

# Using Quartus II, UrJTAG and Krypton Board

**Wadhwani Electronics Lab**

Department of Electrical Engineering  
Indian Institute of Technology Bombay  
Feb 2020

# Before reading this document...

Make sure that

- \* you have installed Quartus II and ModelSim on your Laptop/PC.
- \* you have successfully compiled your design.
- \* you have successfully simulated it.
- \* you have installed UrJTAG as per the instruction handout uploaded on Moodle
- \* you have downloaded Test files for Krypton board from Moodle.

# Lab session

In this session, you will

- 1 be introduced to Krypton board
- 2 generate .svf file. svf stands for *Serial Vector Format*
- 3 load .svf file into the CPLD
- 4 test your Krypton with the test files provided, in order to confirm its working
- 5 verify your full adder design by loading your fulladder.svf file and testing using the switches and LEDs

# Introduction to Krypton

You will be briefly introduced with Krypton in this presentation. For more information please refer to the Lab manual and user manual of Krypton.

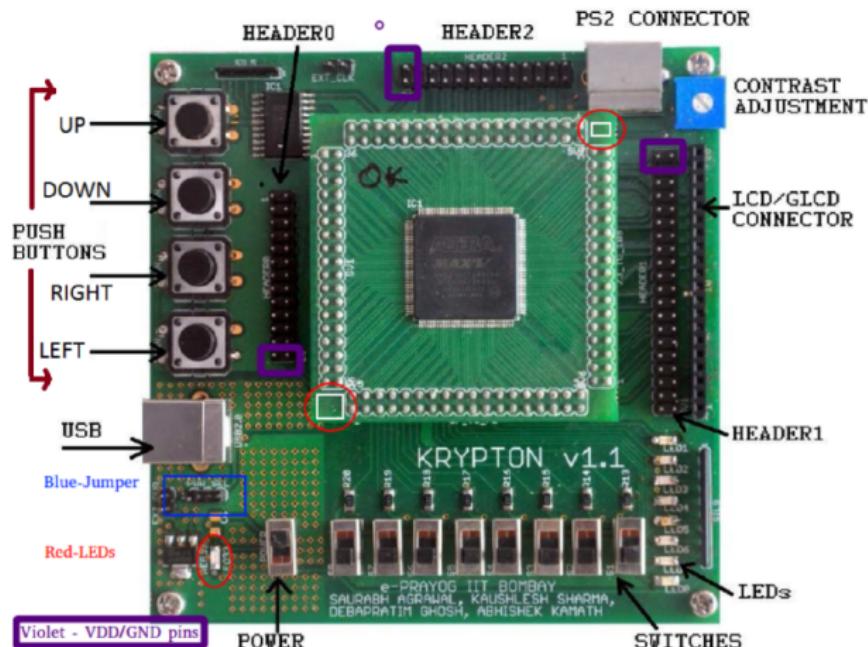


Figure: Caption

# Krypton features

Main features of the board:

- ① CPLD used: 5M1270ZT144C5N, Altera's MAX V family with 980 macro-cells/1270 logic elements
- ② USB programmable and powered, with provision for external DC supply
- ③ Preconfigured on-board I/Os- 8 switches, 8 LEDs and 4 push-buttons
- ④ On-board clock of 1Hz and 50MHz, and provision for external clock source connection
- ⑤ Connectors provided on-board to interface standard peripherals directly (LCD/GLCD/PS2)
- ⑥ Large number of on-board I/Os (86) provided for various applications

# Testing the Krypton board and UrJTAG

- Today we are going to test:
  - On-board I/Os - 8 switches, 8 LEDs and 4 push-buttons.
  - CPLD card headers.
- To test the above peripherals we need to load the Serial Vector File(.svf) file to the Krypton Board.
- To load the .svf file on the board, we will use UrJTAG.
- Now connect your Krypton board to your Laptop/PC.

# Testing the Krypton board: loading .svf File

- Now open the terminal and type:  
**jtag**
- Now you will be presented with a command console. Type:  
**cable ft2232 vid=0x0403 pid=0x6010**  
An indication should now appear that the driver is connected. You can use **lsusb** command after connecting the board to know pid and vid.
- Now type:  
**detect**  
which displays the detected CPLD device (Details like IRChain length, Manufacturer, Device ID, Stepping etc.)
- Load the svf file into the CPLD device by typing  
**svf <path>/On\_board\_peripheral.svf progress**  
Make sure that you specify full path of .svf file.

## Testing the Krypton board: On\_board\_peripheral

- Once the file is loaded, initially all switches are to be kept off. For each test, **exactly one of the switches** is to be turned on. The tests are as follows:
- Switch s1:** The eight LED's will be lit up to display an 8-bit count which is updated every 0.04194s.
- Switch s2:** The eight LED's will be lit up to display an 8 bit count which is updated every second.
- Switch s3:** All LEDs are lit up. There are four push buttons. You can check each of them by pressing them one at a time and checking the observations which should be as follows:
  - If push-button up is pressed, LED 1 and 5 will turn off
  - If push-button down is pressed, LED 2 and LED 6 will turn off and so on.
- Switches s4,s5,s6,s7,s8 (one at a time): The corresponding LED (i.e. LED-4 for switch s4 etc.) will turn on.

## Testing the CPLD headers: IO

- There are three headers in the Krypton. We will check each header by writing to each pin and reading from each pin.
- We test one header at a time by connecting its two rows using strip cables. Before starting the test, we program the CPLD card using the svf file IO.svf.
- For HEADER0 and HEADER2, Pin 25 and 26 is GND and VDD respectively. For HEADER1 Pin 39 is GND and 40 is VDD.
- The switch settings for the different tests are described below:
  - ① Switch s1 : HEADER0 Data sent from first row to second row
  - ② Switch s2 : HEADER0 Data sent from second row to first row
  - ③ Switch s3 : HEADER1 Data sent from first row to second row
  - ④ Switch s4 : HEADER1 Data sent from second row to first row
  - ⑤ Switch s5 : HEADER2 Data sent from first row to second row
  - ⑥ Switch s6 : HEADER2 Data sent from second row to first row
- In each case, the output displayed on LED-8 down to LED-1 will be 0x66(01100110, LED-8 is the most-significant bit) if the test has passed.

