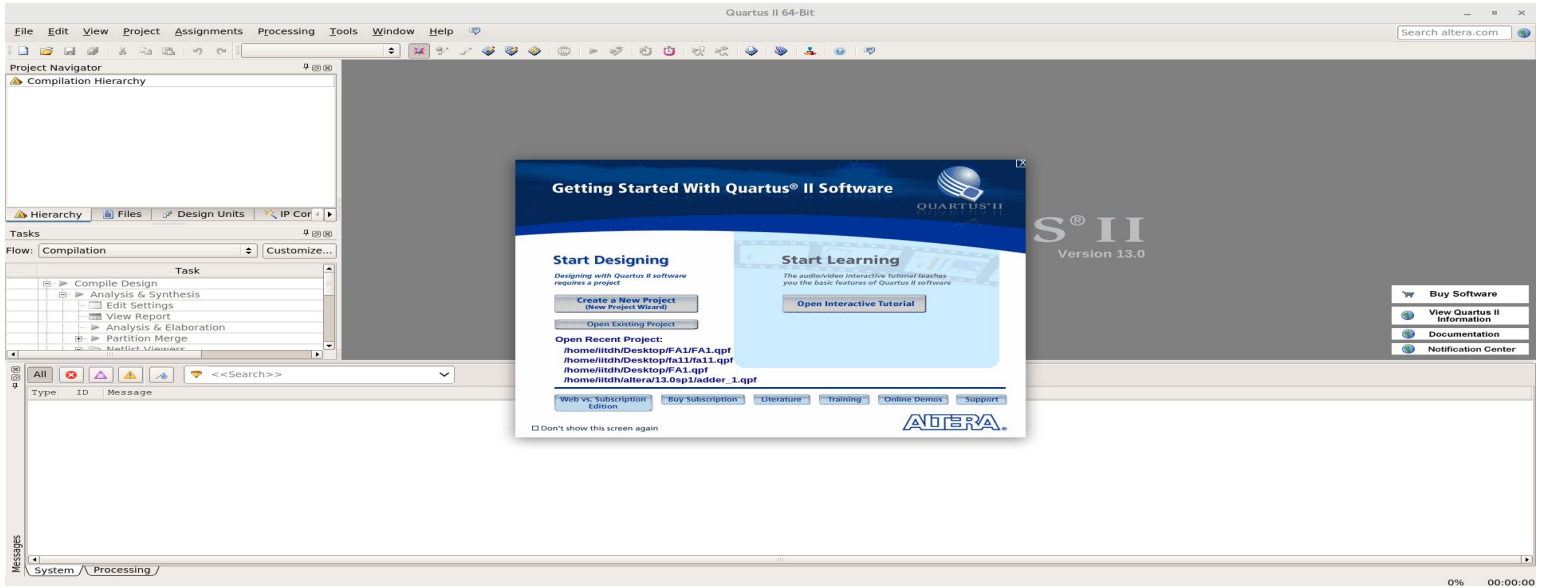
Introduction to CPLD & VHDL

Introduction Lab

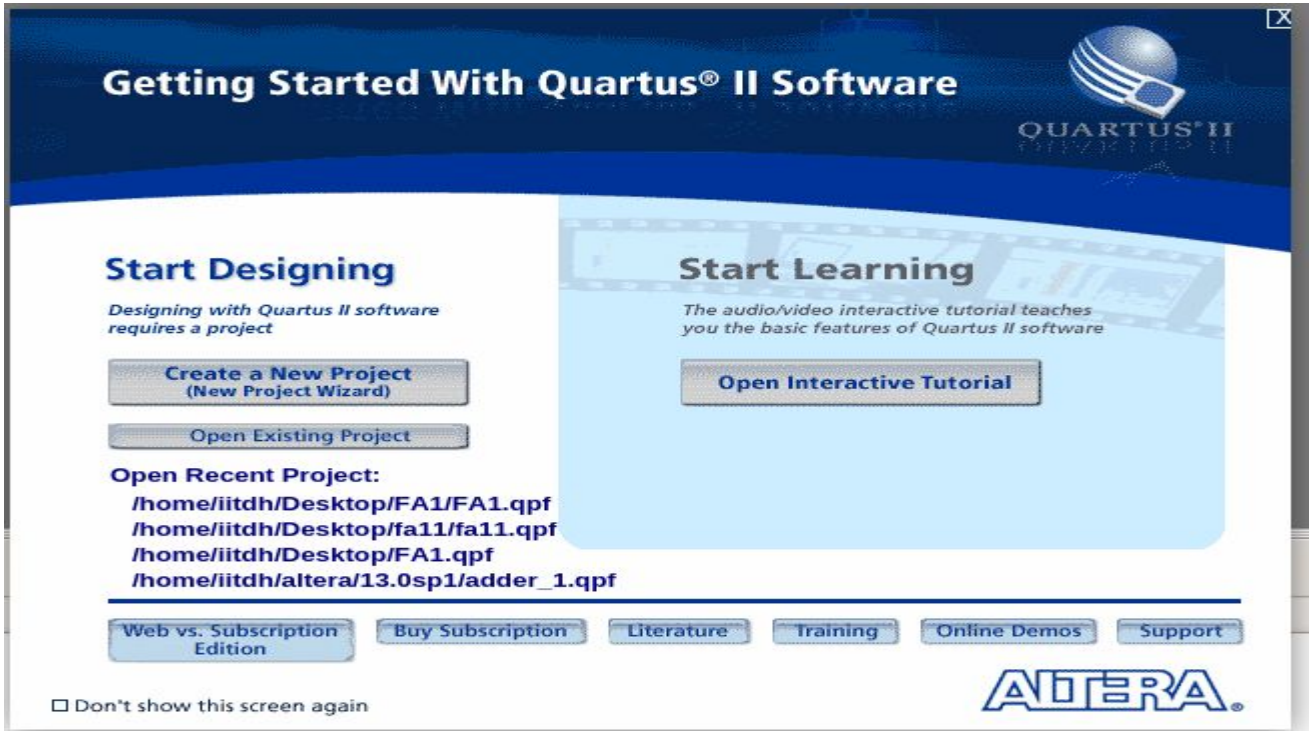
Steps to write VHDL code, dump it in CPLD and verify the written code functionality.

Step 1: Open Quartus II from Desktop.

Note: Switch to Centos operating system.



Step 2: Creating new project



Click on ” Create a new Project”

Step 2: Creating new project

“New project wizard” opens.

Click on”Next”

Provide the directory where the project must be placed.

Provide the project name and the entity name.

