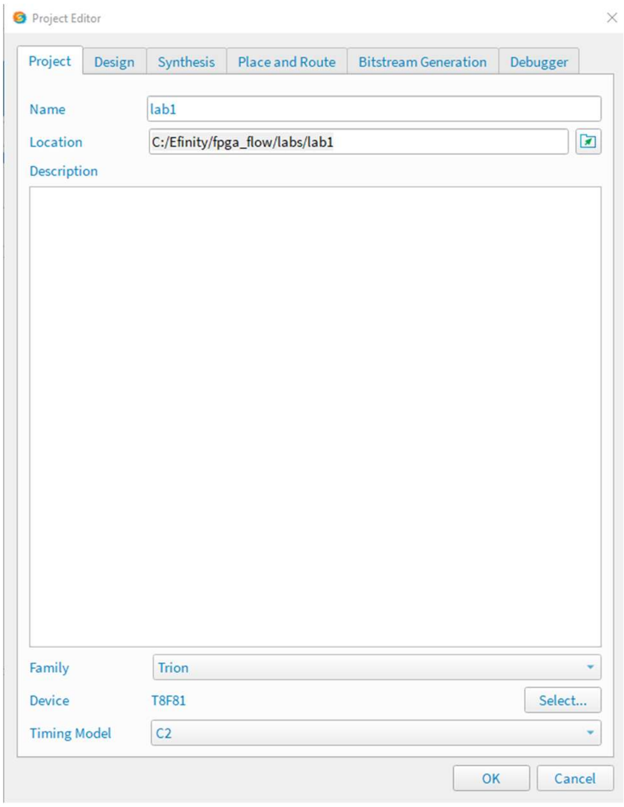
Efinix Design Flow

Steps

Create a Efinity Project

1. Open Efinity by selecting Start > Efinity 2022.1 > Efinity 2022.1.
2. Click File > Create Project… to start the wizard. You will see Project Editor dialog box.
3. Click the Browse button of the *Project location* field of the **New Project** form, browse to **C:/Efinity/fpga_flow/labs/lab1**.
4. Enter lab1 in the Project name field. Make sure that the Family field selected Trion and device field selected the T8F8. Click Design tab.

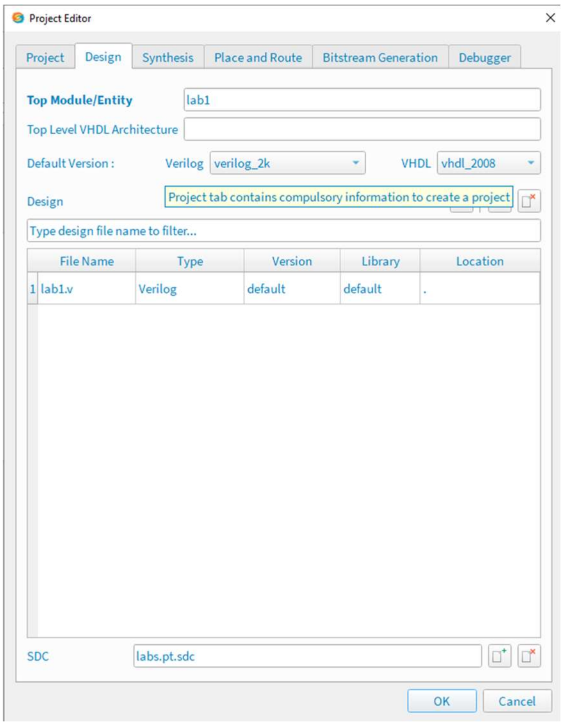


Name: lab1
Location: C:/Efinity/fpga_flow/labs/lab1
Description
Family: Trion
Device: T8F81
Timing Model: C2

Project Name and Location entry

5. Enter lab1 in the Top Module/Entity field. Select **verilog_2k** as the Target Language.
6. Click on the Add design file button and browse to the C:\Efinity\fpga_flow\sources\lab1 directory, select lab1.v.

