De0 Nano Board Demo

Sonal Gupta
Vineesh VS

March 3, 2015

OUTLINE

Altera De0-Nano FPGA board is equipped with:

- **Featured device**
  - Altera Cyclone IV EP4CE22F17C6N FPGA
  - 153 maximum FPGA I/O pins
- **Expansion header**
  - Two 40 pin Headers (GPIOs) provide 72 I/O pins, 5V power pins, two 3.3V power pins and four ground pins
- **General user input/output**
  - 8 green LEDs
  - 2 debounced pushbuttons
  - 4 position DIP switch
- **Clock system**
  - On-board 50MHz clock oscillator
- **Power Supply**
  - USB Type mini-AB port (5V)
  - DC 5V pin for each GPIO header (2 DC 5V pins)
  - 2 pin external power header (3.6-5.7V)

Board Layout

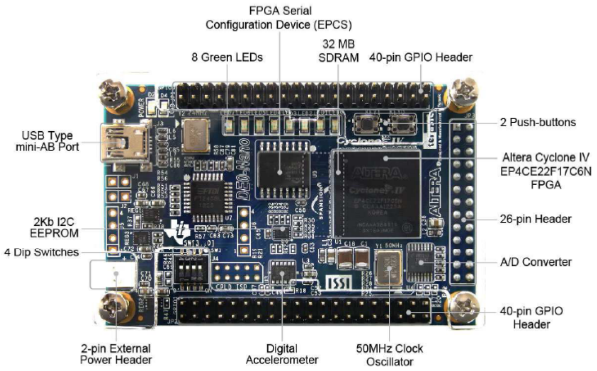


Figure 1 : De0-Nano Board front layout

Ref: Chapter 2 of De0-Nano User Manual

Board Layout

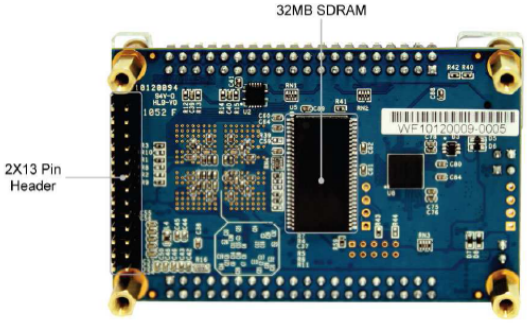


Figure 2 : De0-Nano Board back layout

Ref: Chapter 2 of De0-Nano User Manual

General User Input/Output

- The 40 pin GPIO expansion headers can be used for input-output to the implemented hardware.
- 8 on chip LEDs can be used to display output.
- 2 push buttons and 4 dip switches can be used for input.

Refer to pages 12-20 of De0-Nano User Manual for more information on GPIO.

OUTLINE

Startup and Installation

- For using pc lab machines:
  1. ssh -X student@10.107.32.<21-45>, password: student
  2. Type quartus on terminal
- For installing quartus on Linux:
  1. Copy setup using the following command: scp -r student@10.107.32.50/51:~/Desktop/Quartus ~/Downloads
  2. Refer to installation manual put on moodle.

Make a Project

1. Start quartus and select *File > New Project Wizard*.
2. Enter the directory name for project files and the name of your project (project name should be same as top level entity name).
3. Add or create new VHDL design file.
4. Select **Cyclone IV E** in *Family* and **EP4CE22F17C6** in *device* and click *finish*.

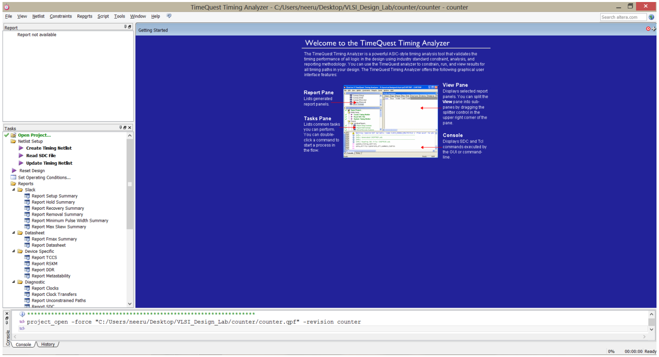
Ref: Page 46-49 from De0-Nano User Manual

Simulate using Modelsim

1. On left hand side double click your design file under *Project Navigator* > *Files* to view and edit the code.
2. Select *Flow* as *RTL Simulation* in *Tasks* pane and double click RTL Simulation.
3. Quartus will compile the design and open Modelsim window.
4. Switch to Modelsim window, go to *Compile* > *Compile*, select your testbench and hit *compile*.
5. In *Library* pane, right click testbench entity name under *work* and select *Simulate*.
6. In *Objects* pane, right click signal and select *Add Wave* to add signals to wave.
7. Type run 100ns in transcript window to simulate for 100 ns.