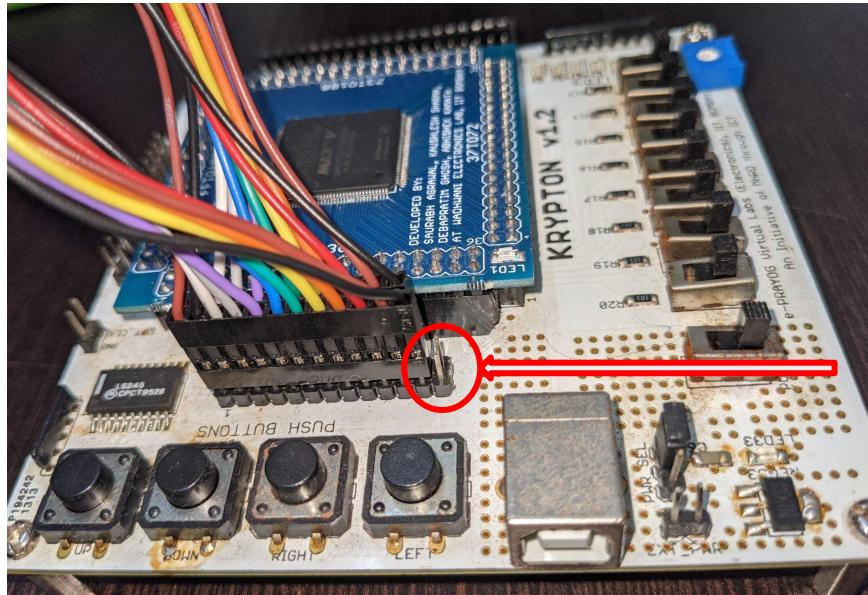
After sliding the switch, press UP push button to start the test.

**NOTE:**

- **Do not connect** Pin 25 to Pin 26 for Header0 and Header1. Only connect Pin1-Pin24 using ribbon cable.
- Similarly do **not connect** Pin 39 to Pin 40 for Header2. Only connect Pin 1-Pin 38 using ribbon cable.

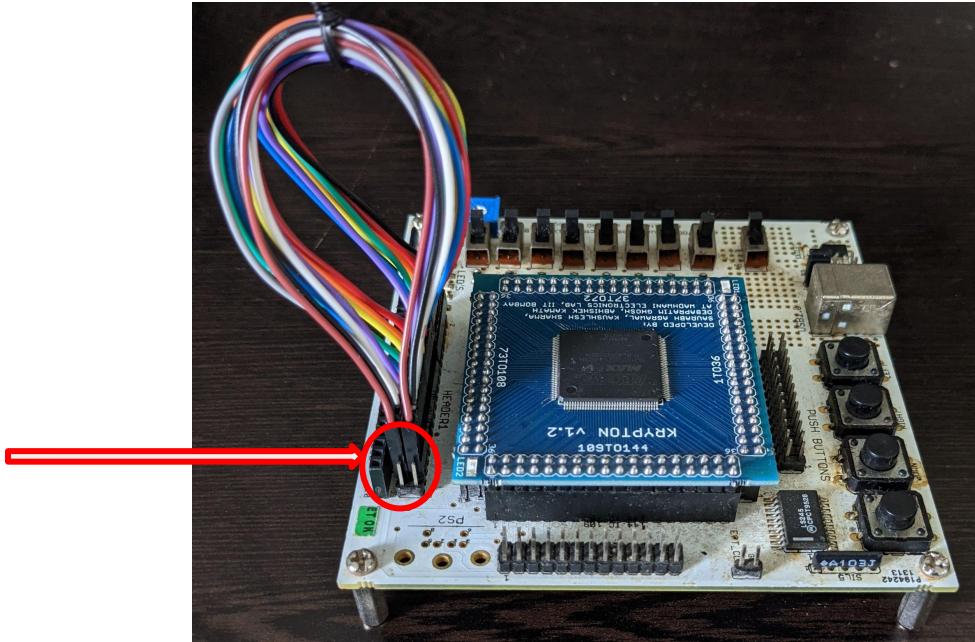
**If you connect VDD to GND. Board will short circuit.**

# HEADER0 Connection



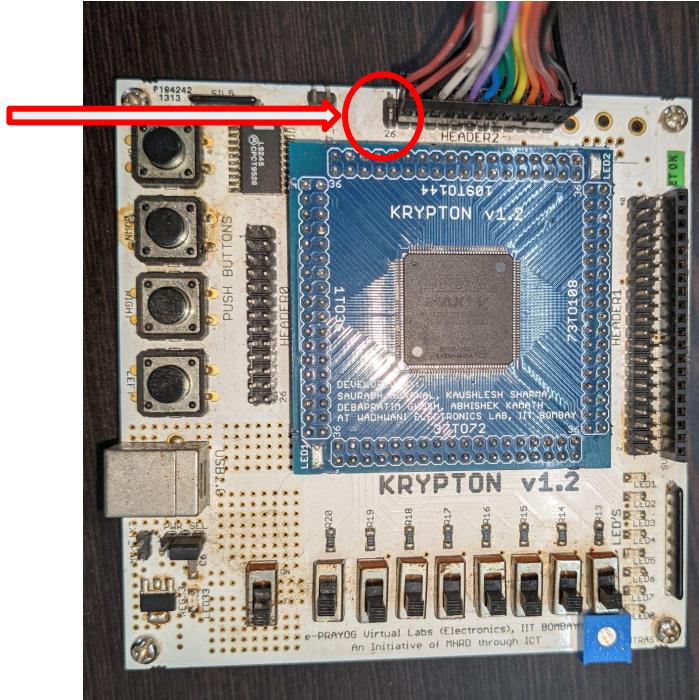
Notice how Pin 25 is not connected to Pin 26

