

TUTORIALS FOR MAX3000A CPLD DEVICES

WEL LAB

Practical-10

3-BIT-BINARY- ADDER

By-21012021003_AMIT GOSWAMI

CONTENTS:

INTRODUCTION

GETTING STARTED.....

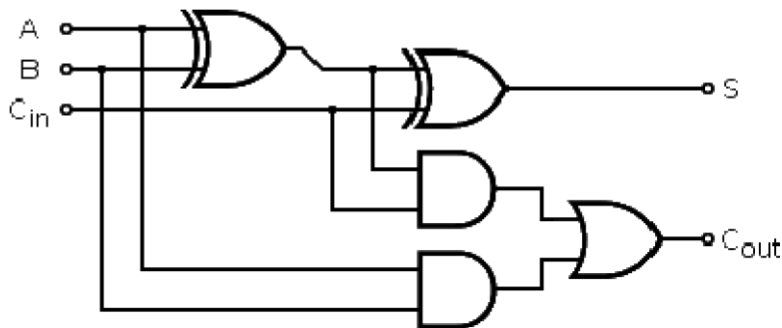
DESIGN ENTRY.....

PIN ASSIGNMENT.....

PROGRAMMING CPLD DEVICE.....

REFERENCES.....

Introduction: In electronics, an **adder** or **summer** is a digital circuit that performs addition of numbers. In modern computers adders reside in the arithmetic logic unit (ALU) where other operations are performed. Although adders can be constructed for many numerical representations, such as Binary-coded decimal or excess-3, the most common adders operate on binary numbers. In cases where two's complement or one's complement is being used to represent negative numbers, it is trivial to modify an adder into an adder-subtractor. Other signed number representations require a more complex adder.



Get Started:

You begin this tutorial by creating a new Quartus II project. A project is a set of files that maintain information about your FPGA design. The Quartus II Settings File (.qsf) and Quartus II Project File (.qpf) files are the primary files in a Quartus II project. To compile a design or make pin assignments, you must first create a project.

1. In the Quartus II software, Select **File > New Project Wizard**. The **Introduction** page opens. See **fig-1**.

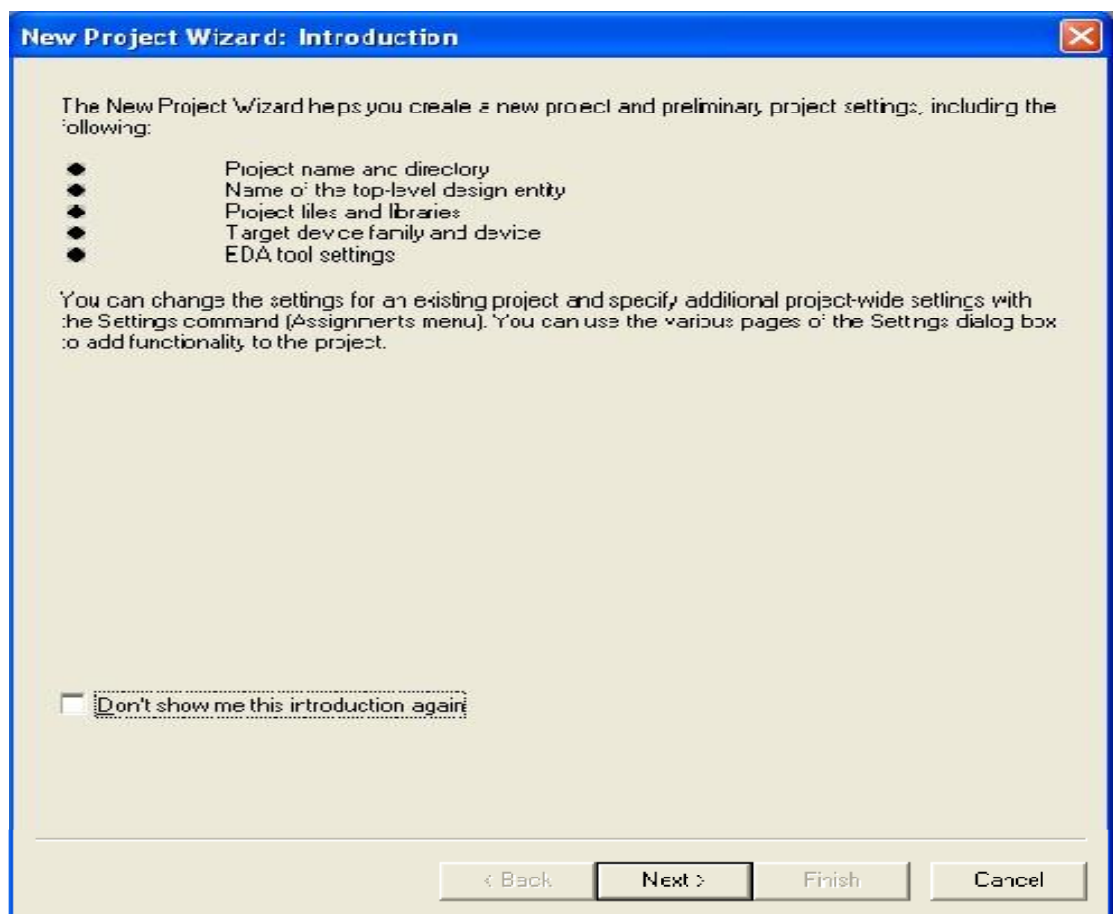


Fig-1

2. Click Next.

3. Enter the following information about your project:

a. What is the working directory for this project?