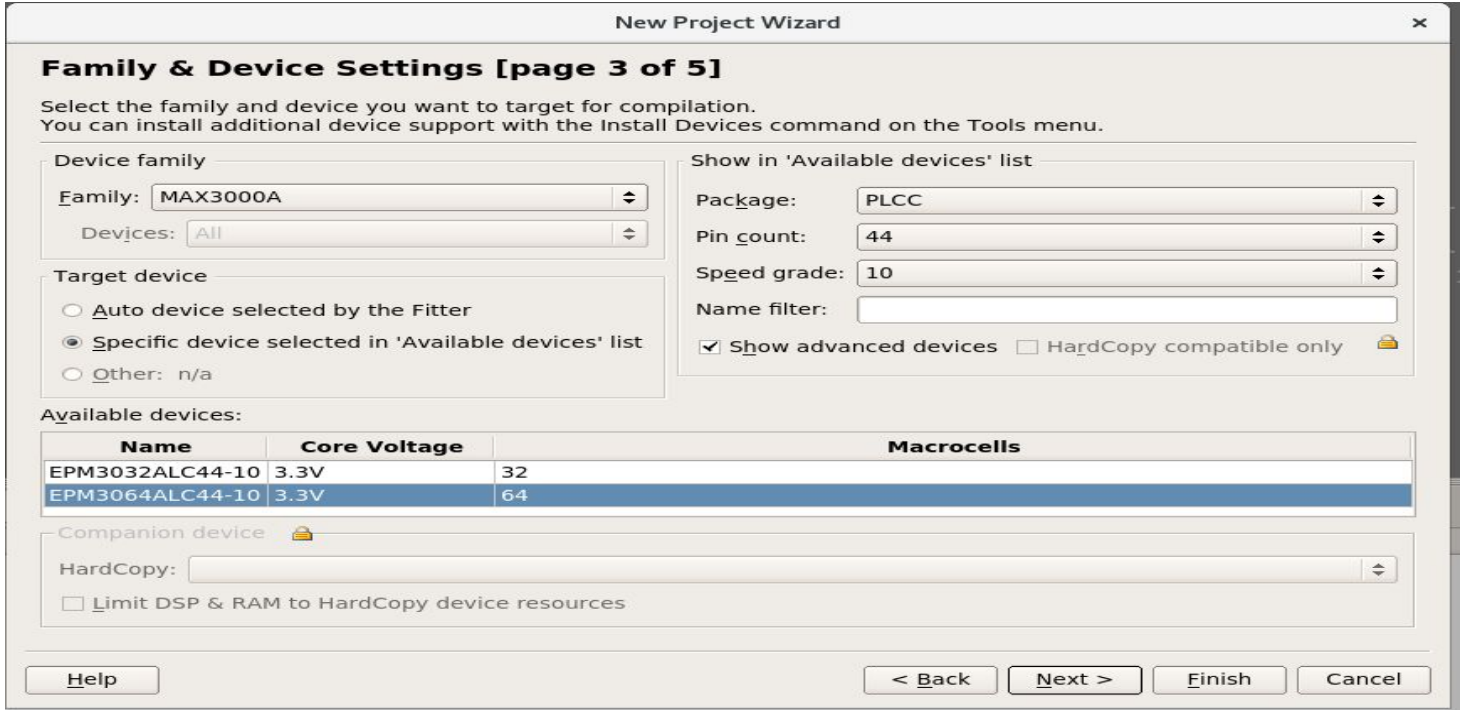
Note that project name and entity name must be same.

Click on “Next”.

Click on “Next”.

Step 2: Creating new project

Select the device, package,pin count, speed and core voltage as shown in the figure.