# HEADER1 Connection



Notice how Pin 39 is not connected Pin 40

# HEADER2 Connection



Notice how Pin 25 is not connected Pin 26

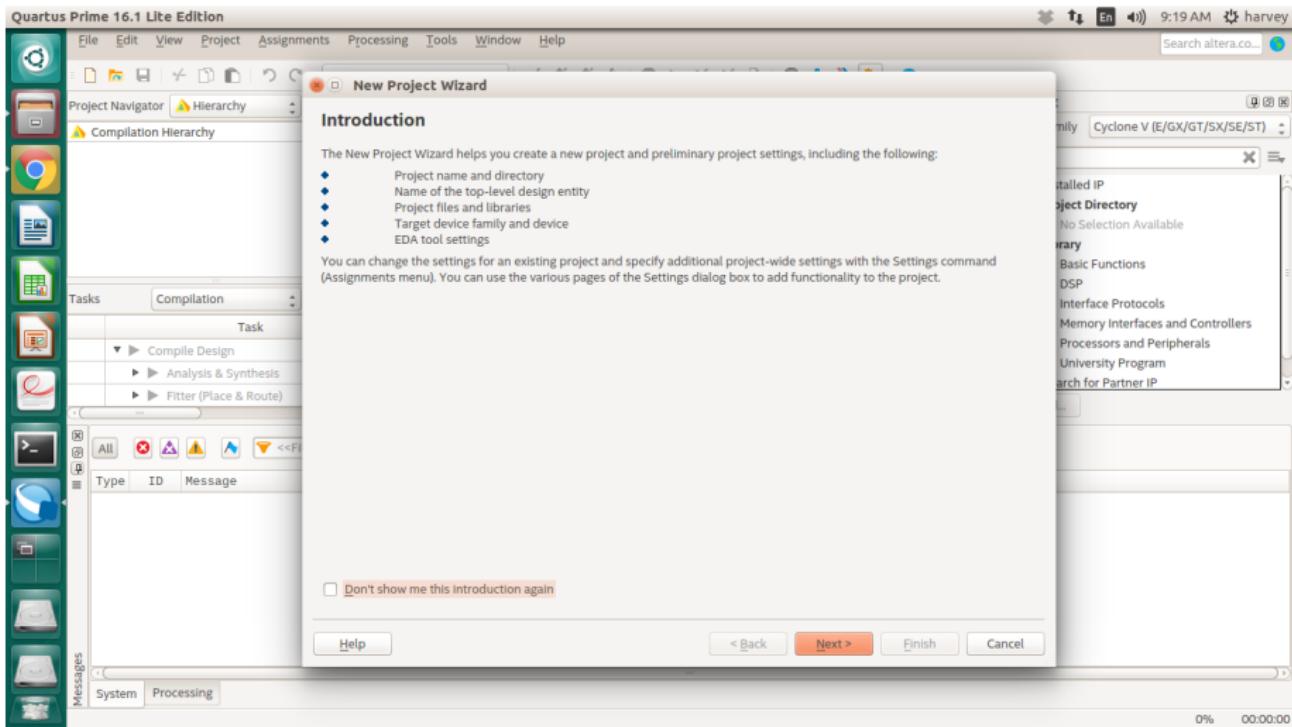
# Testing the LCD module

- Load the LCD Test file for your type of board(White/Blue)
- Slide up Switch S1
- Press and hold UP Push button for 2 secs to Reset the LCD Display
- Wait till the message “ KRYPTON ” is displayed on the LCD Display