7. Click on the Add SDC file button and browse to the C:\Efinity\fpga_flow\sources\lab1 directory, select labs.pt.sdc, click **OK**.



Top Module/Entity: lab1

Top Level VHDL Architecture:

Default Version : Verilog verilog_2k VHDL vhdl_2008

Design

Project tab contains compulsory information to create a project

Type design file name to filter...

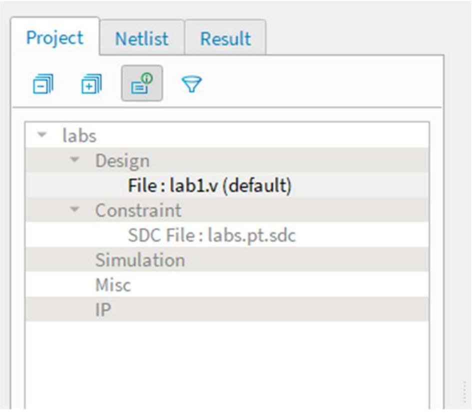
	File Name	Type	Version	Library	Location
1	lab1.v	Verilog	default	default	.

SDC: labs.pt.sdc

Add design and SDC files

Open the lab1.v source and analyze the content.

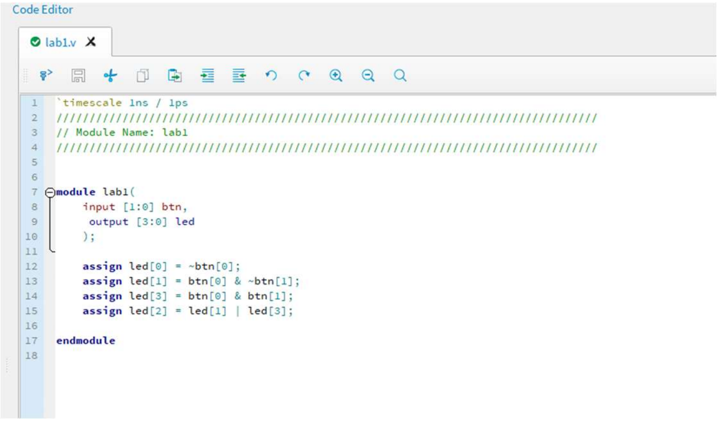
1. In the Design pane, double-click the lab1.v entry to open the file in text mode.



labs
- Design
  - File : lab1.v (default)
- Constraint
  - SDC File : labs.pt.sdc
- Simulation
- Misc
- IP

Opening the source file

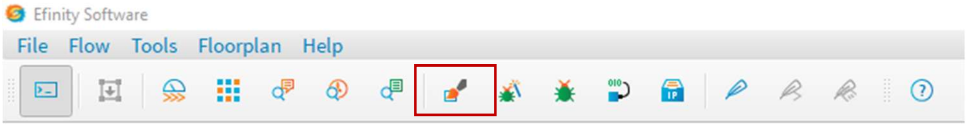
2. Notice in the Verilog code that the first line defines the timescale directive for the simulator. Lines 2-4 are comment lines describing the module name and the purpose of the module. Line 7 defines the beginning (marked with keyword module) and Line 17 defines the end of the module (marked with keyword endmodule). Lines 8-9 defines the input and output ports whereas lines 12-15 defines the actual functionality.



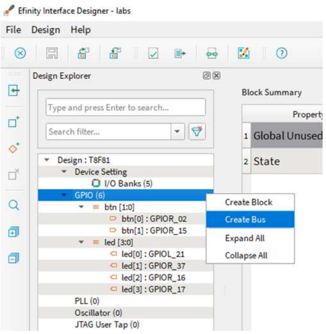
```
`timescale 1ns / 1ps
///////////////////////////////////////////////////////////////////////////////
// Module Name: lab1
///////////////////////////////////////////////////////////////////////////////


module lab1(
    input [1:0] btn,
     output [3:0] led
    );

    assign led[0] = ~btn[0];
    assign led[1] = btn[0] & ~btn[1];
    assign led[3] = btn[0] & btn[1];
    assign led[2] = led[1] | led[3];

endmodule
```

Create the labs.pt.sdc source.

