LAPLAND UNIVERSITY OF APPLIED SCIENCES

Embedded Systems Programming

Experiment Reports

A1301794
Mert Ertugrul Musul

TABLE OF CONTENTS

REPORT NO : 7

We supposed to use 7 segment display,

I modified all work, because in my opinion, it's easy to use a chip with codes. Altera is like Arduino very good stable chipset. Instead I imagined that what if I'd like to build that kind of basic thing, connect each other and see them work. I used National Instruments, virtual lab environment to preaper those, on a breadboard.

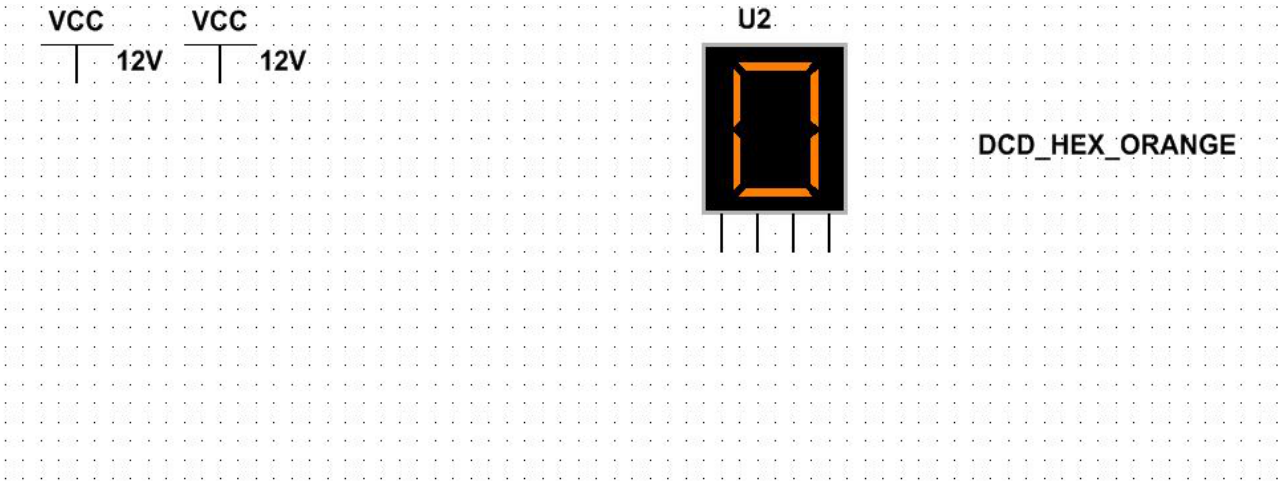
Bindary-coded-decimal (BCD) is a common way of encoding decimal numbers with 4 binary bits as shown below:

Decimal digit	0	1	2	3	4
BCD code	0000	0001	0010	0011	0100

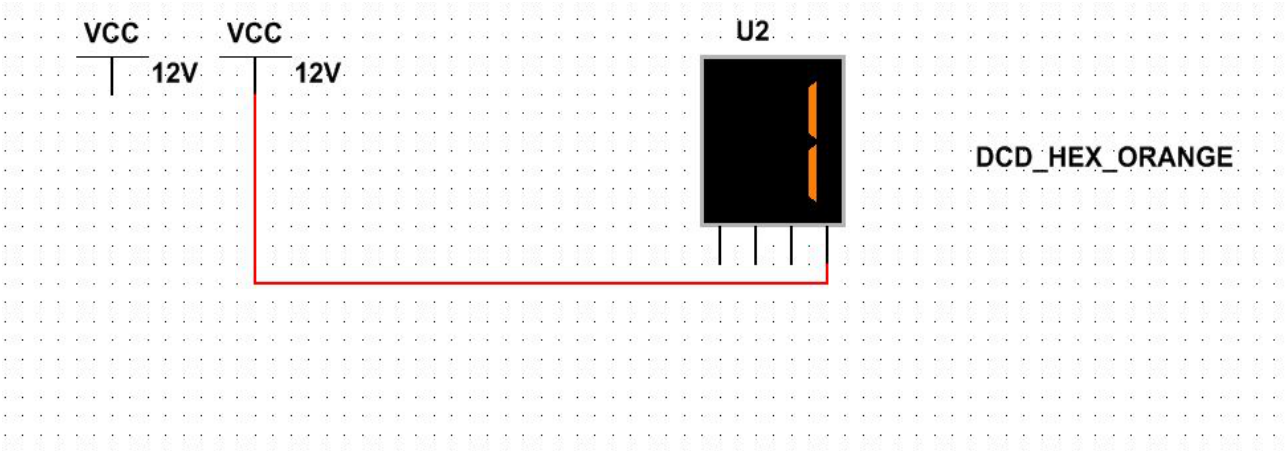
Decimal digit	5	6	7	8	9
BCD code	0101	0110	0111	1000	1001

We are going to follow this chart to print our numbers. Normally 7segment displays have 8 legs. 7 pins for each segment and 1 leg is called common cathode which is GND (Ground).

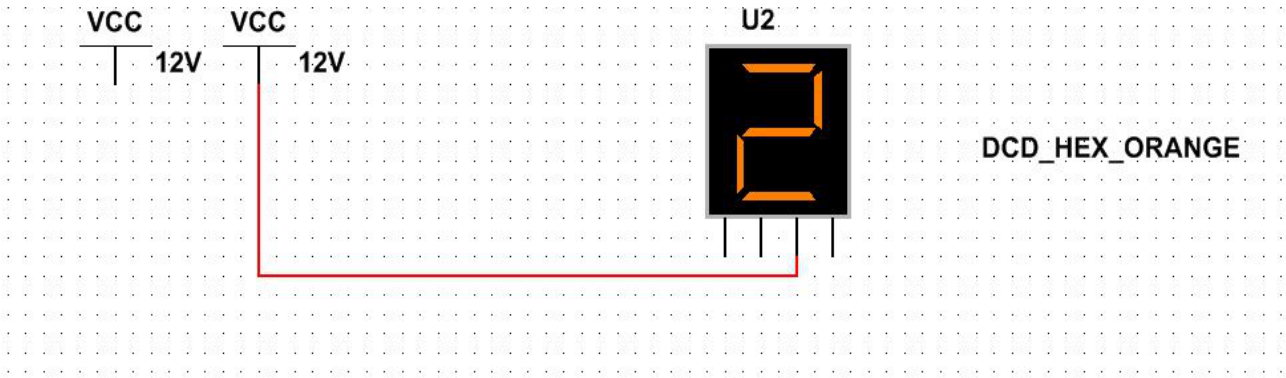
But here in this work, I prefer to use BCD segment, which we can also learn a little bit conversion from decimal to BCD code.



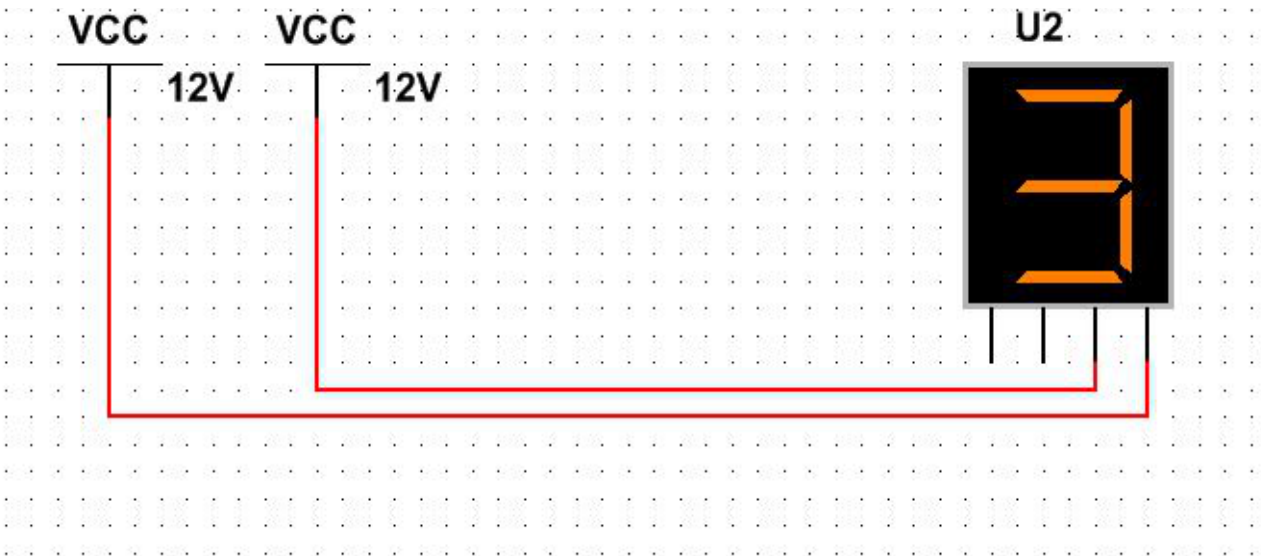
To print 0, we left all legs empty.