Create Timing Constraints

1. Switch back to Quartus window.
2. Select *Flow* as *Compilation* and press Ctrl+L for full compilation.
3. Open *Tools* > *TimeQuest Timing Analyzer*.



Welcome to the TimeQuest Timing Analyzer

The TimeQuest Timing Analyzer is a powerful ASIC-style timing analysis tool that validates the timing performance of all logic in the design using industry standard constraint, analysis, and reporting methodology. You can use the TimeQuest analyzer to constrain, run, and view results for all timing paths in your design. The TimeQuest Timing Analyzer offers the following graphical user interface features:

Report Pane Lists generated report panels.

Tasks Pane Lists common tasks you can perform. You can double-click a command to start a process in the flow.

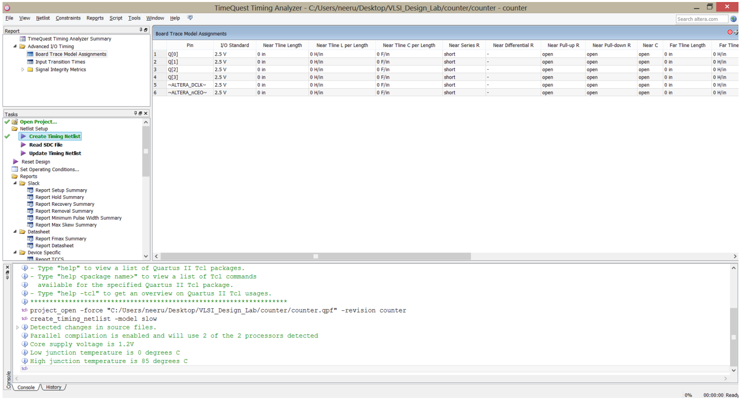
View Pane Displays selected report panels. You can split the **View** pane into sub-panes by dragging the splitter control in the upper right corner of the pane.

Console Displays SDC and Tcl commands executed by the GUI or command-line.

```
*********************************************************************
tcl> project_open -force "C:/Users/neeru/Desktop/VLSI_Design_Lab/counter/counter.qpf" -revision counter
tcl>
```

Create Timing Constraints

In Timing Analyzer window *Tasks* pane double click on *Create Timing Netlist*.



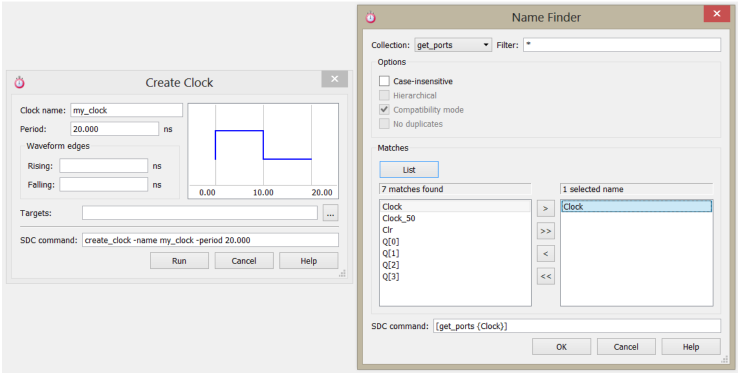
Board Trace Model Assignments

	Pin	I/O Standard	Near Tline Length	Near Tline L per Length	Near Tline C per Length	Near Series R	Near Differential R	Near Pull-up R	Near Pull-down R	Near C	Far Tline Length	Far Tline
1	Q[0]	2.5 V	0 in	0 H/in	0 F/in	short	-	open	open	open	0 in	0 H/in
2	Q[1]	2.5 V	0 in	0 H/in	0 F/in	short	-	open	open	open	0 in	0 H/in
3	Q[2]	2.5 V	0 in	0 H/in	0 F/in	short	-	open	open	open	0 in	0 H/in
4	Q[3]	2.5 V	0 in	0 H/in	0 F/in	short	-	open	open	open	0 in	0 H/in
5	~ALTERA_DCLK~	2.5 V	0 in	0 H/in	0 F/in	short	-	open	open	open	0 in	0 H/in
6	~ALTERA_nCEO~	2.5 V	0 in	0 H/in	0 F/in	short	-	open	open	open	0 in	0 H/in

```
- Type "help" to view a list of Quartus II Tcl packages.
- Type "help <package name>" to view a list of Tcl commands
  available for the specified Quartus II Tcl package.
- Type "help -tcl" to get an overview on Quartus II Tcl usages.
******************************************************************
tcl> project_open -force "C:/Users/neeru/Desktop/VLSI_Design_Lab/counter/counter.qpf" -revision counter
tcl> create_timing_netlist -model slow
Detected changes in source files.
Parallel compilation is enabled and will use 2 of the 2 processors detected
Core supply voltage is 1.2V
Low junction temperature is 0 degrees C
High junction temperature is 85 degrees C
tcl>
```

Create Timing Constraints

1. *Constraints* > *Create clock*. Click on 3 dots near Target > *List* > add clock port.
2. Write my_clk in clock name, specify time period and click *Run*.