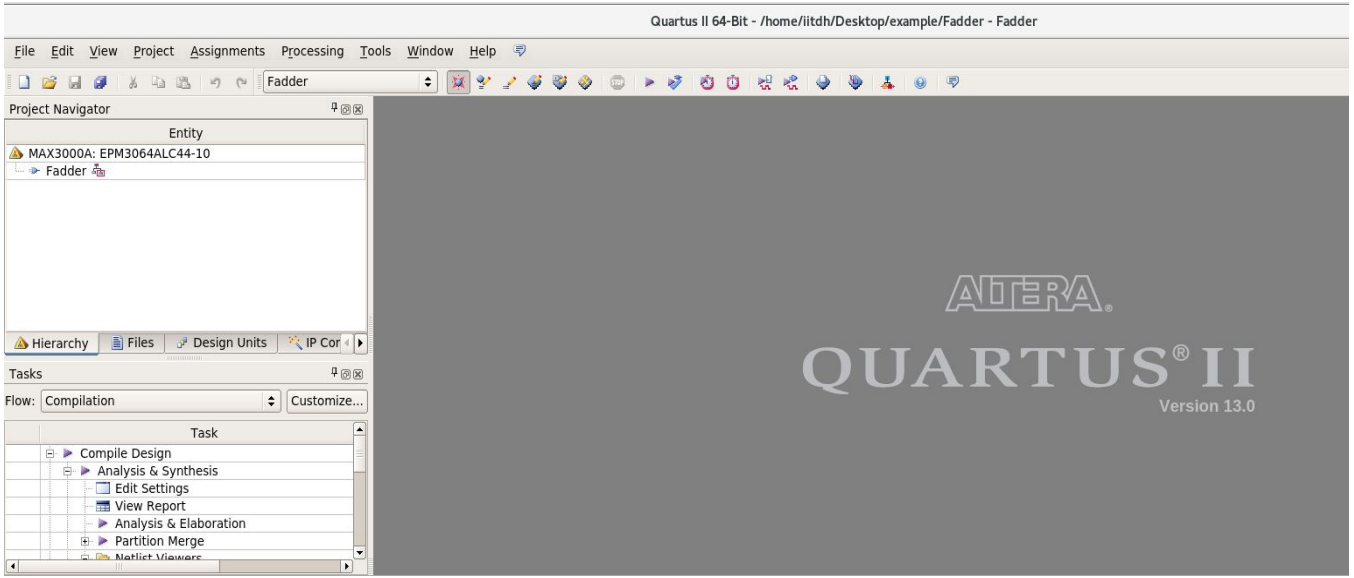


Step 2: Creating new project

Click on”Next”

Click on”Finish”.

Check if the project created appears in the project window.



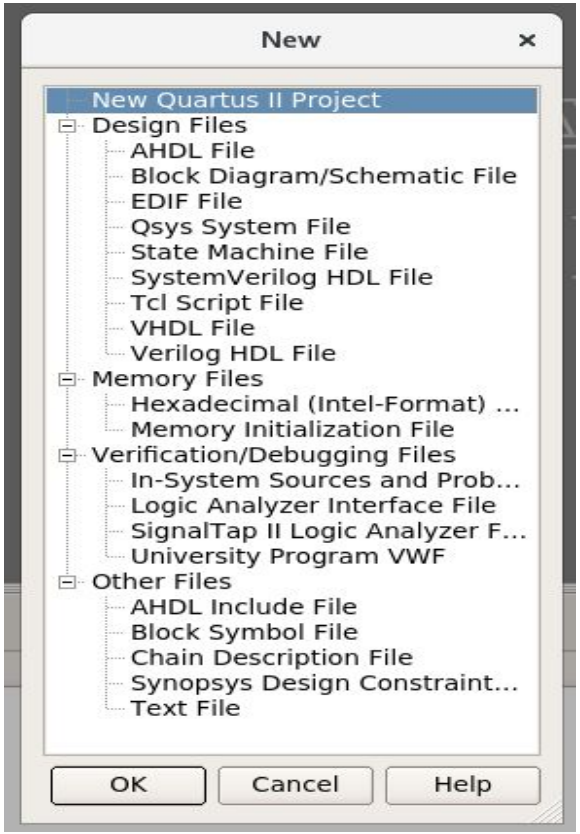
Step3: Creating and compiling a File.

File -> New

Select “vhdl file” under Design files category.

A vhdl file with “.vhd” extension opens.

Type the VHDL code and save the file in the project directory.



VHDL Code for a full adder

```
library ieee;

use ieee.std_logic_1164.all;

entity fulladder is

port(a, b, c: in bit;

sum, carry : out bit);

end fulladder;

architecture dataflow of fulladder is

begin

sum <= a xor b xor c;

carry <= (a and b) or ((a xor b) and c);

end dataflow;
```

Step3: Creating and compiling a File.

To compile the code:

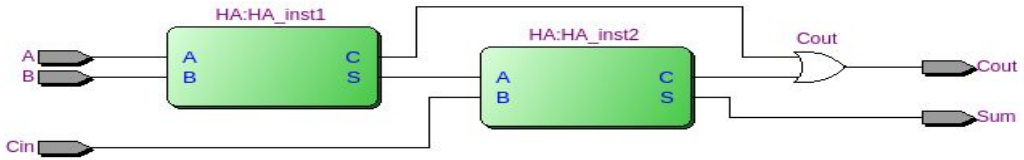
Processing-> start compilation.

The compilation will be done and if any errors (syntax) will be listed down in the messages window. If no errors then successful compilation message appears.

Step4: View RTL schematic of the written vhdl code.

Tools->Netlist viewer->RTL viewer.

This provides the equivalent high level circuit for the written code.