**CAUTION: DO NOT UNPLUG THE LCD MODULE**

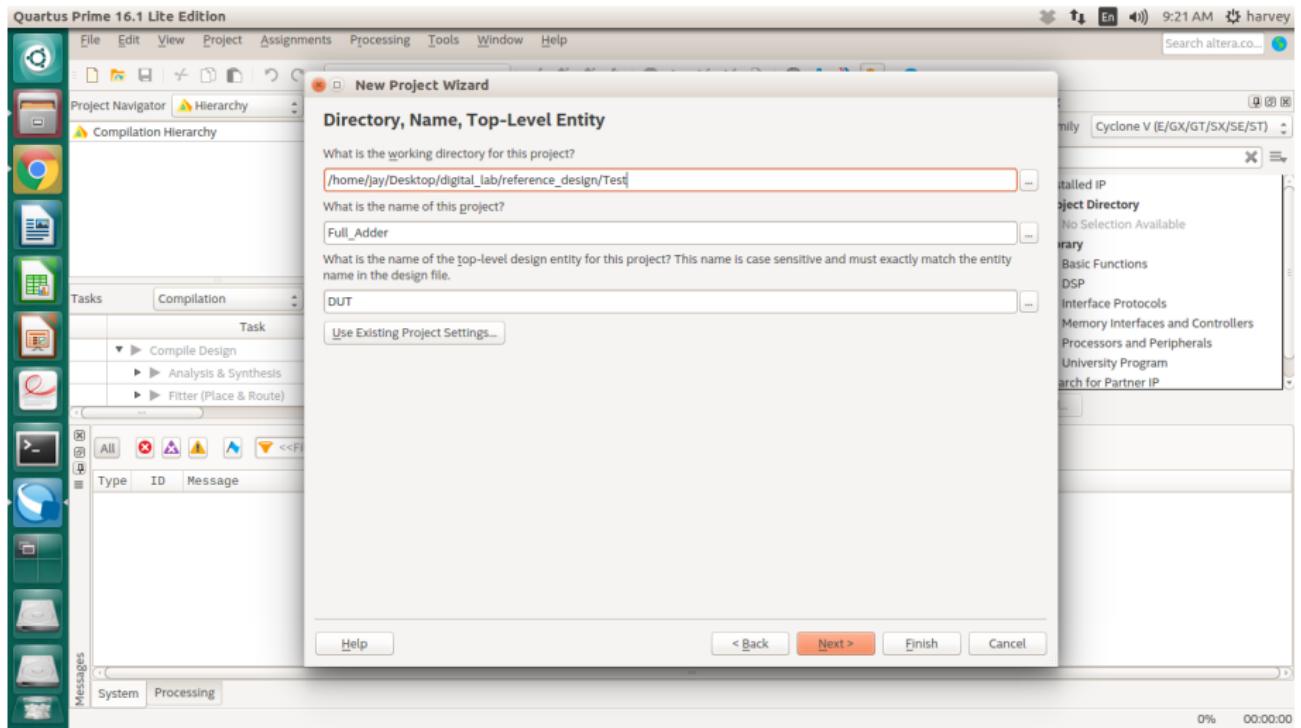
# Using Quartus II - New Project

In the introductory page, click Next.



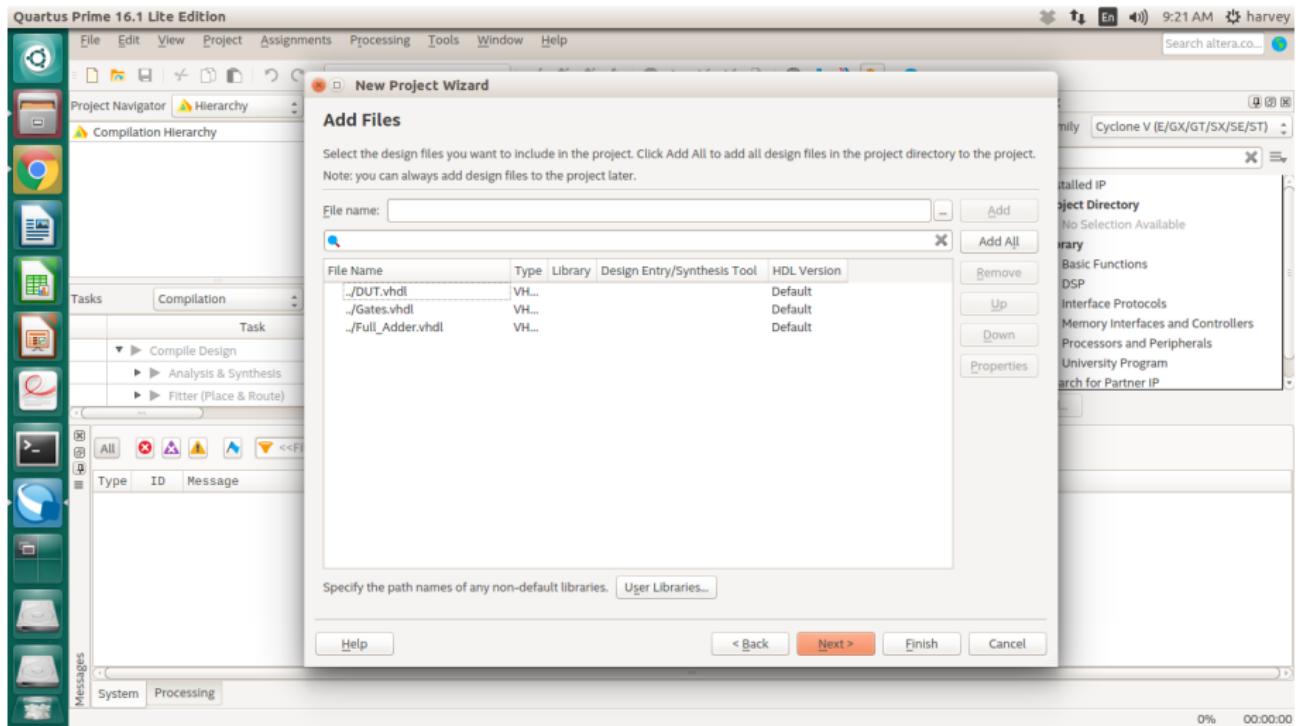
# Using Quartus II- Project Directory and Top-level Module

In this page, specify a working directory for your project. It is a good practice to open a new folder for every new project.