Create Clock

Clock name: my_clock

Period: 20.000 ns

Waveform edges

Rising: ns

Falling: ns

Targets: ...

SDC command: create_clock -name my_clock -period 20.000

Name Finder

Collection: get_ports Filter: *

Options

- [ ] Case-insensitive
- [ ] Hierarchical
- [✓] Compatibility mode
- [ ] No duplicates

Matches

7 matches found: Clock, Clock_50, Clr, Q[0], Q[1], Q[2], Q[3]

1 selected name: Clock

SDC command: [get_ports {Clock}]

Create Timing Constraints

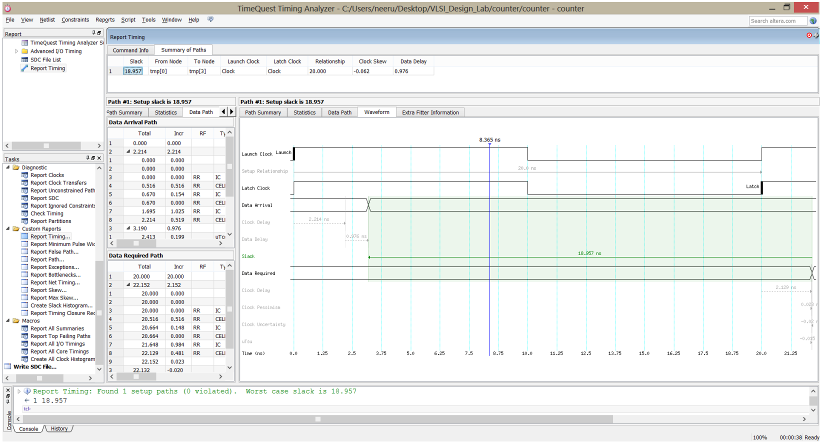
1. Double click on *Read SDC File*, *Update Timing Netlist* and *Write SDC File* in order.
2. It will create *.sdc* (Synopsys Design Constraints) file.

Timing Report

1. Switch back to Quartus window and add the *.sdc* file manually to the project, if not done by the tool.
2. Compile (ctrl+L) the design again (It will consider the .sdc file now).
3. Go back to TimeQuest Timing Analyzer and double click on *Report Timing* under *Custom Reports* in the *Tasks* pane.
4. Select From clock and To clock as my_clk and click *Report Timing*.

Timing Report

Slack positive for clk = 20ns



TimeQuest Timing Analyzer - C:/Users/neeru/Desktop/VLSI_Design_Lab/counter/counter - counter

Report Timing — Summary of Paths

	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	18.957	tmp[0]	tmp[3]	Clock	Clock	20.000	-0.062	0.976

Path #1: Setup slack is 18.957

Data Arrival Path

	Total	Incr	RF	Ty
1	0.000	0.000		
2	2.214	2.214		
1	0.000	0.000		
2	0.000	0.000		
3	0.000	0.000	RR	IC
4	0.516	0.516	RR	CELL
5	0.670	0.154	RR	IC
6	0.670	0.000	RR	CELL
7	1.695	1.025	RR	IC
8	2.214	0.519	RR	CELL
3	3.190	0.976		
1	2.413	0.199		uTco

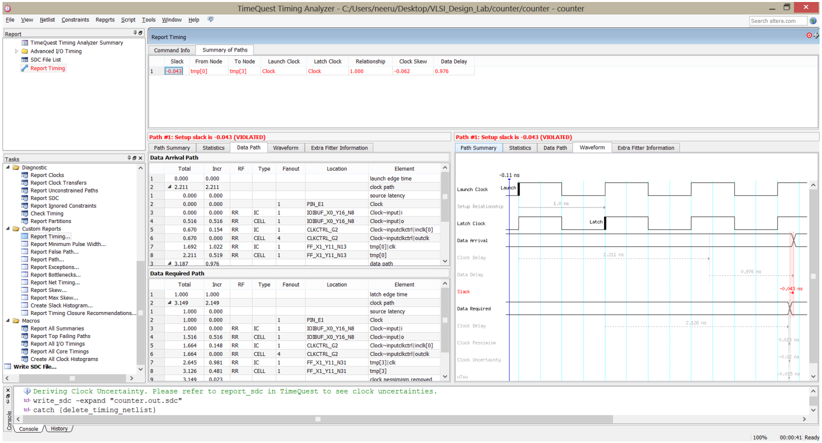
Data Required Path

	Total	Incr	RF	Ty
1	20.000	20.000		
2	22.152	2.152		
1	20.000	0.000		
2	20.000	0.000		
3	20.000	0.000	RR	IC
4	20.516	0.516	RR	CELL
5	20.664	0.148	RR	IC
6	20.664	0.000	RR	CELL
7	21.648	0.984	RR	IC
8	22.129	0.481	RR	CELL
9	22.152	0.023		
3	22.132	-0.020		

```
Report Timing: Found 1 setup paths (0 violated).  Worst case slack is 18.957
1 18.957
tcl>
```