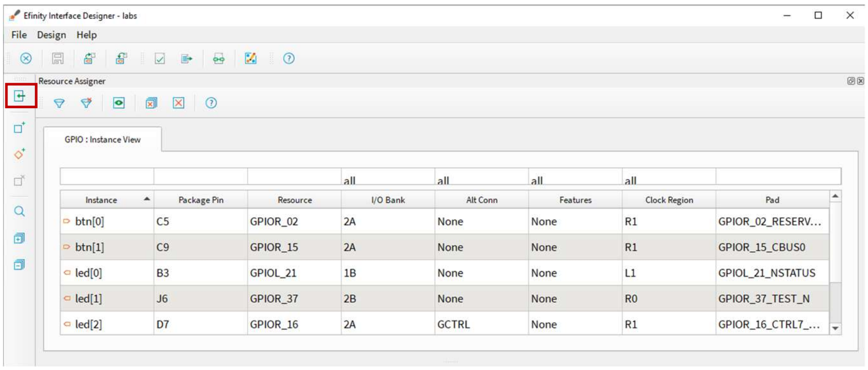
1. Open Efinity Interface Designer.



2. Right click to GPIO and select create bus. Enter the bus name as btn and select MSB as 1, LSB as 0.
3. Right click to GPIO and select create bus. Enter the bus name as led and select MSB as 3, LSB as 0.

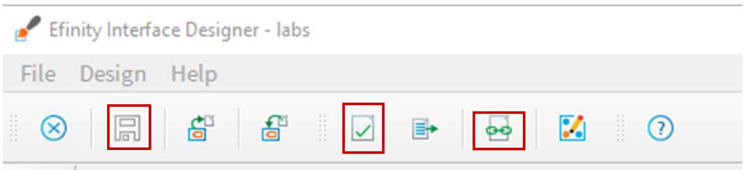


4. Click show/hide GPIO Resource Assigner. Enter the resource part according to the datasheet.



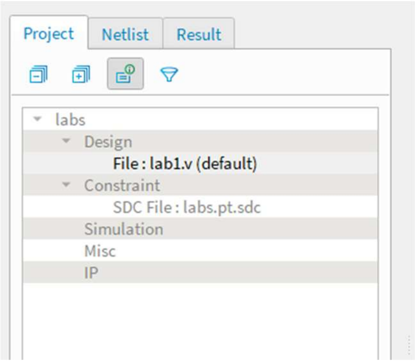
			all	all	all	all	
Instance	Package Pin	Resource	I/O Bank	Alt Conn	Features	Clock Region	Pad
btn[0]	C5	GPIOR_02	2A	None	None	R1	GPIOR_02_RESERV...
btn[1]	C9	GPIOR_15	2A	None	None	R1	GPIOR_15_CBUS0
led[0]	B3	GPIOL_21	1B	None	None	L1	GPIOL_21_NSTATUS
led[1]	J6	GPIOR_37	2B	None	None	R0	GPIOR_37_TEST_N
led[2]	D7	GPIOR_16	2A	GCTRL	None	R1	GPIOR_16_CTRL7_...

5. Save and check the design then click Generate Efinity constraint file.

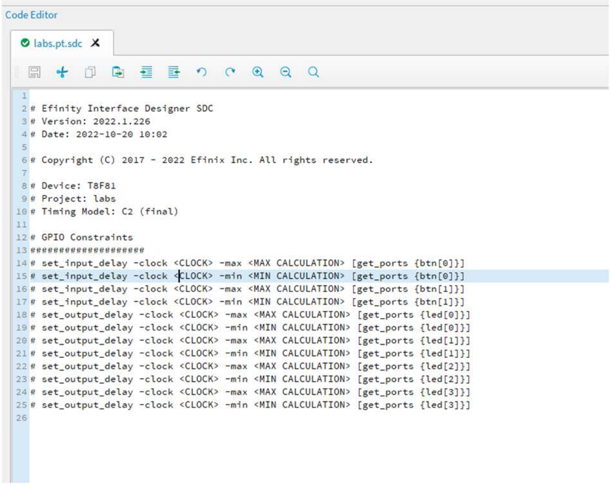


Open the labs.pt.sdc source and analyze the content.