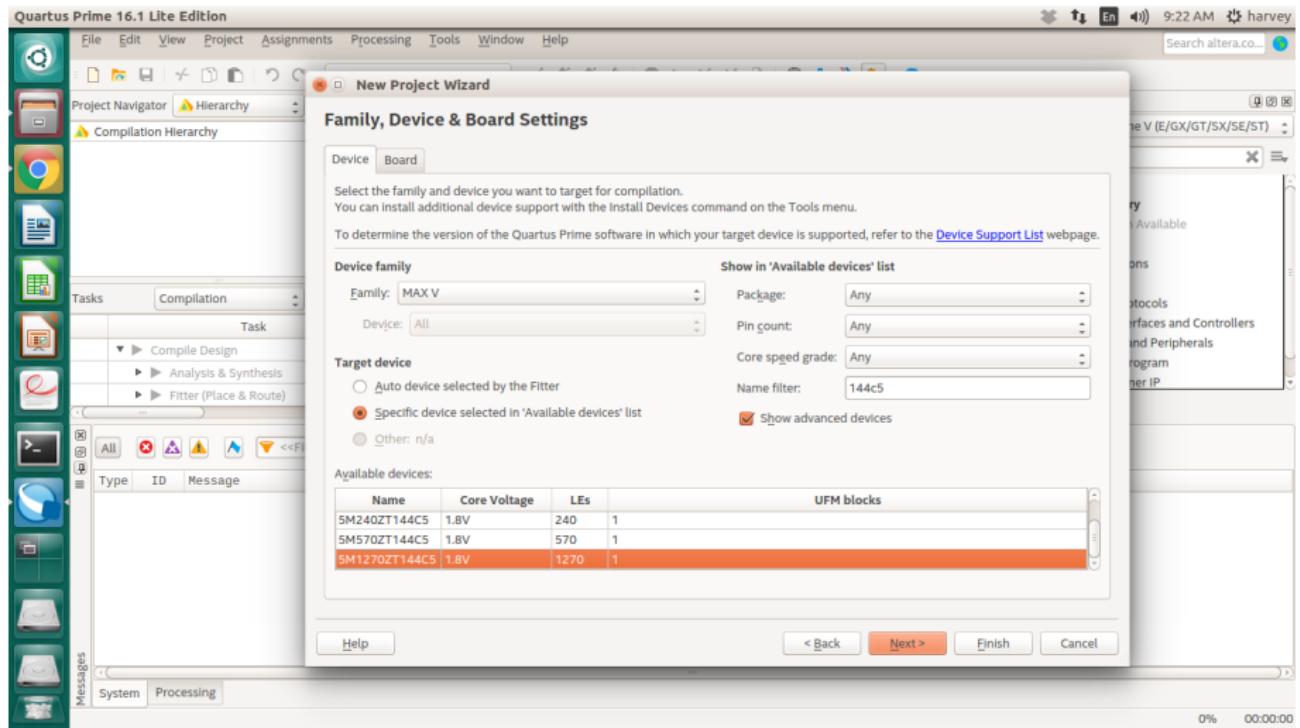
# Using Quartus II- Adding Files to Project

Next page may be skipped, In this page add all relevant files to your project.



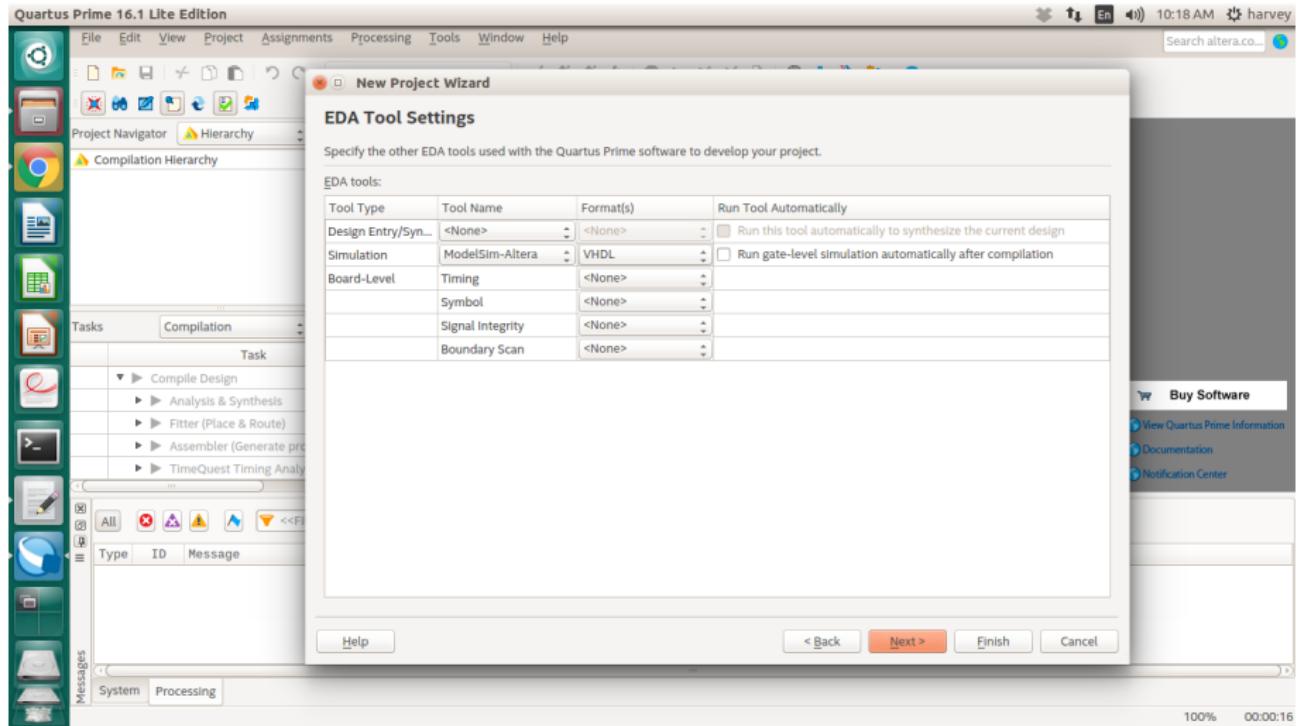
# Using Quartus II- Device Selection

In this page, select the target CPLD. Select Max V from Device family. Then type 144c5 in Name fitter and select last one.



# Using Quartus II- Simulation tool and HDL Selection

In this page, select the target simulation tool as Modelsim-Altera and language as VHDL.



# Using Quartus II- Summary

This page shows you a project summary- the project name, top level module, selected device etc. If there are mistakes, you can go back and change them.