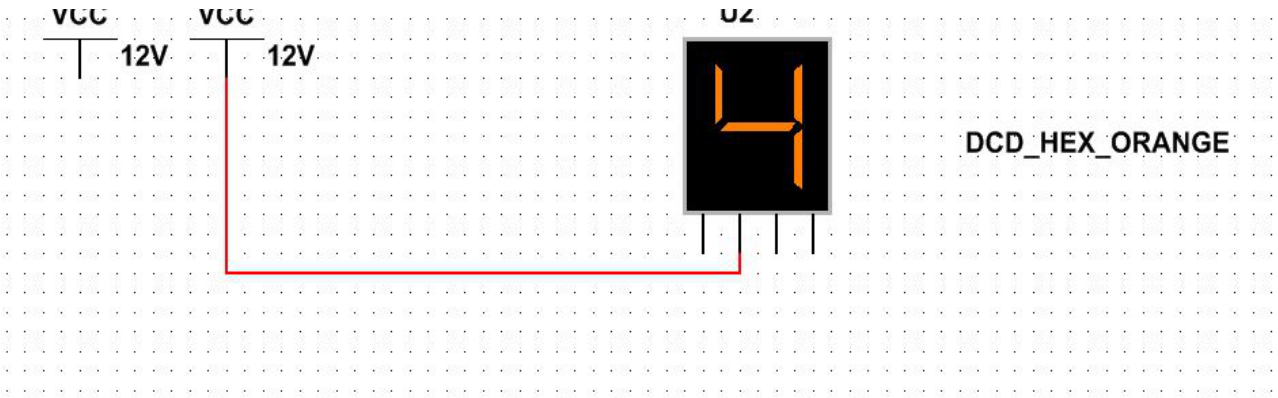
To print 1, we just connect the leg (0 , 0 , 0 , **1**) ,



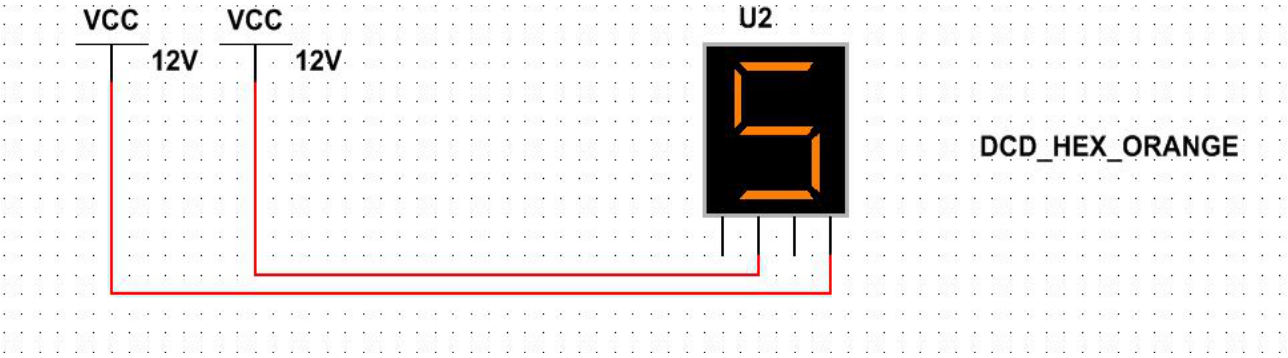
To print 2, we connect (0 , 0, 1, 0),



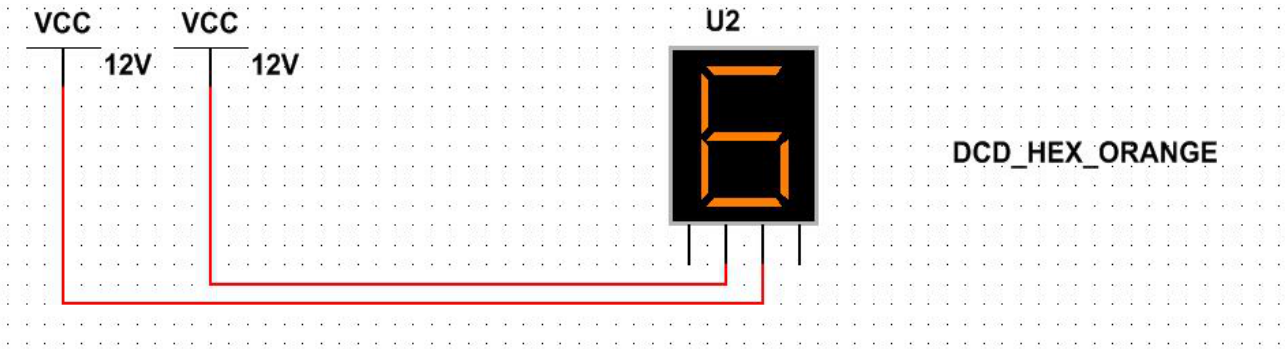
To print 3, we connect two legs (0 , 0 , 1, 1)