Timing Report

Slack negative for clk = 1ns



	Slack	From Node	To Node	Launch Clock	Latch Clock	Relationship	Clock Skew	Data Delay
1	-0.043	tmp[0]	tmp[3]	Clock	Clock	1.000	-0.062	0.976

Path #1: Setup slack is -0.043 (VIOLATED)

Data Arrival Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	0.000	0.000					launch edge time
2	◢ 2.211	2.211					clock path
1	0.000	0.000					source latency
2	0.000	0.000			1	PIN_E1	Clock
3	0.000	0.000	RR	IC	1	IOIBUF_X0_Y16_N8	Clock~input|i
4	0.516	0.516	RR	CELL	1	IOIBUF_X0_Y16_N8	Clock~input|o
5	0.670	0.154	RR	IC	1	CLKCTRL_G2	Clock~inputclkctrl|inclk[0]
6	0.670	0.000	RR	CELL	4	CLKCTRL_G2	Clock~inputclkctrl|outclk
7	1.692	1.022	RR	IC	1	FF_X1_Y11_N13	tmp[0]|clk
8	2.211	0.519	RR	CELL	1	FF_X1_Y11_N13	tmp[0]
3	◢ 3.187	0.976					data path

Data Required Path

	Total	Incr	RF	Type	Fanout	Location	Element
1	1.000	1.000					latch edge time
2	◢ 3.149	2.149					clock path
1	1.000	0.000					source latency
2	1.000	0.000			1	PIN_E1	Clock
3	1.000	0.000	RR	IC	1	IOIBUF_X0_Y16_N8	Clock~input|i
4	1.516	0.516	RR	CELL	1	IOIBUF_X0_Y16_N8	Clock~input|o
5	1.664	0.148	RR	IC	1	CLKCTRL_G2	Clock~inputclkctrl|inclk[0]
6	1.664	0.000	RR	CELL	4	CLKCTRL_G2	Clock~inputclkctrl|outclk
7	2.645	0.981	RR	IC	1	FF_X1_Y11_N31	tmp[3]|clk
8	3.126	0.481	RR	CELL	1	FF_X1_Y11_N31	tmp[3]
9	3.149	0.023					clock pessimism removed

Path #1: Setup slack is -0.043 (VIOLATED)

```
Deriving Clock Uncertainty. Please refer to report_sdc in TimeQuest to see clock uncertainties.
write_sdc -expand "counter.out.sdc"
catch {delete_timing_netlist}
```

Other Timing Constraints

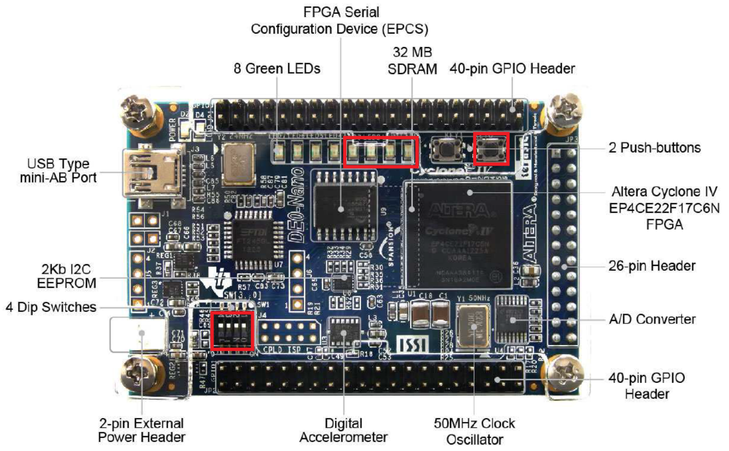
In addition to clock period, the following can be set under *Constraints* tab:

- *Set Input Delay*: Input to clock delay
- *Set Output Delay*: Clock to output delay
- *Set Maximum/Minimum Delay*: Input to output delay in case of combinational path

Experiment with these and observe changes in Timing Report (*project_name.sta.rpt* in *output_files* folder).

Programming DE0 Nano FPGA

Highlighted are the switches and LEDs we are going to use for the *counter*.