The screenshot shows the Quartus Prime 16.1 Lite Edition interface. A central dialog box titled "New Project Wizard" is open, specifically the "Summary" step. The dialog lists various project settings:

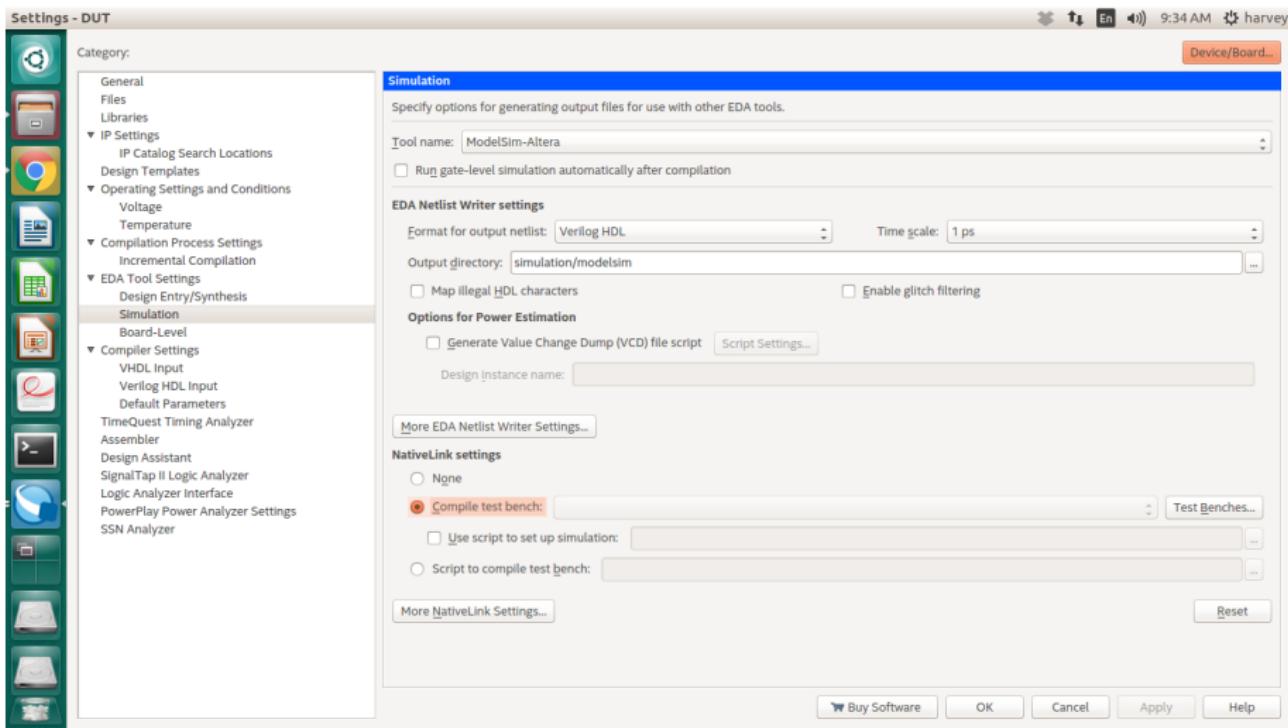
Setting	Value
Project directory:	/home/jay/Desktop/digital_lab/reference_design/Test
Project name:	Full_Adder
Top-level design entity:	DUT
Number of files added:	3
Number of user libraries added:	0
Device assignments:	
Design template:	n/a
Family name:	MAX V
Device:	5M1270ZT144C5
Board:	n/a
EDA tools:	
Design entry/synthesis:	<None> (<None>)
Simulation:	ModelSim-Altera (Verilog HDL)
Timing analysis:	0
Operating conditions:	
VCCINT voltage:	1.8V
Junction temperature range:	0-85 °C

At the bottom of the dialog are "Help", "< Back", "Next >", "Finish" (highlighted in orange), and "Cancel" buttons. The main application window shows a toolbar, a menu bar, and several toolbars on the left. The status bar at the bottom right shows "0%" and "00:00:00".

- Once you have created project and added files start compilation.
- Make sure that you have selected proper Top-Level entity and did the full compilation.**
- If you are getting any errors resolve them. Warning can be ignored as of now.
- Now the next step is Gate Level Simulation.

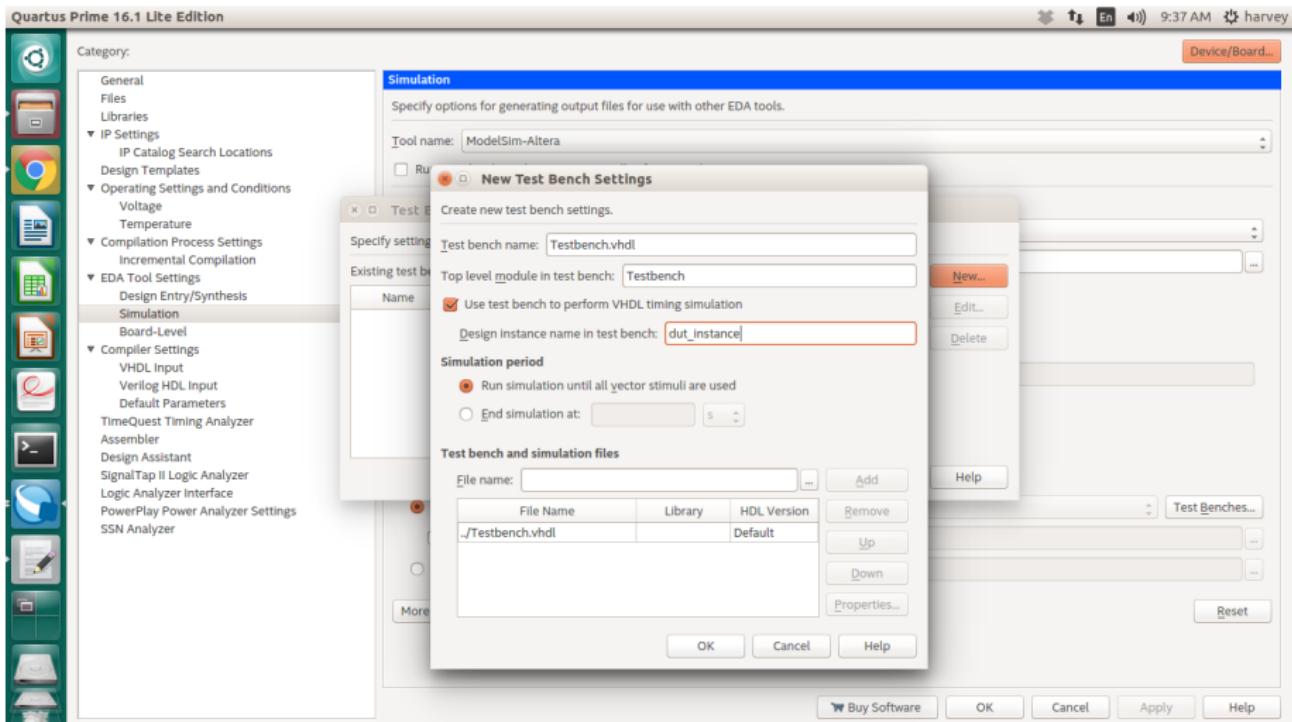
# Using Quartus II- Compiling Test Bench

Add the given Test bench in Compile the Test bench Section. (i.e. Assignmenets > Settings > Simulation). Then select Test Benches and Select New.



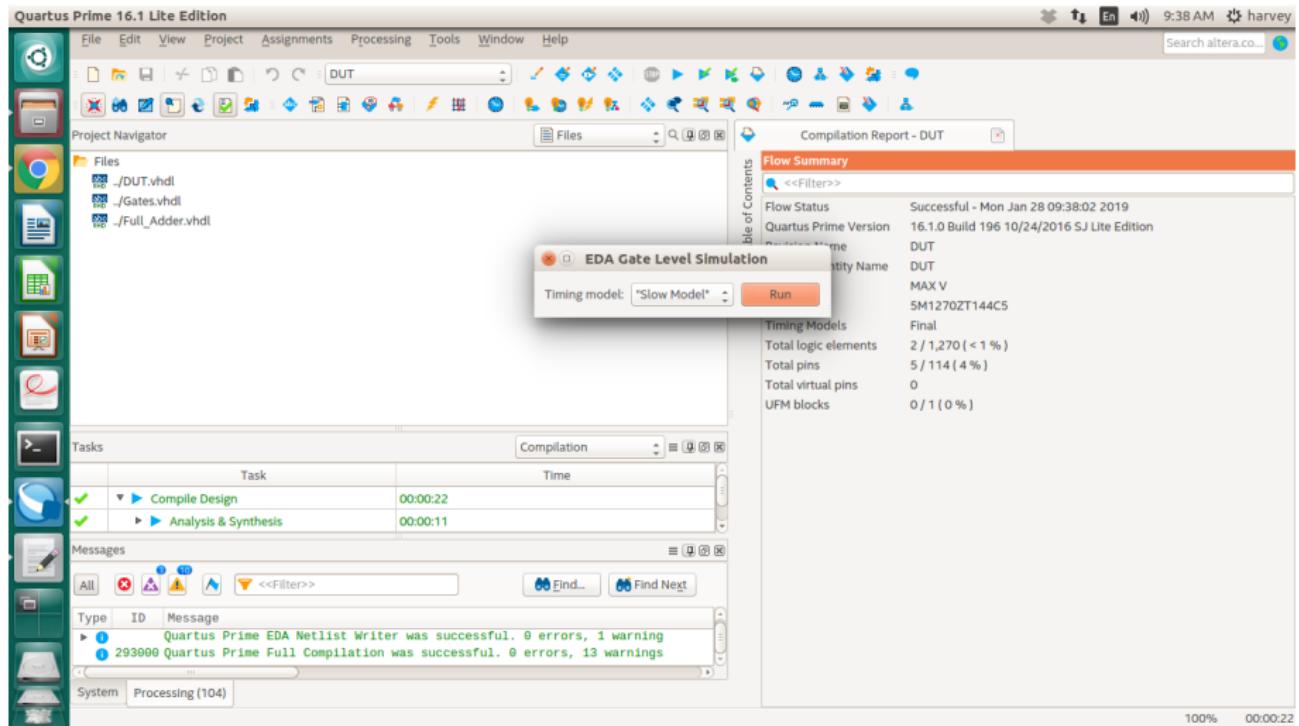
# Using Quartus II- Compiling Test Bench

Add the Test bench file and specify Top level module in the test bench file. Tick the Use tench bench to perform timing simulation and select the instance of design file mentioned in test file.



# Using Quartus II- Gate Level Simulation

Once you are done with setting up the test bench file run gate level simulation.  
(Tools > Run Simulation Tool > Gate Level Simulation)



# Using Quartus II- Generating SVF file

- After successfully completing Gate level simulation, now we will generate svf file which will be loaded on the board.
- **Gate Level Simulation successfully Completed??** If no then change the settle time in the Testbench.vhdl to 20 ns. (Line no. 110)
- Now open pin planner from Assignments > Pin Planner or Ctrl + Shift + N.
- Please note that signals which are mentioned as input and output in the Top level entity will come in the Pin Planner.

# Using Quartus II- Pin Planning

If you have followed all the steps correctly then you will be able to see all 5 signals of DUT come to list.

