Lab2: Push Buttons and LED Control

National Chiao Tung University

Chun-Jen Tsai

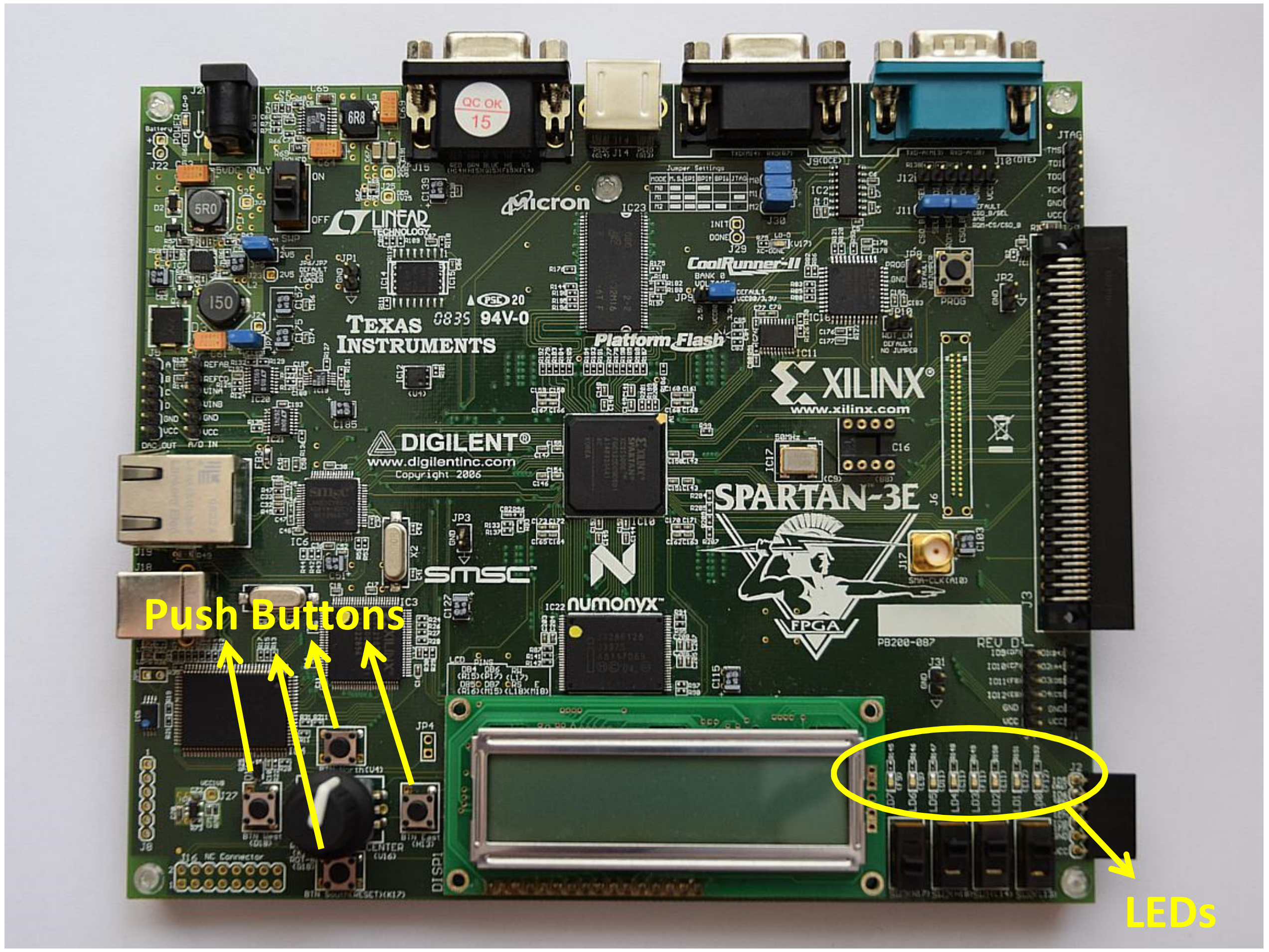
9/27/2016

Lab 2: Push Buttons and LED Control

- In this lab, you will use the Spartan-3E board to implement a simple I/O control circuit
  - There are 5 push-buttons and 8 LED lights on the board
  - You must design a synchronous circuit that reads the push-button inputs and use the inputs to increase/decrease the value of a 4-bit counter
  - You must display the counter value using the 8 LEDs in sign-extended 2’s complement binary format at any time
- You will demo the design to your TA during the lab hours on 10/4

Buttons and LEDs on the Board

- The Spartan-3E Starter Kit



The System Behavior of Lab2

- Your circuit has a 4-bit counter register
  - The counter value is set to zero upon reset
  - The counter value is in 2’complement from –8 ~ 7
- Two push-buttons are used to increase/decrease the counter:
  - Push the west/east buttons increase/decrease the counter by 1
  - If the counter value becomes grater than 7, it is truncated to 7; if the value is smaller than –8, it is set to –8
- The south button is used to reset the counter to zero
- The counter register value will be displayed in sign-extended binary format using the 8 LEDs at any time

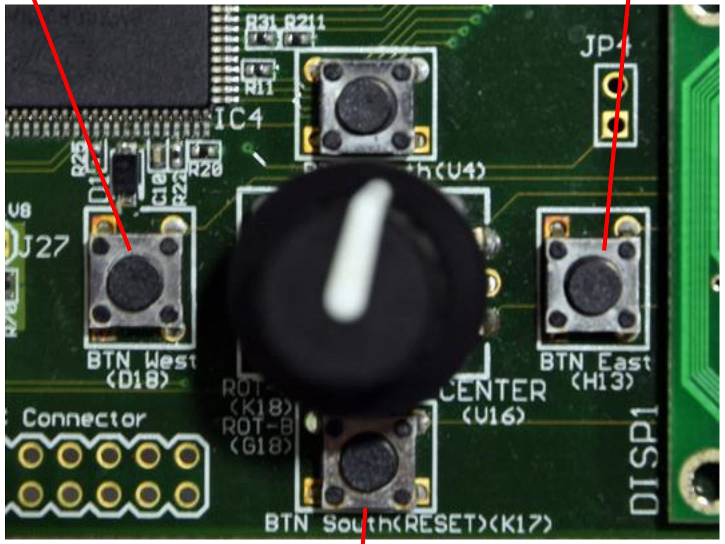
Push buttons and LEDs Control

When a push button is pressed, the following actions is applied to the counter register:

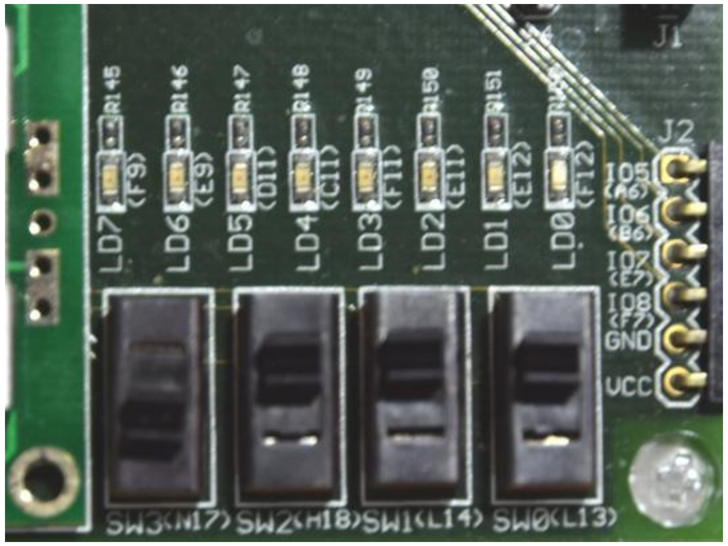
Use the LED lights to display the value of the counter register:
“on” means “1” and “off” means “0”.
The most significant bit is LD7.

Counter + 1

Counter – 1



Counter reset



User I/O Pins of an FPGA IC

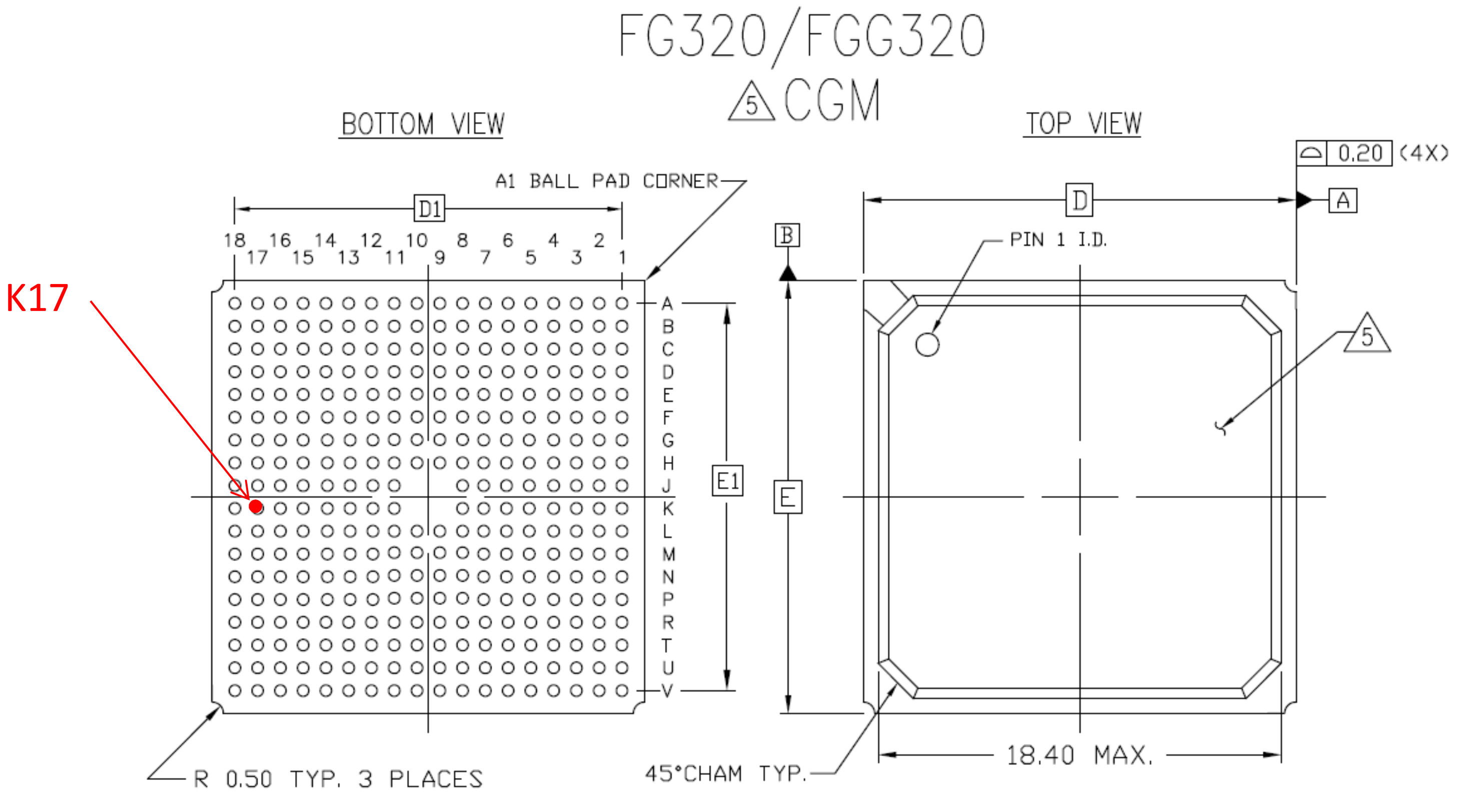
- There are many “FPGA” pins that are used as user I/O pins: each pin connects to an I/O device such as the push-buttons or the LEDs:

Bottom-view of an FPGA IC
(shows IC pins)

Top-view of an FPGA IC
(shows IC markings)

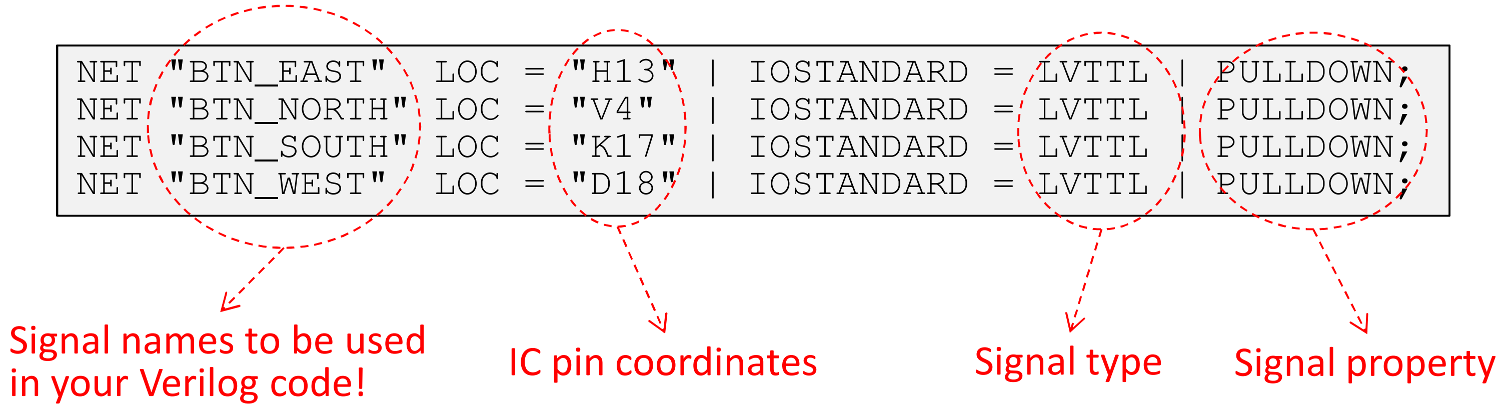
FPGA Pin Coordinates

- Each pin at the bottom of the FPGA has a coordinate. For example, “K17” is the coordinate of the red pin of our Spartan-3E IC:



Use the I/O Pin Signal in Verilog

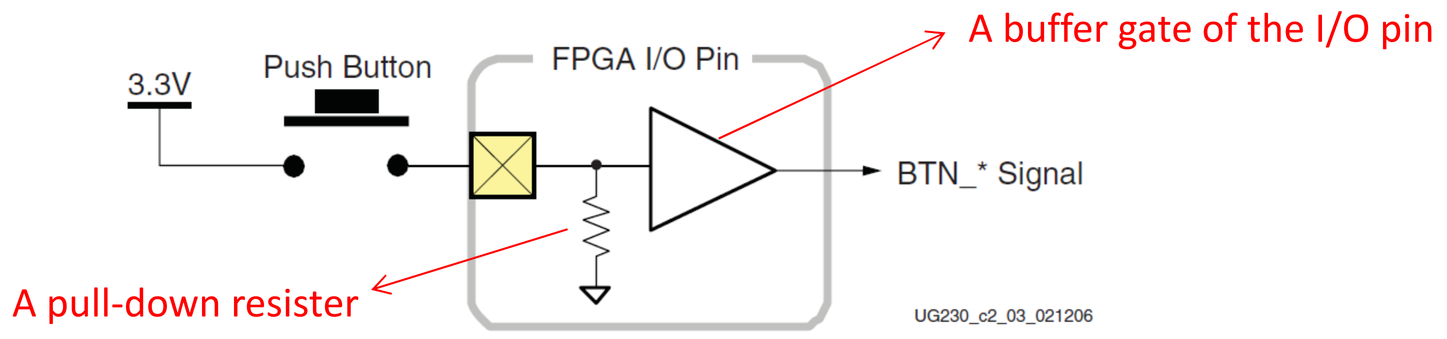
- ❑ To read/write the I/O pins, we must map the pin coordinates to Verilog signals in our code
  - ■ User constraints are used to do the job
- ❑ A user constraint is a text command that specifies the physical property in an HDL code. For example, for the four push-buttons, their mapping to Verilog signals can be as follows:



```
NET "BTN_EAST"  LOC = "H13" | IOSTANDARD = LVTTL | PULLDOWN;
NET "BTN_NORTH" LOC = "V4"  | IOSTANDARD = LVTTL | PULLDOWN;
NET "BTN_SOUTH" LOC = "K17" | IOSTANDARD = LVTTL | PULLDOWN;
NET "BTN_WEST"  LOC = "D18" | IOSTANDARD = LVTTL | PULLDOWN;
```

How to Read the Input Push-Button

- The physical connection from an FPGA I/O pin to a push-button is as follows:

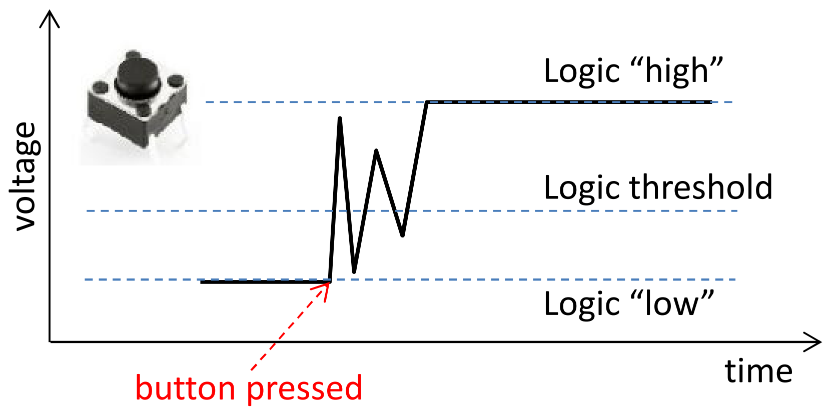


- Ideally, when a push-button is pushed (the circuit is closed), the FPGA pin that connects to the button becomes high voltage and the corresponding signal in Verilog reads “1”, otherwise it reads “0”.

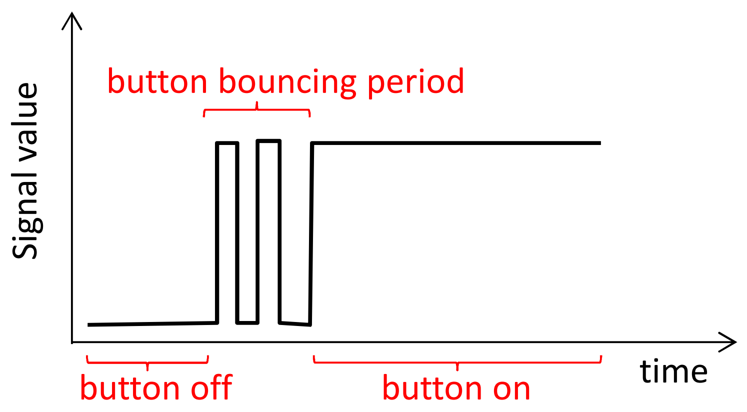
The Bouncing Problem

- In reality, however, the signal value oscillates between 0 and 1 several times before it stabilizes. This is called the bouncing behavior of a hardware button.

The physical voltage values:



The actual digital signal:



The De-bouncing Circuit

- To detect whether the button has been pressed, you cannot simply check the button signals:

```
always @(posedge clk, posedge reset)
  if (reset == 1)
    button_north_is_pressed = 0;
  else
    button_north_is_pressed = (BTN_NORTH)? 1 : 0;
```

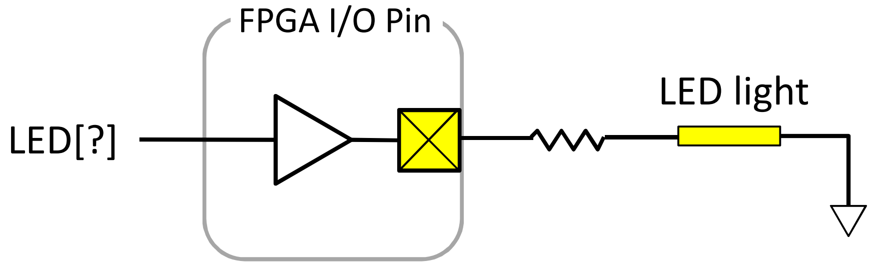
✗

  - This circuit will catch all the state changes during the bouncing period $\rightarrow$ a single button click will be treated as multiple clicks!
- You must find a way to average-out the noises of the push-button signal during the bouncing period
  - Hint: you can use a shift register to accumulate the input signal; or a timer to wait out the bouncing period

Turn On/Off the LEDs

- The LEDs can be turned on or off by writing 1 or 0 to the corresponding Verilog signals (`reg [7:0] LED`):

```
NET "LED[7]" LOC = "F9"  | IOSTANDARD = LVTTL | DRIVE = 8;
NET "LED[6]" LOC = "E9"  | IOSTANDARD = LVTTL | DRIVE = 8;
NET "LED[5]" LOC = "D11" | IOSTANDARD = LVTTL | DRIVE = 8;
NET "LED[4]" LOC = "C11" | IOSTANDARD = LVTTL | DRIVE = 8;
NET "LED[3]" LOC = "F11" | IOSTANDARD = LVTTL | DRIVE = 8;
NET "LED[2]" LOC = "E11" | IOSTANDARD = LVTTL | DRIVE = 8;
NET "LED[1]" LOC = "E12" | IOSTANDARD = LVTTL | DRIVE = 8;
NET "LED[0]" LOC = "F12" | IOSTANDARD = LVTTL | DRIVE = 8;
```