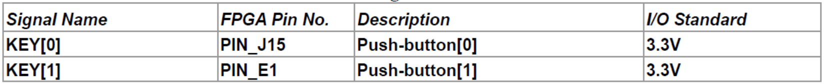


Ref: Pages 76-82 of De0-Nano User Manual

Programming DE0 Nano FPGA

Locations of the push button switches:

Table 3-1 Pin Assignments for Push-buttons



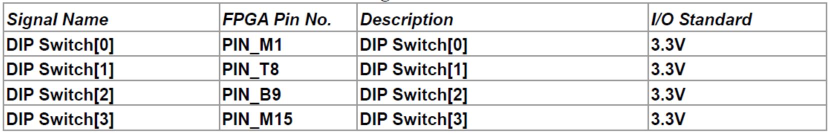
Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
KEY[0]	PIN_J15	Push-button[0]	3.3V
KEY[1]	PIN_E1	Push-button[1]	3.3V

Ref: Pages 12-20 of De0-Nano User Manual

Programming DE0 Nano FPGA

Locations of the DIP switches:

Table 3-3 Pin Assignments for DIP Switches



Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
DIP Switch[0]	PIN_M1	DIP Switch[0]	3.3V
DIP Switch[1]	PIN_T8	DIP Switch[1]	3.3V
DIP Switch[2]	PIN_B9	DIP Switch[2]	3.3V
DIP Switch[3]	PIN_M15	DIP Switch[3]	3.3V

Ref: Pages 12-20 of De0-Nano User Manual

Programming DE0 Nano FPGA

Locations of the LEDs:

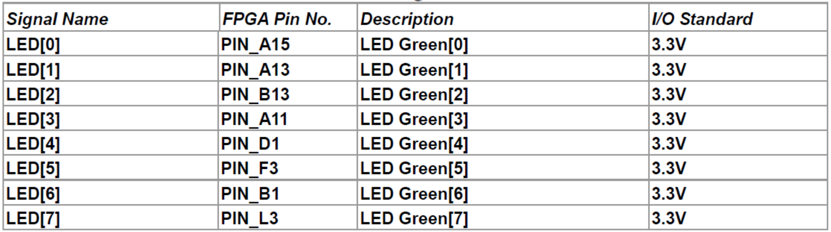


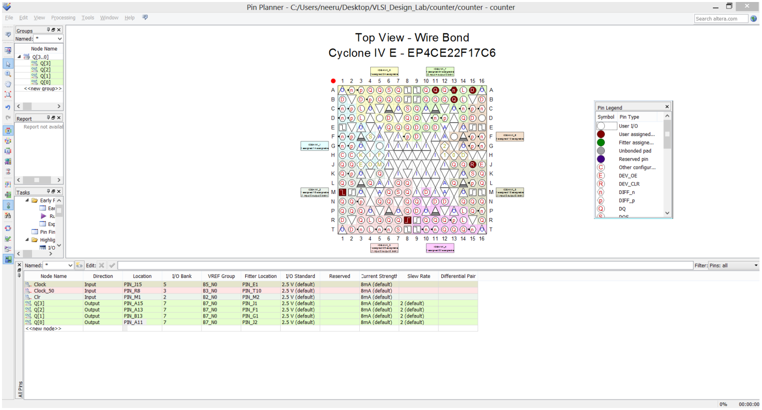
Table 3-2 Pin Assignments for LEDs

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
LED[0]	PIN_A15	LED Green[0]	3.3V
LED[1]	PIN_A13	LED Green[1]	3.3V
LED[2]	PIN_B13	LED Green[2]	3.3V
LED[3]	PIN_A11	LED Green[3]	3.3V
LED[4]	PIN_D1	LED Green[4]	3.3V
LED[5]	PIN_F3	LED Green[5]	3.3V
LED[6]	PIN_B1	LED Green[6]	3.3V
LED[7]	PIN_L3	LED Green[7]	3.3V

Ref: Pages 12-20 of De0-Nano User Manual

Programming DE0 Nano FPGA

From *Quartus* main window, Assignments > Pin Planner.
Enter the pin numbers in the *Location* column. (*Clock_50 must be connected to PIN_R8* (50MHz clock))


Pin Planner - C:/Users/neeru/Desktop/VLSI_Design_Lab/counter/counter - counter

Top View - Wire Bond
Cyclone IV E - EP4CE22F17C6

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair
Clock	Input	PIN_J15	5	B5_N0	PIN_E1	2.5 V (default)		8mA (default)		
Clock_50	Input	PIN_R8	3	B3_N0	PIN_T10	2.5 V (default)		8mA (default)		
Clr	Input	PIN_M1	2	B2_N0	PIN_M2	2.5 V (default)		8mA (default)		
Q[3]	Output	PIN_A15	7	B7_N0	PIN_J1	2.5 V (default)		8mA (default)	2 (default)	
Q[2]	Output	PIN_A13	7	B7_N0	PIN_F1	2.5 V (default)		8mA (default)	2 (default)	
Q[1]	Output	PIN_B13	7	B7_N0	PIN_G1	2.5 V (default)		8mA (default)	2 (default)	
Q[0]	Output	PIN_A11	7	B7_N0	PIN_J2	2.5 V (default)		8mA (default)	2 (default)	
<<new node>>										