Pin Planner - /home/Jay/Desktop/adder/Full\_Adder - DUT

File Edit View Processing Tools Window Help

Report

Report not available

Groups Report

Tasks

- Early Pin Planning
  - Early Pin Planning...
  - Run I/O Assignment Analy...
  - Export Pin Assignments...
  - Pin Finder...
- Highlight Pins
- I/O Banks

Top View

Wire Bond

MAX V

5M1270ZT144C5

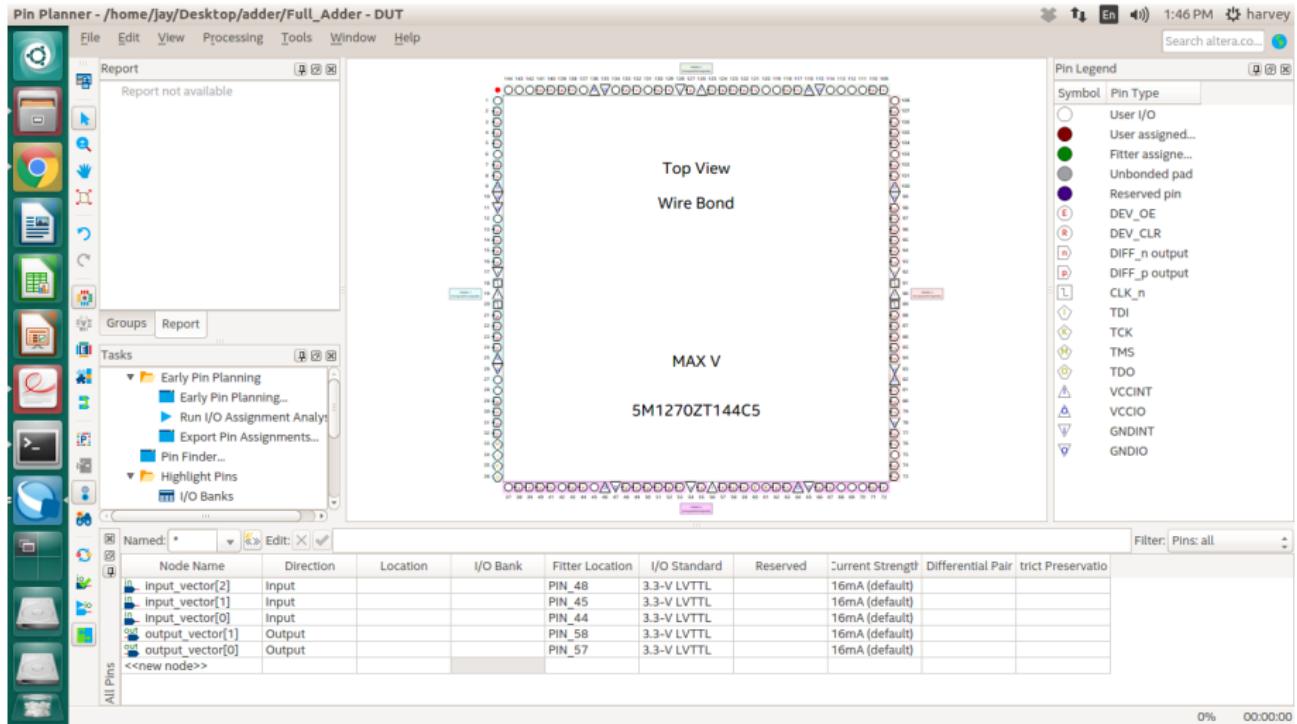
Pin Legend

Symbol	Pin Type
○	User I/O
●	User assigned...
●	Filter assign...
○	Unbonded pad
●	Reserved pin
○	DEV_OE
○	DEV_CLR
○	DIFF_n output
○	DIFF_p output
L	CLK_n
○	TDI
○	TCK
○	TMS
○	TDO
○	VCCINT
○	VCCIO
○	GNDINT
○	GNDIO

Named: \* Edit:

Node Name	Direction	Location	I/O Bank	Filter Location	I/O Standard	Reserved	Current Strength	Differential Pair	strict Preservation
input_vector[2]	Input			PIN_48	3.3-V LVTTL		16mA (default)		
input_vector[1]	Input			PIN_45	3.3-V LVTTL		16mA (default)		
input_vector[0]	Input			PIN_44	3.3-V LVTTL		16mA (default)		
output_vector[1]	Output			PIN_58	3.3-V LVTTL		16mA (default)		
output_vector[0]	Output			PIN_57	3.3-V LVTTL		16mA (default)		
<<new node>>									

0% 00:00:00



# Using Quartus II- Pin Planning

- Now we need to assign Input/Output pins to the signals. We have 3 inputs which we will assign to 3 switches and 2 outputs will be assigned to LEDs.
- For assigning pin, write pin number in Location against the respective Input/Output.
- Once pin assignments is done, close the window and compile it again (use Ctrl L).

Switch	CPLD Pin No.	LED	CPLD Pin No.
S1	48	LED1	58
S2	45	LED2	57
S3	44	LED3	55
S4	43	LED4	53
S5	42	LED5	52
S6	41	LED6	51
S7	40	LED7	50
S8	39	LED8	49

Table 2: Pin mapping for on-board Switches and LEDs

You can refer Krypton User manual for other pin configuration.

# Using Quartus II- Generation of SVF file

- For generating svf (Serial Vector File) go to Tools > Programmer.
- You should be able to see your device with DUT.pof file.
- If this window is **coming blank then go to Add File** from left side panel. Then go to output\_files and select .pof file.
- Once you are ready with .pof file then go to File > Create JAM,JBC,SVf or ISC file.
- Select File format as Serial Vector Format(.svf) and give File Name as Full\_Adder.svf and select OK to generate .svf file.
- Generally .svf is generated under output\_files folder.

# Using Quartus II- Generation of SVF file

- Once .svf is generated load that file to Krypton board using UrJAG.
- Check for all 8 combinations using switches and see the correponding outputs on LEDs.