Clock and Reset Pins

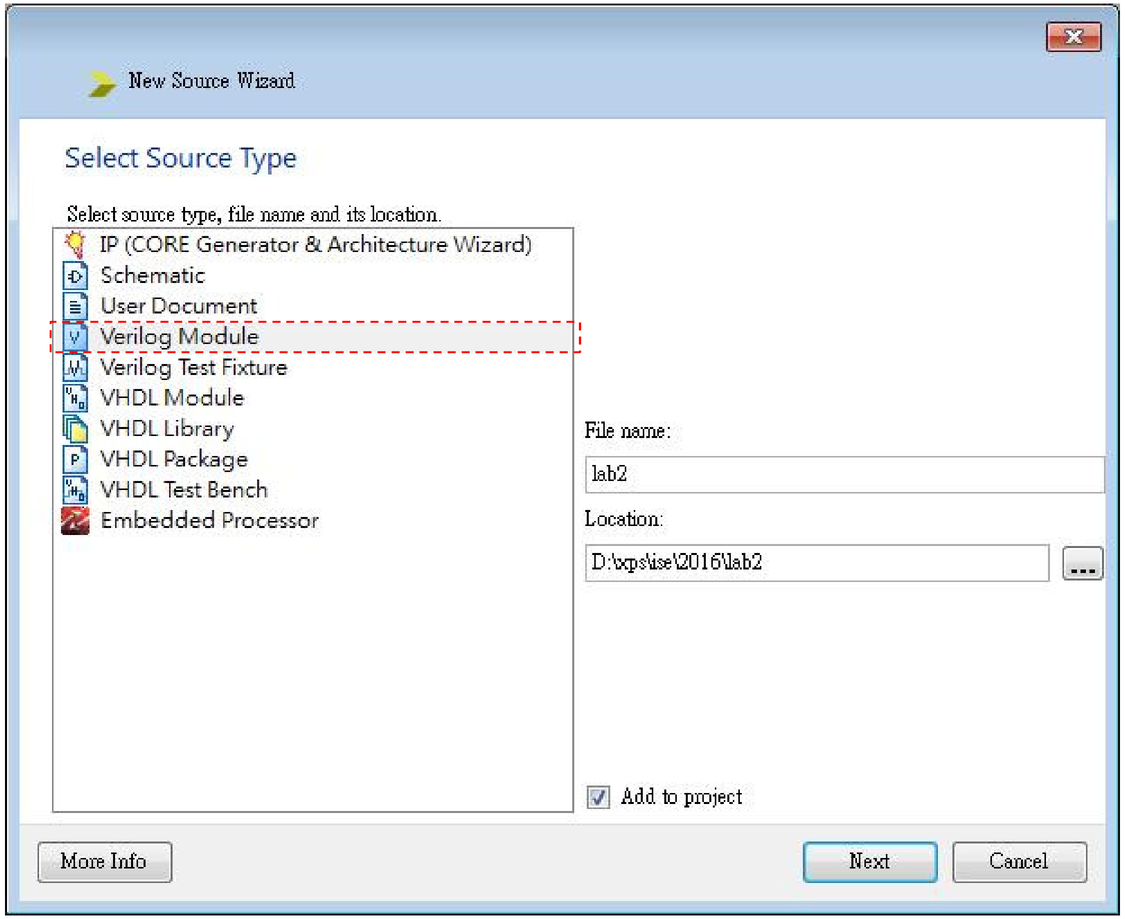
- For synchronous design, you need a clock signal for your circuit
  - The clock signal usually comes from an on-board oscillator
  - There is an FPGA pin that connects to the oscillator

```
NET "CLK" LOC = "C9" | IOSTANDARD = LVCMOS33;

# Define clock period for 50 MHz oscillator
NET "CLK" PERIOD = 20.0ns HIGH 40%;
```

- For the Spartan-3E Starter board, the south push-button (K17) is used as the reset pin
  - The reset pin is connected to an active de-bouncing IC so you do not have to de-bounce its input!

Create ISE Projects for the Board

- As before, we create a new ISE project, then click the "New Source Wizard" icon:



New Source Wizard

Select Source Type

Select source type, file name and its location.

- IP (CORE Generator & Architecture Wizard)
- Schematic
- User Document
- Verilog Module
- Verilog Test Fixture
- VHDL Module
- VHDL Library
- VHDL Package
- VHDL Test Bench
- Embedded Processor

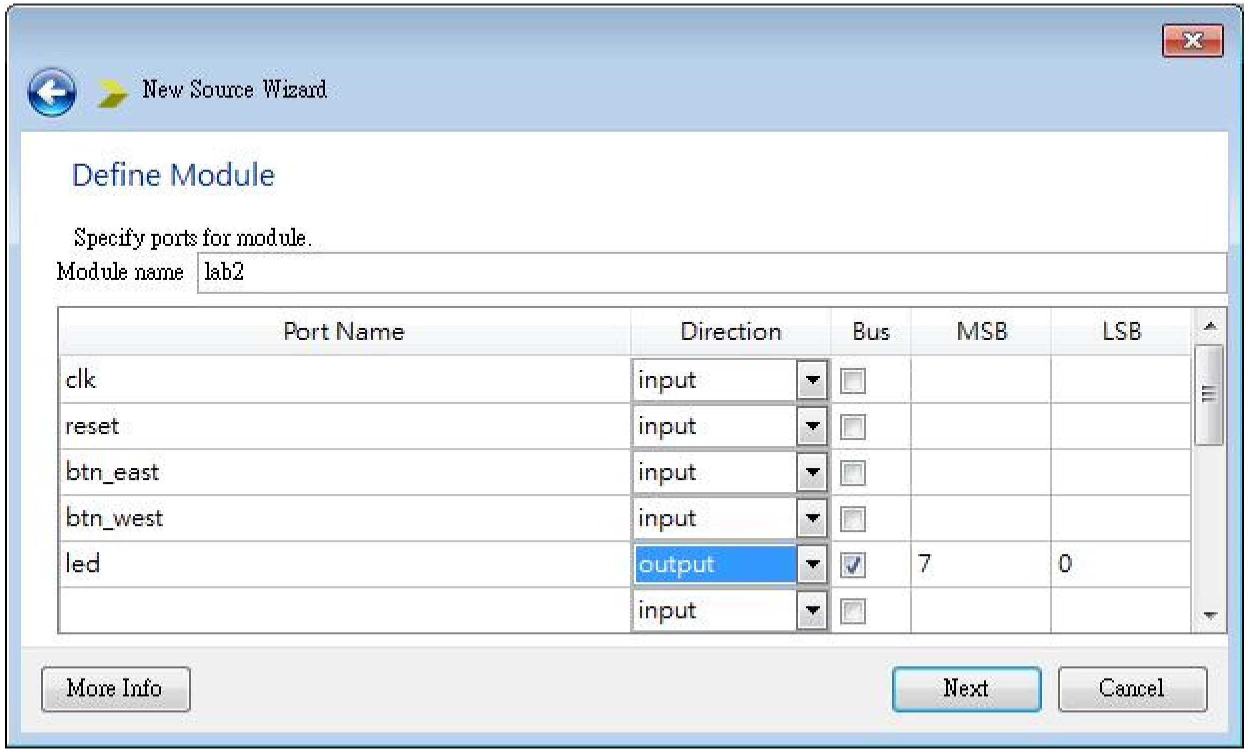
File name: lab2

Location: D:\xps\ise\2016\lab2

☑ Add to project

Specify I/O Ports

- But this time, we specify the I/O ports of the module as follows:



Define Module

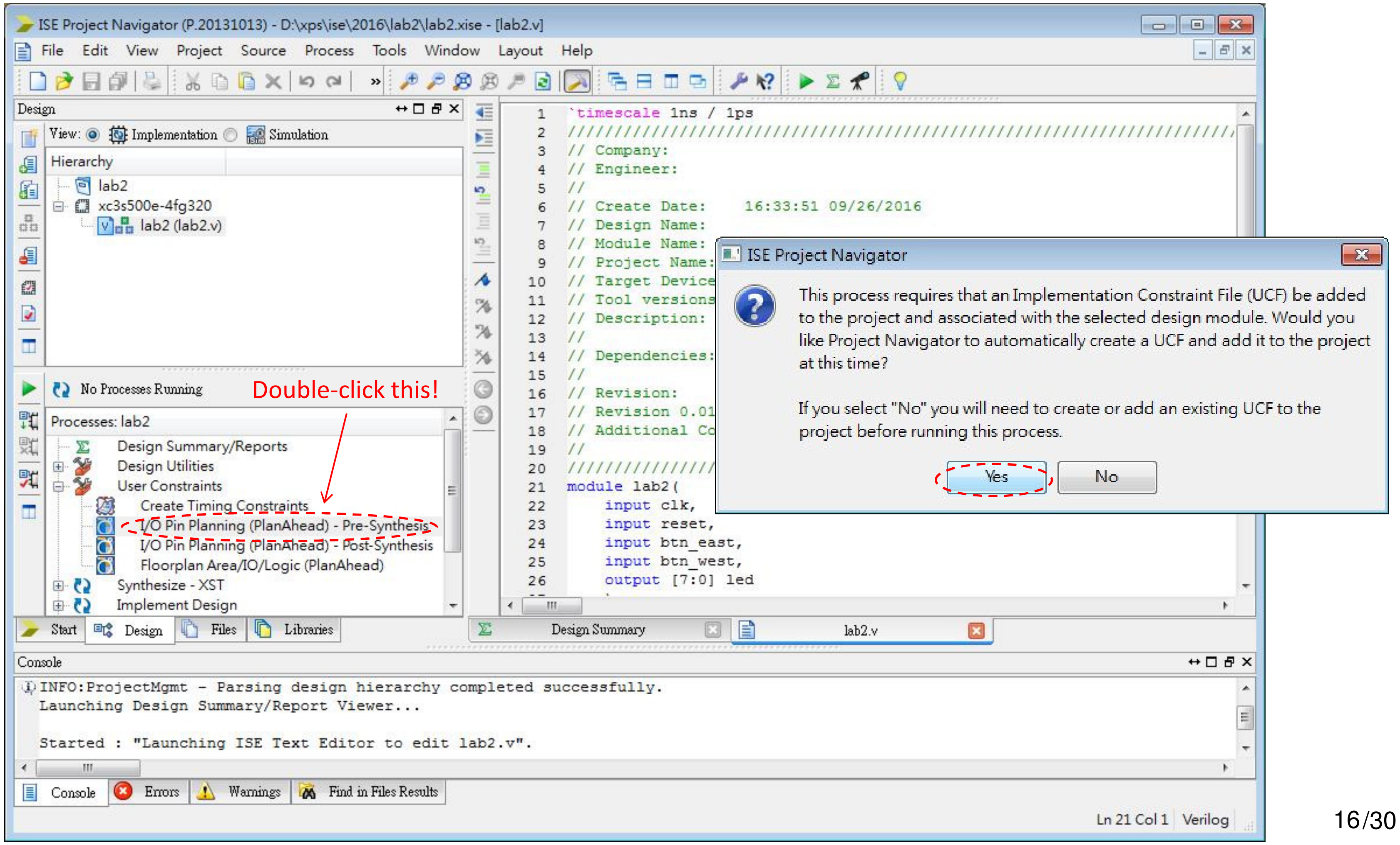
Specify ports for module.