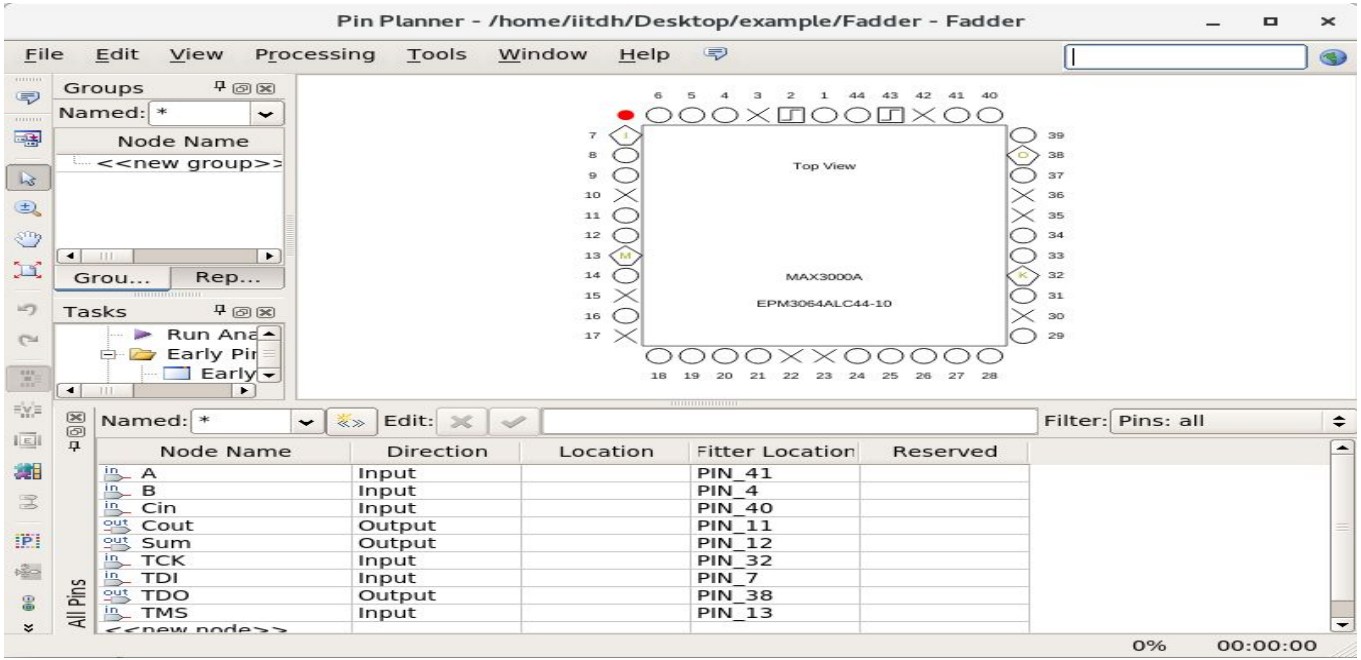


Sample RTL for a Full adder VHDL code.

Step5: Creating .svf file for dumping in the CPLD.

Assignment-> pin planner.

Assign switches pin numbers to inputs and leds pin numbers to outputs.