Ref: Pages 76-82 of De0-Nano User Manual

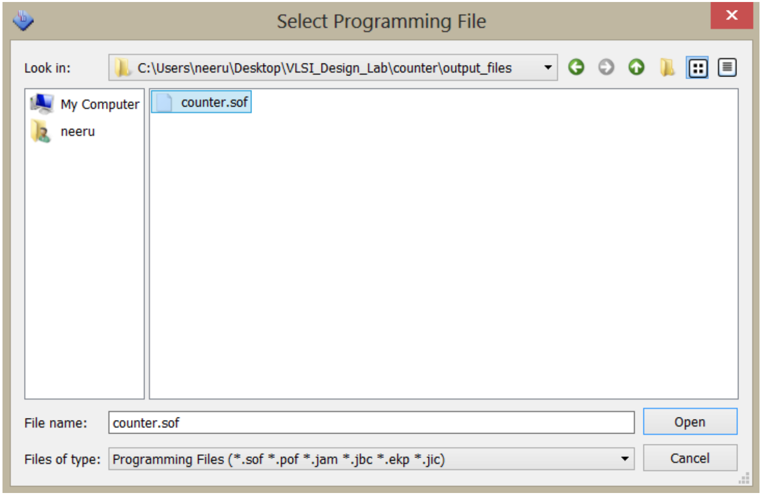
Programming DE0 Nano FPGA

- Recompile (Ctrl+L)
- From *Quartus* main window, *Tools* > *Programmer*.
- Connect the board using USB cable.
- Select *USB-Blaster* in *Hardware Setup*.
- In case you face problem with Hardware Setup, refer to installation manual on moodle and run the Hardware Setup script as instructed.

Ref: Pages 76-82 of De0-Nano User Manual

Programming DE0 Nano FPGA

Click on *Add File* and select the *.sof (SRAM Object File)* file


Select Programming File

Look in: C:\Users\neeru\Desktop\VLSI_Design_Lab\counter\output_files

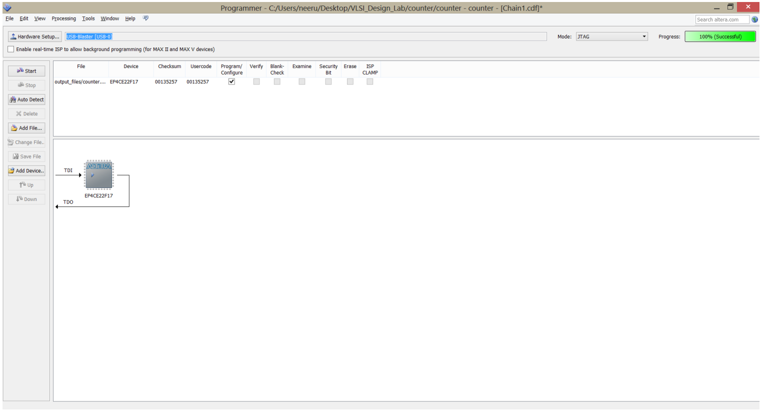
File name: counter.sof

Files of type: Programming Files (*.sof *.pof *.jam *.jbc *.ekp *.jic)

Ref: Pages 76-82 of De0-Nano User Manual

Programming DE0 Nano FPGA

Click on *Start* to program the device


Programmer - C:/Users/neeru/Desktop/VLSI_Design_Lab/counter/counter - counter - [Chain1.cdf]*

Hardware Setup... USB-Blaster [USB-0] Mode: JTAG Progress: 100% (Successful)

Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
output_files/counter....	EP4CE22F17	00135257	00135257	✓						

Ref: Pages 76-82 of De0-Nano User Manual

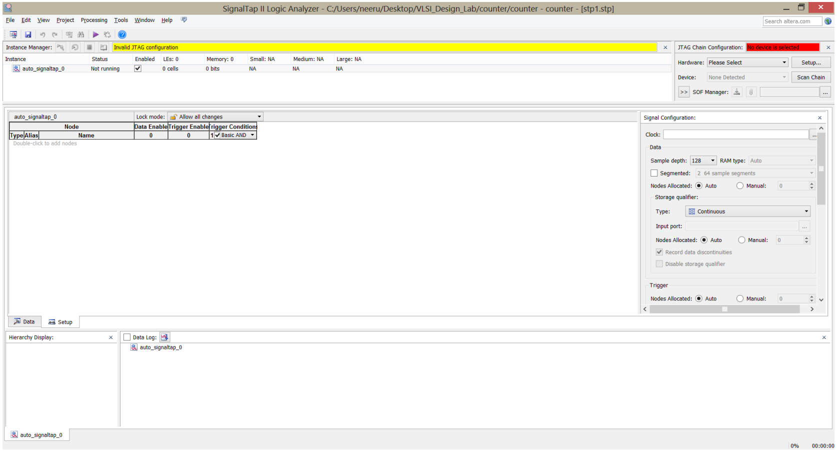
Verifying the design in FPGA
Setup

- Go to *Tools* > *Options* > *Internet Connectivity* > *TalkBack Options*
- Tick on *Enable sending TalkBack data to Altera*