1. In the Sources pane, expand the Constraints folder and double-click the labs.pt.sdc entry to open the file in text mode.



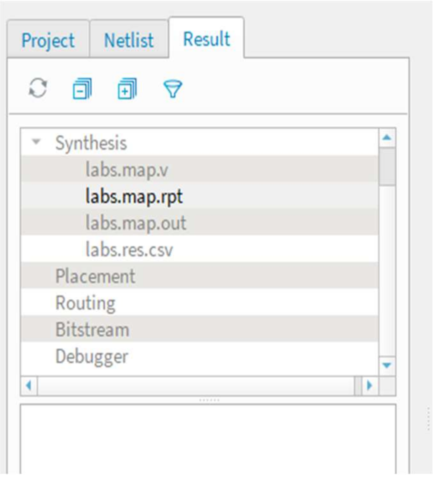
2. Lines 14-17 define the pin locations for the input buttons and lines 18-25 define pin locations for output LEDs.

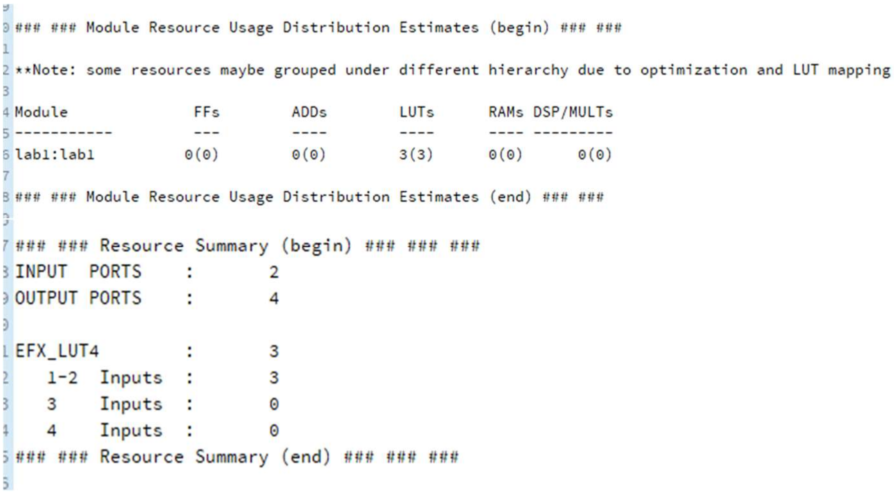


```
 1
 2 # Efinity Interface Designer SDC
 3 # Version: 2022.1.226
 4 # Date: 2022-10-20 10:02
 5
 6 # Copyright (C) 2017 - 2022 Efinix Inc. All rights reserved.
 7
 8 # Device: T8F81
 9 # Project: labs
10 # Timing Model: C2 (final)
11
12 # GPIO Constraints
13 ####################
14 # set_input_delay -clock <CLOCK> -max <MAX CALCULATION> [get_ports {btn[0]}]
15 # set_input_delay -clock <CLOCK> -min <MIN CALCULATION> [get_ports {btn[0]}]
16 # set_input_delay -clock <CLOCK> -max <MAX CALCULATION> [get_ports {btn[1]}]
17 # set_input_delay -clock <CLOCK> -min <MIN CALCULATION> [get_ports {btn[1]}]
18 # set_output_delay -clock <CLOCK> -max <MAX CALCULATION> [get_ports {led[0]}]
19 # set_output_delay -clock <CLOCK> -min <MIN CALCULATION> [get_ports {led[0]}]
20 # set_output_delay -clock <CLOCK> -max <MAX CALCULATION> [get_ports {led[1]}]
21 # set_output_delay -clock <CLOCK> -min <MIN CALCULATION> [get_ports {led[1]}]
22 # set_output_delay -clock <CLOCK> -max <MAX CALCULATION> [get_ports {led[2]}]
23 # set_output_delay -clock <CLOCK> -min <MIN CALCULATION> [get_ports {led[2]}]
24 # set_output_delay -clock <CLOCK> -max <MAX CALCULATION> [get_ports {led[3]}]
25 # set_output_delay -clock <CLOCK> -min <MIN CALCULATION> [get_ports {led[3]}]
26
```

