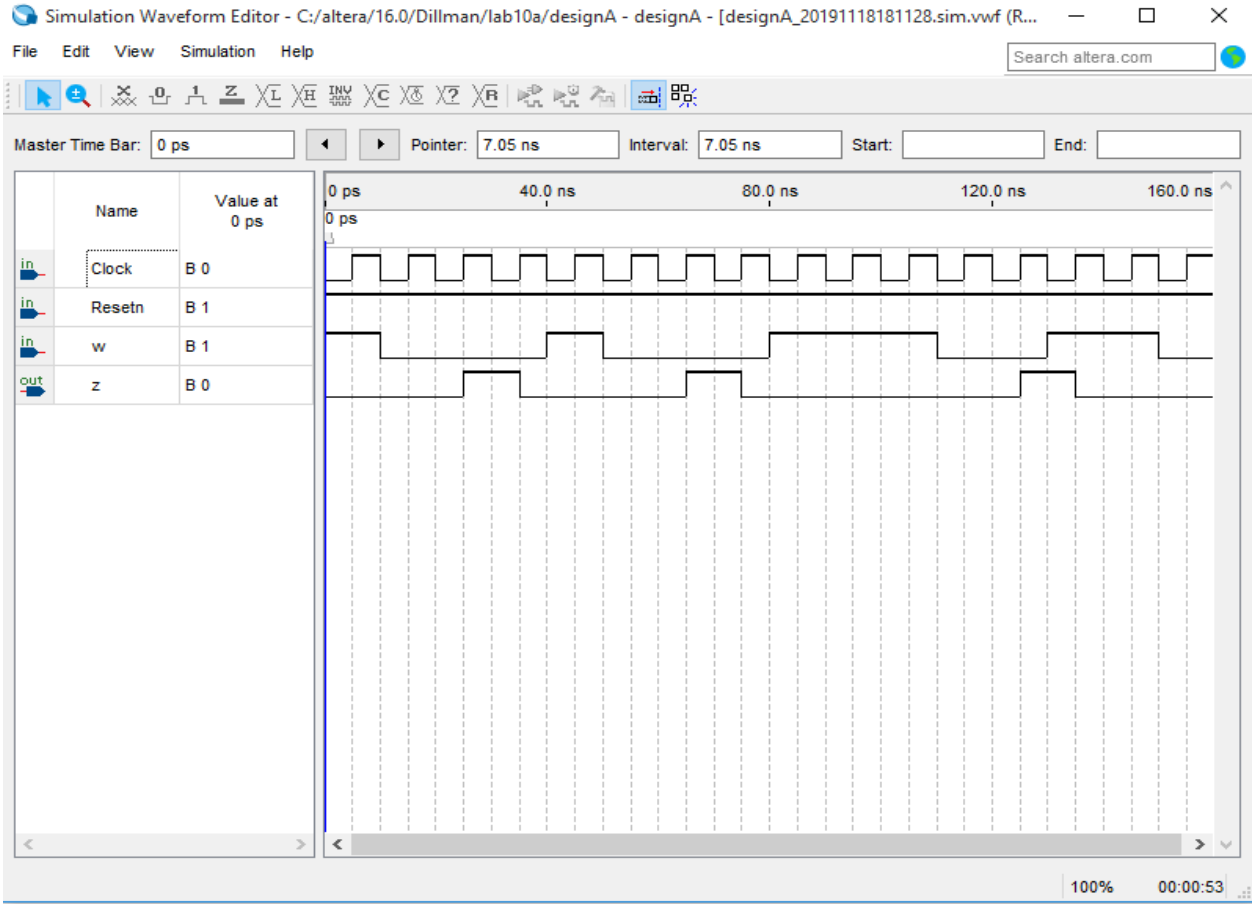
In the prelab, we draw the bubble diagram of two machine. In the picture, diagram A is Moore-type finite state machine and diagram B is Mealy-type finite state machine.



a. Design A

In the design A, we redesign the VHDL code and make some change to let the machine can detected the sequence.