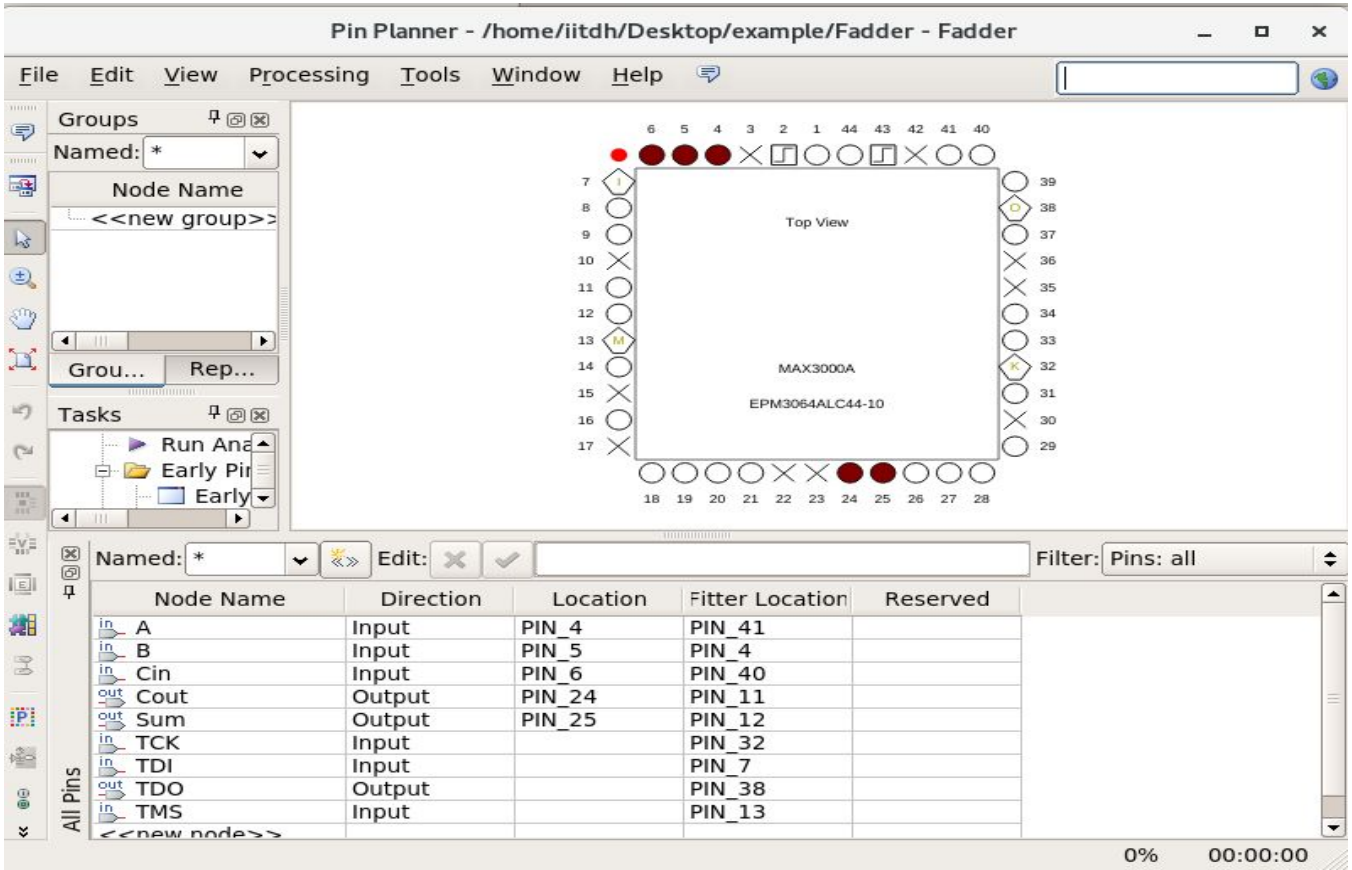
Step5: Creating .svf file for dumping in the CPLD.

INPUT AND OUTPUT PINS DETAILS OF MAX 3000A

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

Outputs	Pin No.
LED1	24
LED 2	25
LED 3	26
LED 4	27
LED 5	28
LED 6	29
LED 7	31
LED 8	33