Synthesize the Design

1. Before performing the synthesis process, we stop the synthesis, placement, routing and bitstream flow by pressing the toogle automated button.
2. Click Synthesis under the dashboard.
3. When the process is completed. Click Result tab and expand Synthesis. Open labs.map.rpt file as we want to look at the synthesis output before progressing to the placement and routing the stage.

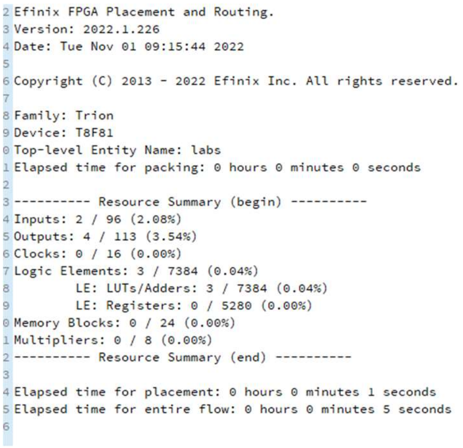




```
### ### Module Resource Usage Distribution Estimates (begin) ### ###

**Note: some resources maybe grouped under different hierarchy due to optimization and LUT mapping

Module               FFs         ADDs        LUTs       RAMs DSP/MULTs
-----------          ---         ----        ----       ---- ---------
lab1:lab1           0(0)        0(0)        3(3)       0(0)      0(0)

### ### Module Resource Usage Distribution Estimates (end) ### ###

### ### Resource Summary (begin) ### ### ###
INPUT  PORTS    :        2
OUTPUT PORTS    :        4

EFX_LUT4        :        3
   1-2  Inputs  :        3
   3    Inputs  :        0
   4    Inputs  :        0
### ### Resource Summary (end) ### ### ###
```

Place the Design

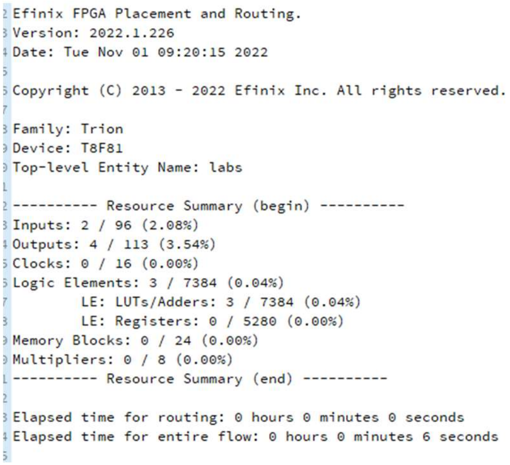
1. Click Placement under the dashboard.
2. When the process is completed. Click Result tab and expand Placement. Open labs.place.rpt file as we want to look at the placement output before progressing to the Routing stage.