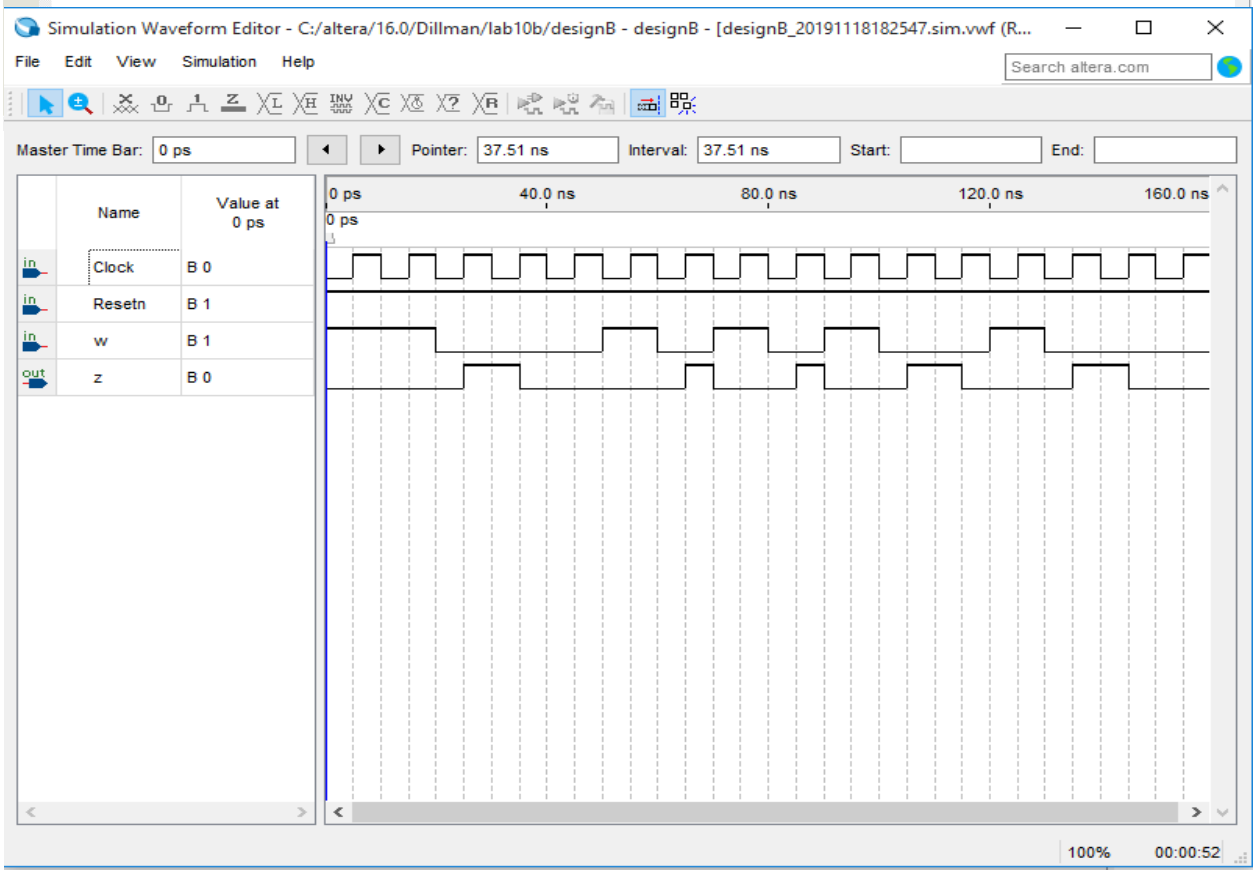
```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;
ENTITY designA IS
    PORT ( Clock, Resetn, w : IN STD_LOGIC ;
           z : OUT STD_LOGIC ) ;
END designA ;
ARCHITECTURE Behavior OF designA IS
    TYPE State_type IS (A, B, C, D) ;
    SIGNAL y : State_type ;
BEGIN
    PROCESS ( Resetn, Clock )
    BEGIN
        IF Resetn = '0' THEN
            y <= A ;
        ELSIF (Clock'EVENT AND Clock = '1') THEN
            CASE y IS
                WHEN A =>
                    IF w = '1' THEN
                        y <= B ;
                    ELSE
                        y <= A ;
                    END IF ;
                WHEN B =>
                    IF w = '0' THEN
                        y <= C ;
                    ELSE
                        y <= B ;
                    END IF ;
                WHEN C =>
                    IF w = '0' THEN
                        y <= D ;
                    ELSE
                        y <= B ;
                    END IF ;
                WHEN D =>
                    IF w = '0' THEN
                        y <= A;
                    ELSE
                        y <= B;
                    END IF;
            END CASE ;
        END IF ;
END PROCESS ;
z <= '1' WHEN y = D ELSE '0' ;
END Behavior ;
```