Module name: lab2

Port Name	Direction	Bus	MSB	LSB
clk	input	☐		
reset	input	☐		
btn_east	input	☐		
btn_west	input	☐		
led	output	☑	7	0
	input	☐		

Create User Constraints

- Select “I/O Pin Planning” to create user constraints:


ISE Project Navigator (P.20131013) - D:\xps\ise\2016\lab2\lab2.xise - [lab2.v]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Hierarchy
lab2
xc3s500e-4fg320
lab2 (lab2.v)
No Processes Running
Double-click this!
Processes: lab2
Design Summary/Reports
Design Utilities
User Constraints
Create Timing Constraints
I/O Pin Planning (PlanAhead) - Pre-Synthesis
I/O Pin Planning (PlanAhead) - Post-Synthesis
Floorplan Area/IO/Logic (PlanAhead)
Synthesize - XST
Implement Design
Start
Design
Files
Libraries
Design Summary
lab2.v
Console
Errors
Warnings
Find in Files Results
Ln 21 Col 1
Verilog
ISE Project Navigator
Yes
No

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    16:33:51 09/26/2016
// Design Name:
// Module Name:
// Project Name:
// Target Device
// Tool versions
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01
// Additional Co
//
///////////////
module lab2(
    input clk,
    input reset,
    input btn_east,
    input btn_west,
    output [7:0] led
```

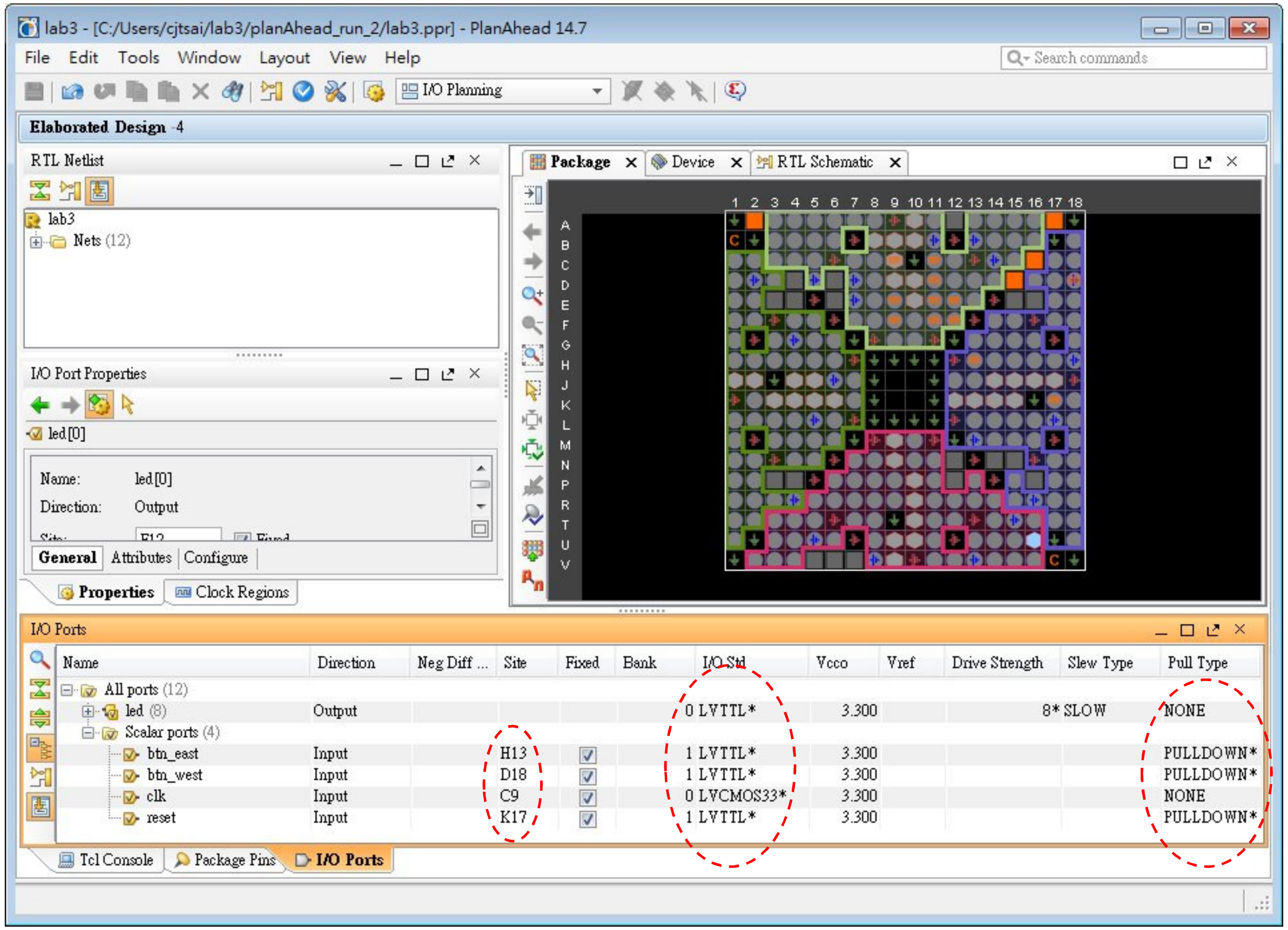
This process requires that an Implementation Constraint File (UCF) be added to the project and associated with the selected design module. Would you like Project Navigator to automatically create a UCF and add it to the project at this time?

If you select "No" you will need to create or add an existing UCF to the project before running this process.

```
INFO:ProjectMgmt - Parsing design hierarchy completed successfully.
Launching Design Summary/Report Viewer...

Started : "Launching ISE Text Editor to edit lab2.v".
```

The User Constraint Editor

- ❑ Assign the Verilog I/O ports to pin coordinates:

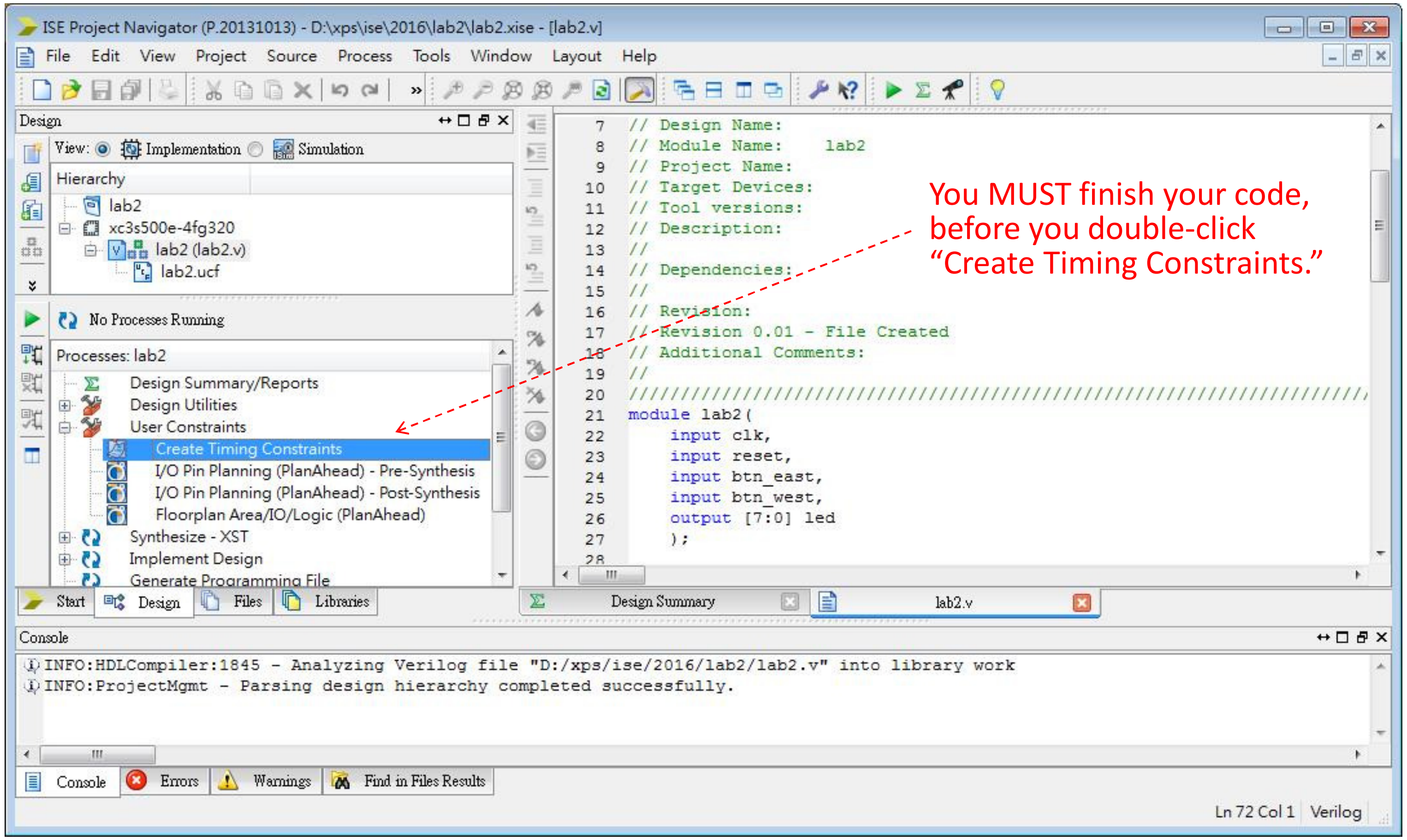

lab3 - [C:/Users/cjtsai/lab3/planAhead_run_2/lab3.ppr] - PlanAhead 14.7
File Edit Tools Window Layout View Help
Search commands
I/O Planning
Elaborated Design - 4
RTL Netlist
lab3
Nets (12)
I/O Port Properties
led[0]
General Attributes Configure
Properties Clock Regions
Package Device RTL Schematic
I/O Ports
Tcl Console Package Pins I/O Ports

Name: led[0]
Direction: Output
Site: F12 Fixed

Name	Direction	Neg Diff ...	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type
All ports (12)											
led (8)	Output				0	LVTTL*	3.300		8*	SLOW	NONE
Scalar ports (4)											
btn_east	Input		H13	✓	1	LVTTL*	3.300				PULLDOWN*
btn_west	Input		D18	✓	1	LVTTL*	3.300				PULLDOWN*
clk	Input		C9	✓	0	LVCMOS33*	3.300				NONE
reset	Input		K17	✓	1	LVTTL*	3.300				PULLDOWN*

Adding Clock Signal Constraints (1/3)

- Now, the signal “`clk`” has a default timing constraint
  - we can modify it once the design is ready for synthesis:


ISE Project Navigator (P.20131013) - D:\xps\ise\2016\lab2\lab2.xise - [lab2.v]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Hierarchy
lab2
xc3s500e-4fg320
lab2 (lab2.v)
lab2.ucf
No Processes Running
Processes: lab2
Design Summary/Reports
Design Utilities
User Constraints
Create Timing Constraints
I/O Pin Planning (PlanAhead) - Pre-Synthesis
I/O Pin Planning (PlanAhead) - Post-Synthesis
Floorplan Area/IO/Logic (PlanAhead)
Synthesize - XST
Implement Design
Generate Programming File
Start Design Files Libraries
Design Summary
lab2.v
Console
Console Errors Warnings Find in Files Results
Ln 72 Col 1 Verilog

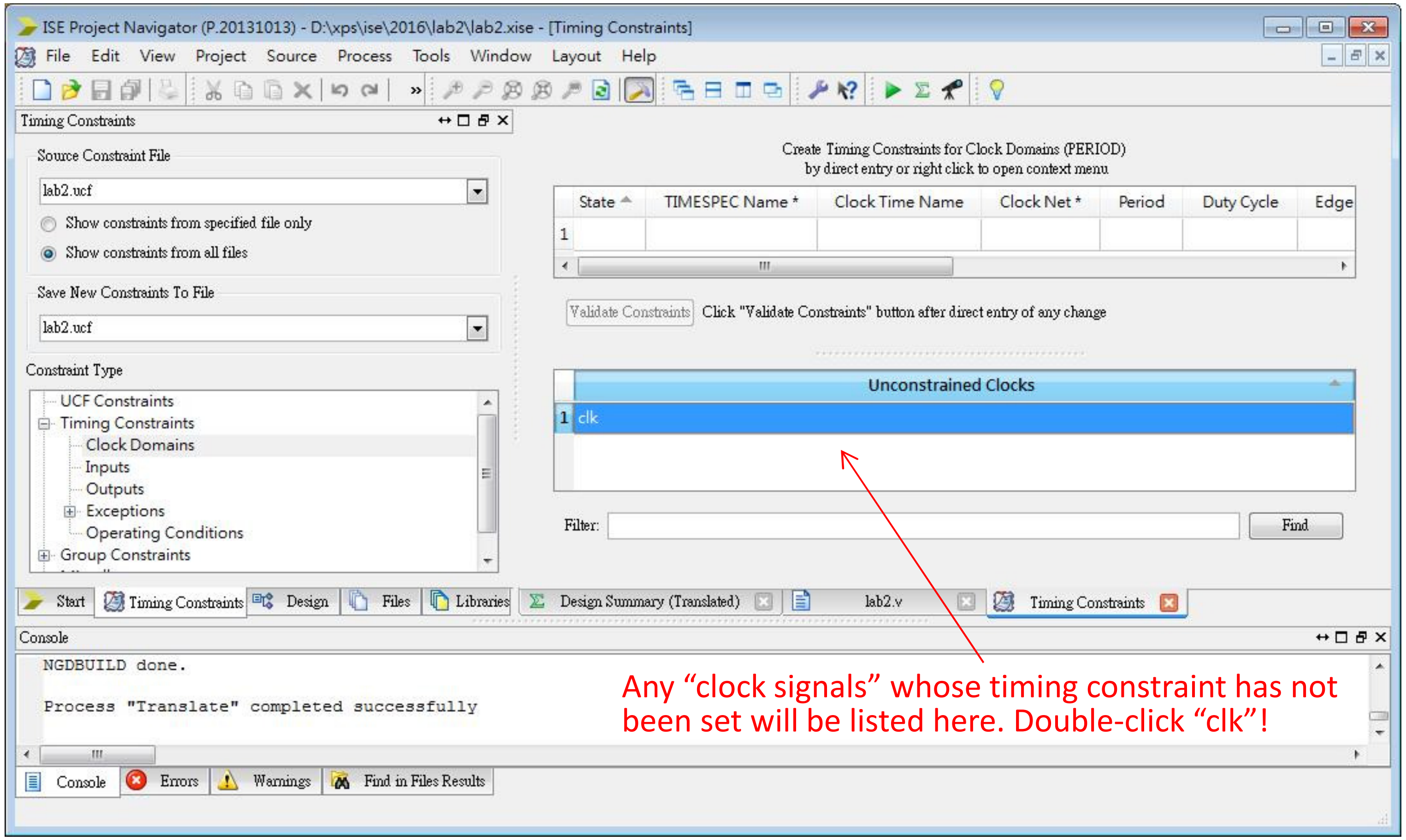
```
// Design Name:
// Module Name:    lab2
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module lab2(
    input clk,
    input reset,
    input btn_east,
    input btn_west,
    output [7:0] led
    );
```