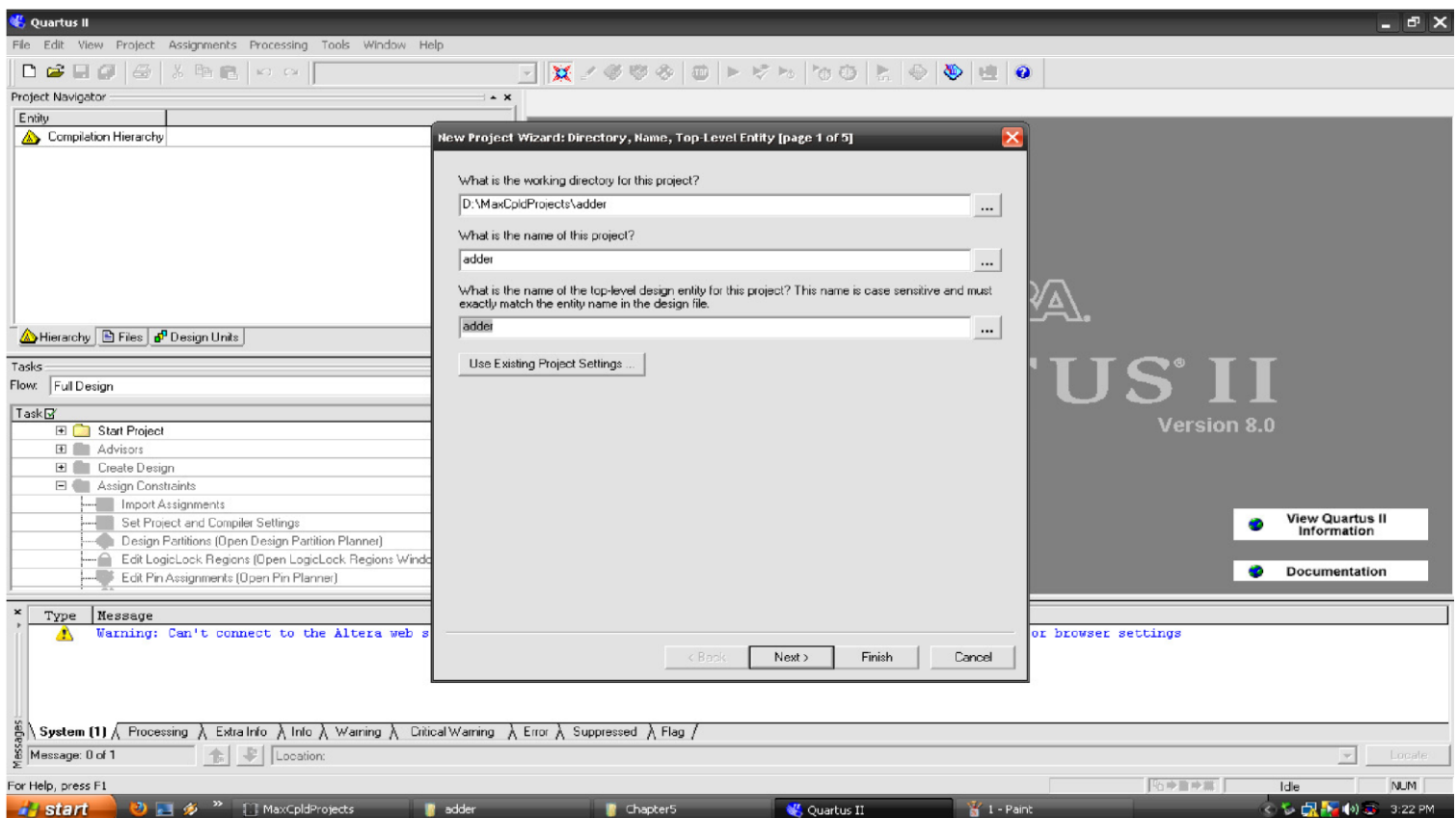
Enter a directory in which you will store your Quartus II project files for this design, for example, c:\altera\4-bit-binary-adder.

b. What is the name of this project?

Type adder.

c. What is the name of the top-level design entity for this project?

Type adder. See fig-2.



Note: File names, project names, and directories in the Quartus II software cannot contain spaces.

And the name of top module in the program should be same as given in the above window.

4. If you have any existing file of VHDL, or VERILOG HDL file then you can add those files to your project by simply clicking (...) bar and then click **Add** button and if you want to add more than one file simply click **Add All** button. Otherwise leave it blank and click next tab. In fig -3.