b. Design B

In the design B, we just design the same finite state machine but in the Mealy type.

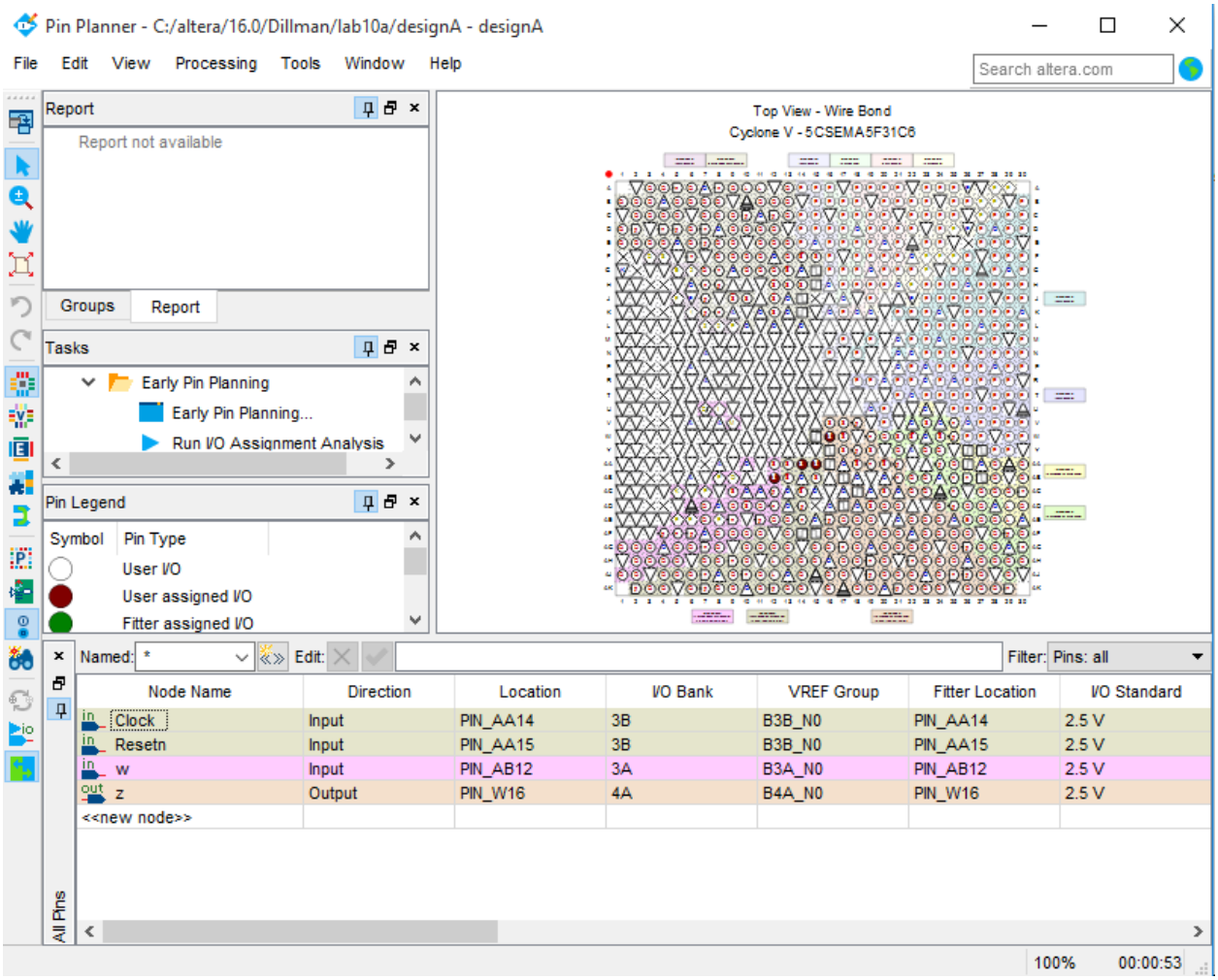
```
1   LIBRARY ieee ;
2   USE ieee.std_logic_1164.all ;
3  ENTITY designB IS
4  PORT ( Clock, Resetn, w : IN STD_LOGIC ;
5  z : OUT STD_LOGIC ) ;
6  END designB ;
7  ARCHITECTURE Behavior OF designB IS
8  TYPE State_type IS (A, B, C) ;
9  SIGNAL y : State_type ;
10 BEGIN
11 PROCESS ( Resetn, Clock )
12 BEGIN
13 IF Resetn = '0' THEN
14 y <= A ;
15 ELSIF (Clock'EVENT AND Clock = '1') THEN
16 CASE y IS
17 WHEN A =>
18 IF w = '0' THEN y <= A ;
19 ELSE y <= B ;
20 END IF ;
21 WHEN B =>
22 IF w = '0' THEN y <= C ;
23 ELSE y <= B ;
24 END IF ;
25 WHEN C =>
26 IF w = '0' THEN y <= A;
27 ELSE y<= B;
28 END IF;
29 END CASE ;
30 END IF ;
31 END PROCESS ;
32 PROCESS ( y, w )
33 BEGIN
34 CASE y IS
35 WHEN C =>
36 z <= NOT w ;
37 WHEN B =>
38 z <= '0' ;
39 WHEN A =>
40 z <= '0';
41 END CASE ;
42 END PROCESS ;
43 END Behavior ;
```

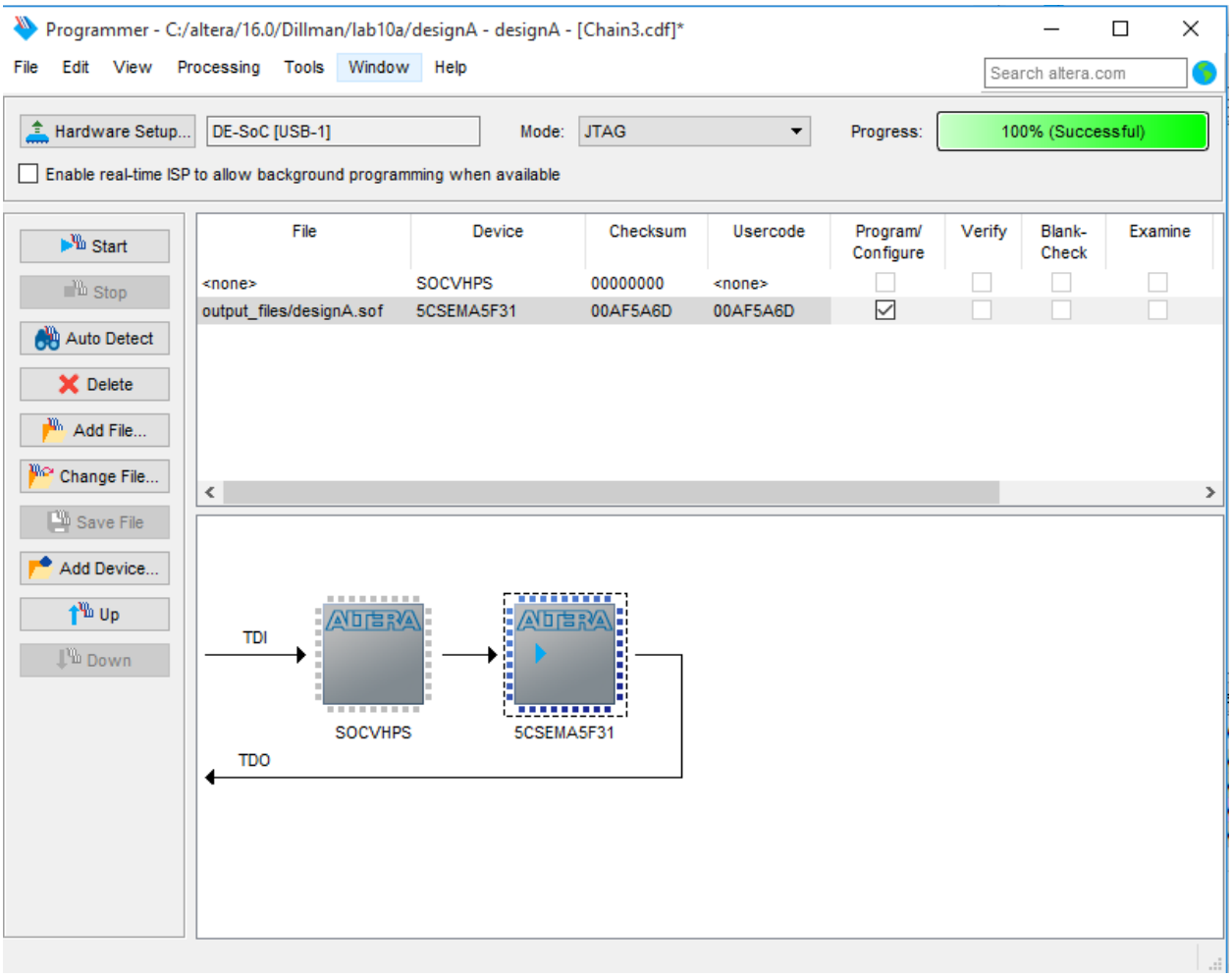


B. During the lab

a. Design A

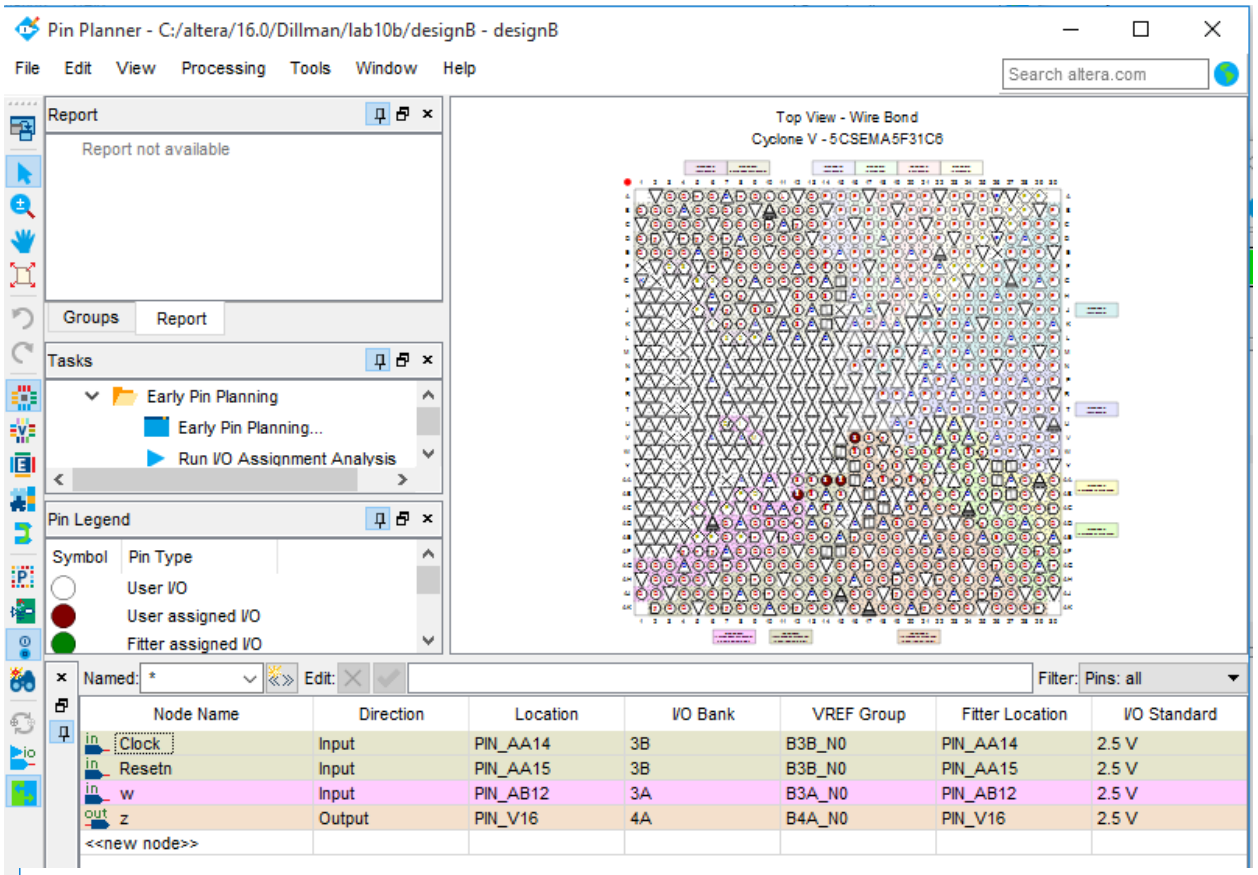
During the lab, we assign the pins of the design to the DE1 board. Then we upload the design to the lab and finally we do the test manually.