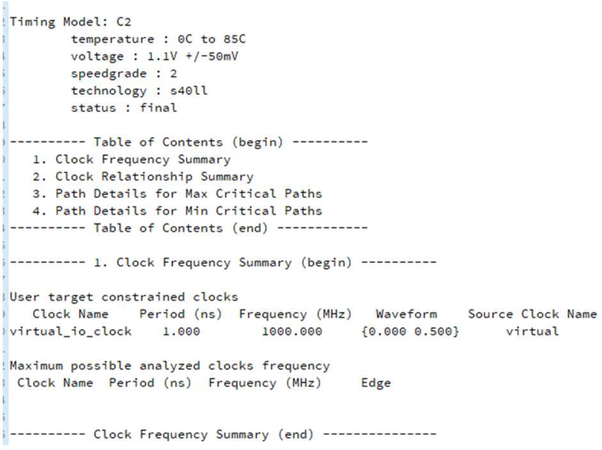
```
Efinix FPGA Placement and Routing.
Version: 2022.1.226
Date: Tue Nov 01 09:15:44 2022

Copyright (C) 2013 - 2022 Efinix Inc. All rights reserved.

Family: Trion
Device: T8F81
Top-level Entity Name: labs
Elapsed time for packing: 0 hours 0 minutes 0 seconds

---------- Resource Summary (begin) ----------
Inputs: 2 / 96 (2.08%)
Outputs: 4 / 113 (3.54%)
Clocks: 0 / 16 (0.00%)
Logic Elements: 3 / 7384 (0.04%)
        LE: LUTs/Adders: 3 / 7384 (0.04%)
        LE: Registers: 0 / 5280 (0.00%)
Memory Blocks: 0 / 24 (0.00%)
Multipliers: 0 / 8 (0.00%)
---------- Resource Summary (end) ----------

Elapsed time for placement: 0 hours 0 minutes 1 seconds
Elapsed time for entire flow: 0 hours 0 minutes 5 seconds
```

Route the Design

1. Click Routing under the dashboard.
2. When the process is completed. Click Result tab and expand Routing. Open labs.route.rpt and labs.timig.rpt files as we want to look at the routing output before progressing to the Bitstream stage.



```
Efinix FPGA Placement and Routing.
Version: 2022.1.226
Date: Tue Nov 01 09:20:15 2022

Copyright (C) 2013 - 2022 Efinix Inc. All rights reserved.

Family: Trion
Device: T8F81
Top-level Entity Name: labs

---------- Resource Summary (begin) ----------
Inputs: 2 / 96 (2.08%)
Outputs: 4 / 113 (3.54%)
Clocks: 0 / 16 (0.00%)
Logic Elements: 3 / 7384 (0.04%)
        LE: LUTs/Adders: 3 / 7384 (0.04%)
        LE: Registers: 0 / 5280 (0.00%)
Memory Blocks: 0 / 24 (0.00%)
Multipliers: 0 / 8 (0.00%)
---------- Resource Summary (end) ----------

Elapsed time for routing: 0 hours 0 minutes 0 seconds
Elapsed time for entire flow: 0 hours 0 minutes 6 seconds
```



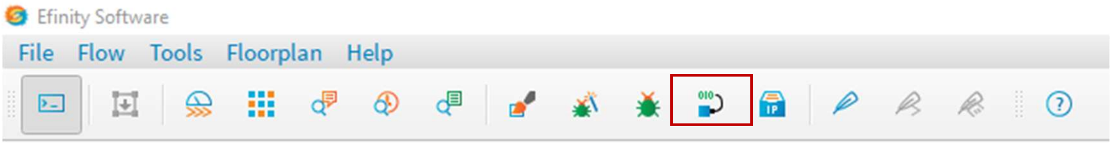
```
Timing Model: C2
        temperature : 0C to 85C
        voltage : 1.1V +/-50mV
        speedgrade : 2
        technology : s40ll
        status : final

---------- Table of Contents (begin) ----------
   1. Clock Frequency Summary
   2. Clock Relationship Summary
   3. Path Details for Max Critical Paths
   4. Path Details for Min Critical Paths
---------- Table of Contents (end) ------------

---------- 1. Clock Frequency Summary (begin) ----------

User target constrained clocks
   Clock Name     Period (ns)  Frequency (MHz)   Waveform    Source Clock Name
virtual_io_clock    1.000        1000.000     {0.000 0.500}      virtual

Maximum possible analyzed clocks frequency
 Clock Name   Period (ns)  Frequency (MHz)     Edge


---------- Clock Frequency Summary (end) ---------------
```