```
INFO:HDLCompiler:1845 - Analyzing Verilog file "D:/xps/ise/2016/lab2/lab2.v" into library work
INFO:ProjectMgmt - Parsing design hierarchy completed successfully.
```

You MUST finish your code, before you double-click “Create Timing Constraints.”

Adding Clock Signal Constraints (2/3)

- The default constraint should work fine, but we can change the duty cycle to better match the board

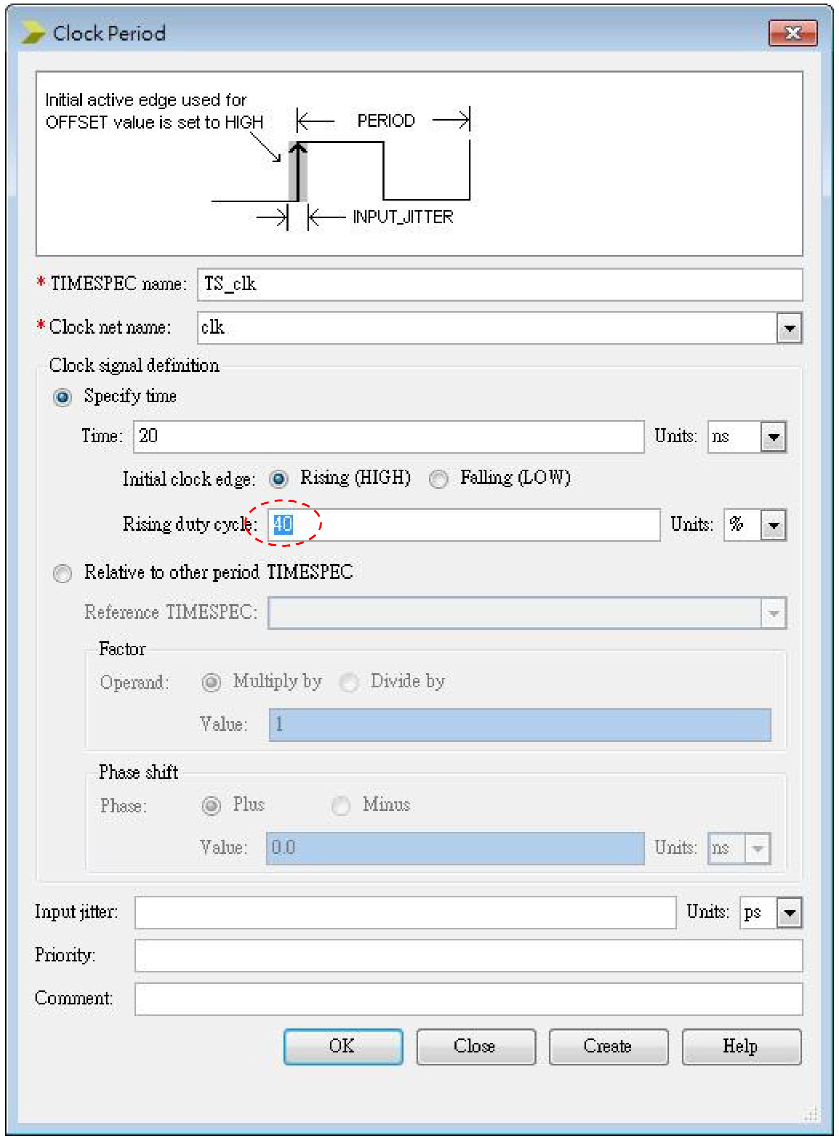

ISE Project Navigator (P.20131013) - D:\xps\ise\2016\lab2\lab2.xise - [Timing Constraints]
File Edit View Project Source Process Tools Window Layout Help
Timing Constraints
Source Constraint File
lab2.ucf
Show constraints from specified file only
Show constraints from all files
Save New Constraints To File
lab2.ucf
Constraint Type
UCF Constraints
Timing Constraints
Clock Domains
Inputs
Outputs
Exceptions
Operating Conditions
Group Constraints
Create Timing Constraints for Clock Domains (PERIOD)
by direct entry or right click to open context menu
State
TIMESPEC Name *
Clock Time Name
Clock Net *
Period
Duty Cycle
Edge
Validate Constraints
Click "Validate Constraints" button after direct entry of any change
Unconstrained Clocks
1 clk
Filter:
Find
Start
Timing Constraints
Design
Files
Libraries
Design Summary (Translated)
lab2.v
Timing Constraints
Console
Errors
Warnings
Find in Files Results

```
NGDBUILD done.