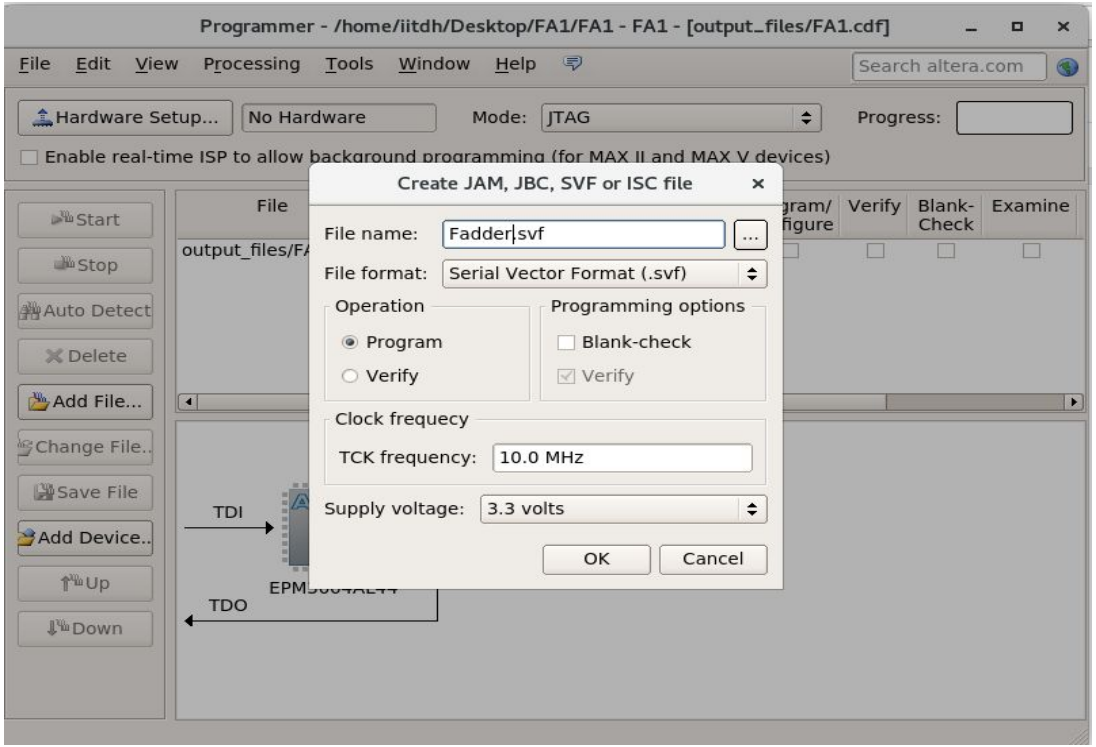
Step5: Creating .svf file for dumping in the CPLD.



After assigning the pin numbers, close the window.
Compile the code again.
Processing-> start compilation.

Step5: Creating .svf file for dumping in the CPLD.

Tools->Programmer
“Programmer” window opens.
Select the device opted during project creation.
File->Create JAM, SVF,JBC,ISC file.
A small window pops up.
Provide the file name and select svf as file format.



Step6: Verifying the code on CPLD

Connect the CPLD board to the system and turn it on.

Open the terminal & type the commands sequentially.

1. sudo jtag
2. Enter the password: iitdh@walmi

You will enter the jtag shell.

3. cable ft2232 vid=0x0403 pid=0x6010

“Connected to libftdi driver” message appears.

4. detect

Checks if the device is connected and gives its signature.

5. svf <svf file location> progress

Provide the svf file path and file name to be dumped in the CPLD.

The file will now be loaded, and the output of the VHDL/ Verilog program may be observed on the board.

Step6: Verifying the code on CPLD

iitdh@localhost:~

File Edit View Search Terminal Help

```
[iitdh@localhost ~]$ sudo jtag
[sudo] password for iitdh:

UrJTAG 2017.10 #
Copyright (C) 2002, 2003 ETC s.r.o.
Copyright (C) 2007, 2008, 2009 Kolja Waschk and the respective authors

UrJTAG is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
There is absolutely no warranty for UrJTAG.

warning: UrJTAG may damage your hardware!
Type "quit" to exit, "help" for help.

jtag> cable ft2232 vid=0x0403 pid=0x6010
Connected to libftdi driver.
jtag> detect
IR length: 10
Chain length: 1
Device Id: 00010111000001100100000011011101 (0x170640DD)
  Manufacturer: Altera (0x0DD)
  Part(0):      EPM3064A (0x7064)
  Stepping:     1
  Filename:     /usr/share/urjtag/altera/epm3064a/epm3064a
jtag> svf /home/iitdh/Desktop/example/Fadder1.svf progress
detail: Parsing     20/520 (  3%)warning: unimplemented mode 'ABSENT' for TRST
detail: Parsing    520/520 (100%)detail:
detail: Scanned device output matched expected TDO values.
jtag>
```

Terminal screenshot

Repeat the steps for a Full subtractor.