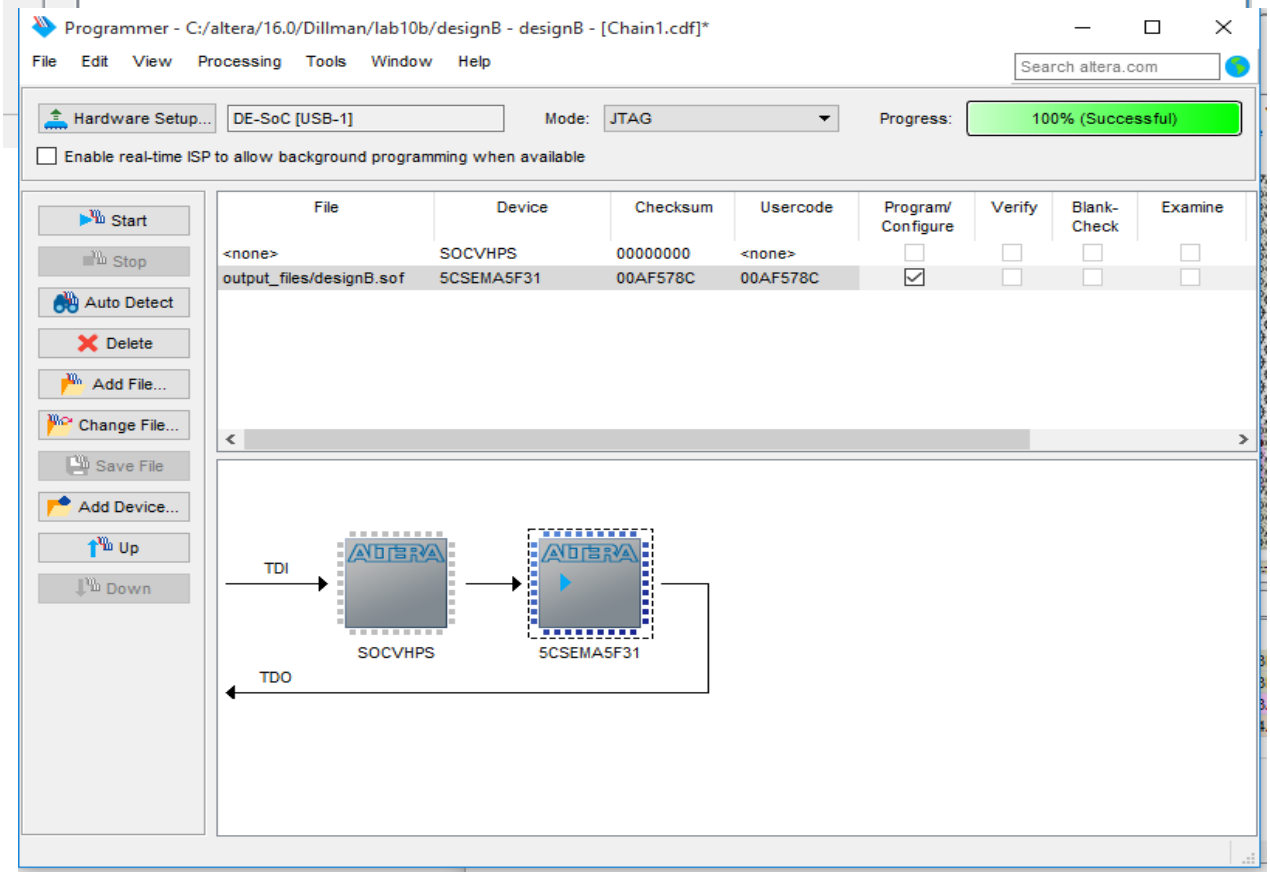




b. Design B

In the design B, we repeat the same procedure in the design A.


Pin Planner - C:/altera/16.0/Dillman/lab10b/designB - designB
File Edit View Processing Tools Window Help
Search altera.com
Report
Report not available
Groups Report
Tasks
Early Pin Planning
Early Pin Planning...
Run I/O Assignment Analysis
Pin Legend
Symbol Pin Type
User I/O
User assigned I/O
Fitter assigned I/O
Top View - Wire Bond
Cyclone V - 5CSEMA5F31C6
Named: * Edit: Filter: Pins: all
Node Name Direction Location I/O Bank VREF Group Fitter Location I/O Standard
Clock Input PIN_AA14 3B B3B_N0 PIN_AA14 2.5 V
Resetn Input PIN_AA15 3B B3B_N0 PIN_AA15 2.5 V
w Input PIN_AB12 3A B3A_N0 PIN_AB12 2.5 V
z Output PIN_V16 4A B4A_N0 PIN_V16 2.5 V
<<new node>>


Programmer - C:/altera/16.0/Dillman/lab10b/designB - designB - [Chain1.cdf]*
File Edit View Processing Tools Window Help
Search altera.com
Hardware Setup... DE-SoC [USB-1] Mode: JTAG Progress: 100% (Successful)
Enable real-time ISP to allow background programming when available
Start
Stop
Auto Detect
Delete
Add File...
Change File...
Save File
Add Device...
Up
Down
File Device Checksum Usercode Program/ Configure Verify Blank-Check Examine
<none> SOCVHPS 00000000 <none>
output_files/designB.sof 5CSEMA5F31 00AF578C 00AF578C
TDI
SOCVHPS
5CSEMA5F31
TDO

Result

In the result of the board is same as we test in the prelab.

Conclusion

You need to complete VHDL codes and functional simulations in Quartus, before your lab sessions.

Pre-Lab (30 pts)	**Score**	**TA initial**
30 pts Designs	A	

Report (70 pts)	
10 pts Introduction	
10 pts Procedures	
20 pts Results	
30 pts Conclusions	
Lab Grade (100 pts)	