Process "Translate" completed successfully
```

Any "clock signals" whose timing constraint has not been set will be listed here. Double-click "clk"!

Adding Clock Signal Constraints (3/3)

- Change the rising duty cycle of the clock signal from the default 50% to 40% duty cycle.
- After saving the clock constraint, close the constraint editor and go back to the main screen of ISE project navigator.

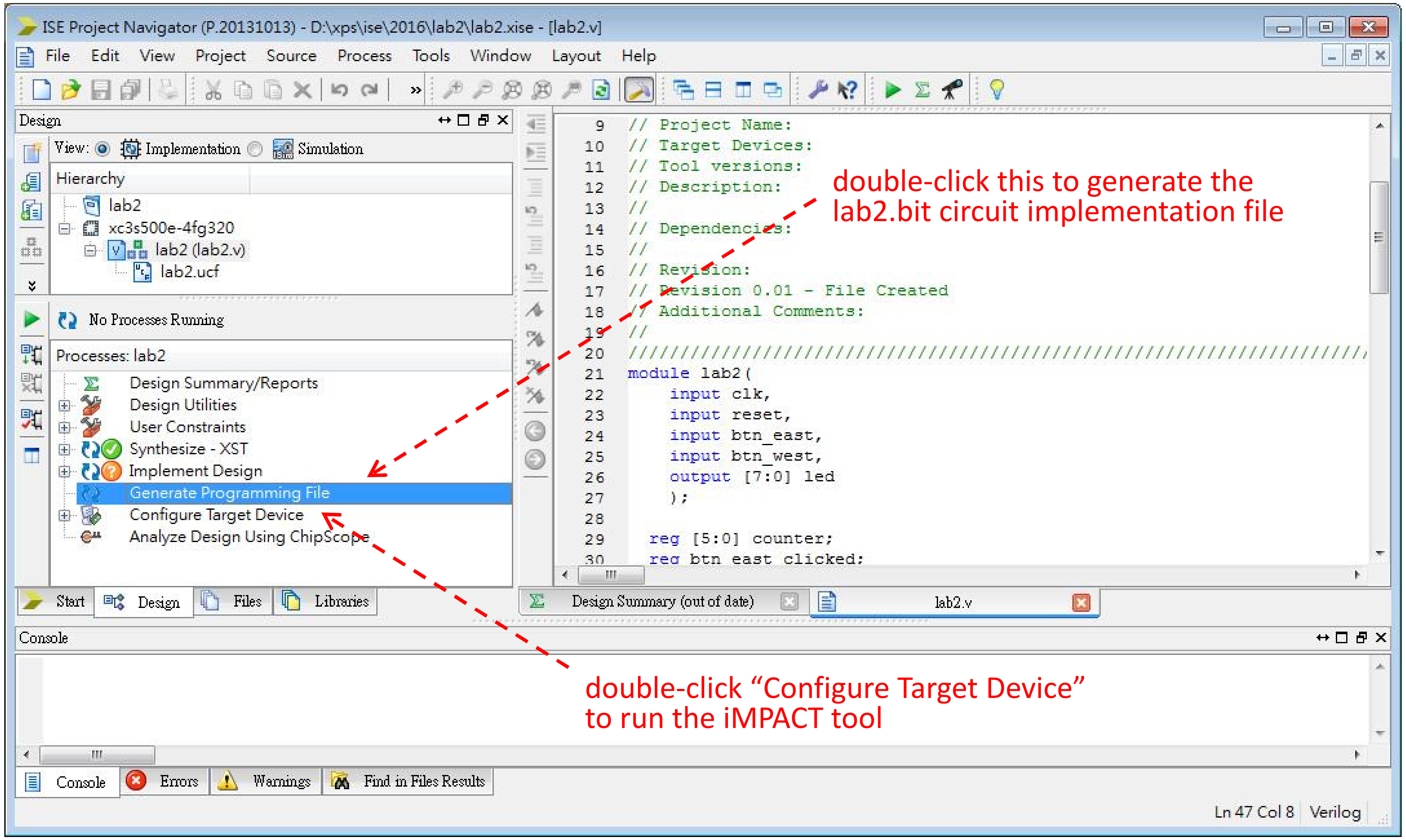


* TIMESPEC name: TS_clk
* Clock net name: clk

Clock signal definition
[✓ Specify time]
Time: 20 Units: ns
Initial clock edge: [✓ Rising (HIGH)] Falling (LOW)
Rising duty cycle: 40 Units: %
Relative to other period TIMESPEC
Reference TIMESPEC:
Factor
Operand: [✓ Multiply by] Divide by
Value: 1
Phase shift
Phase: [✓ Plus] Minus
Value: 0.0 Units: ns
Input jitter: Units: ps
Priority:
Comment:

Generate the Programming File

❑ Now, create the programming file "lab2.bit" for FPGA:



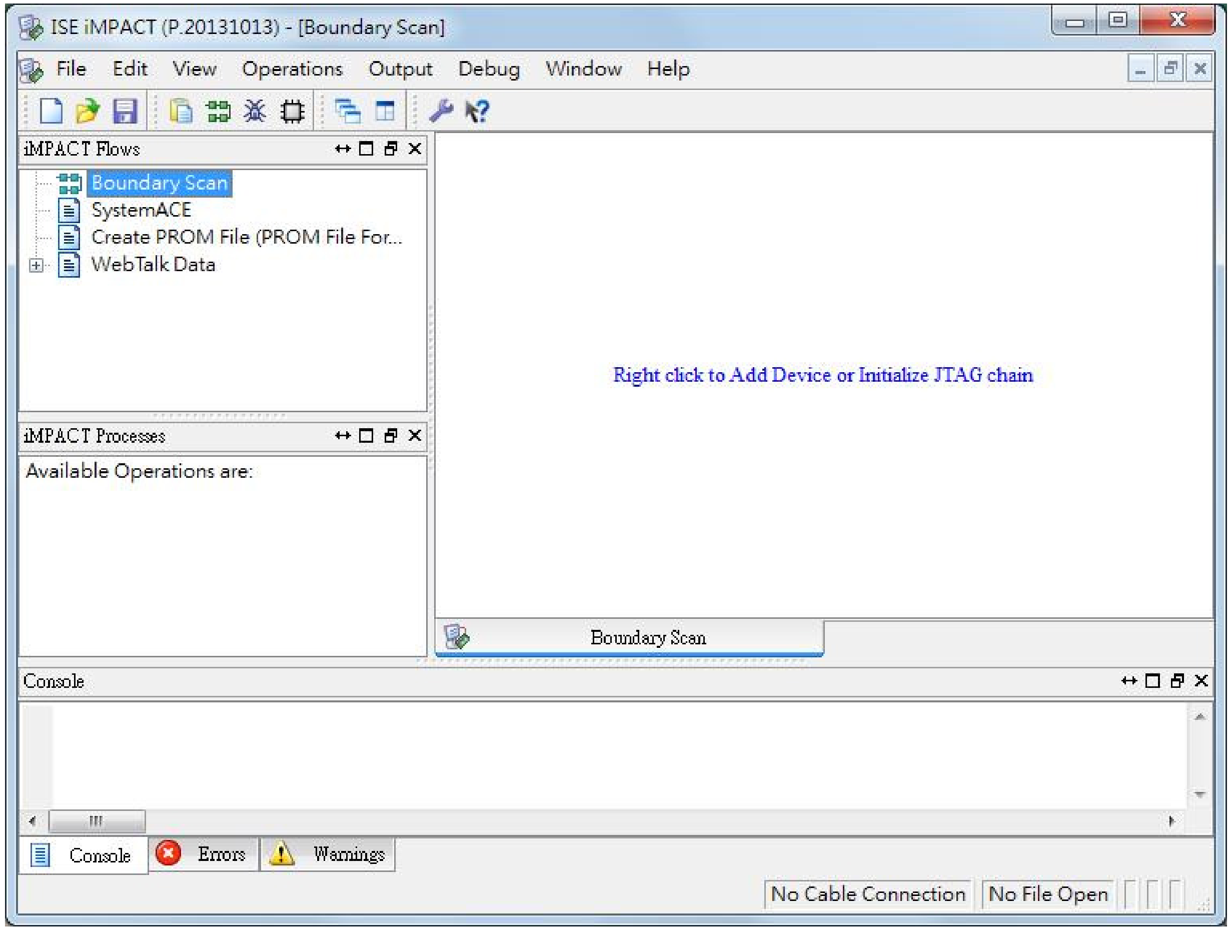
```
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////
module lab2(
    input clk,
    input reset,
    input btn_east,
    input btn_west,
    output [7:0] led
    );