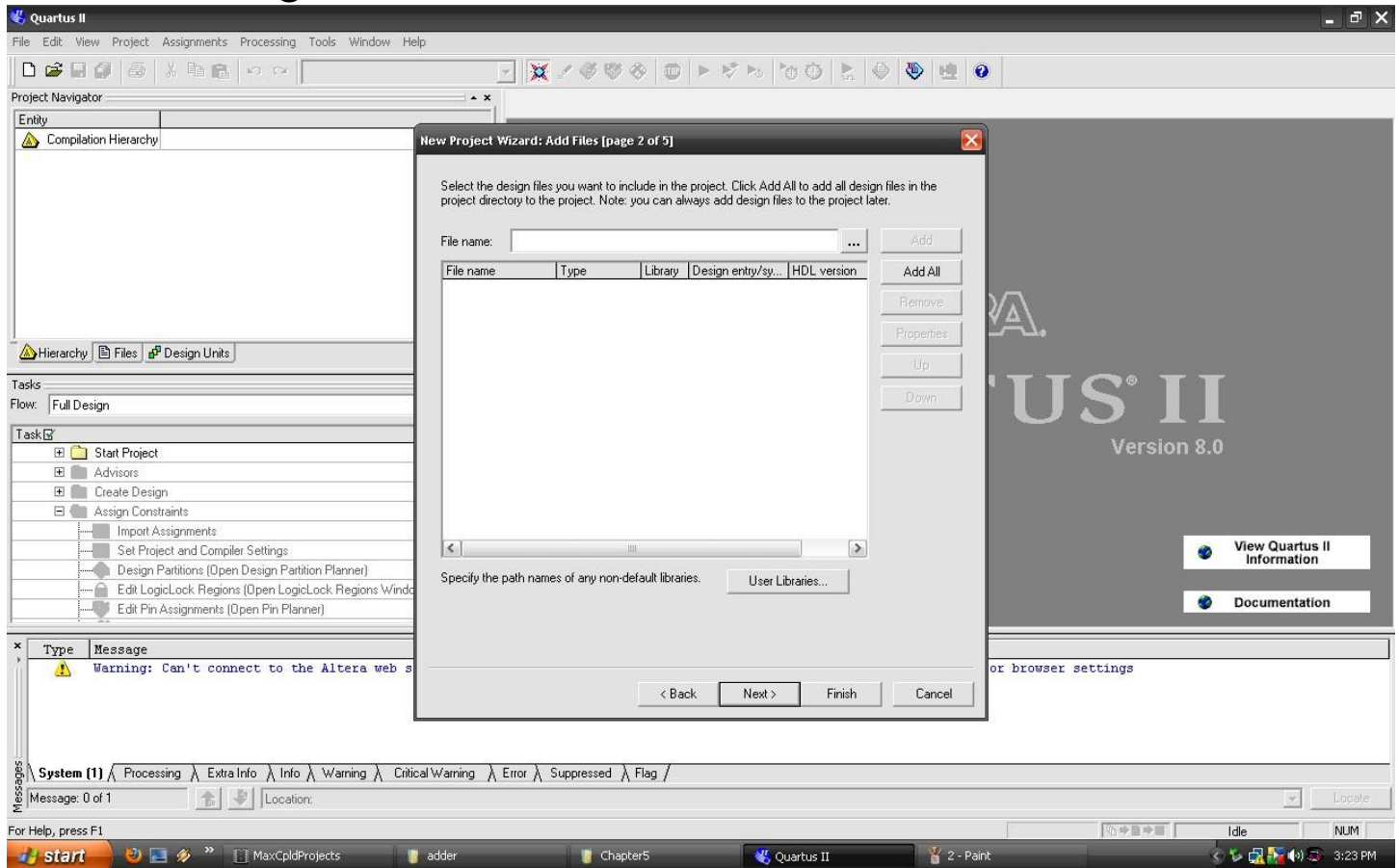


Fig-3

5. In the next step we have to select the **FPGA** or **CPLD** device which we are using on our board (select MAX 3000A) as shown in fig -4 make all the settings as show--n in the Fig-4.

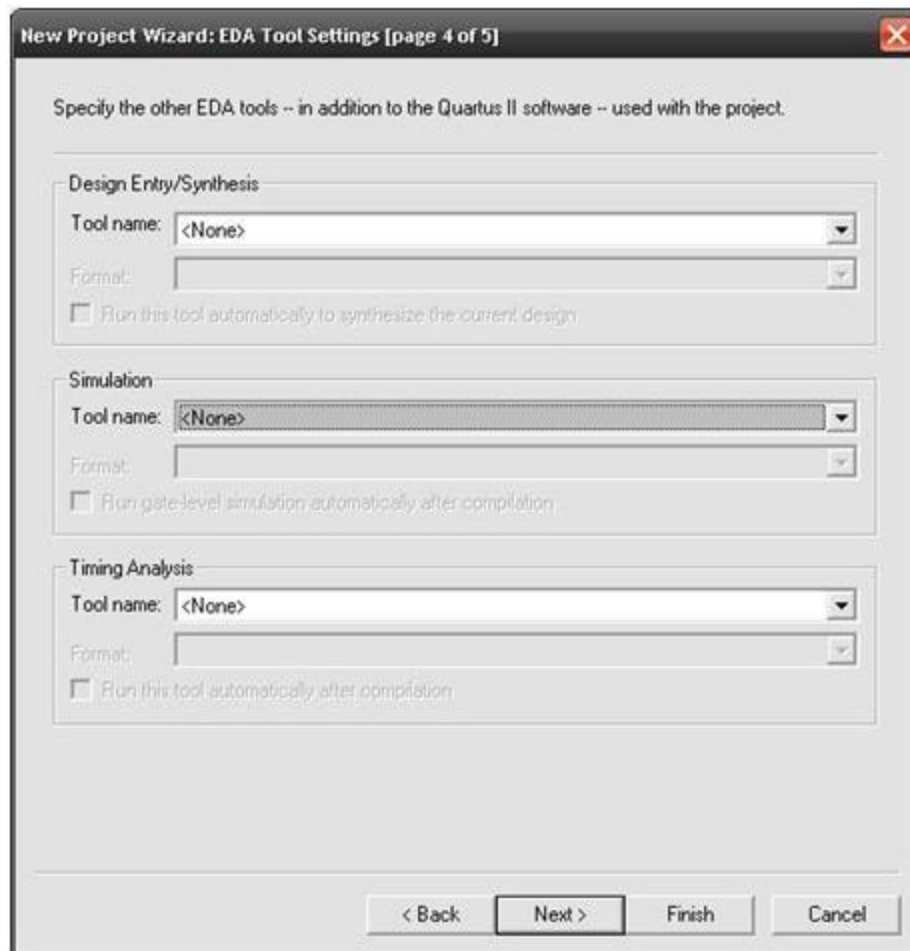


Fig-5

7. Click next , summary of project and its details are shown in the next window.fig-6

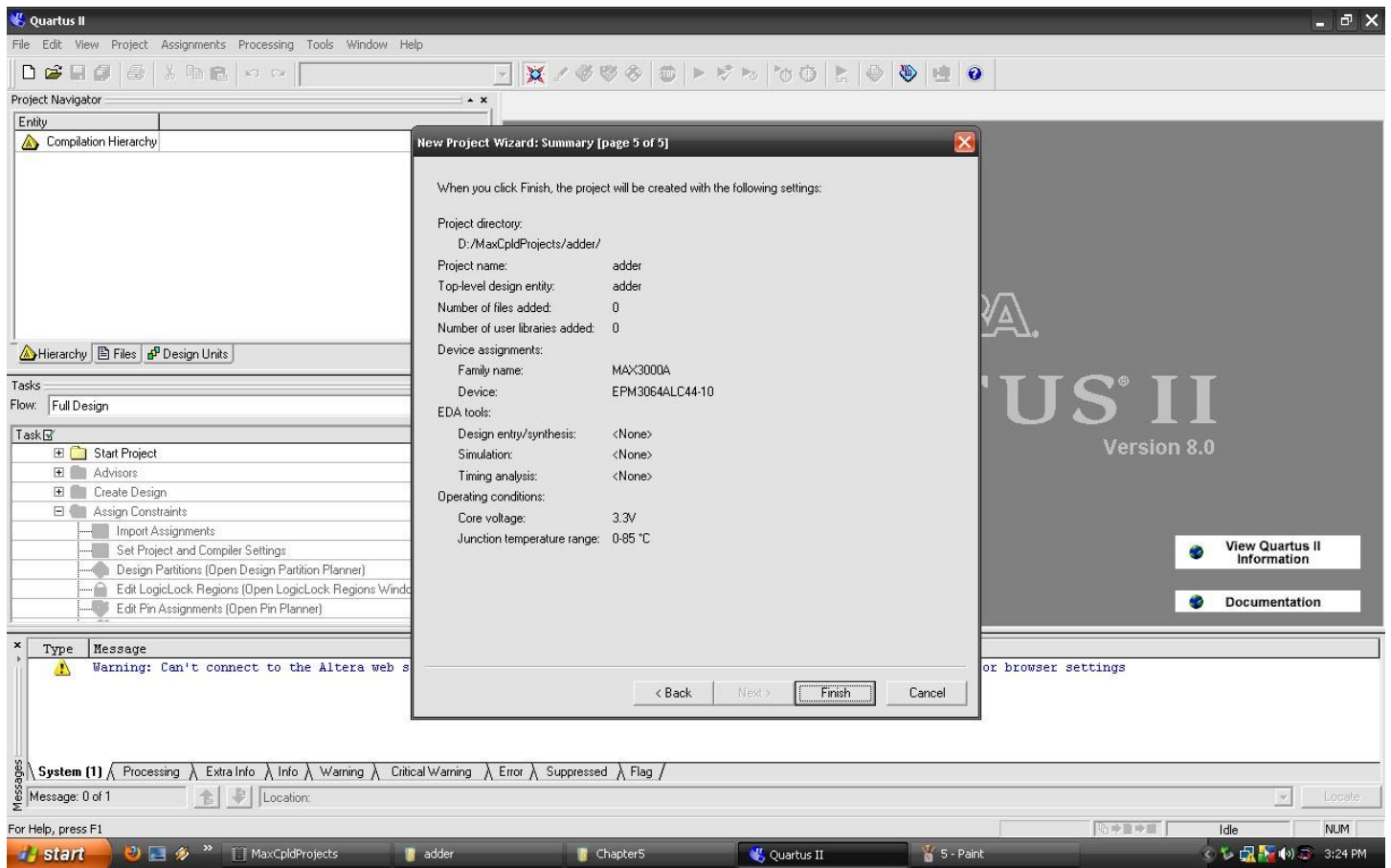


Fig-6

Design Entry: In the design entry phase, you use RTL or schematic entry to create the logic to be implemented in the device. You also make pin assignments, including pin placement information, and timing constraints that might be necessary for building a functioning design.

In the design entry step you create a schematic or Block Design File (**.bdf**) or Hdl file (Vhdl, Verilog) that is the top-level design. You will add library of parameterized modules (LPM) functions and use Verilog HDL code to add a logic block.

When creating your own designs, you can choose any of these methods or a combination of them.

- a. Choose **File > New > Verilog Hdl File/Schematic File** (see **Fig-7**) to create a new file, **Verilog.v file** which you will save as the toplevel design.



Fig-7

b. Click OK.

c. Choose File > Save As and enter the following information (see fig-8).

File name: multiplier.

Save as type: Verilog Hdl File(*.v).

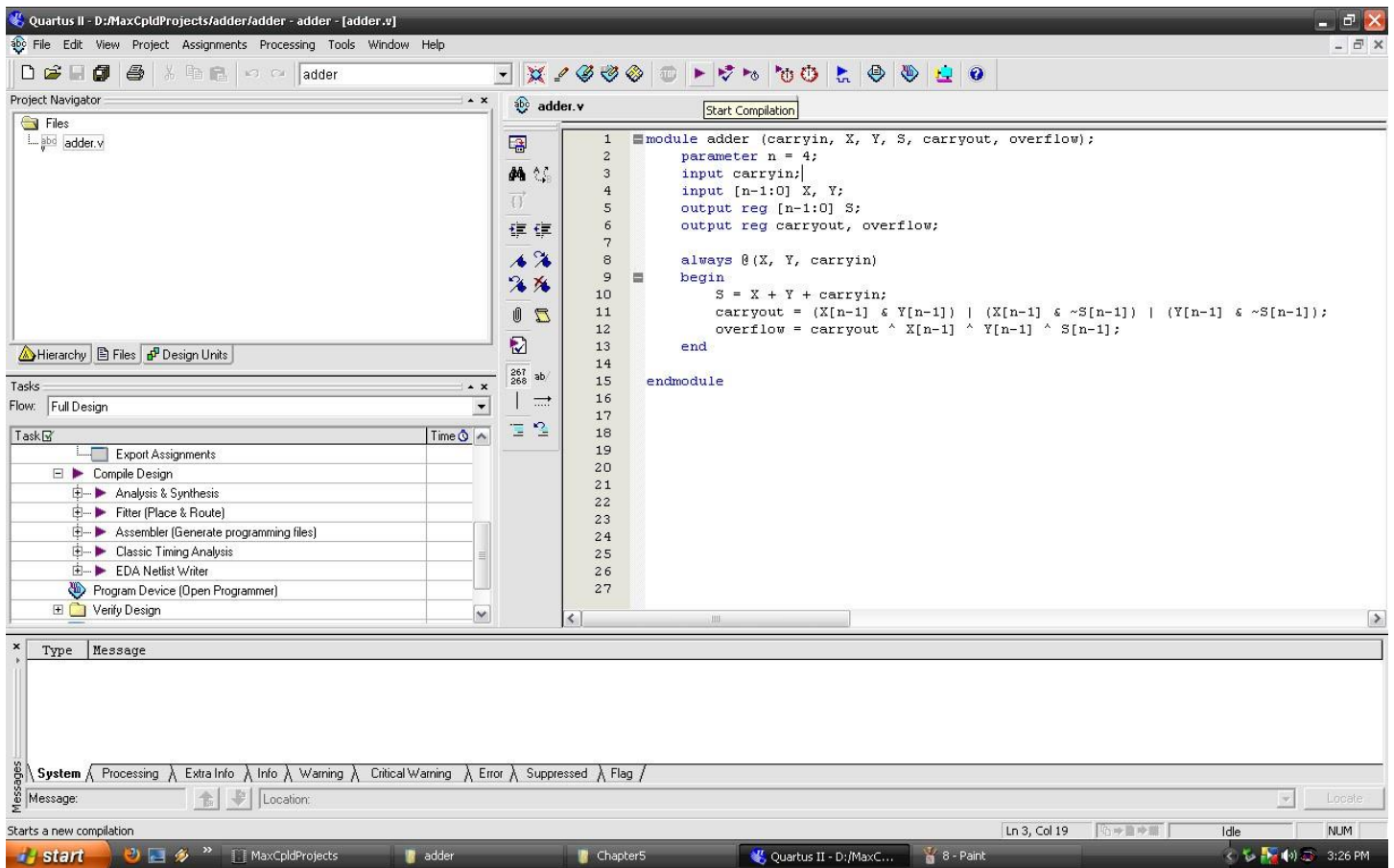


Fig-8

d. Type the following Verilog HDL code into the blank **adder.v** file and save it. (see fig -9).

```

////////////////////////////////////
timescale 1ns / 100ps module adder (carryin, X, Y,
S, carryout, overflow);
    parameter n = 3;
    input carryin; input [n-
1:0] X, Y;    output
reg [n-1:0] S; output
reg carryout, overflow;

    always @(X, Y, carryin)
begin

```

```

S = X + Y + carryin;
carryout = (X[n-1] & Y[n-1]) | (X[n-1] & ~S[n-1]) | (Y[n-1]
          & ~S[n-1]);
overflow = carryout ^ X[n-1] ^ Y[n-1] ^ S[n-1];
end

endmodule

```

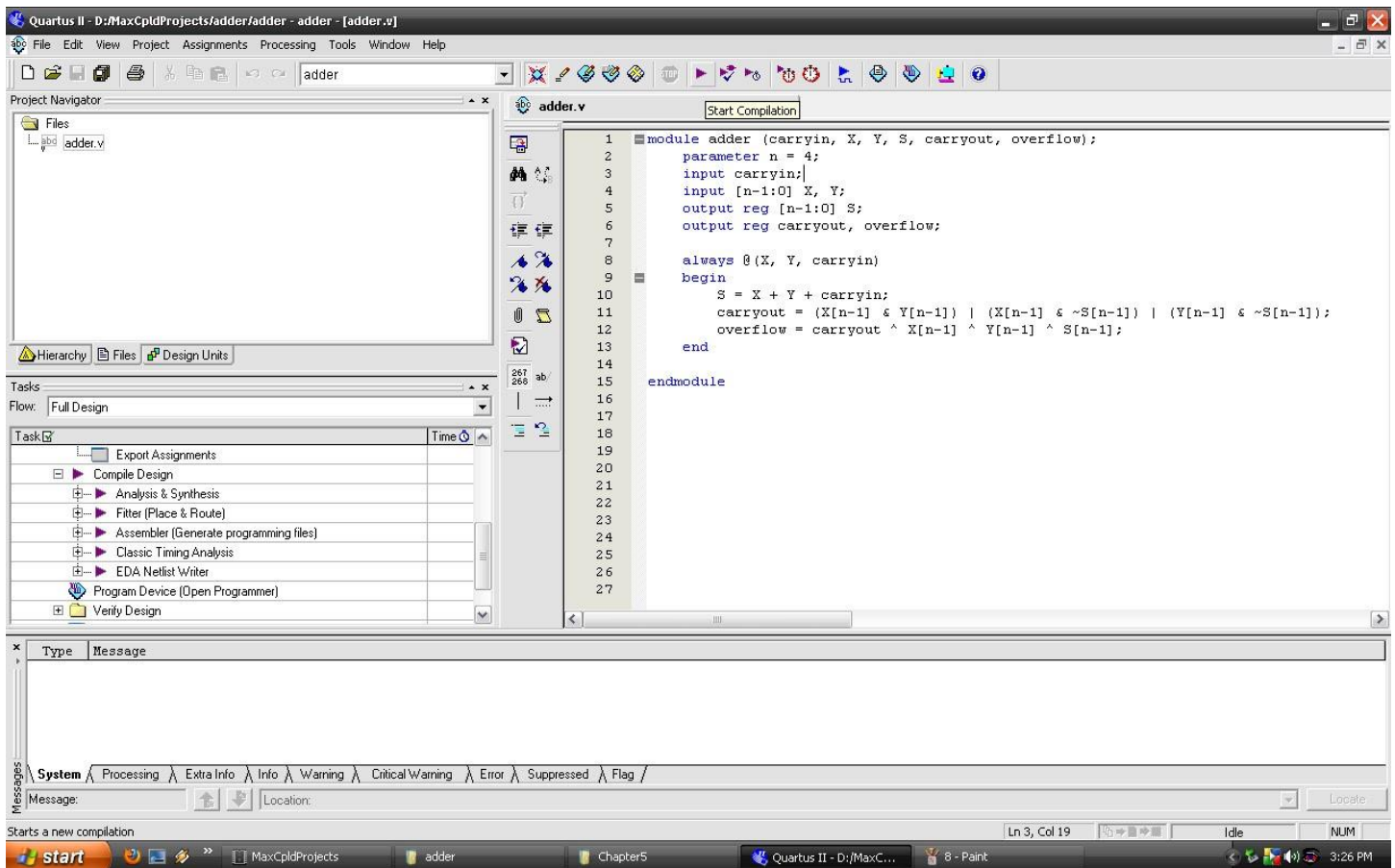


Fig-9

Now click the **start compilation** button in tool menu as shown in fig-9

- e. If the simulation is successful then a green tick mark is displayed in front of Compile Design in the task bar as shown below.

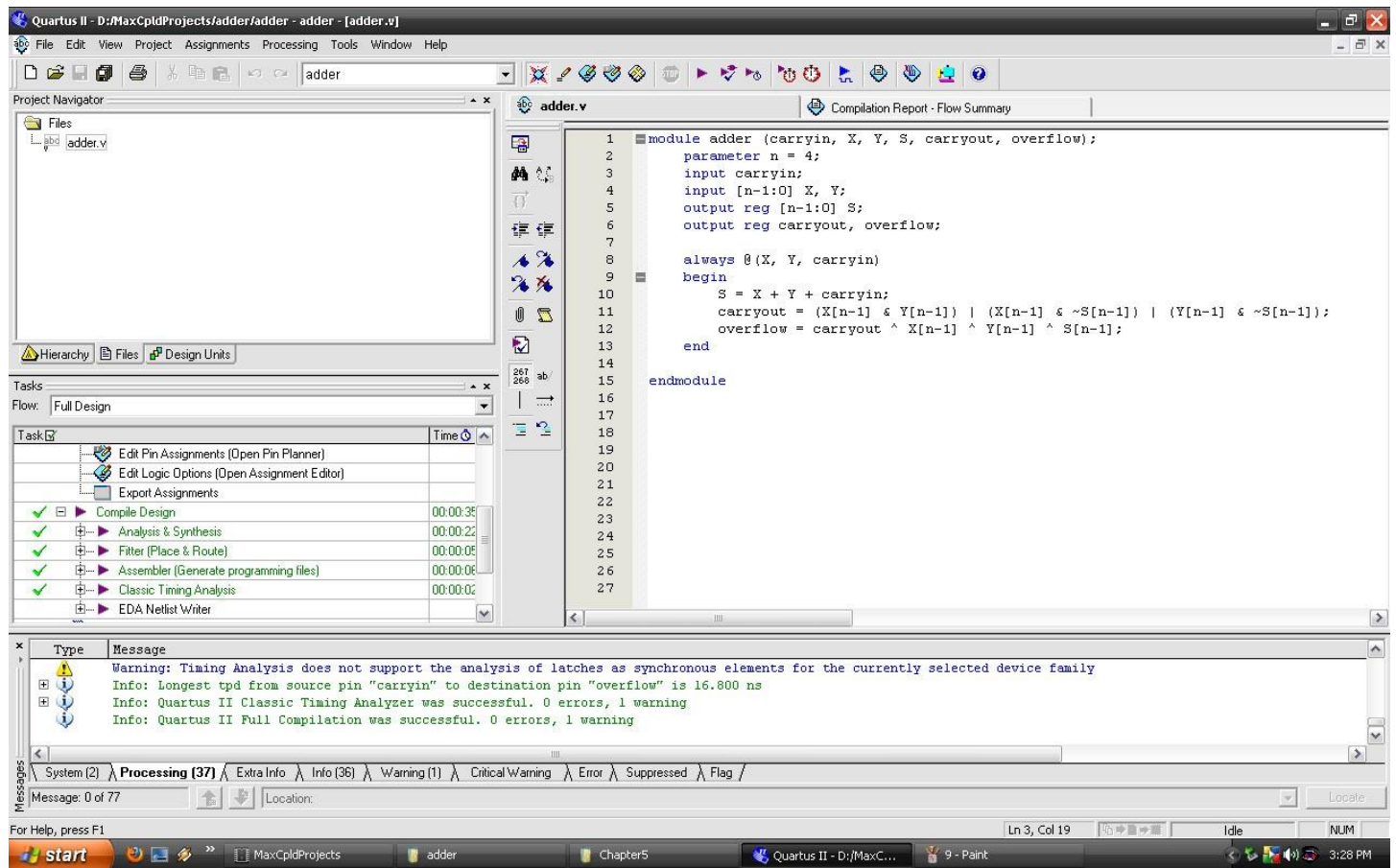


Fig-10

Assignment of Pins:

In this section, you will make pin assignments. Follow the steps given below for making assignment.

1. Once the design is compiled, you can choose what pins to assign as input/output from “Pin Planner” under the Assignments tab in the Menu bar. It opens windows as shown below. Fig-11

After making the pin assignments recompile the design.

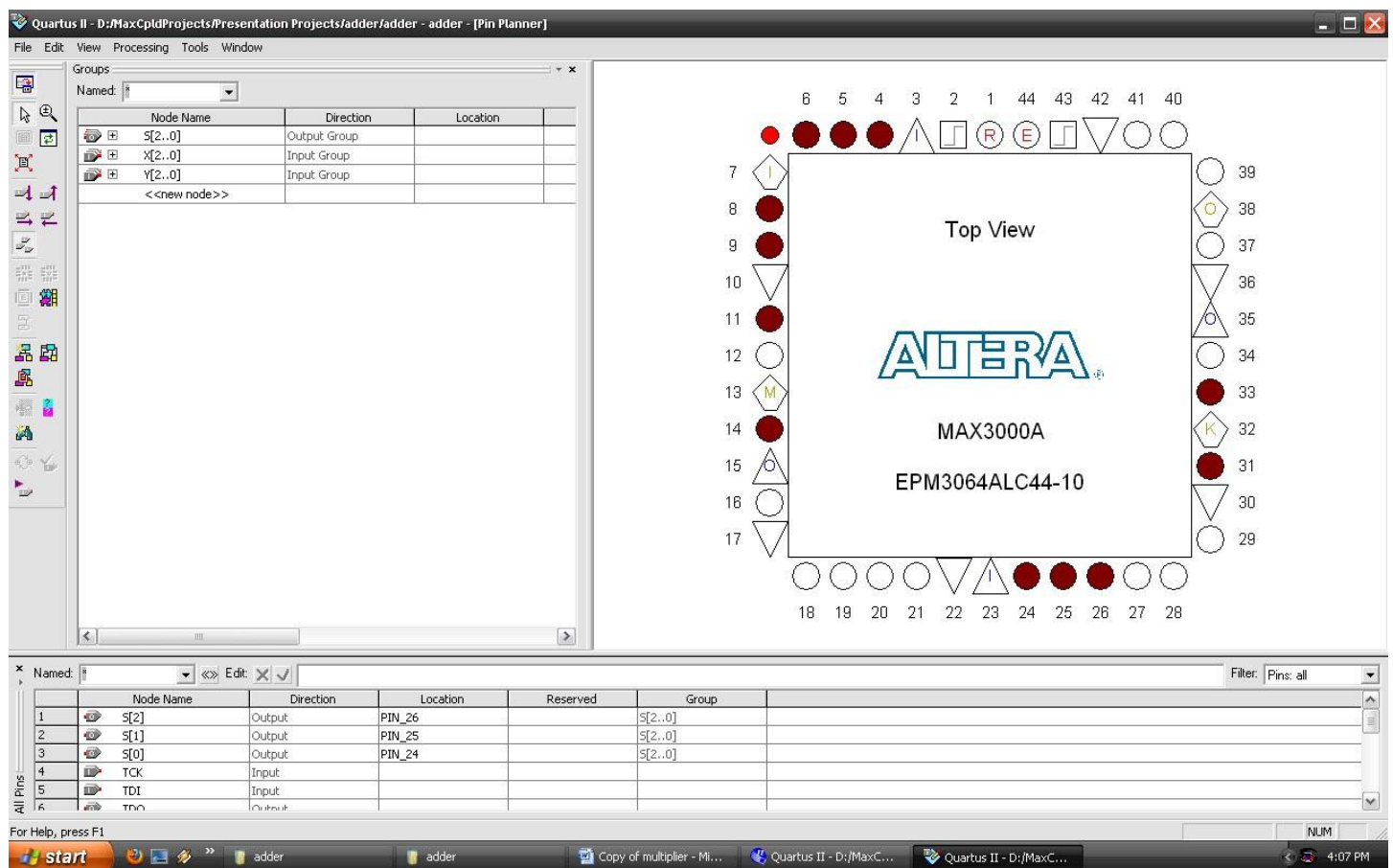


Fig-11

2. Re-run the compiler and make sure that there are no errors.

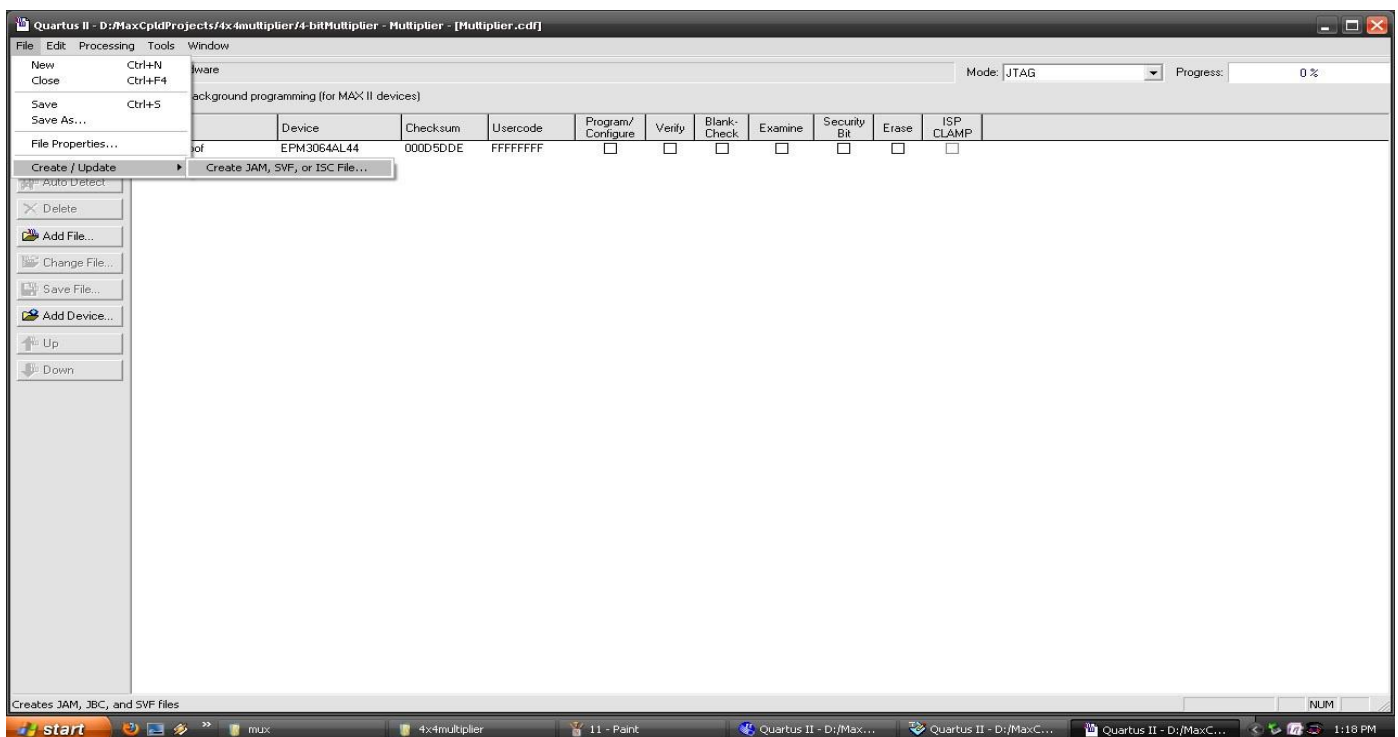
3. The pins which have been configured as input and output on the board are listed. Onboard clock is on pin no 43.

Inputs	Pin No.
SW1	4
SW2	5
SW3	6
SW4	8
SW5	9
SW6	11
SW7	12
SW8	14

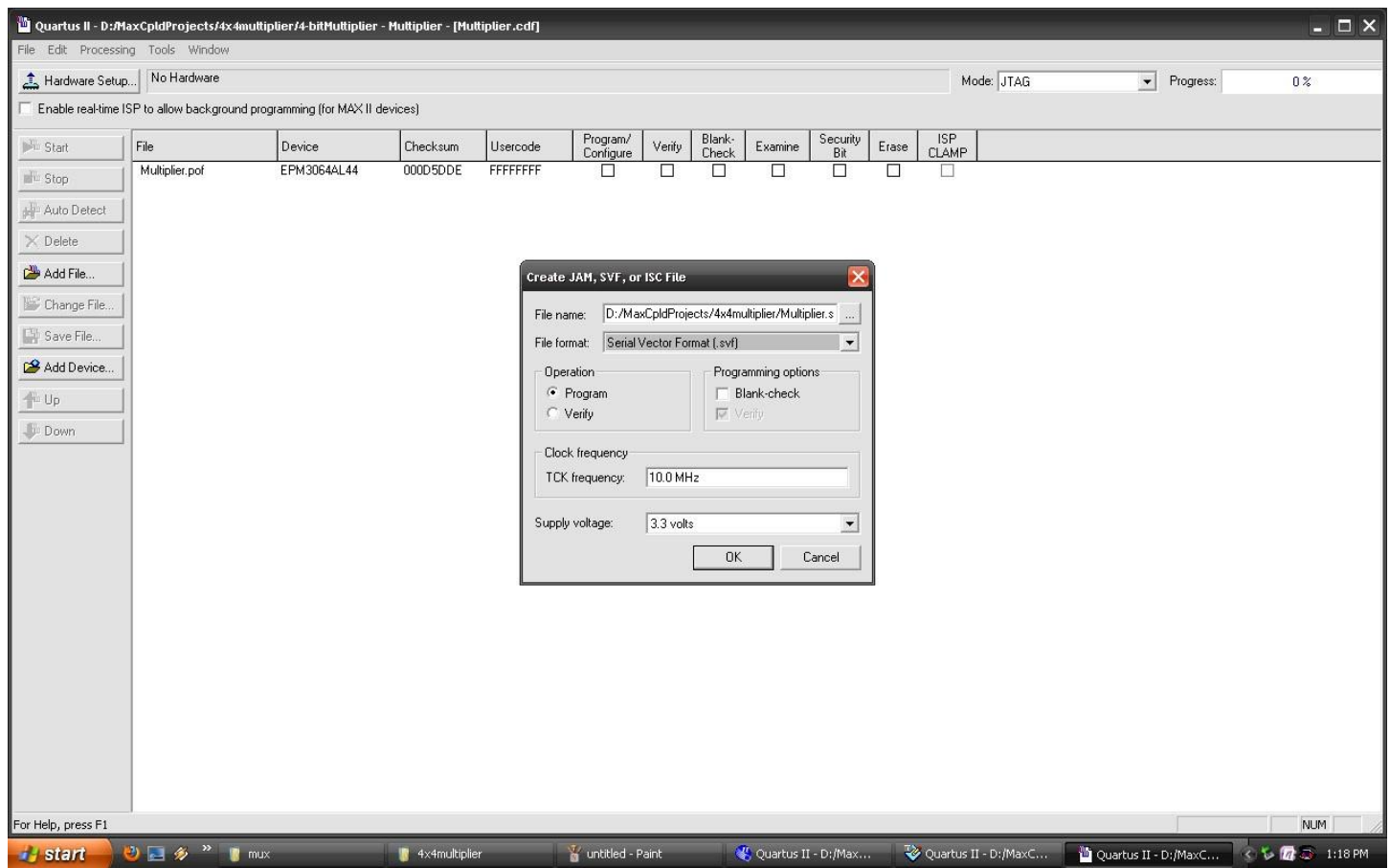
Outputs	Pin No.
LED1	24
LED2	25
LED3	26
LED4	27
LED5	28
LED6	29
LED7	31
LED8	33

Programming CPLD :

1. For programming the CPLD device open the tool option and select **programmer** a window will open as shown in fig(a) fig(b) create the Serial Vector file (.svf) for programming device.

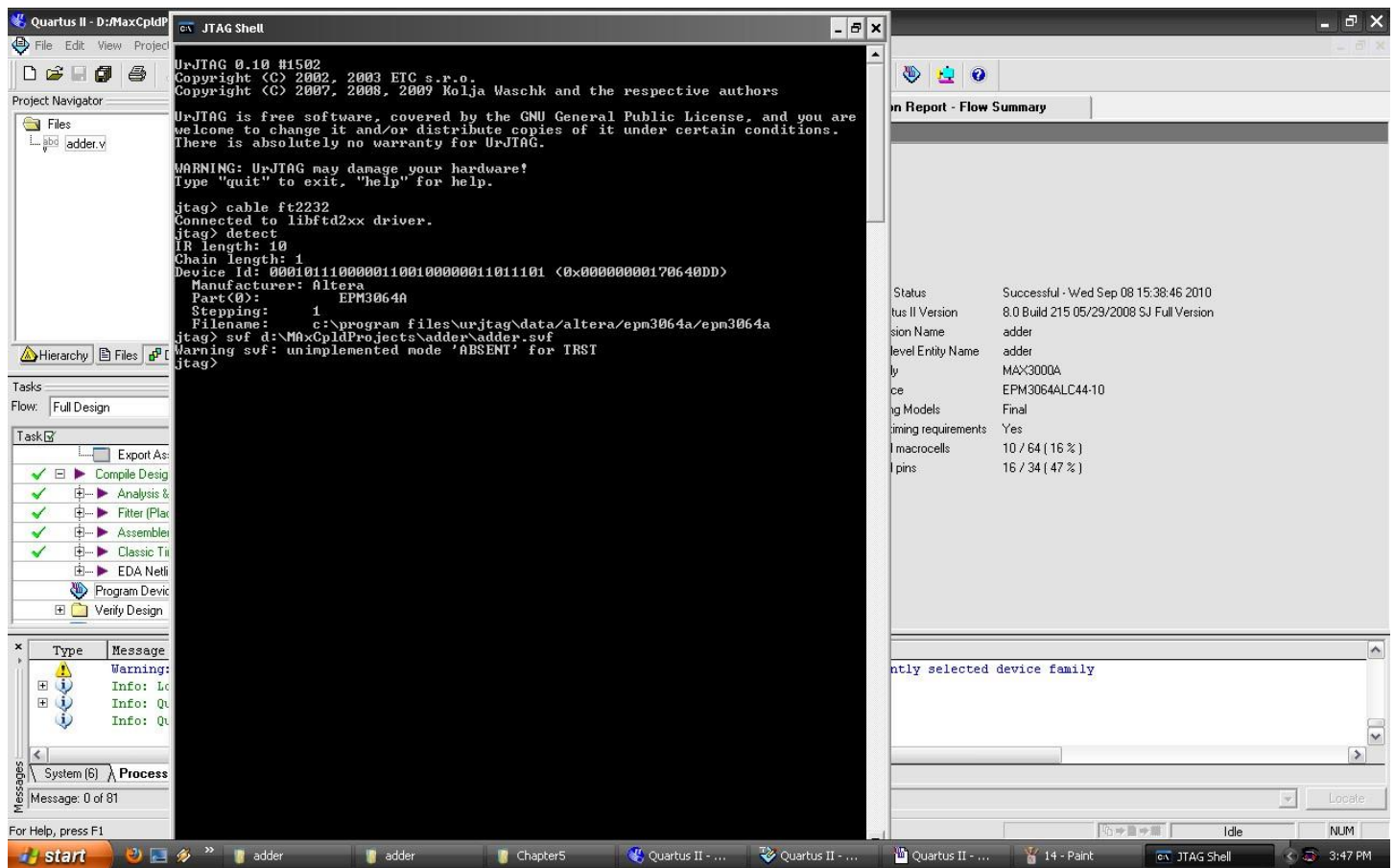


Fig(a)



Fig(b)

2. Run the JTAG shell it will display the window as shown below



3. Type out the command “cable FT2232.
4. The “detect” command should show the IR length of the CPLD connected followed by the details of the CPLD.
5. Since the CPLD is the only device in the JTAG chain, type the “part 0” command. In case you have other devices connected to the JTAG chain, you can choose the appropriate device by specifying the corresponding part number.
6. The command “svf <Address of the .svf file>” will start programming the CPLD.
7. Once the cursor returns, it shows that the device is programmed.

References:

1. www.altera.com
2. www.asic-world.com
3. Fundamentals of Digital Logic with Verilog Design (Stephen Brown) 2E.
4. Digital Logic and Computer Design (M.Morris Mano).