Verifying the design in FPGA

SignalTap II Logic Analyzer

Tools > SignalTap II Logic Analyzer



SignalTap II Logic Analyzer - C:/Users/neeru/Desktop/VLSI_Design_Lab/counter/counter - counter - [stp1.stp]

File Edit View Project Processing Tools Window Help

Instance Manager: Invalid JTAG configuration

JTAG Chain Configuration: No device is selected

Instance	Status	Enabled	LEs: 0	Memory: 0	Small: NA	Medium: NA	Large: NA
auto_signaltap_0	Not running	✓	0 cells	0 bits	NA	NA	NA

Hardware: Please Select — Setup...
Device: None Detected — Scan Chain
SOF Manager:

auto_signaltap_0 — Lock mode: Allow all changes

Node			Data Enable	Trigger Enable	Trigger Conditions
Type	Alias	Name	0	0	1 ✓ Basic AND

Double-click to add nodes

Signal Configuration:
Clock:
Data
Sample depth: 128 RAM type: Auto
Segmented: 2 64 sample segments
Nodes Allocated: ● Auto ○ Manual: 0
Storage qualifier:
Type: Continuous
Input port:
Nodes Allocated: ● Auto ○ Manual: 0
✓ Record data discontinuities
Disable storage qualifier
Trigger
Nodes Allocated: ● Auto ○ Manual: 0

Data | Setup

Hierarchy Display:

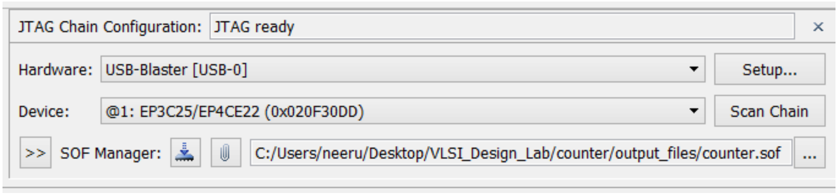
Data Log: auto_signaltap_0

auto_signaltap_0

0% 00:00:00

Verifying the design in FPGA

Select the hardware *USB-Blaster* and *.sof* file.



JTAG Chain Configuration: JTAG ready

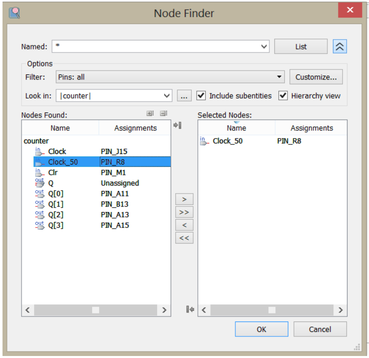
Hardware: USB-Blaster [USB-0]

Device: @1: EP3C25/EP4CE22 (0x020F30DD)

SOF Manager: C:/Users/neeru/Desktop/VLSI_Design_Lab/counter/output_files/counter.sof

Verifying the design in FPGA

In *Signal Configuration* window, select *Clock_50* for *Clock*



Node Finder

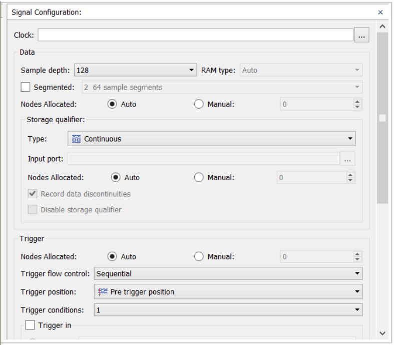
Named: *
Options
Filter: Pins: all
Look in: |counter|
[✓] Include subentities [✓] Hierarchy view

Nodes Found:

Name	Assignments
counter	
Clock	PIN_J15
Clock_50	PIN_R8
Clr	PIN_M1
Q	Unassigned
Q[0]	PIN_A11
Q[1]	PIN_B13
Q[2]	PIN_A13
Q[3]	PIN_A15

Selected Nodes:

Name	Assignments
Clock_50	PIN_R8



Signal Configuration:

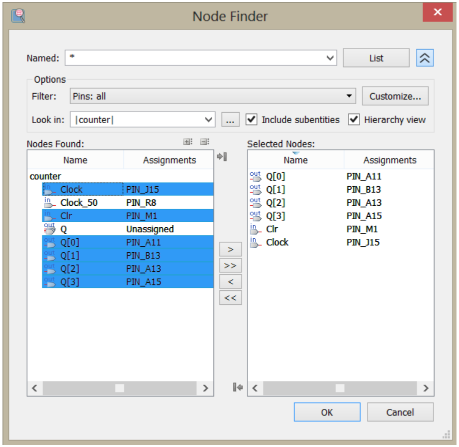
Clock:

Data
Sample depth: 128 RAM type: Auto
Segmented: 2 64 sample segments
Nodes Allocated: [✓ Auto] Manual: 0
Storage qualifier:
Type: Continuous
Input port:
Nodes Allocated: [✓ Auto] Manual: 0
[✓] Record data discontinuities
Disable storage qualifier