  reg [5:0] counter;
  reg btn_east_clicked;
```

double-click this to generate the lab2.bit circuit implementation file

double-click "Configure Target Device" to run the iMPACT tool

Downloading Your Circuit to the Board

- To download your programming file into the FPGA, you must use the "iMPACT" tool:



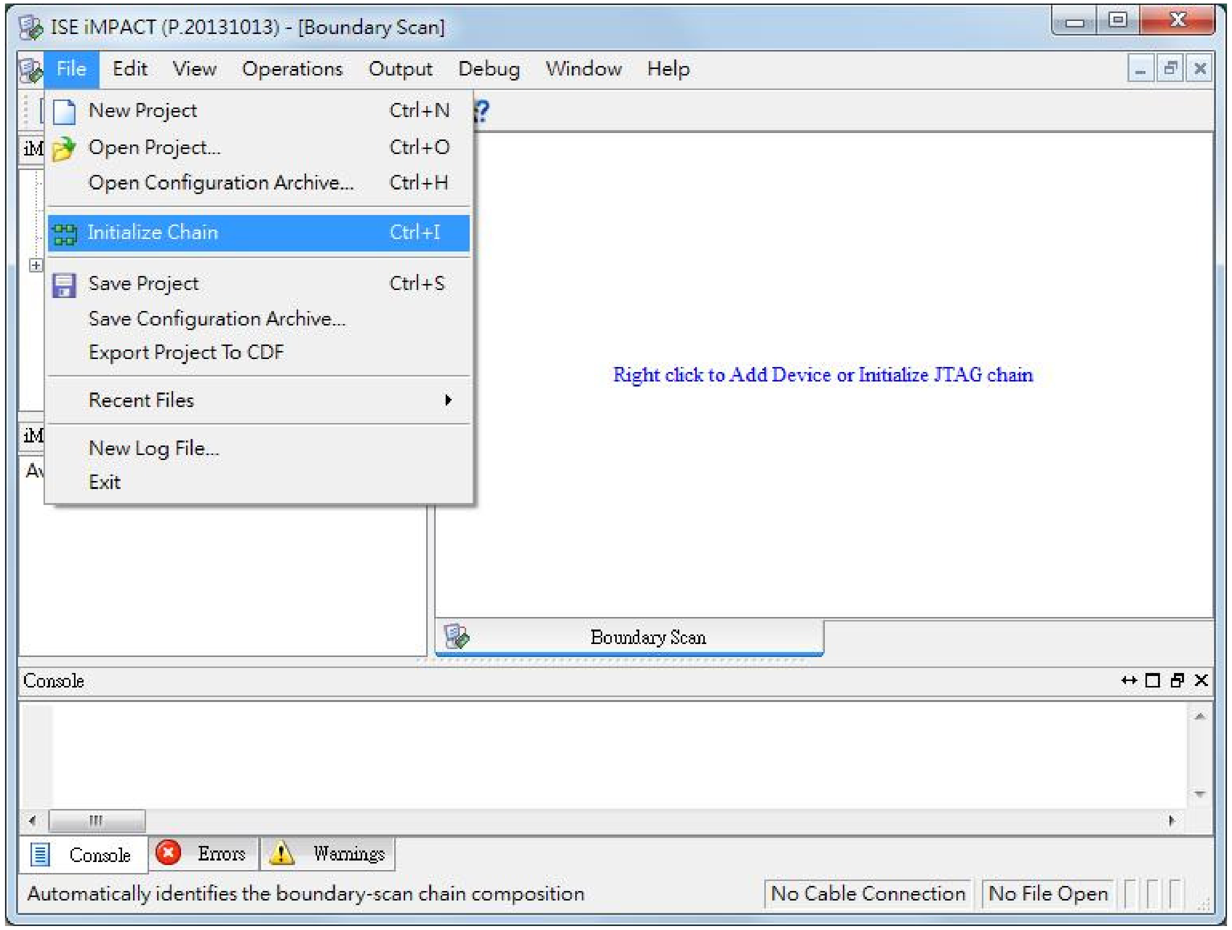
Available Operations are:

Right click to Add Device or Initialize JTAG chain

No Cable Connection | No File Open

Detect the FPGA from iMPACT

- Power on the board and initialize the JTAG chain:

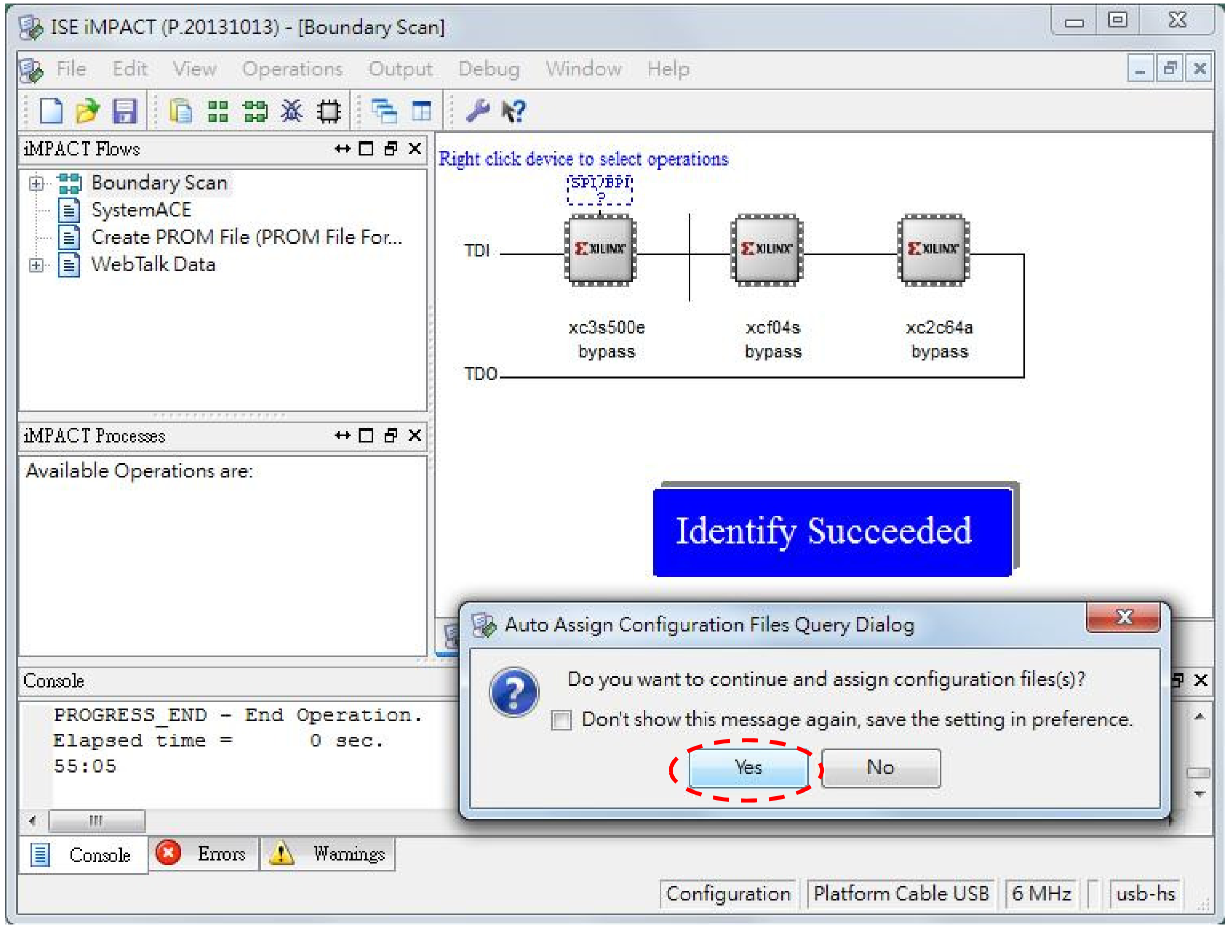


Right click to Add Device or Initialize JTAG chain

Automatically identifies the boundary-scan chain composition

Assign the *.bit File to the FPGA

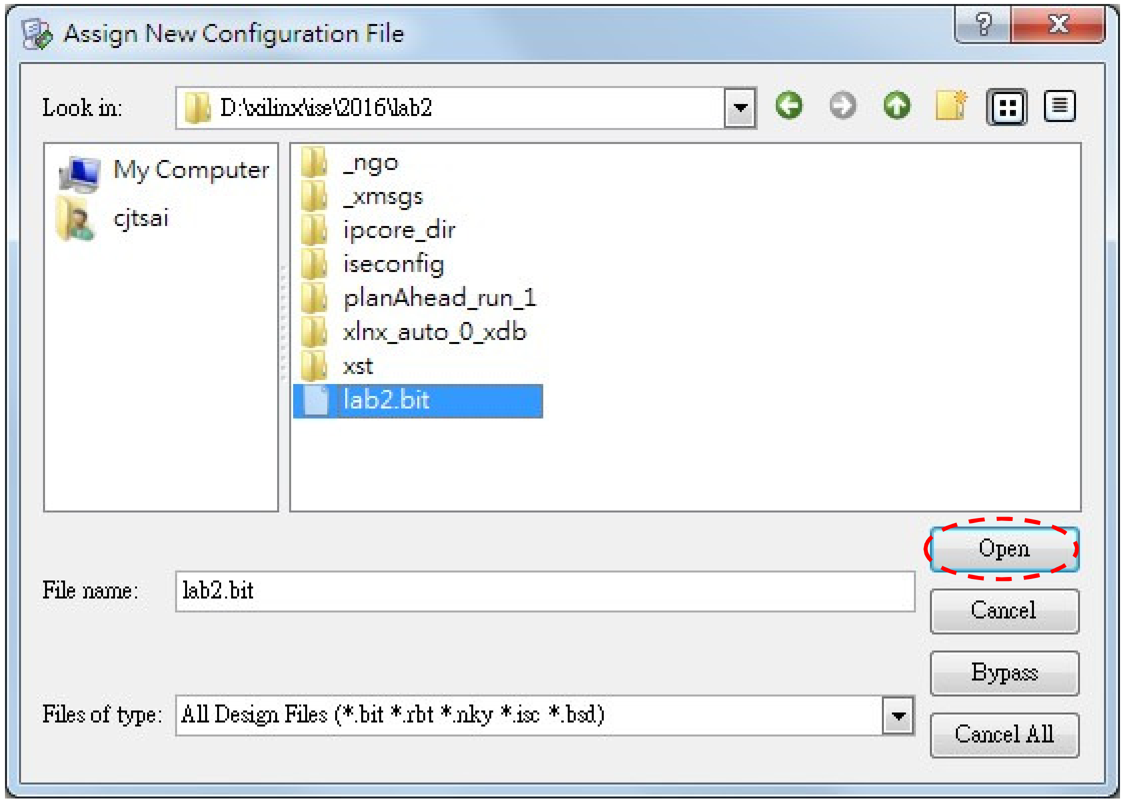
- ❑ JTAG shows three IC's, we only use the first one:


ISE iMPACT (P.20131013) - [Boundary Scan]
File Edit View Operations Output Debug Window Help
iMPACT Flows
Boundary Scan
SystemACE
Create PROM File (PROM File For...
WebTalk Data
iMPACT Processes
Available Operations are:
Right click device to select operations
SPI/BPI
TDI
TDO
xc3s500e bypass
xcf04s bypass
xc2c64a bypass
Identify Succeeded
Auto Assign Configuration Files Query Dialog
Do you want to continue and assign configuration files(s)?
Don't show this message again, save the setting in preference.
Yes
No
Console
Errors
Warnings
Configuration
Platform Cable USB
6 MHz
usb-hs

```
PROGRESS_END - End Operation.
Elapsed time =      0 sec.
55:05
```

Browse to the *.bit File

- Browse to your directory and select the bit file for Lab2:



Assign New Configuration File

Look in: D:\xilinx\ise\2016\lab2

My Computer
cjtsai

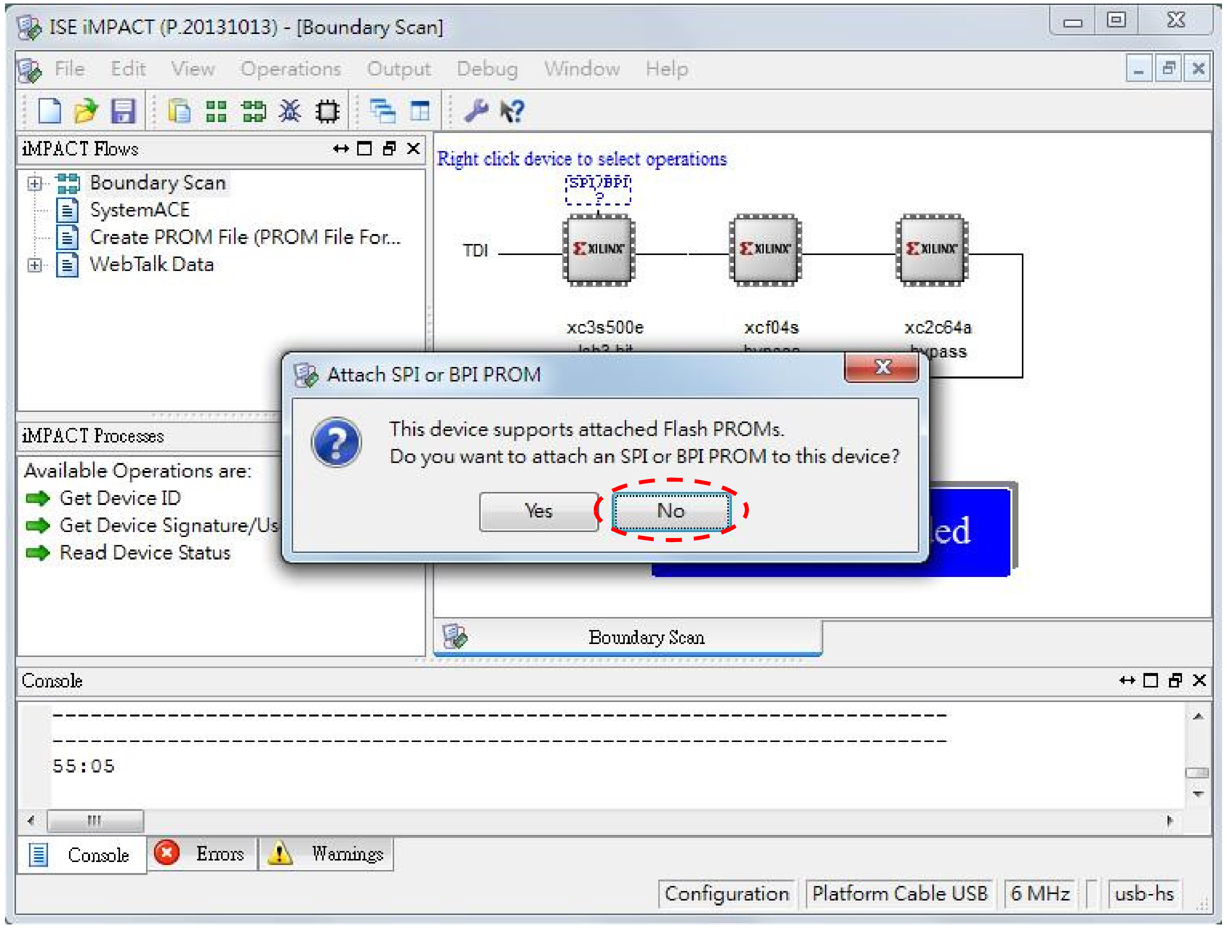
_ngo
_xmsgs
ipcore_dir
iseconfig
planAhead_run_1
xlnx_auto_0_xdb
xst
lab2.bit

File name: lab2.bit

Files of type: All Design Files (*.bit *.rbt *.nky *.isc *.bsd)

Skip Configuration from SPI ROM

- We do not use the SPI flash to configure the FPGA



Attach SPI or BPI PROM

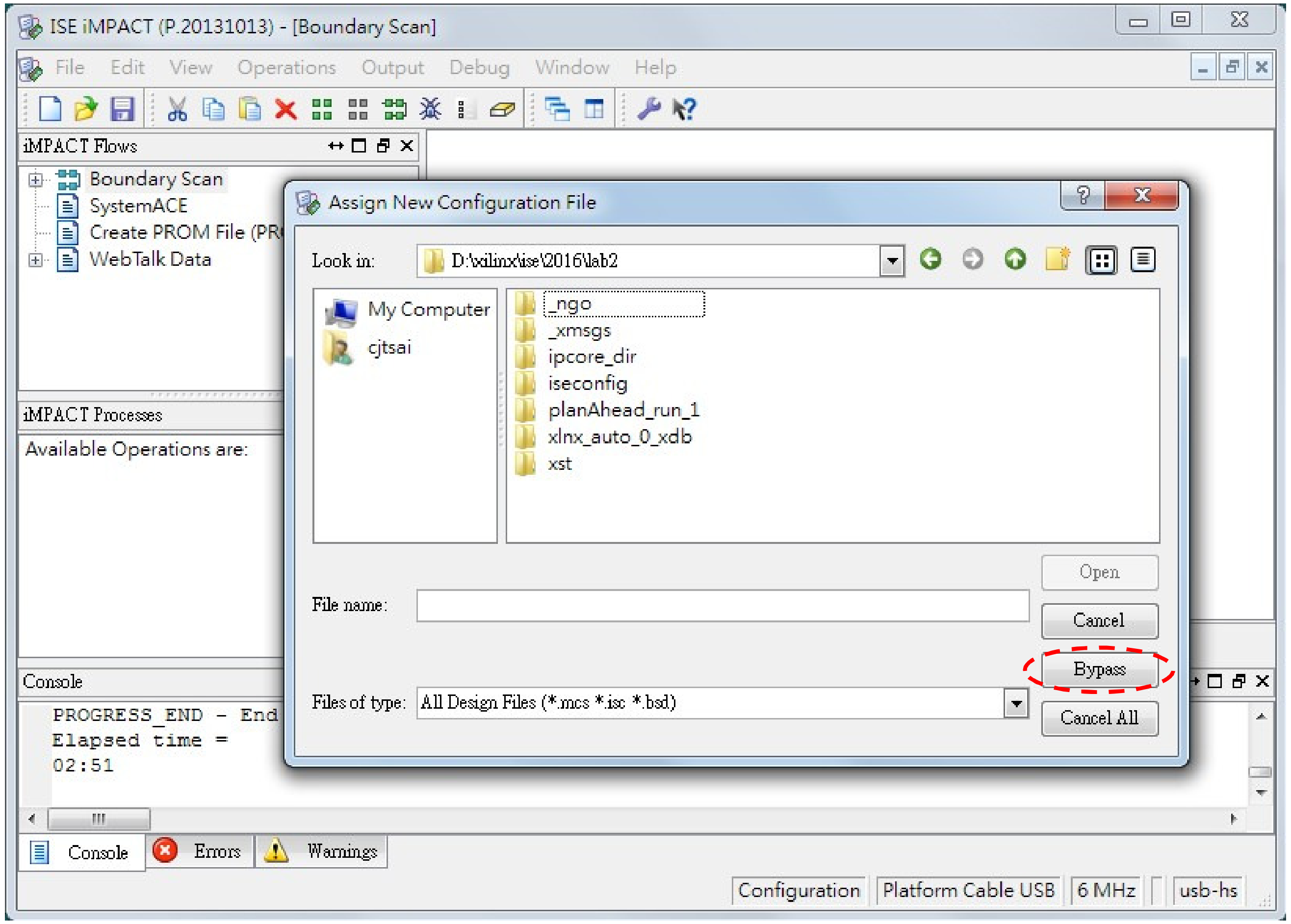
This device supports attached Flash PROMs.
Do you want to attach an SPI or BPI PROM to this device?

Yes [✓ No]

