**Dr.Ahmed Shelpy**

**Dr.Fatma**

**team number**

**22**

We have 7 modules in our project

**1.we have UpDwoncounter module**

In that module we have up down counter that count up and down and it has its input and output

# Inputs

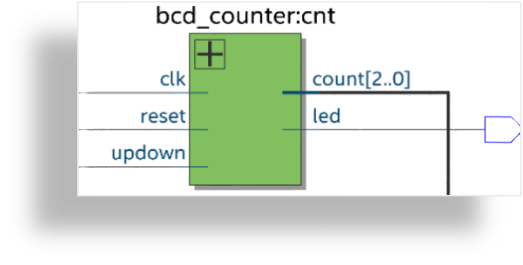
1. Clock which is come from the flip flop and have

## CLK\_divider

2. Up or down switch which identify we will count up or down 3. Reset and that is a reset case in which we back to our initial case(the queue have zero and no tellers)

## Output

1. Count [2:0] which has wire to decoder\_7 seg to display the output on 7\_segment
2. Alarm and it is a led when the queue is full or empty it is on otherwise it is off

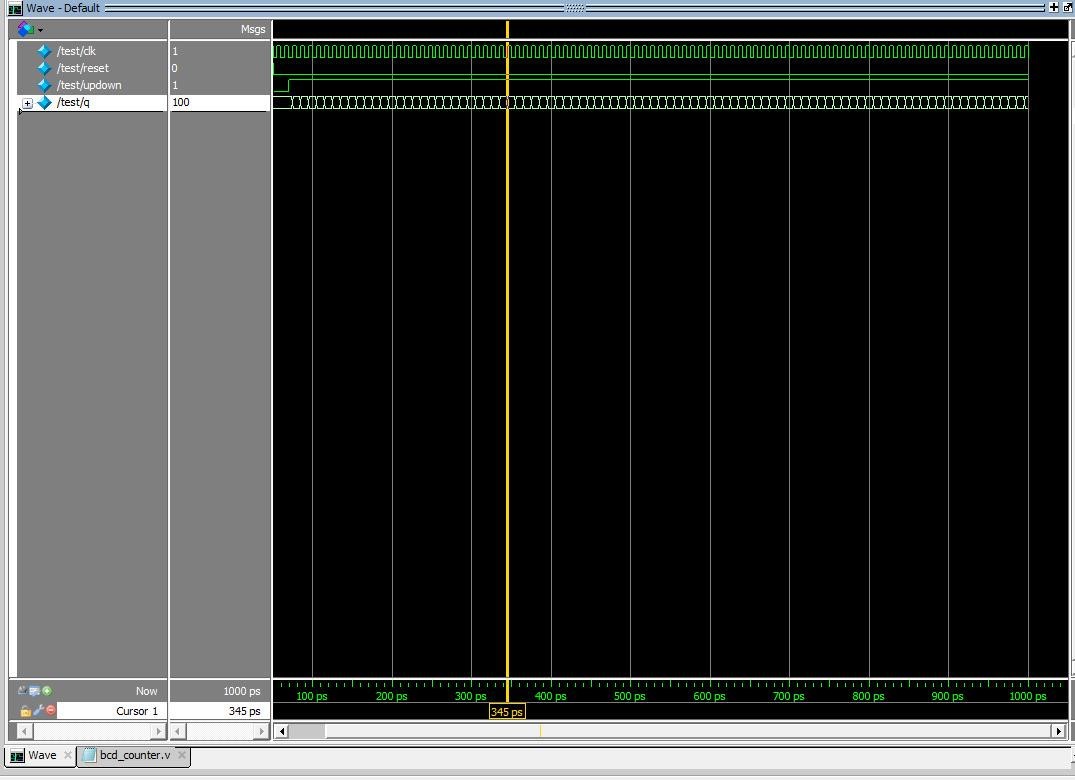
An n-bit up-down counter is to be used to generate

Pcount. The maximum

Pcount value will be (2n - 1), with a default value of 7, where n is a generic value, with a default value of 3.



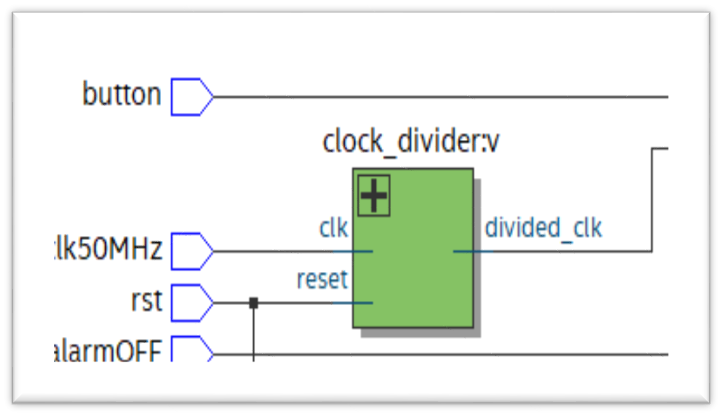
### This is the testbench on modelism For up counter

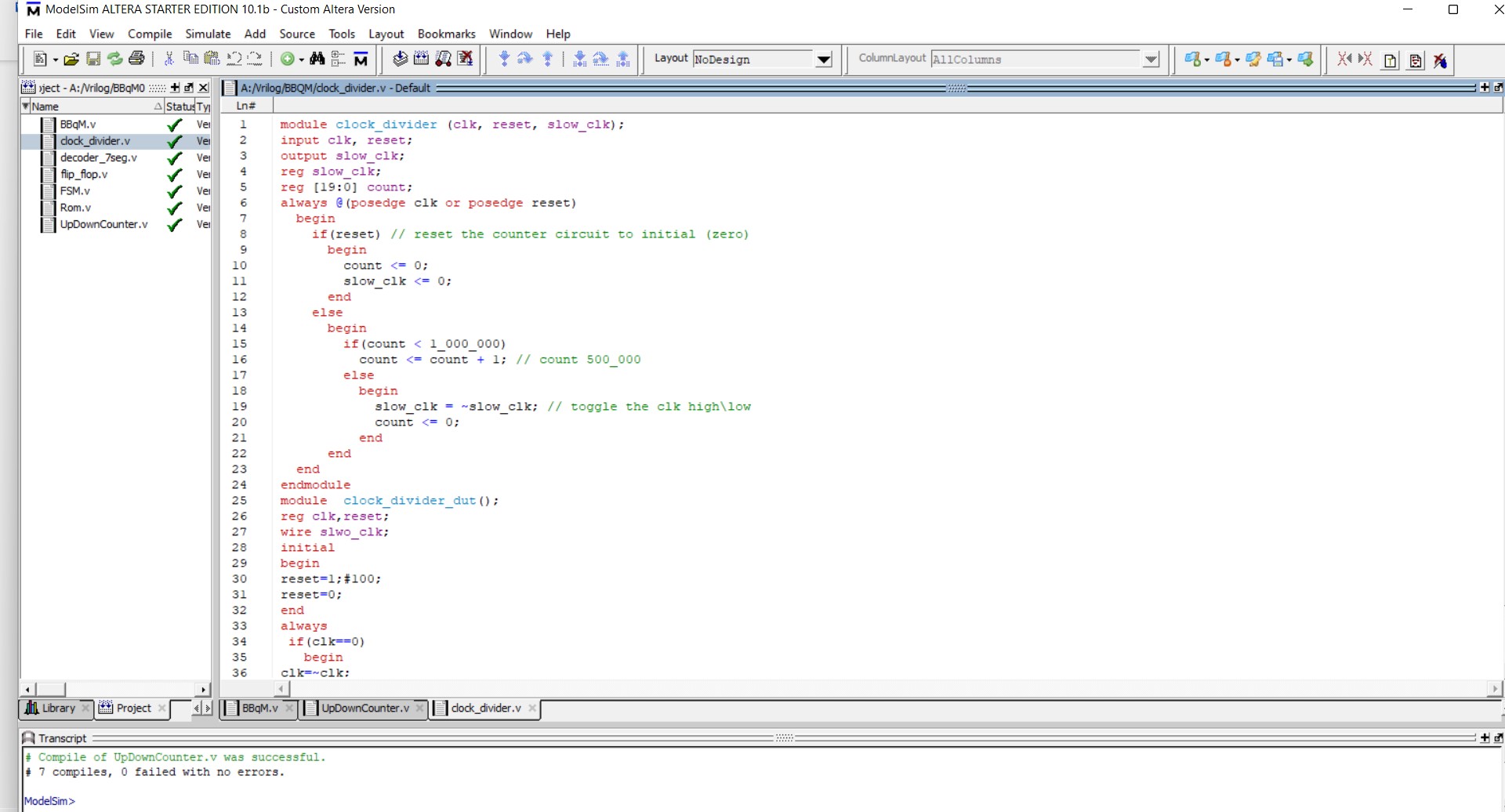


This is the test bench on model sim For down counter

## 2.clock\_divider Module

### It takes input from the clk which exact on FPGA and from reset switch if it up then the system back to initial state

-We use that module to create lower frequency. The clk signal has input from an input clk source. The divider circuit count input clk cyclesand drive output clk low and then high for some number of input clk cycles