Trigger
Nodes Allocated: [✓ Auto] Manual: 0
Trigger flow control: Sequential
Trigger position: Pre trigger position
Trigger conditions: 1
Trigger in

Verifying the design in FPGA

Go to the *Setup* tab and double-click to add the signals to which are to be monitored. In Filter, you may select *Pins: all*

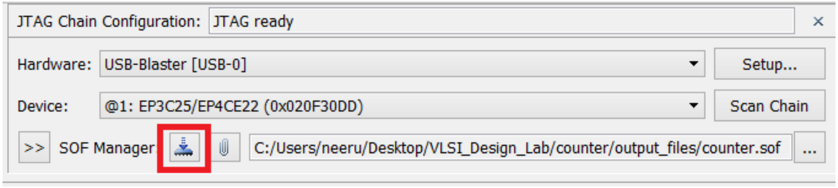


Name	Assignments
counter	
Clock	PIN_J15
Clock_50	PIN_R8
Clr	PIN_M1
Q	Unassigned
Q[0]	PIN_A11
Q[1]	PIN_B13
Q[2]	PIN_A13
Q[3]	PIN_A15

Name	Assignments
Q[0]	PIN_A11
Q[1]	PIN_B13
Q[2]	PIN_A13
Q[3]	PIN_A15
Clr	PIN_M1
Clock	PIN_J15

Verifying the design in FPGA

- Compile the design again by clicking the *Play* button in the SignalTap Logic Analyzer.
- Once compilation is done, click on the button in red square, to program the device



JTAG Chain Configuration: JTAG ready

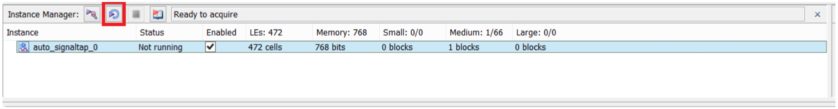
Hardware: USB-Blaster [USB-0]

Device: @1: EP3C25/EP4CE22 (0x020F30DD)

SOF Manager: C:/Users/neeru/Desktop/VLSI_Design_Lab/counter/output_files/counter.sof

Verifying the design in FPGA

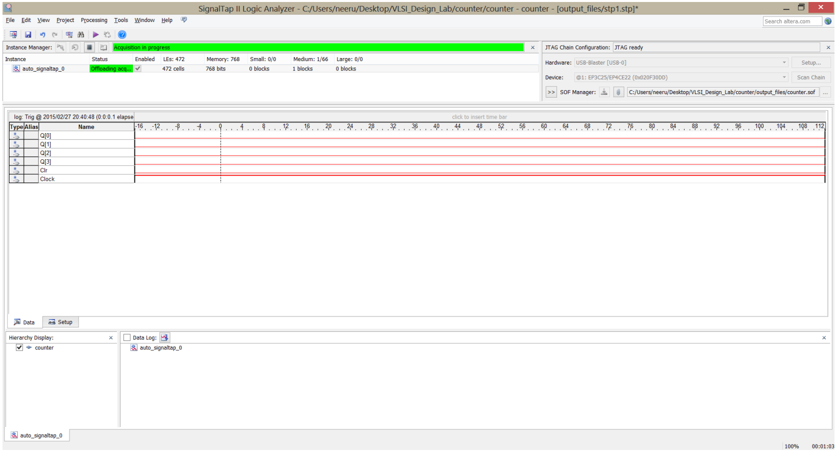
Do *Autorun Analysis* to continuously monitor the signals from the FPGA board



Instance	Status	Enabled	LEs: 472	Memory: 768	Small: 0/0	Medium: 1/66	Large: 0/0
auto_signaltap_0	Not running	✓	472 cells	768 bits	0 blocks	1 blocks	0 blocks

Verifying the design in FPGA

Verify the design


SignalTap II Logic Analyzer - C:/Users/neeru/Desktop/VLSI_Design_Lab/counter/counter - counter - [output_files/stp1.stp]*

Instance Manager: Acquisition in progress

Instance	Status	Enabled	LEs: 472	Memory: 768	Small: 0/0	Medium: 1/66	Large: 0/0
auto_signaltap_0	Offloading acq...	✓	472 cells	768 bits	0 blocks	1 blocks	0 blocks

JTAG Chain Configuration: JTAG ready

Hardware: USB-Blaster [USB-0]

Device: @1: EP3C25/EP4CE22 (0x020F30DD)

SOF Manager: C:/Users/neeru/Desktop/VLSI_Design_Lab/counter/output_files/counter.sof

log: Trig @ 2015/02/27 20:40:48 (0:0:0.1 elapse

Type	Alias	Name
		Q[0]
		Q[1]
		Q[2]
		Q[3]
		Clr
		Clock

Hierarchy Display: ✓ counter

Data Log: auto_signaltap_0

100% 00:01:03

Additional Information

Read about *.sof*, *.cdf*, *.sda* etc from Altera site.