Install Zadig and Connect Xyloni with Micro-USB

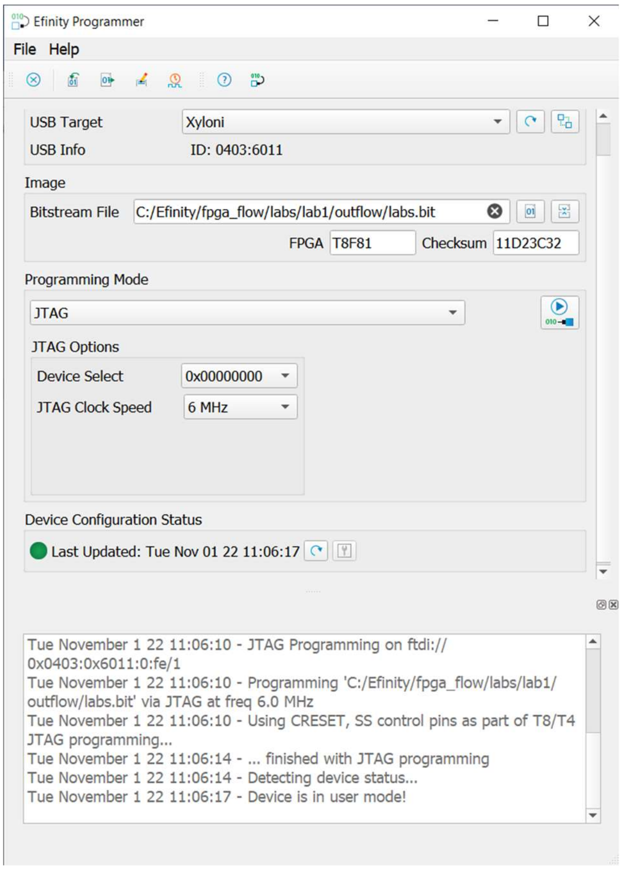
1. Fort the installing zadig, read Xyloni Development Kit User Guide > Install USB Drivers content and install zadig.
2. Connect the board to your computer with the appropriate cable and power it up.
3. Run the Zadig software.
4. Choose Options > List All Devices.
5. Select libusb-win32 or libusbK in the Driver drop-down list. (Do not choose
6. WinUSB.)
7. Click Replace Driver.
8. Close the Zadig software.

Generate the Bitstream and Verify Functionality

1. Click Bitstream under the dashboard.
2. This process will have generated a labs.bit file under the Bitstream.
3. Click Open programmer button.



4. Click Refresh USB Targets and select your Xyloni board.
5. Click Select image file button and add labs.bit file.
6. Under the programming mode select JTAG and click start program button.
7. The Device Configuration Status will lit when the device is programmed.



USB Target: Xyloni

USB Info: ID: 0403:6011

Bitstream File: C:/Efinity/fpga_flow/labs/lab1/outflow/labs.bit

FPGA: T8F81 Checksum: 11D23C32

JTAG

Device Select: 0x00000000

JTAG Clock Speed: 6 MHz

● Last Updated: Tue Nov 01 22 11:06:17

```
Tue November 1 22 11:06:10 - JTAG Programming on ftdi://
0x0403:0x6011:0:fe/1
Tue November 1 22 11:06:10 - Programming 'C:/Efinity/fpga_flow/labs/lab1/
outflow/labs.bit' via JTAG at freq 6.0 MHz
Tue November 1 22 11:06:10 - Using CRESET, SS control pins as part of T8/T4
JTAG programming...
Tue November 1 22 11:06:14 - ... finished with JTAG programming
Tue November 1 22 11:06:14 - Detecting device status...
Tue November 1 22 11:06:17 - Device is in user mode!
```

8. Verify the functionality by pushing the buttons and observing the output on the LEDs (Refer to the earlier logic diagram).
9. When satisfied, power OFF the board.
10. Close the Efinity program by selecting File > Exit.