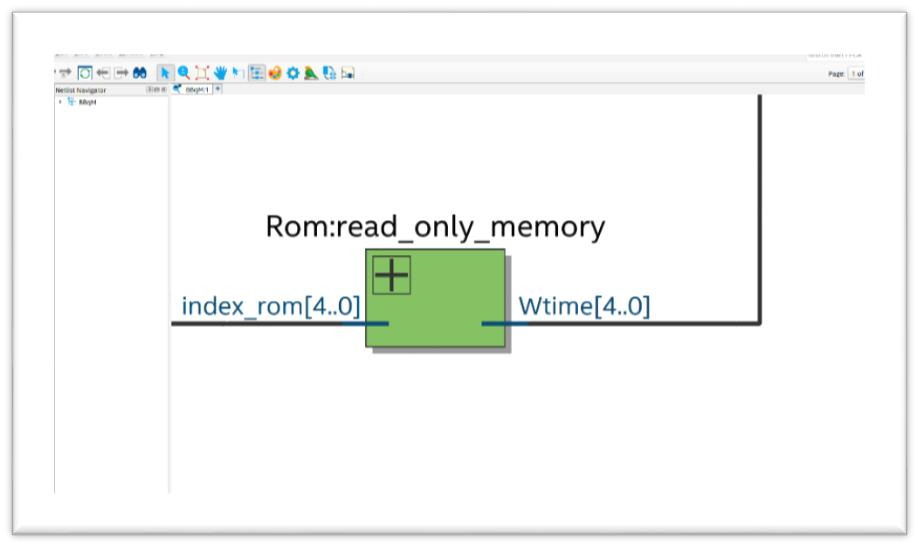


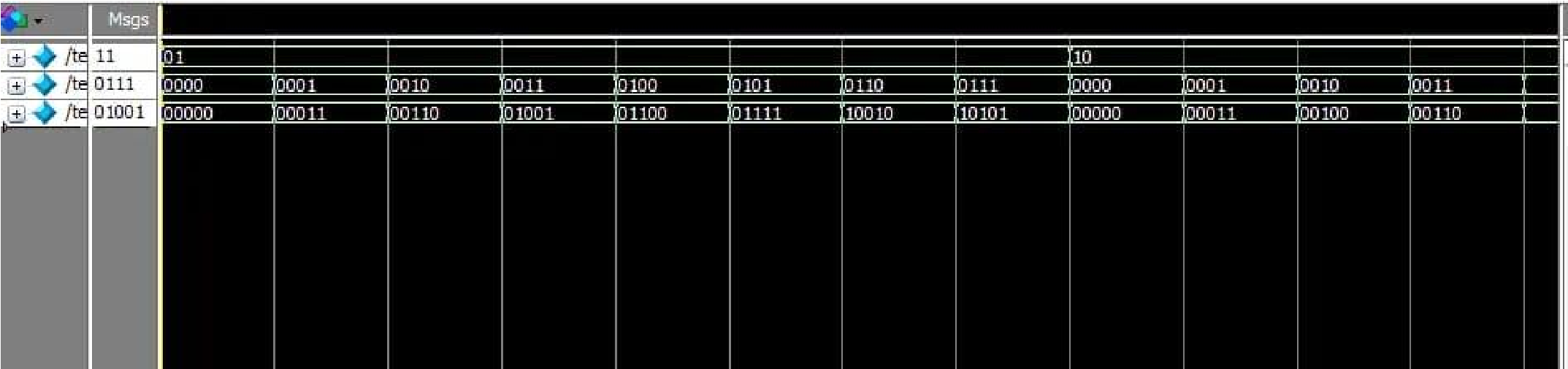
# 3.The Rom module

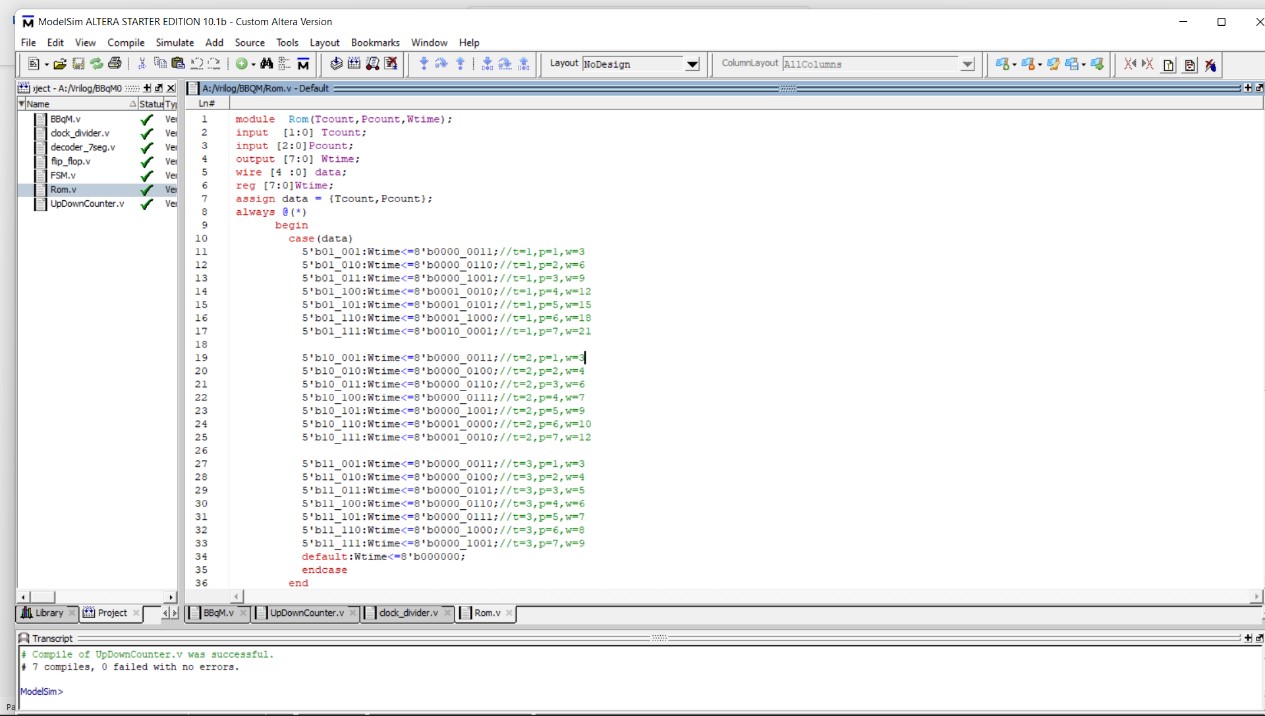
ROM is declared as a one-dimension array of integers holding the Wtime values. The index into this array could be constructed by concatenating Tcount and Pcount. For example, for Pcount = 5 and Tcount = 1, the array index in binary will be “01101”. ‘&’ is the concatenation operator in

Verilog

## • ROM can be implemented using casestatement like decoder modeling Wtime=3\*(Pcount+Tcount-1)/Tcount





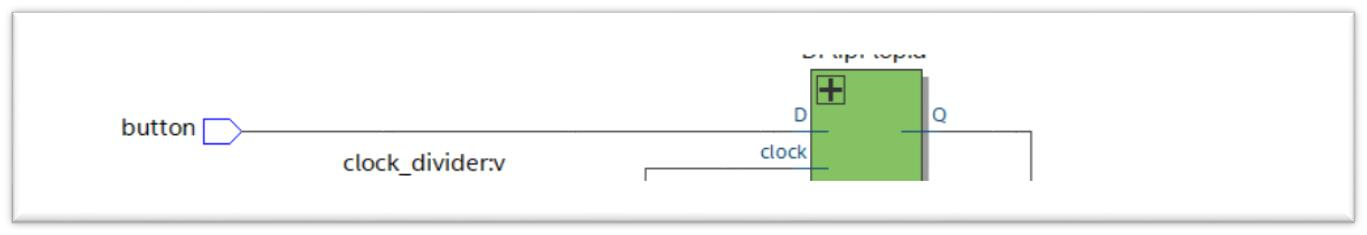


## This is our testbench for rom when t (from 1to3) and pcount (from 0 to 7)



**4.FlipFlop module**

## Input: D(button) Output : Q (which go to counter)

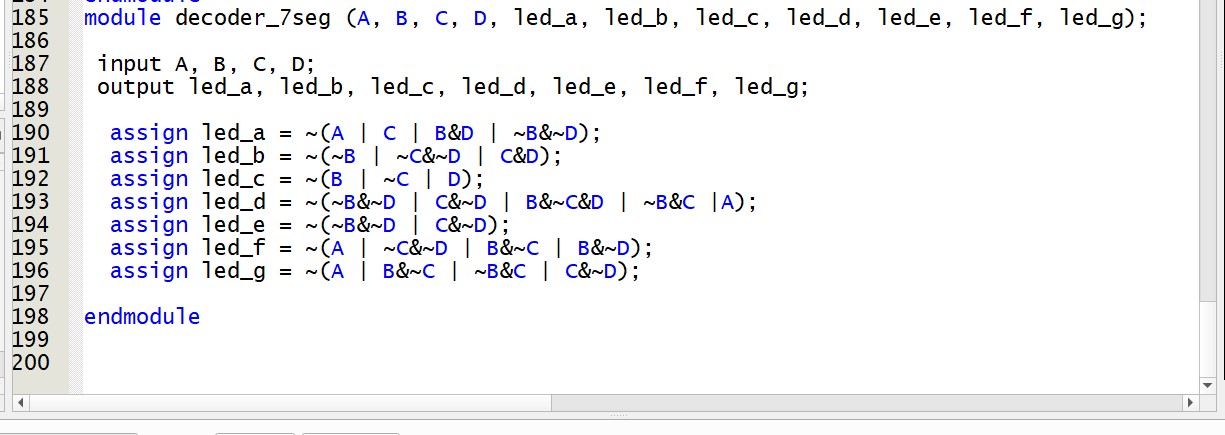




The purpose of that module is to Store binary data then it convert to bcd by decoder (next module)

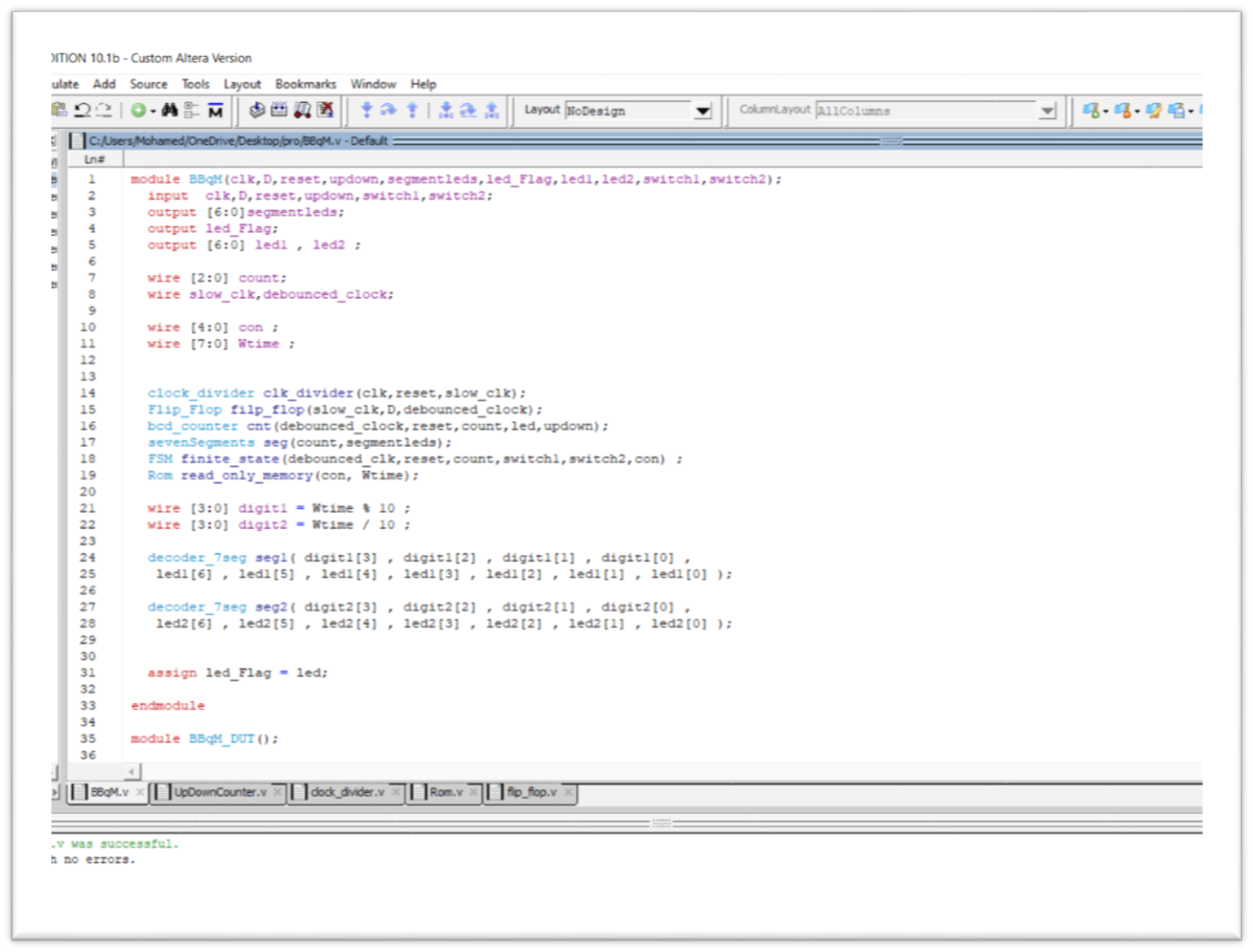
## 5.decoder\_7seg module

### It take the input from the up down counter and ROM and then to decoder then to display it on 7 segment on FPGA

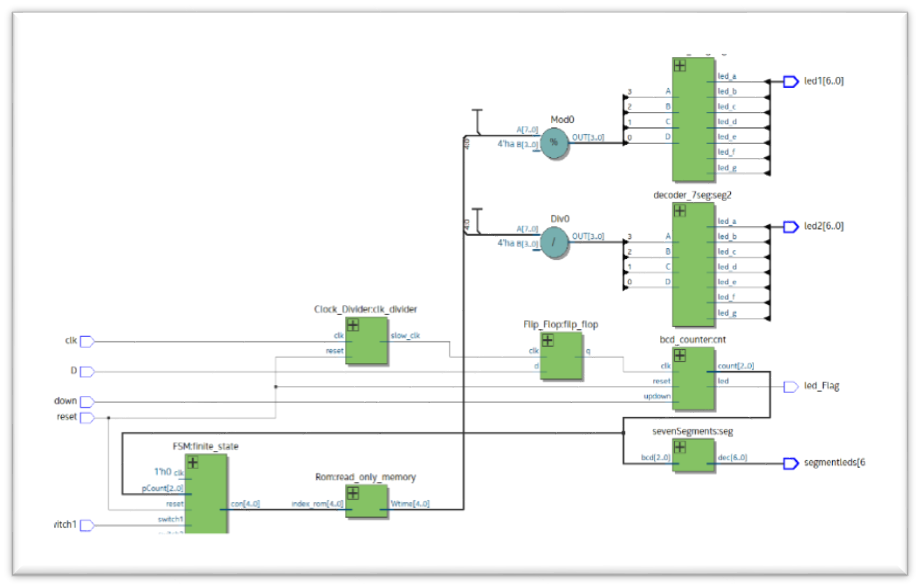


### 6.The Top module

#### In this module connction for the all module to work at the sime module



## All the project on RTL\_Viewer on quartus



## 7.FSM