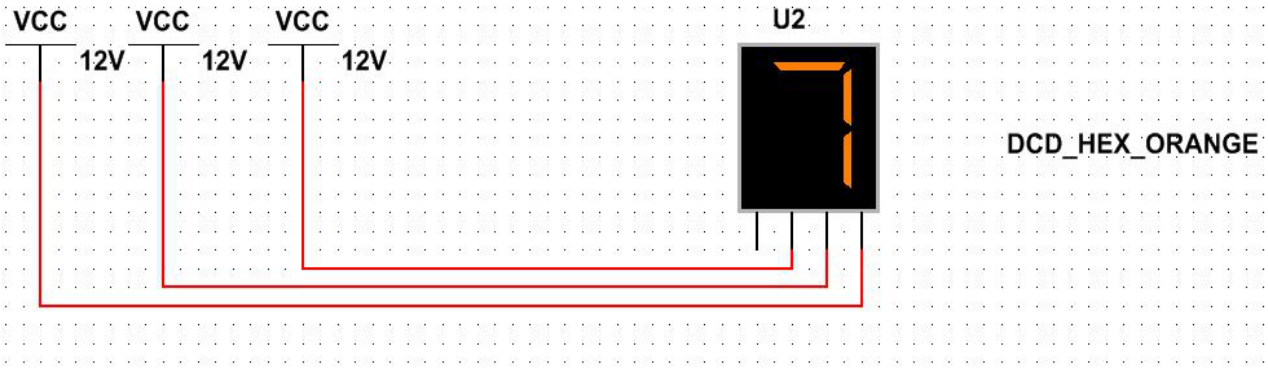
To print 4, we connect (0, 1, 0, 0),



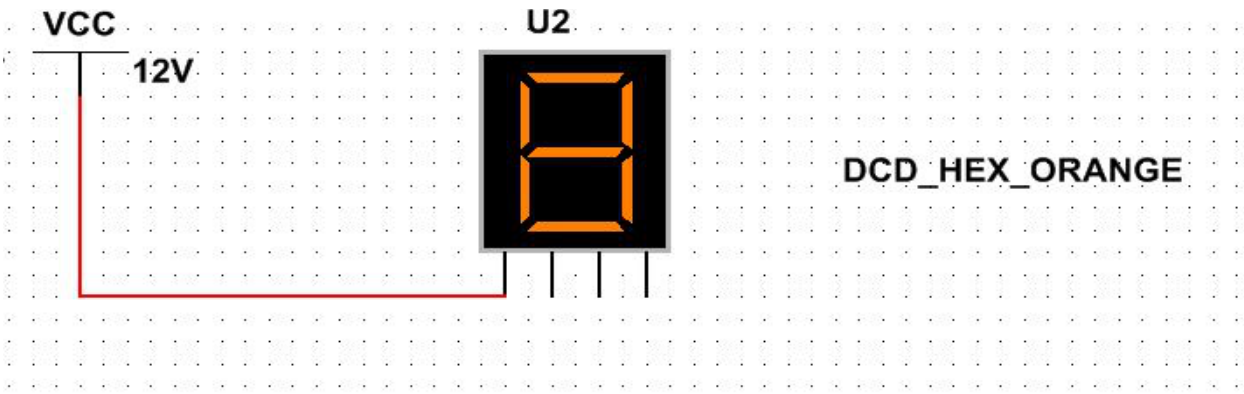
To print 5, we connect (0, 1, 0, 1),



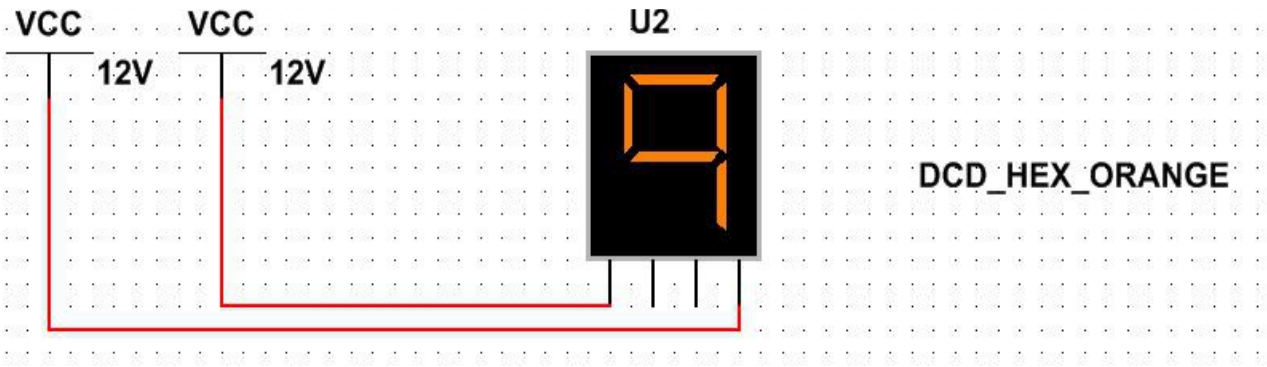
To print 6, (0, 1, 1, 0),



To print 7, we connect (0, 1, 1, 1) ,



To print 8, we connect (1, 0, 0, 1)

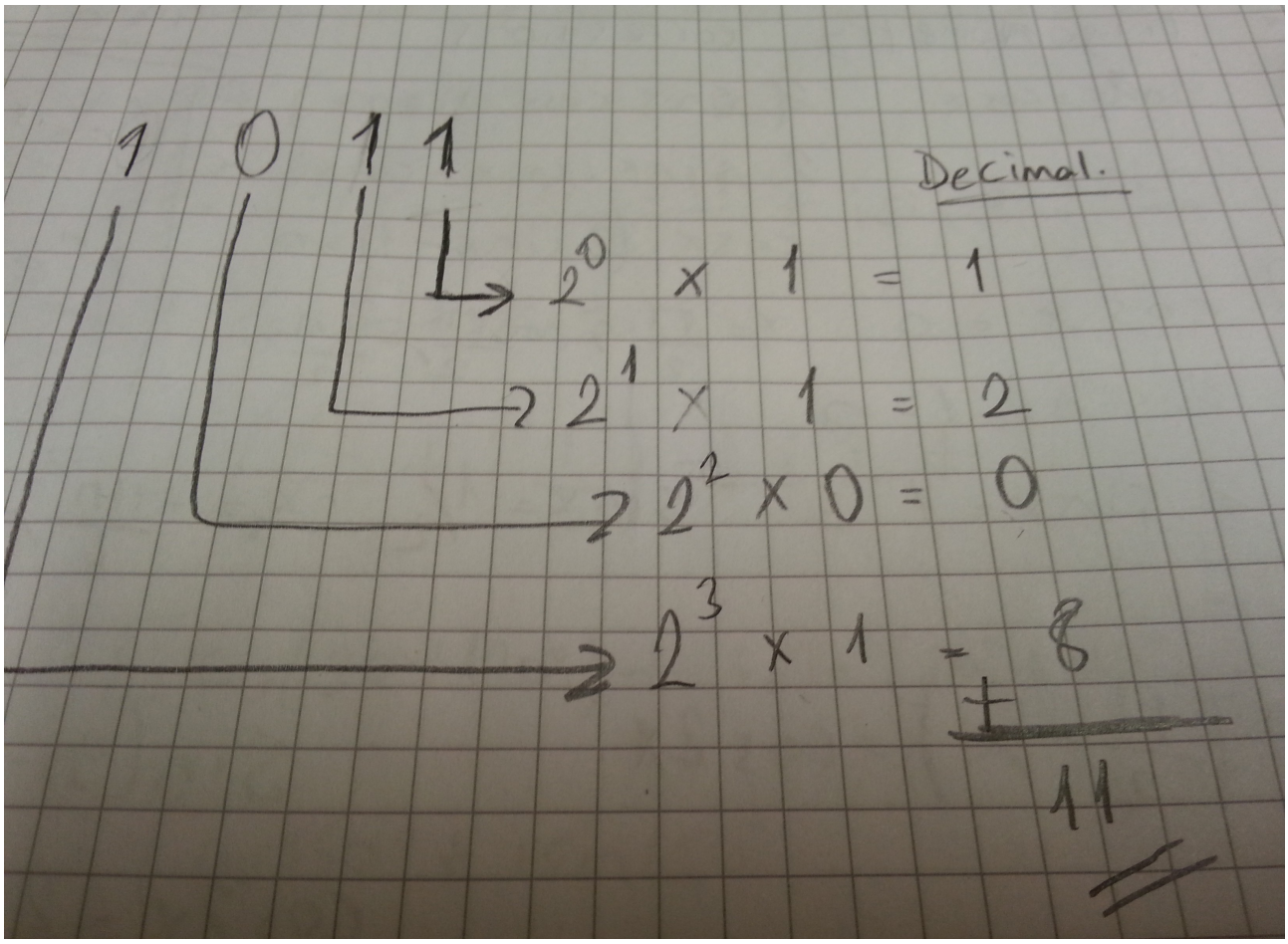


To print 9, we connect (1, 0, 0 1)

So, what we do while converting,

On the internet, I couldn't find a good and easy explanition so I wanted to tell what logic I use.

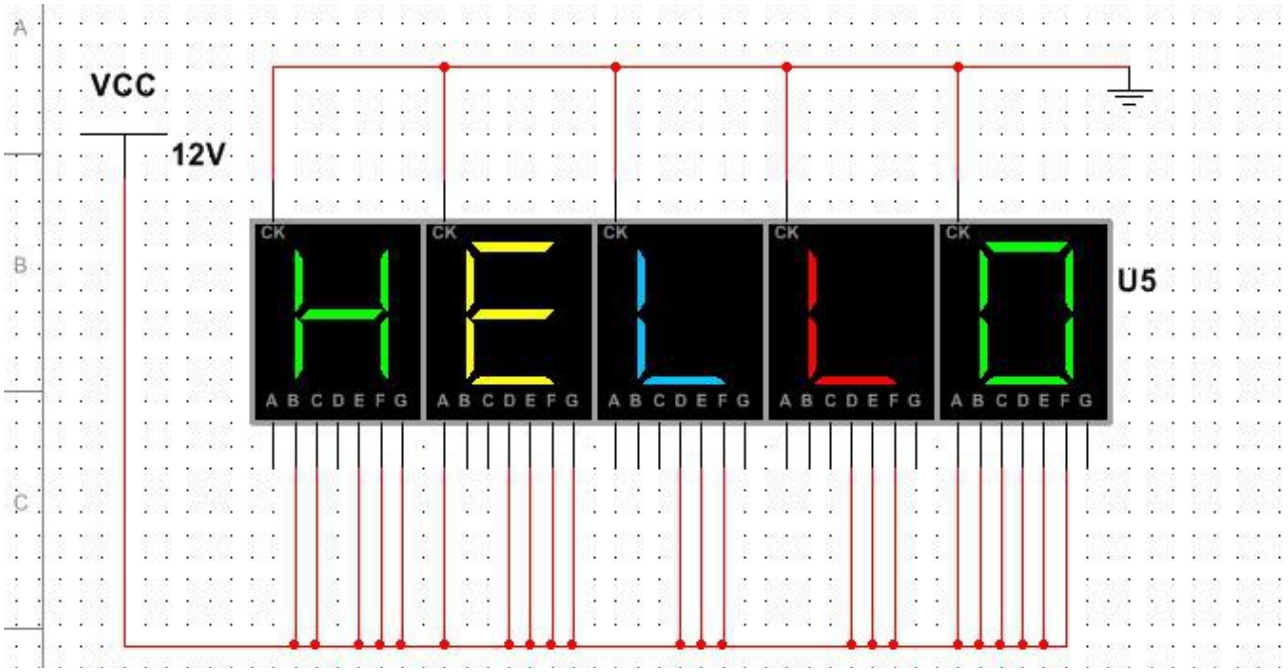
So we connect power to the pins in this row to get what decimal we want.



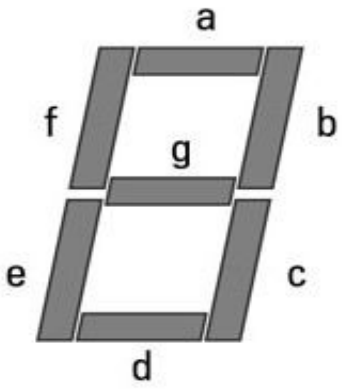
REPORT NO: 8

I modified the all task, what we supposed to do is writing flow chart to print HELLO on display.

I used National Instruments, virtual lab software to build this.



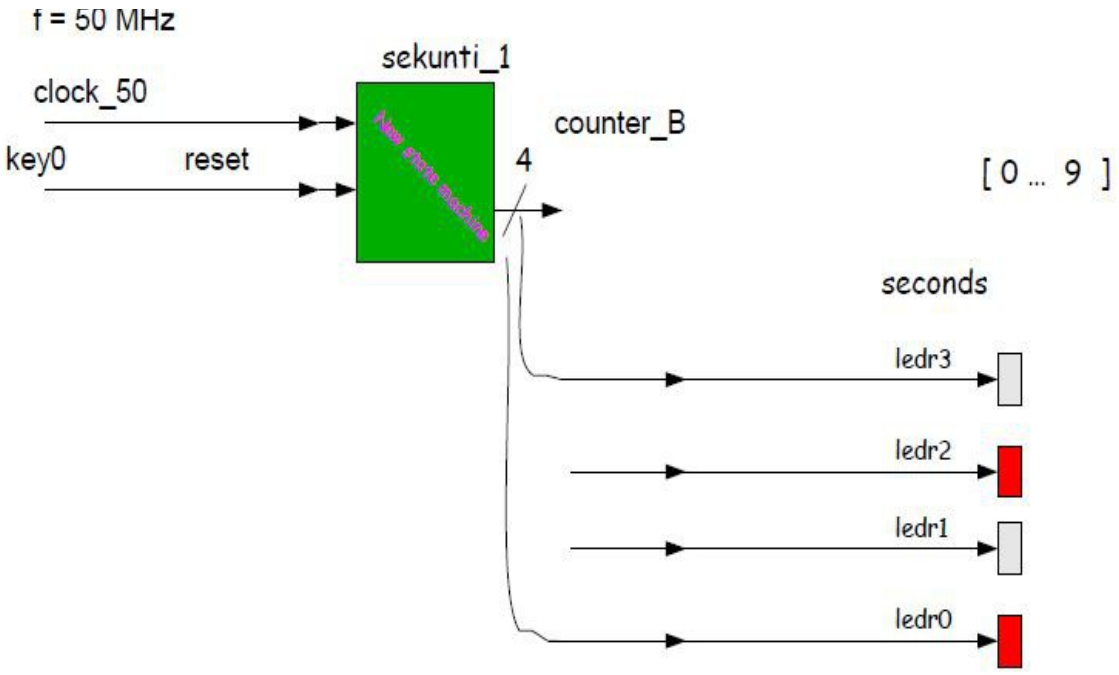
I used here 7 segment display. I connected power to the pins to light up right leds. And also I need to connect to common pins to the ground.

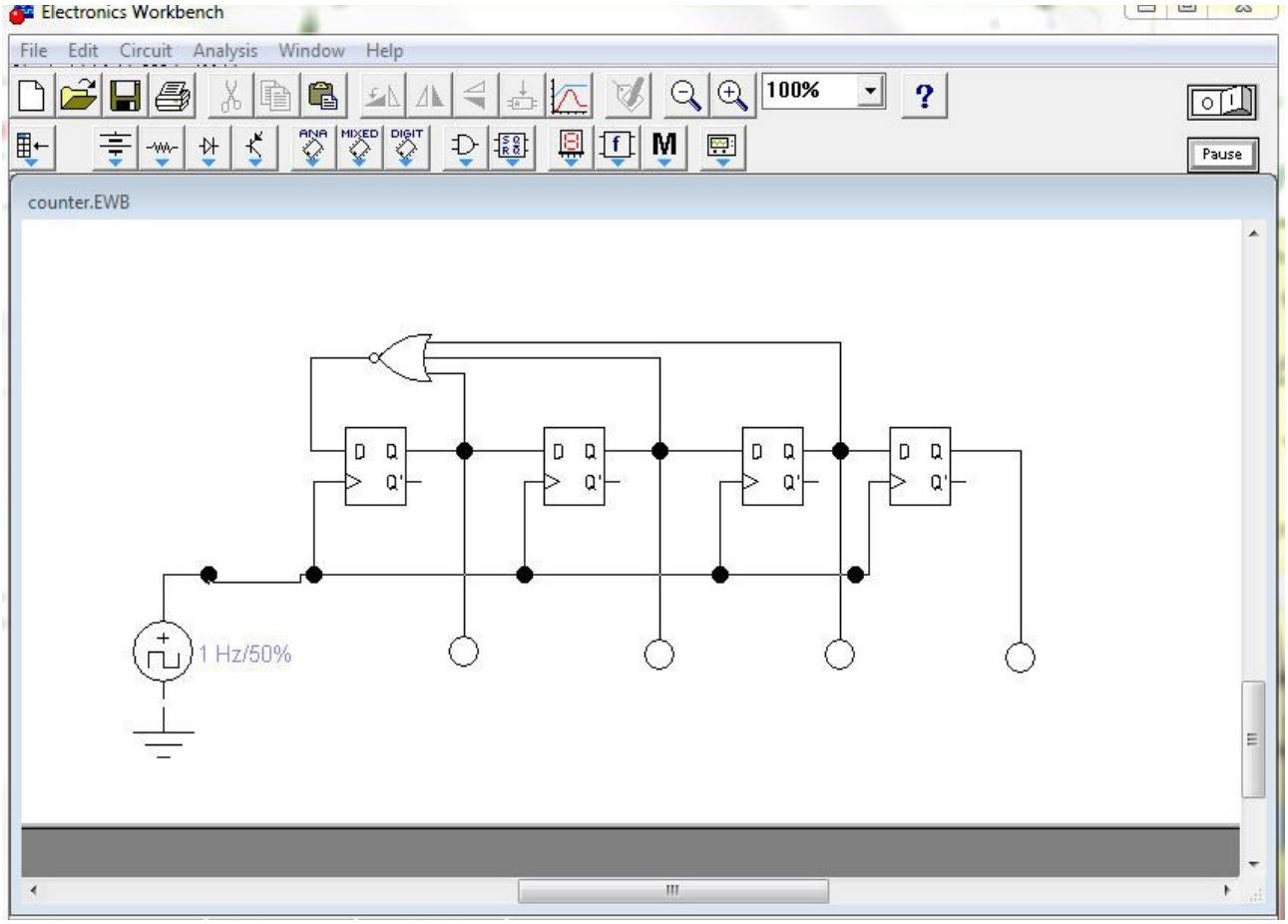


This is our ”led map” that we can understand which pins we should connect power to light up.

REPORT NO : 9

We are supposed to do a counter. As following the guide I've done it on HDL designer but again I wanted to modify all task. I built the counter's data sheet. I used Electronic Workbench virtual circuit.





Let's start which components I use here, D-Flip Flop, Clock, 3 Input NOR Gate and Leds.

What is Flip- Flop ?

In electronics, a **flip-flop** or**latch** is a circuit that has two stable states and can be used to store state information. A flip-flop is a bistable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in sequential logic. Flip-flops and latches are a fundamental building block of digital electronics systems used in computers, communications, and many other types of systems.

And D Flip Flop,

The D flip-flop is widely used. It is also known as a "data" or "delay" flip-flop.

The D flip-flop captures the value of the D-input at a definite portion of the clock cycle (such as the rising edge of the clock). That captured value becomes the Q output. At other times, the output Q does not change.The D flip-flop can be viewed as a memory cell, a zero-order hold, or a delay line.

Truth Table of D flip flop:

Clock	D	Q_{next}
Rising edge	0	0
Rising edge	1	1
Non-Rising	X	Q

NOR Gate :

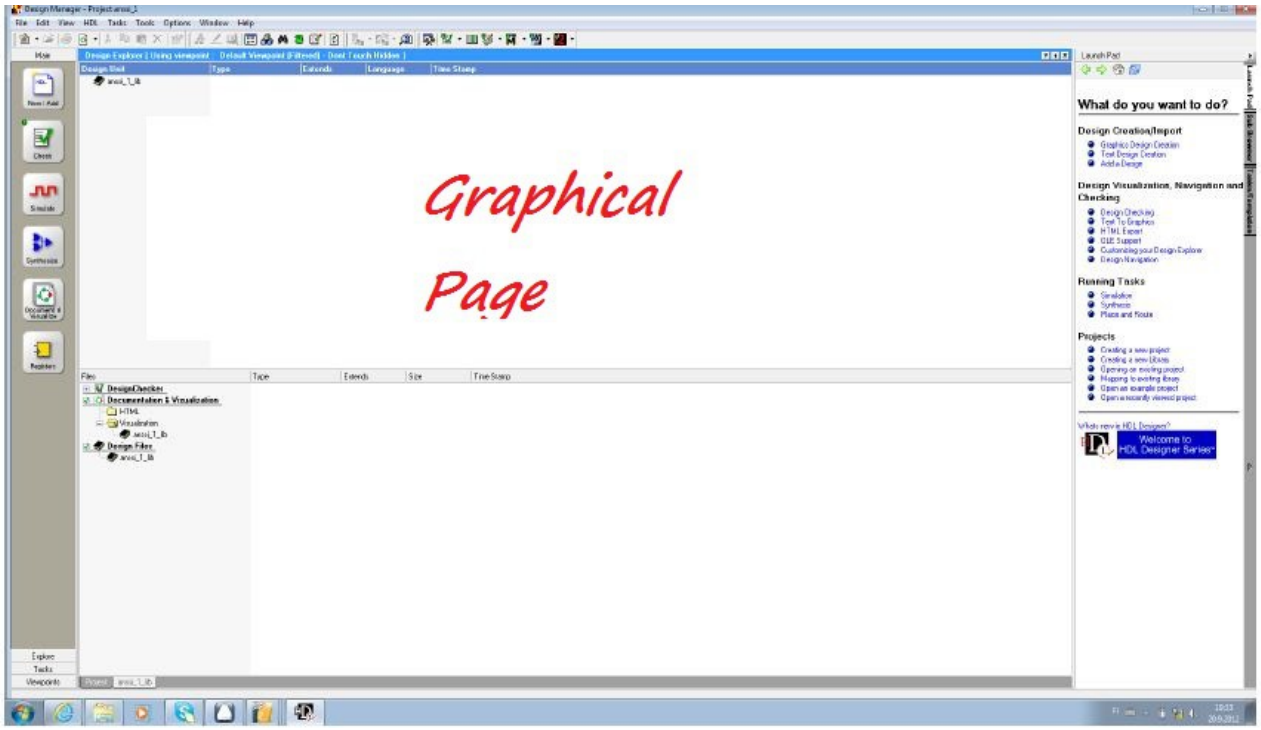
The **NOR gate** is a digital logic gate that implements logical NOR- it behaves according to the truth table to the right. A HIGH output (1) results if both the inputs to the gate are LOW (0); if one or both input is HIGH (1), a LOW output (0) results. NOR is the result of the negation of the OR operator. It can also be seen as an AND gate with all the inputs inverted. NOR is a functionally complete operation—NOR gates can be combined to generate any other logical function. By contrast, the OR operator is *monotonic* as it can only change LOW to HIGH but not vice versa.

Truth table,

INPUT		OUTPUT
A	B	A NOR B
0	0	1
0	1	0
1	0	0
1	1	0

REPORT NO : 1

We learnt how to prepare hdl designer for a new project start.

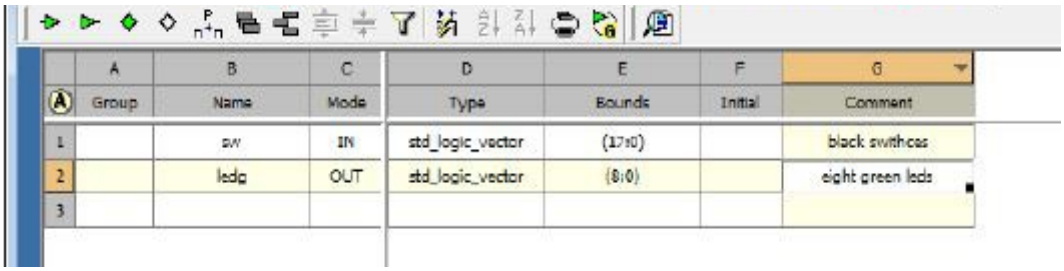


We draw our graphical solutions here, components like flow charts, state machines, block diagrams.

Basically we are getting know with Hdl designer tool.

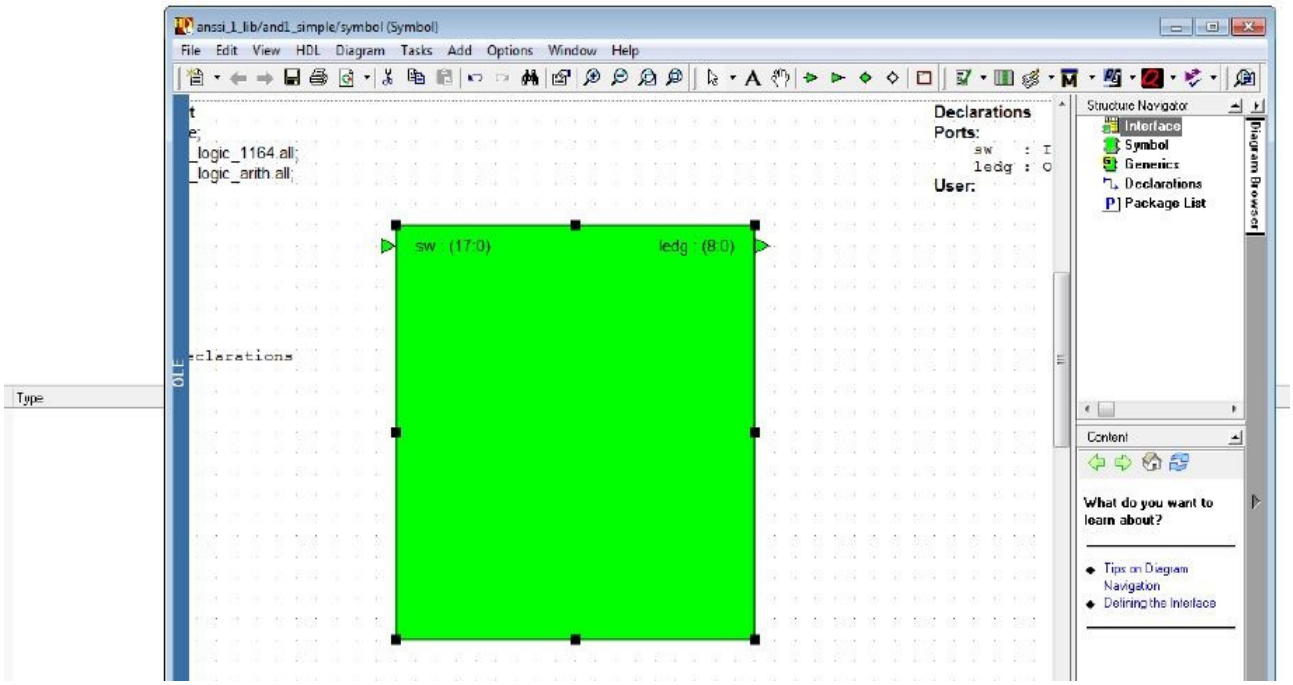
REPORT NO : 2

First time we start to do something serious ! We write the names of input bits and output bits. The picture below.



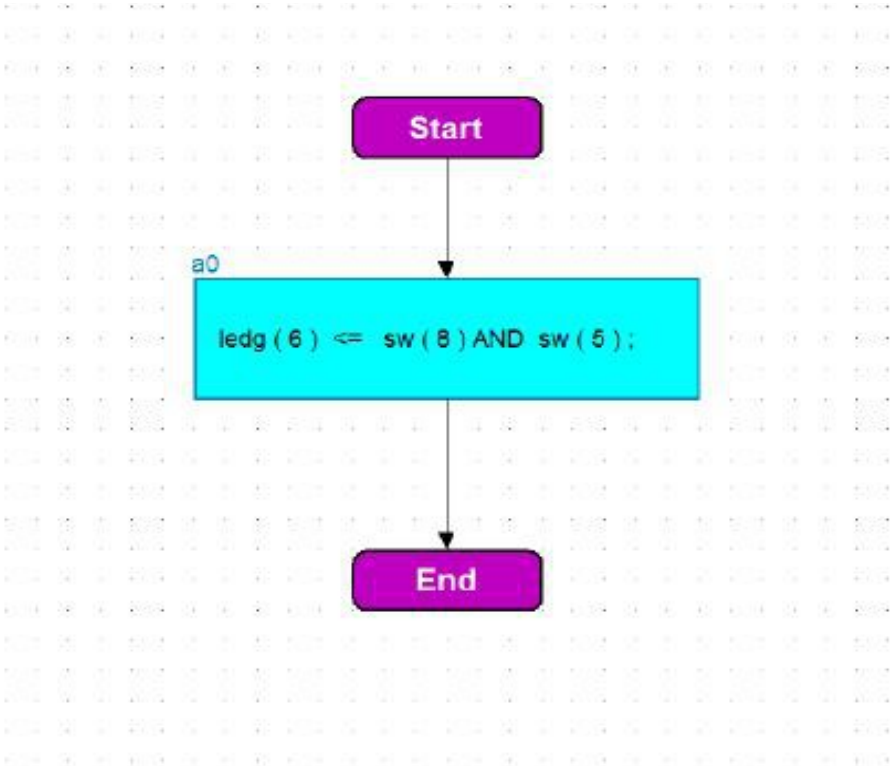
	A	B	C	D	E	F	G
	Group	Name	Mode	Type	Bounds	Initial	Comment
1		sw	IN	std_logic_vector	(17:0)		black swithces
2		ledg	OUT	std_logic_vector	(8:0)		eight green leds
3							

Now we want to see them as graphical scheme,

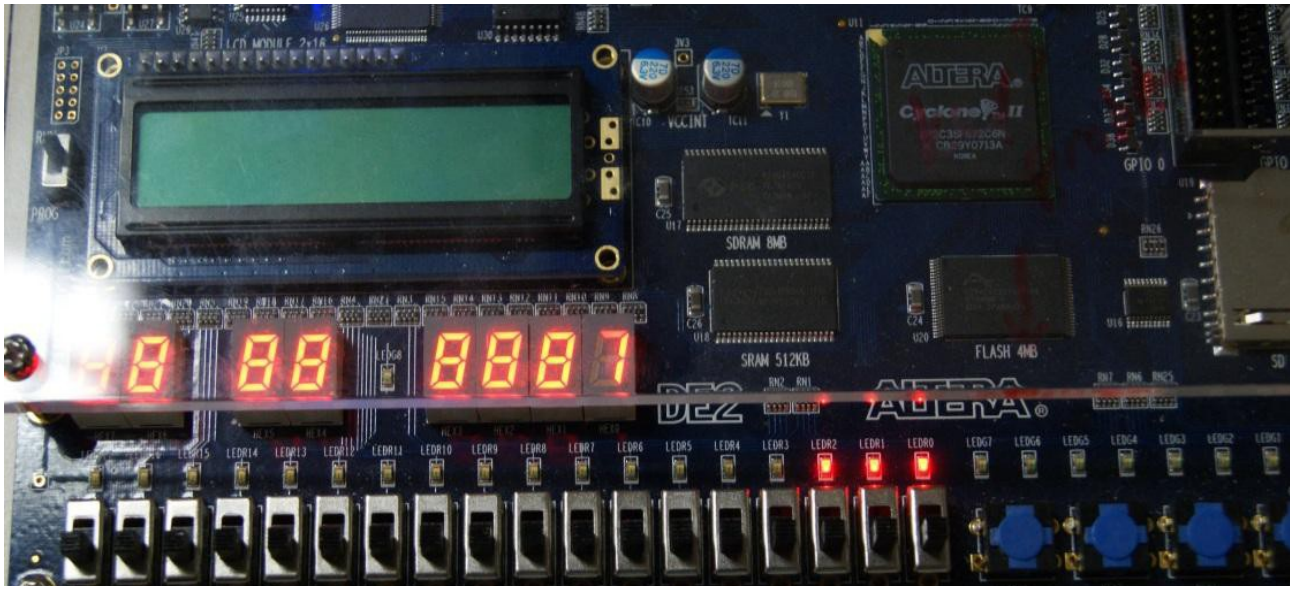


And also we want to use flow chart tool, what is flowchart

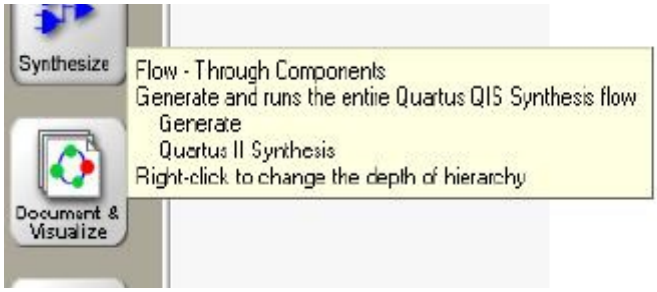
A flowchart is a type of diagram that represents an algorithm or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows. This diagrammatic representation illustrates a solution to a given problem. Process operations are represented in these boxes, and arrows; rather, they are implied by the sequencing of operations. Flowcharts are used in analyzing, designing, documenting or managing aprocess or program in various fields



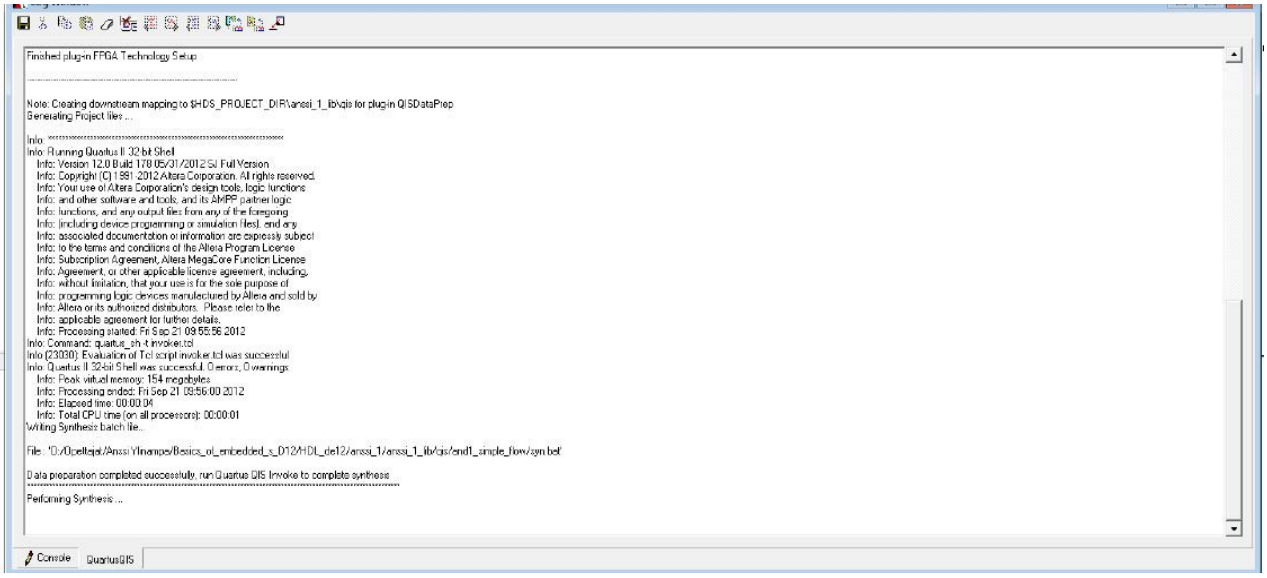
After all, we want to see this program in real life ! And we are going to use Cyclone II FPGA chip



So we are going synthesis this program to make ready for Quartus VHDL synthesis tool.

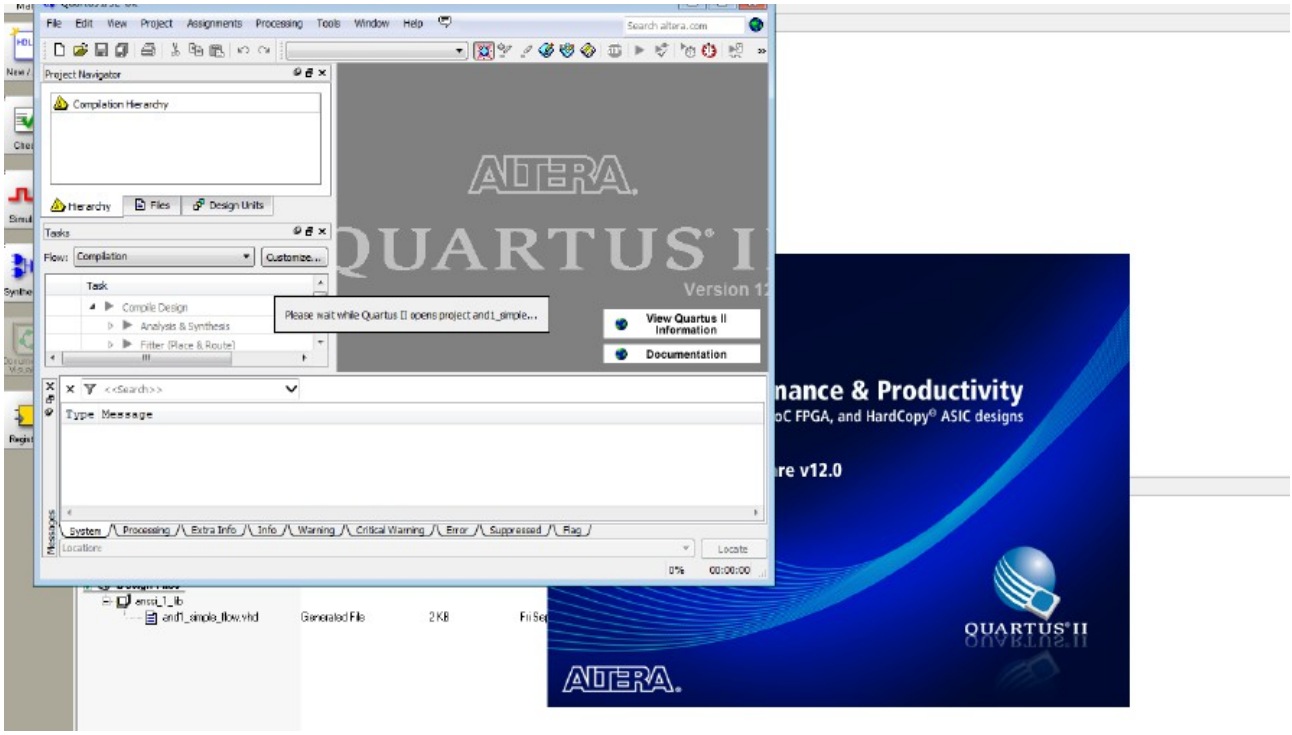


What is VHDL, **VHDL (VHSIC Hardware Description Language**) is a hardware description language used in electronic design automation to describe digital and mixed-signal systems such as field-programmable gate arrays and integrated circuits. VHDL can also be used as a general purpose parallel programming language.

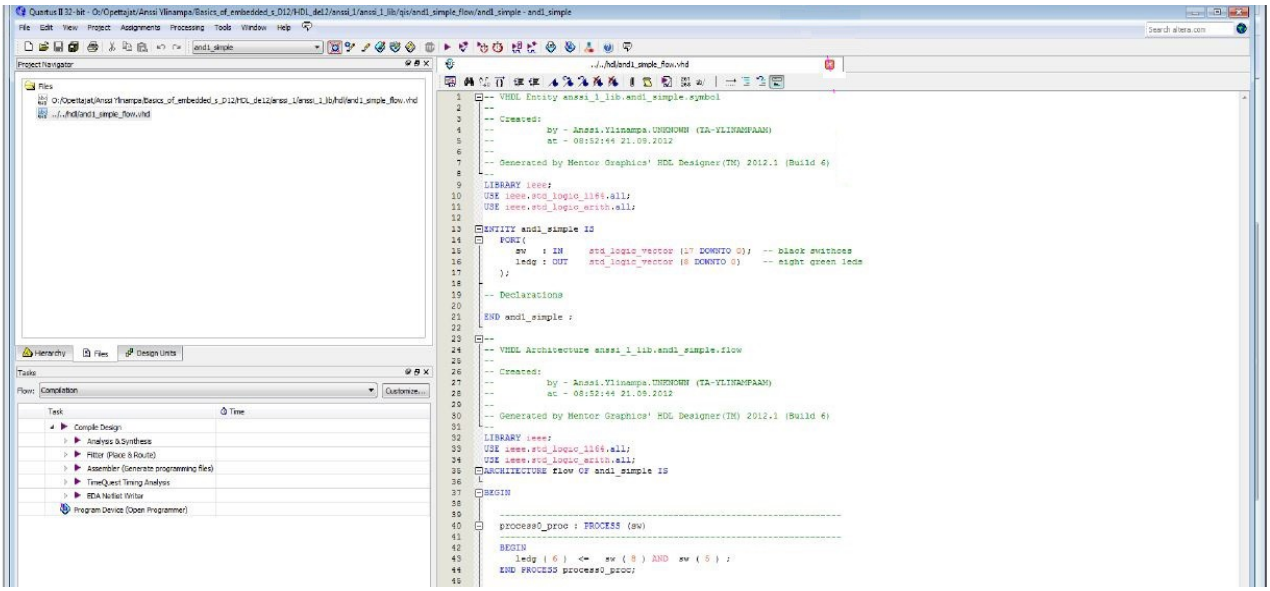


And HDL Designer is generating files,

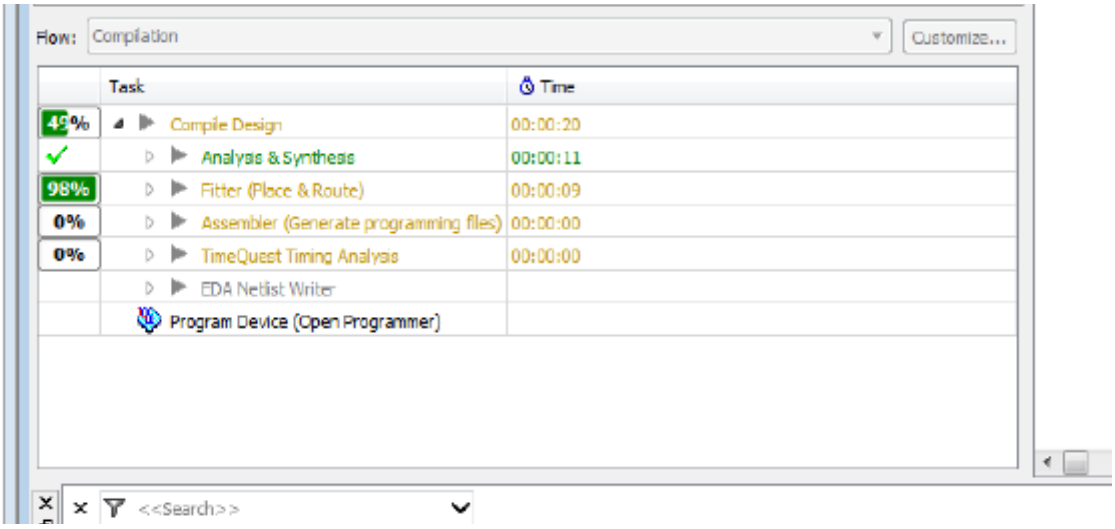
Then we are ready to use Quartus II, to program our chip.



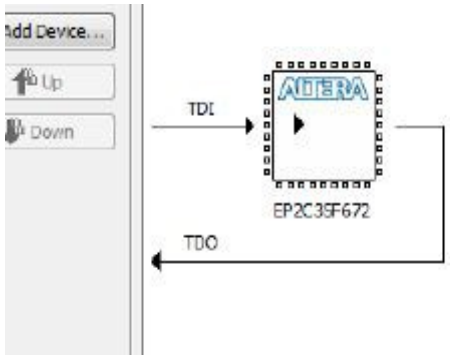
And our codes,



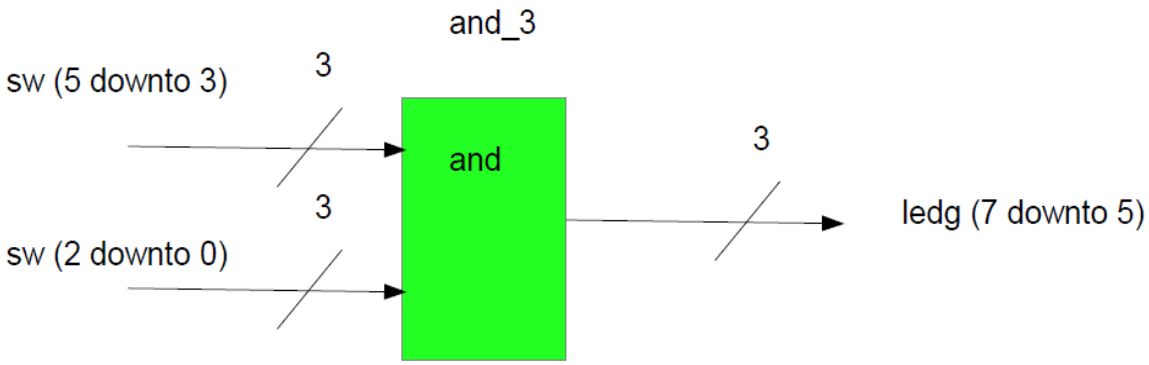
Now we are going to make the compilation to test our program in real cyclone chip.



Coming to the last step, we need to load our program to Altera, so we need , Altera USB buster JTAG programming tool.



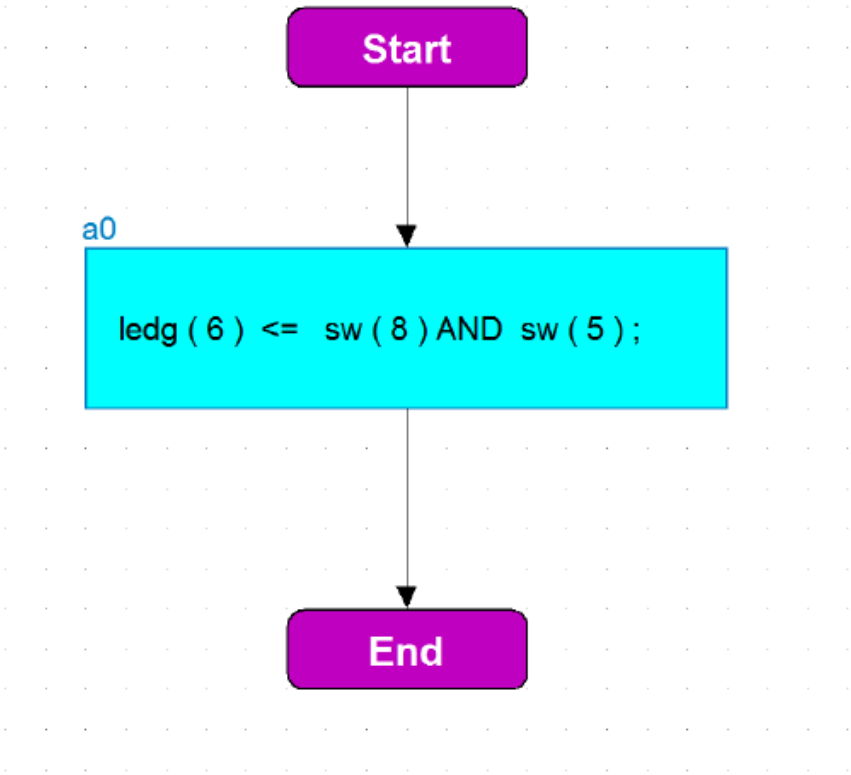
REPORT NO : 3



Now we are going to do this statement, which is very easy.

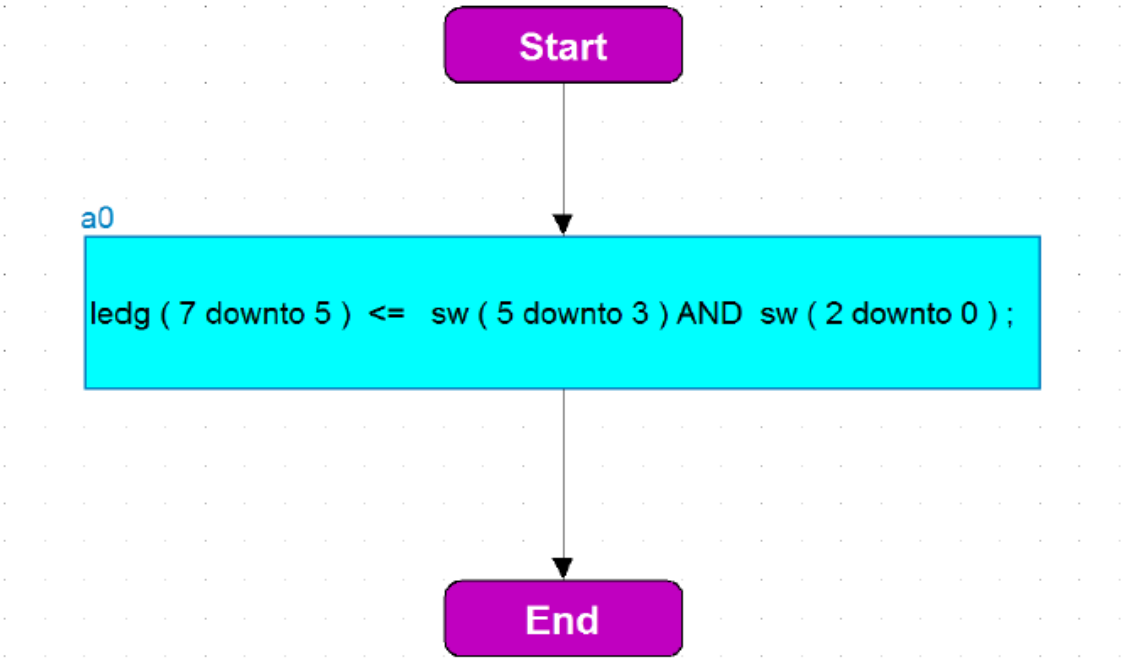
At first we will copy what we have done last and just modify it. It's the magic ! :)

After that let's make this flowchart !



Which probably must look like from old version,

then we will change this into :



Then we will save it and run it.

That's all.