```
55:05
```

Bypass the Next Two IC's

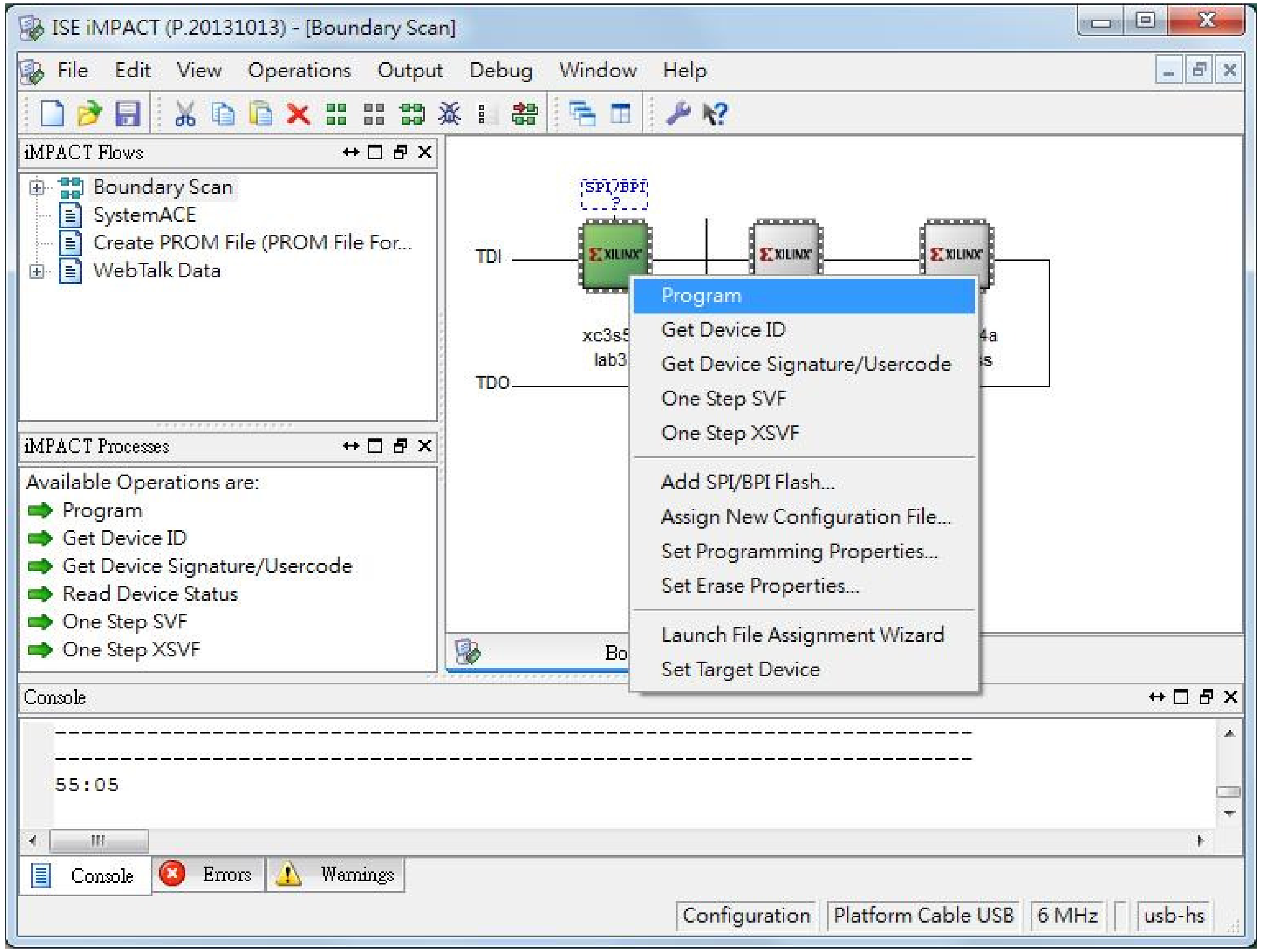
- The next two IC's in the JTAG chain are special flash memory → they are obsolete, bypass next two boxes!



```
PROGRESS_END - End
Elapsed time =
02:51
```

Program the FPGA

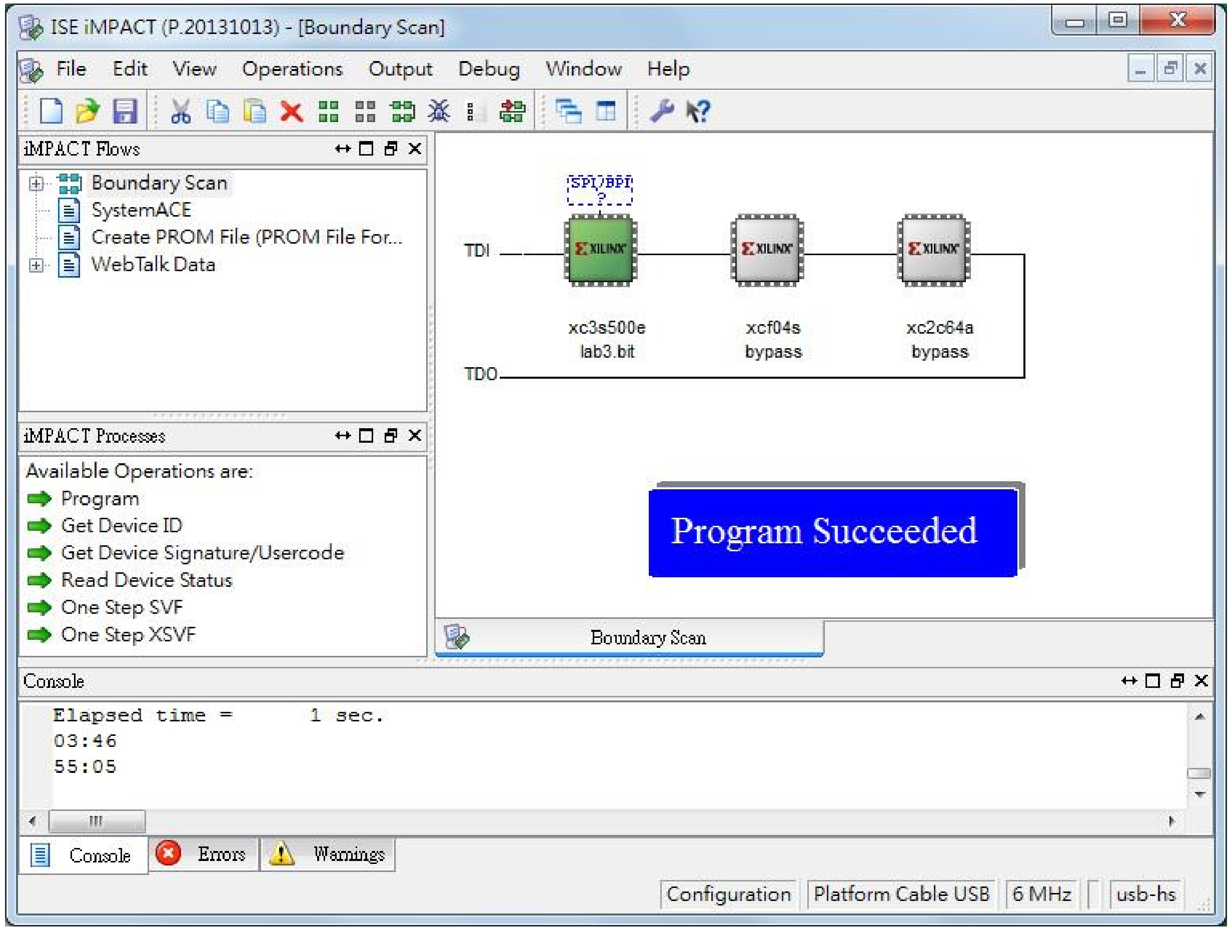
- Finally, we can right-click the green IC to program it!



```
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
55:05
```

Test Your Design

- You can now test your circuit by clicking the buttons on the Spartan3 board and see how the LEDs lights up!



```
Elapsed time =      1 sec.
03:46
55:05
```

Reference

- Xilinx, *Spartan-3E Starter Kit Board User Guide, UG230 (v1.0)*, March 9, 2006. Available from:

  http://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf