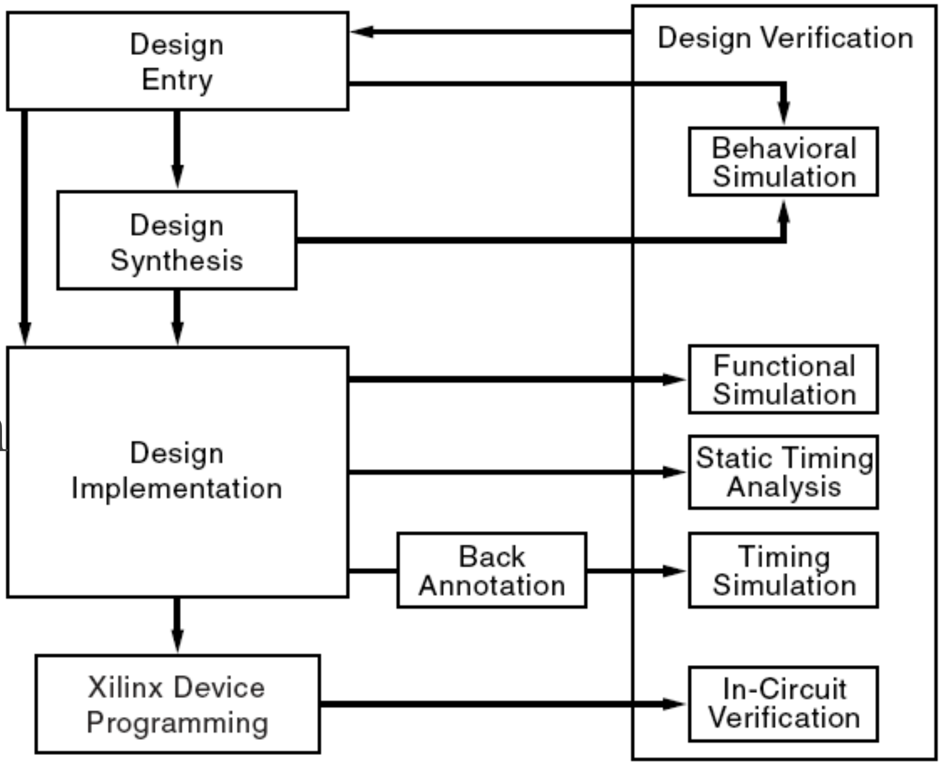
FPGA Emulation

Hsi-Pin Ma

http://lms.nthu.edu.tw/course/24953
Department of Electrical Engineering
National Tsing Hua University

Design Flow

- General design flow
  - Design construction
  - Behavioral simulation
  - Design implementation
  - Timing simulation
- HDL-based design Flow



Important Notes

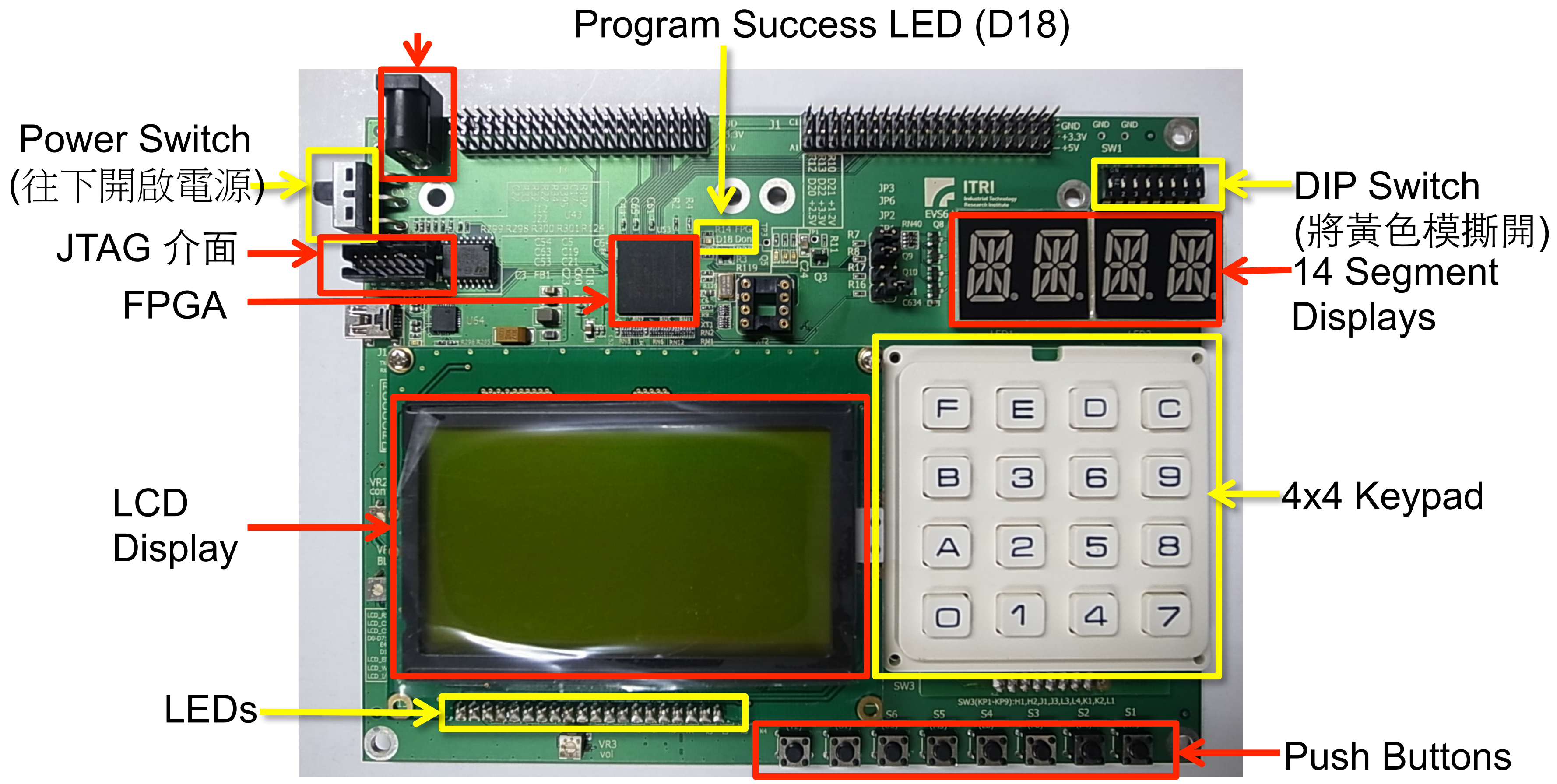
- **Draw schematic first** and then construct Verilog codes.
- Verilog RTL coding philosophy is not the same as C programming
  - Every Verilog RTL construct has its own logic mapping (for synthesis)
  - You should have the logics (draw schematic) first and then the RTL codes
  - You have to write **synthesizable** RTL codes

Notes from Lab1

- Always ‘SAVE’ before next step
- Select the right file for the next step
  - For simulation
  - For implementation
- Do not use wired filename
  - Number, with space, Chinese

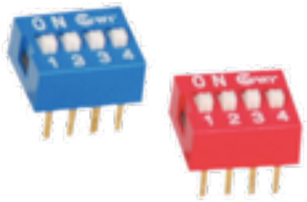
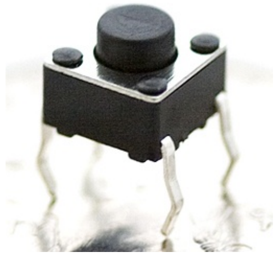
Laboratory for Reliable Computing

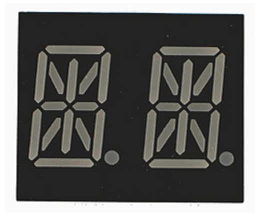
Xilinx XC6SLX16 CS324 Demo Board



I/O Devices

- Clock generation: 40MHz oscillator
- Input devices
  - 8 DIP switches
  - 8 push buttons
  - 1 4x4 keypad
- Output devices
  - 16 user LEDs
  - 4 14-segment display
  - 1 128x64 LCD display
  - Audio interface

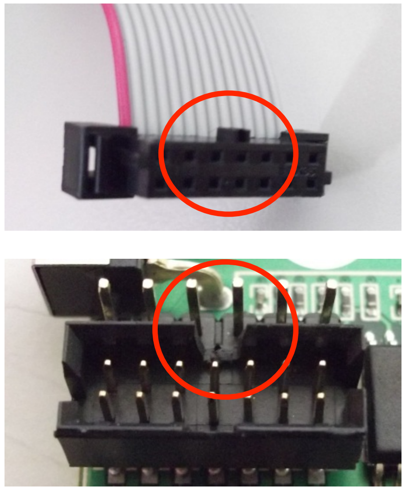


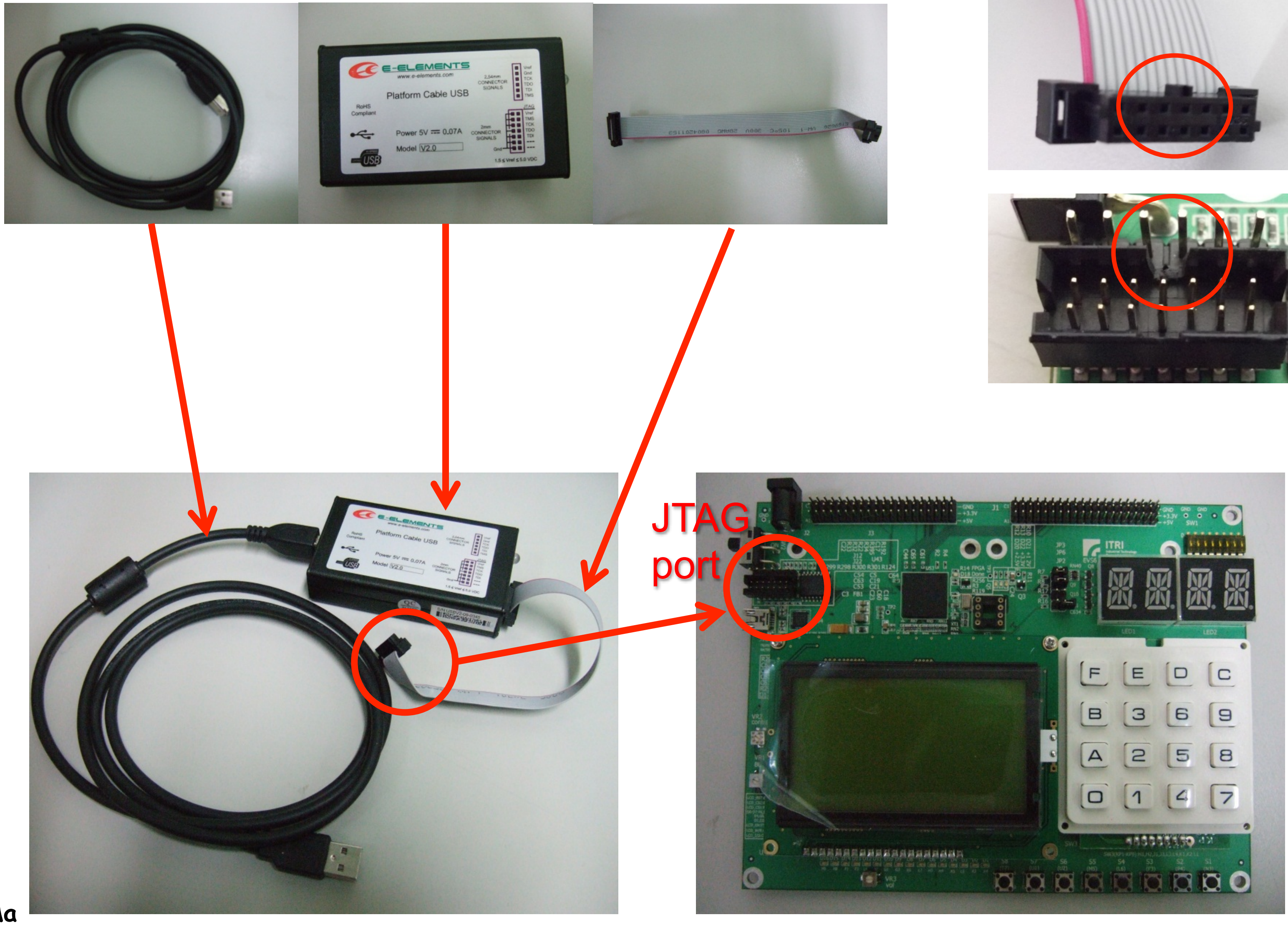
Input/Output Connections

- LEDs are pre-wired such that
  - A HIGH signal from the FPGA to turn it on
- Push buttons and DP switches are pre-wired such that
  - The corresponding input is tied to LOW when a push button is pressed or a DIP switch is turned on.

USB / JTAG Download Cable

注意: 凹槽要對到卡榫

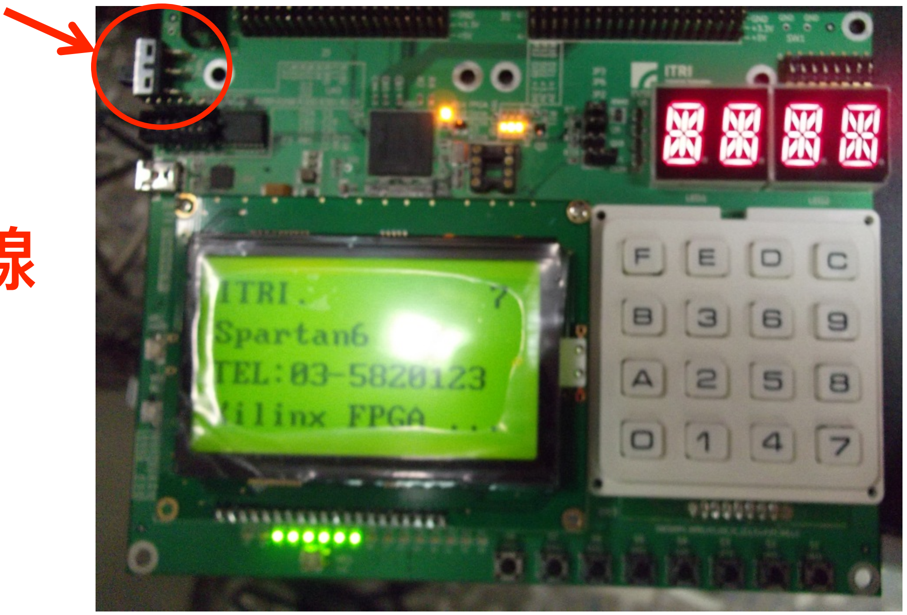




Test Your FPGA Board (1/3)

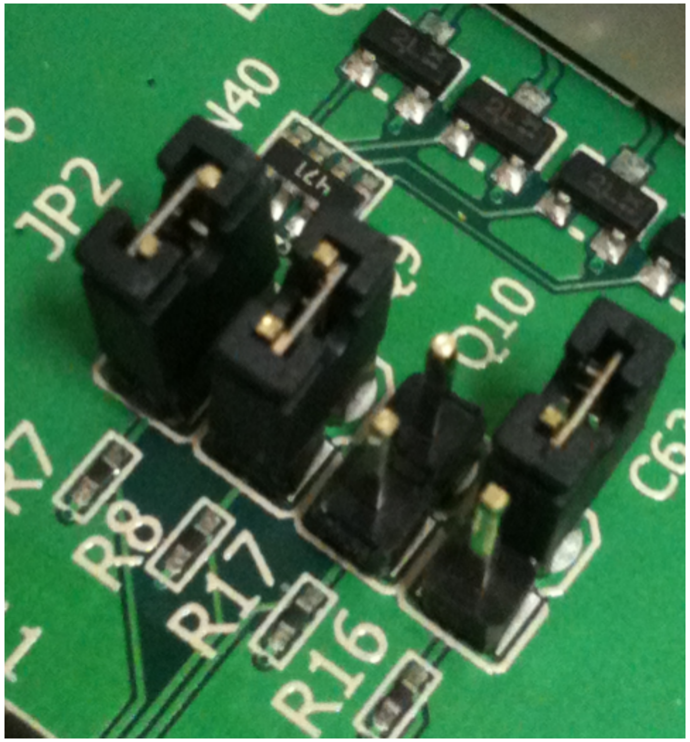
- Connect the demo board to PC and also the power supply

•所有的線接好再開電源
•關閉電源後再拔所有的線
•勿用導體接觸針腳
•插座別插反了
•避免長時間開機

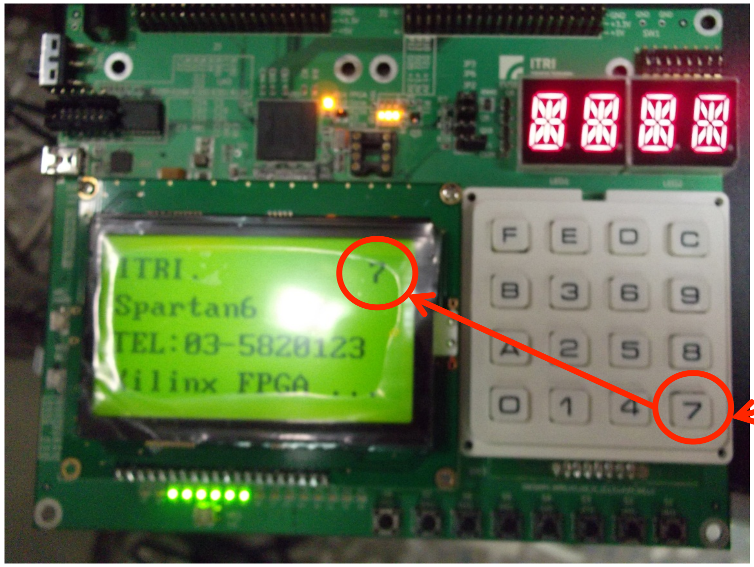


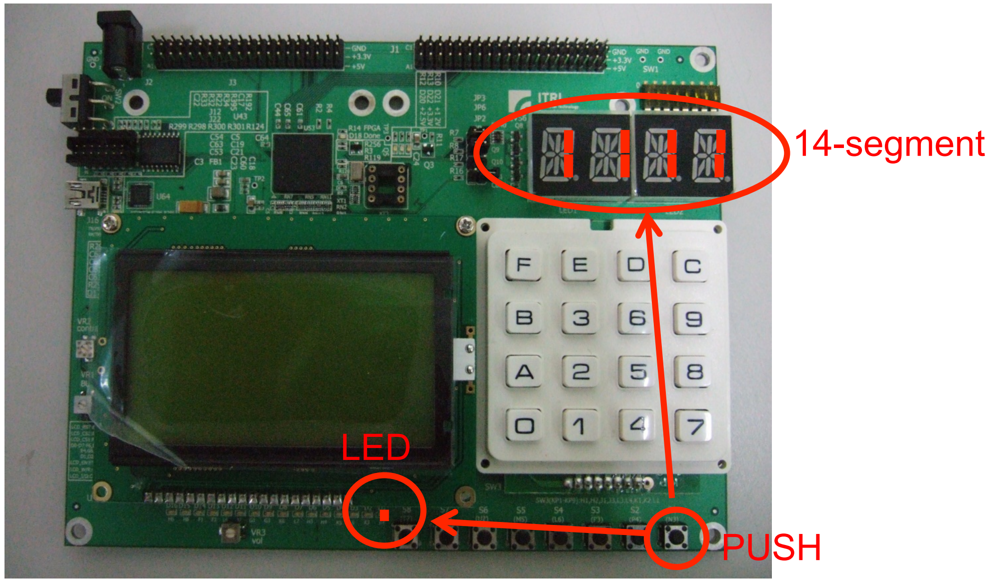
Test Your FPGA Board (2/3)

- Don’t move or remove the jumpers



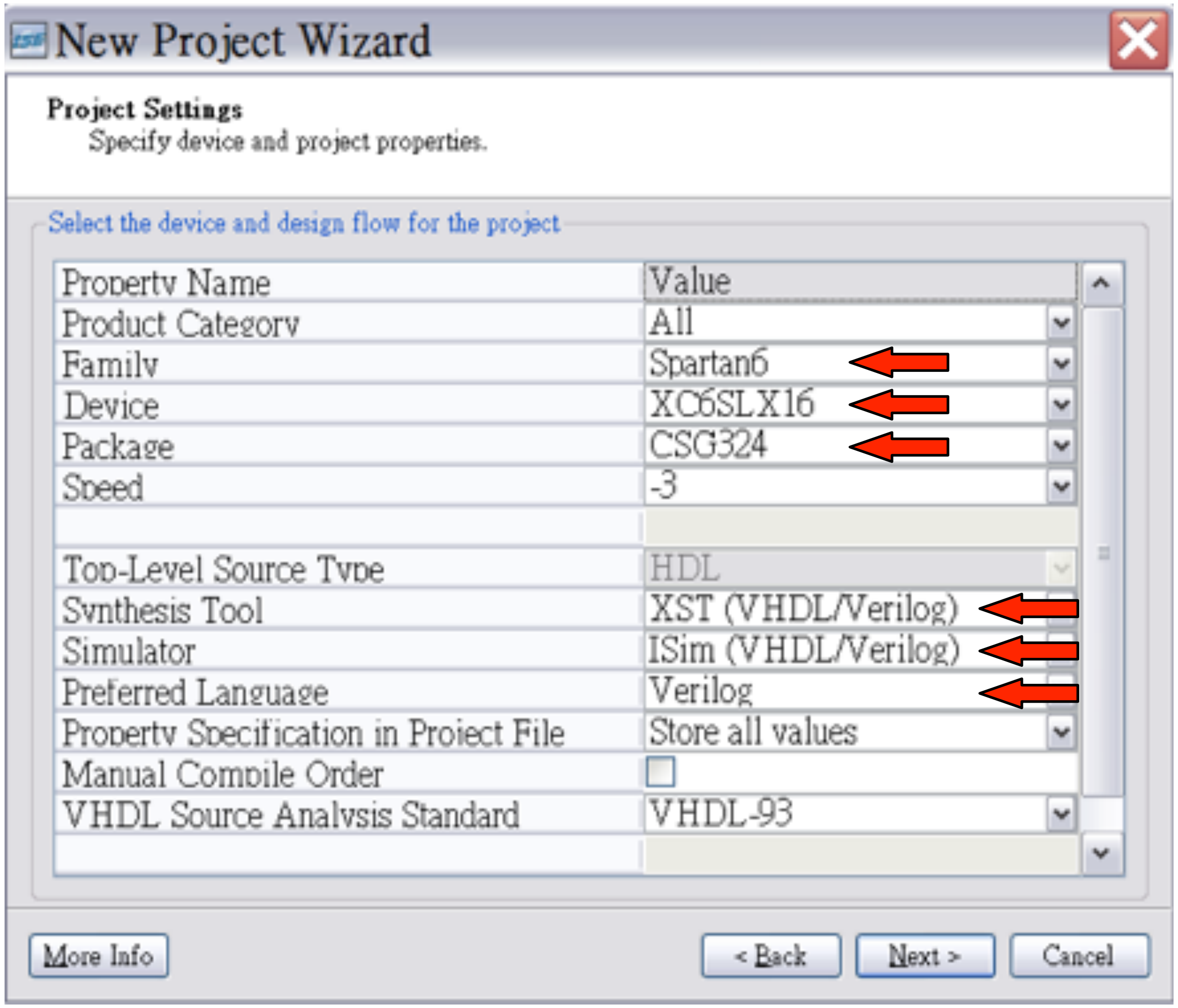
Test Your FPGA Board (3/3)





FPGA Emulation Using Xilinx ISE

Open New Project (2/2)



Project Settings
Specify device and project properties.

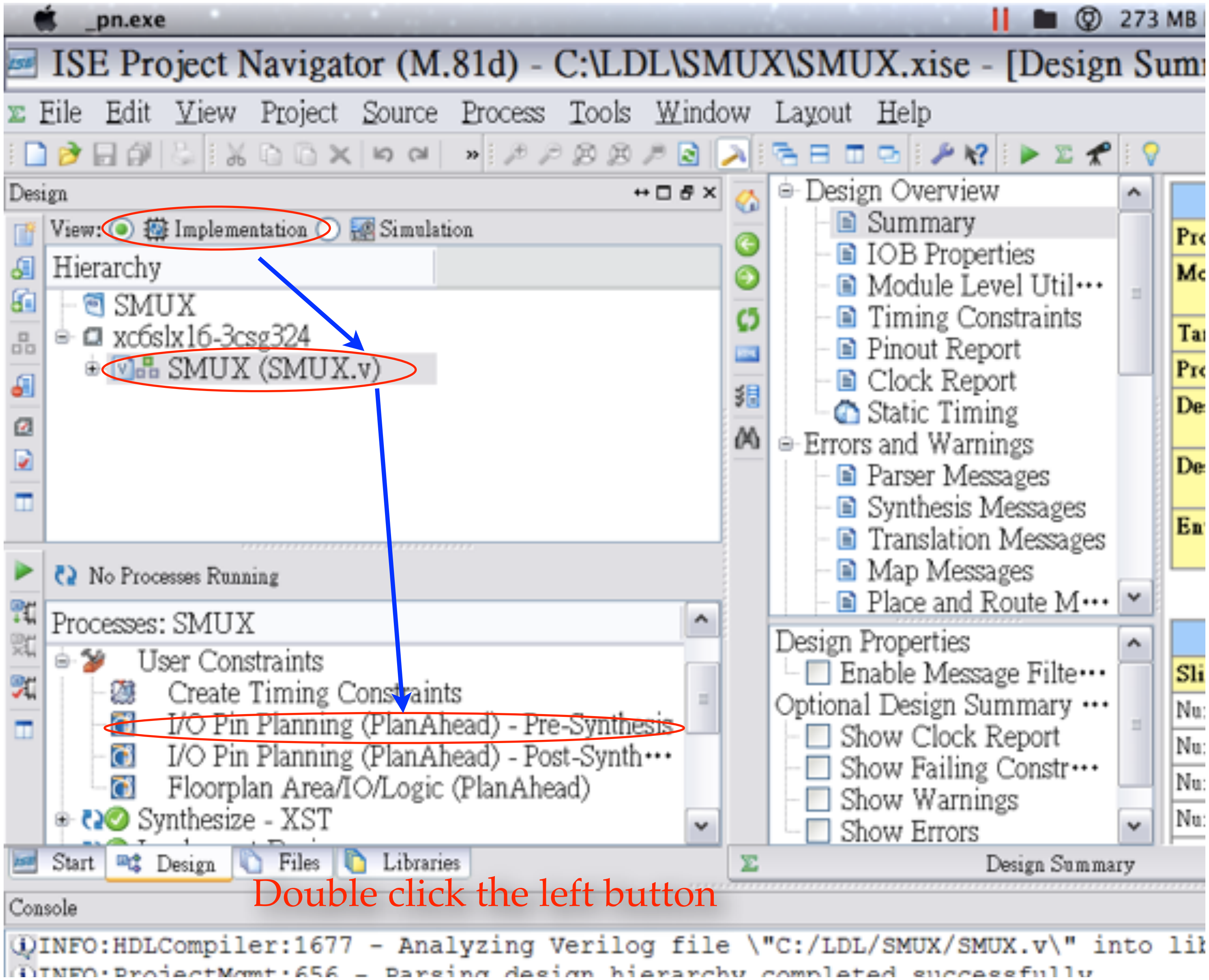
Select the device and design flow for the project

Property Name	Value
Product Category	All
Family	Spartan6
Device	XC6SLX16
Package	CSG324
Speed	-3
Top-Level Source Type	HDL
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	Verilog
Property Specification in Project File	Store all values
Manual Compile Order	☐
VHDL Source Analysis Standard	VHDL-93

NTHU EE

Laboratory for Reliable Computing

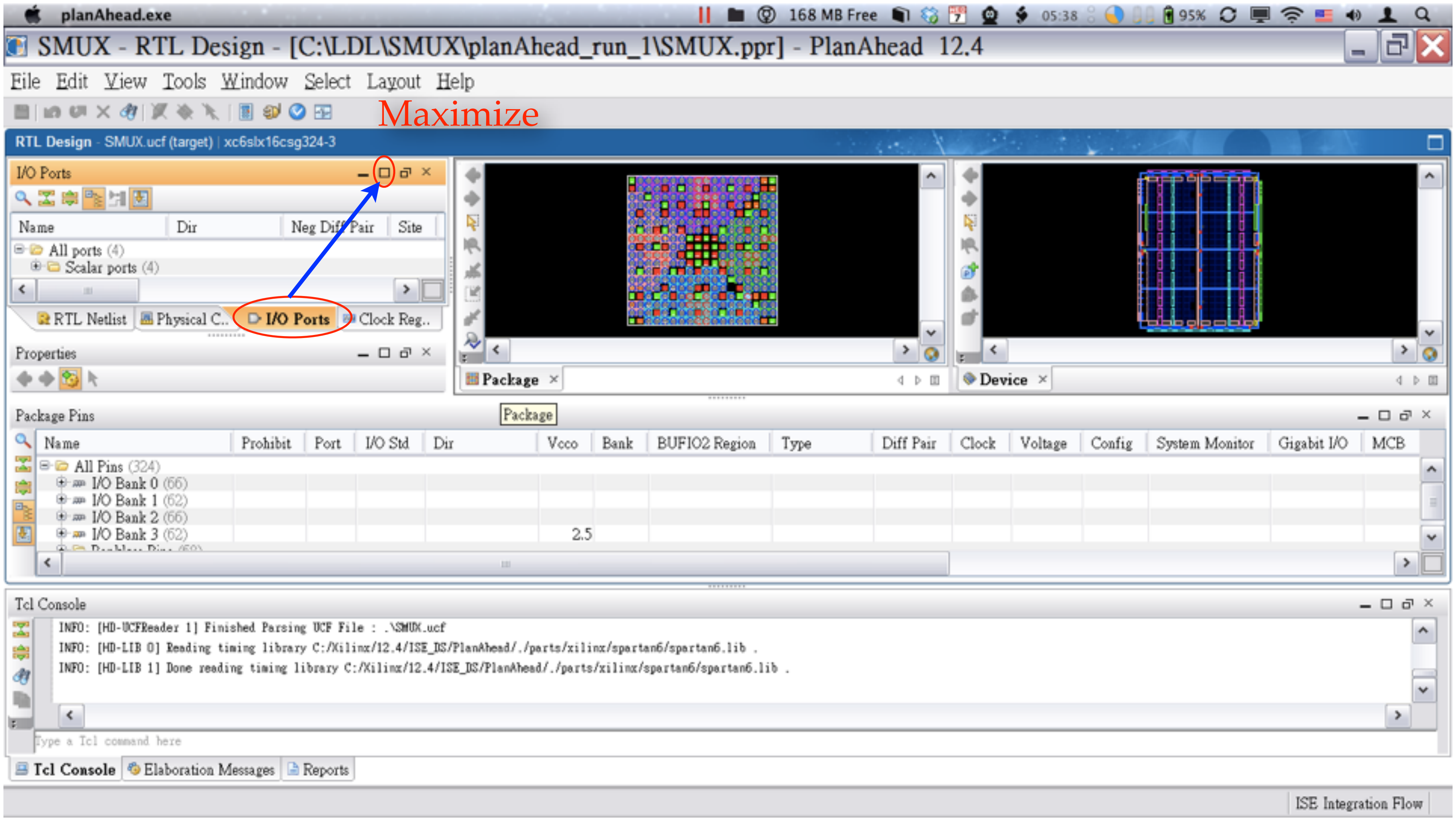
Back to Implementation



```
INFO:HDLCompiler:1677 - Analyzing Verilog file \"C:/LDL/SMUX/SMUX.v\" into lil
INFO:ProjectMgmt:656 - Parsing design hierarchy completed successfully
```

Double click the left button

I/O Pins Assignment: PlanAhead (1/3)

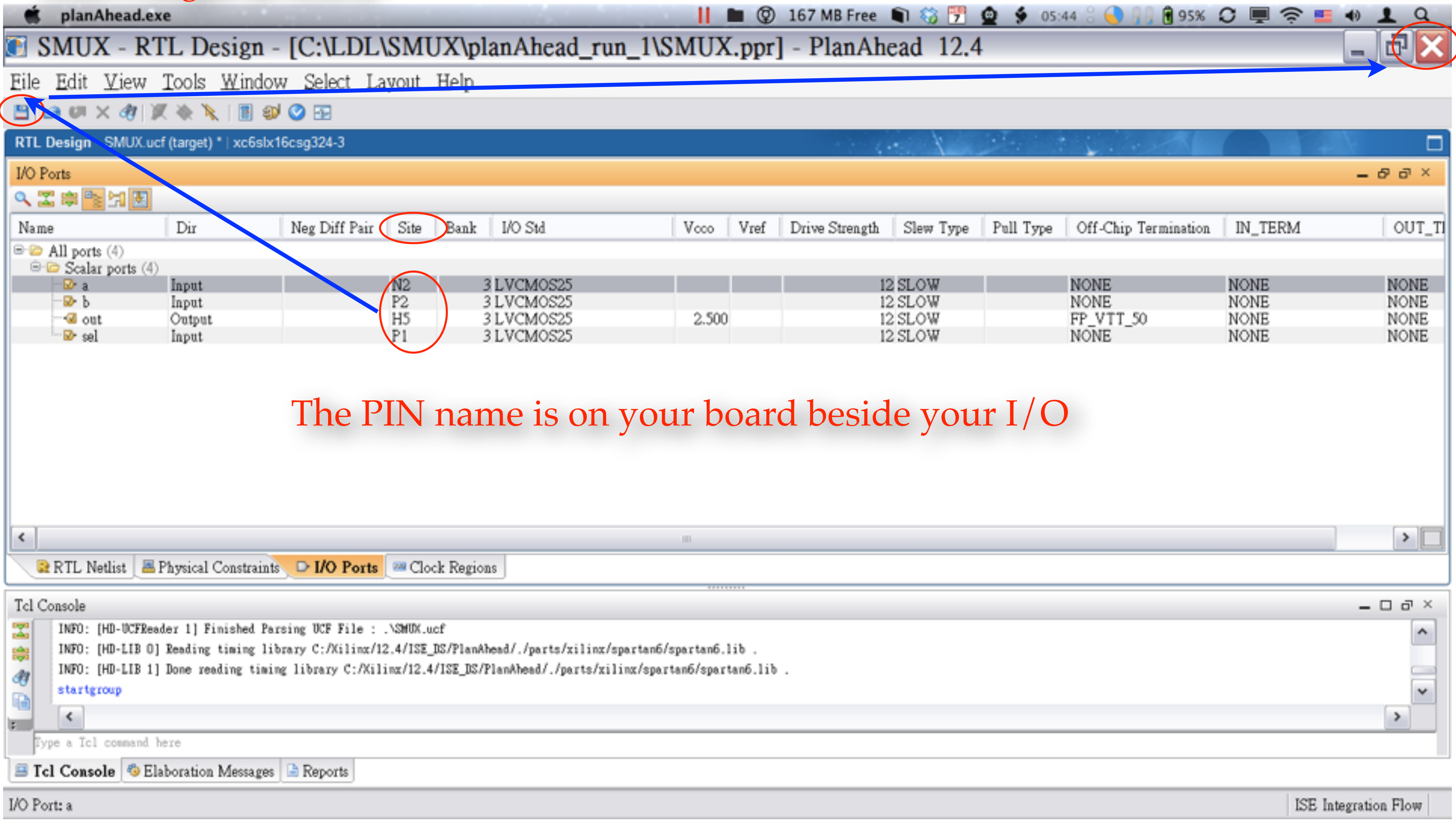


```
INFO: [HD-UCFReader 1] Finished Parsing UCF File : .\SMUX.ucf
INFO: [HD-LIB 0] Reading timing library C:/Xilinx/12.4/ISE_DS/PlanAhead/./parts/xilinx/spartan6/spartan6.lib .
INFO: [HD-LIB 1] Done reading timing library C:/Xilinx/12.4/ISE_DS/PlanAhead/./parts/xilinx/spartan6/spartan6.lib .
```

I/O Pin Assignment: PlanAhead (2/3)

Don't forget to SAVE

close

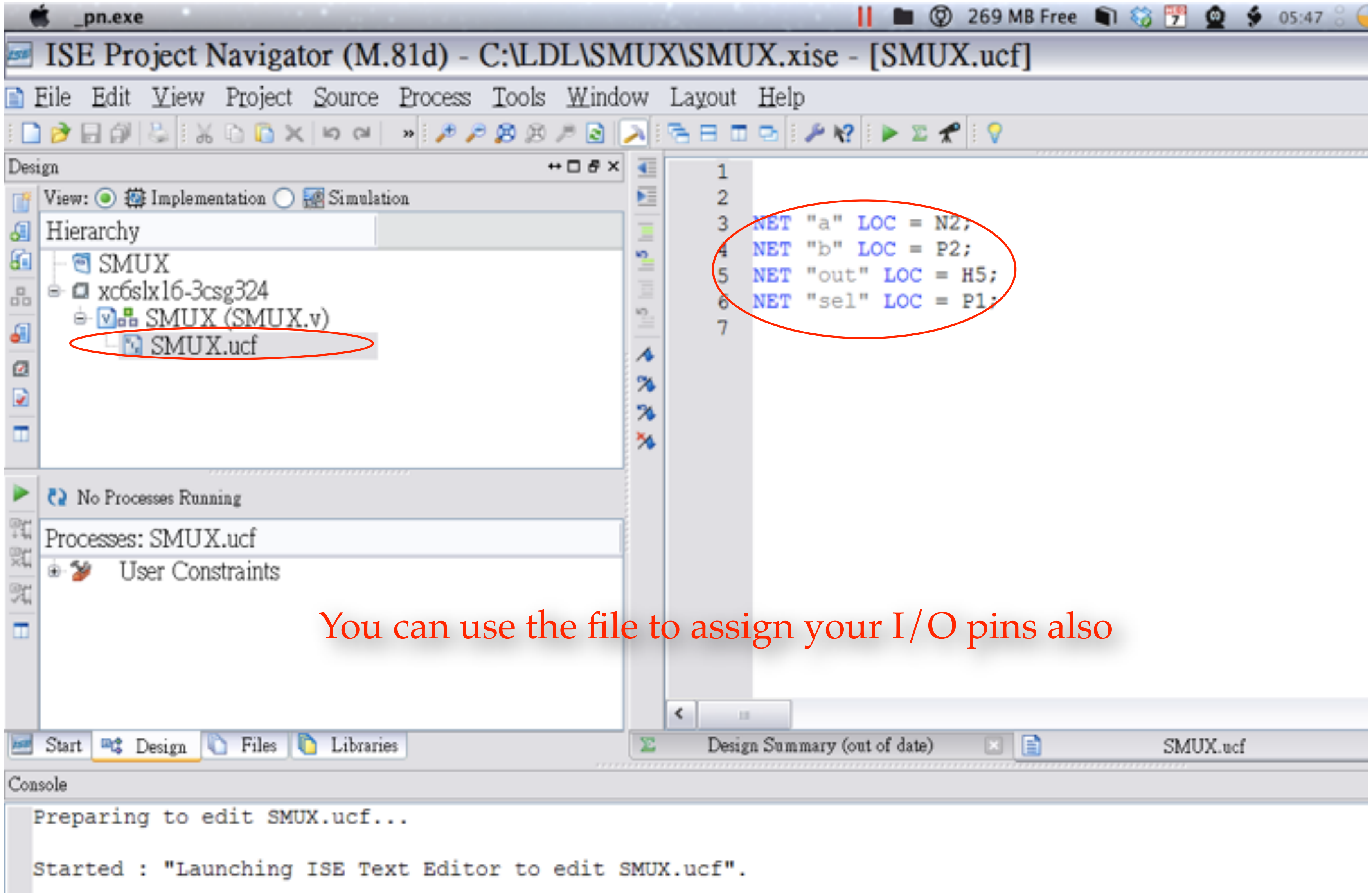


Name	Dir	Neg Diff Pair	Site	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	IN_TERM	OUT_TE
All ports (4)													
Scalar ports (4)													
a	Input		N2	3	LVCMOS25			12	SLOW		NONE	NONE	NONE
b	Input		P2	3	LVCMOS25			12	SLOW		NONE	NONE	NONE
out	Output		H5	3	LVCMOS25	2.500		12	SLOW		FP_VTT_50	NONE	NONE
sel	Input		P1	3	LVCMOS25			12	SLOW		NONE	NONE	NONE

```
INFO: [HD-UCFReader 1] Finished Parsing UCF File : .\SMUX.ucf
INFO: [HD-LIB 0] Reading timing library C:/Xilinx/12.4/ISE_DS/PlanAhead/./parts/xilinx/spartan6/spartan6.lib .
INFO: [HD-LIB 1] Done reading timing library C:/Xilinx/12.4/ISE_DS/PlanAhead/./parts/xilinx/spartan6/spartan6.lib .
startgroup
```

The PIN name is on your board beside your I/O

I/O Pin Assignment: Another Method (.ucf file) (3/3)



```
NET "a" LOC = N2;
NET "b" LOC = P2;
NET "out" LOC = H5;
NET "sel" LOC = P1;
```

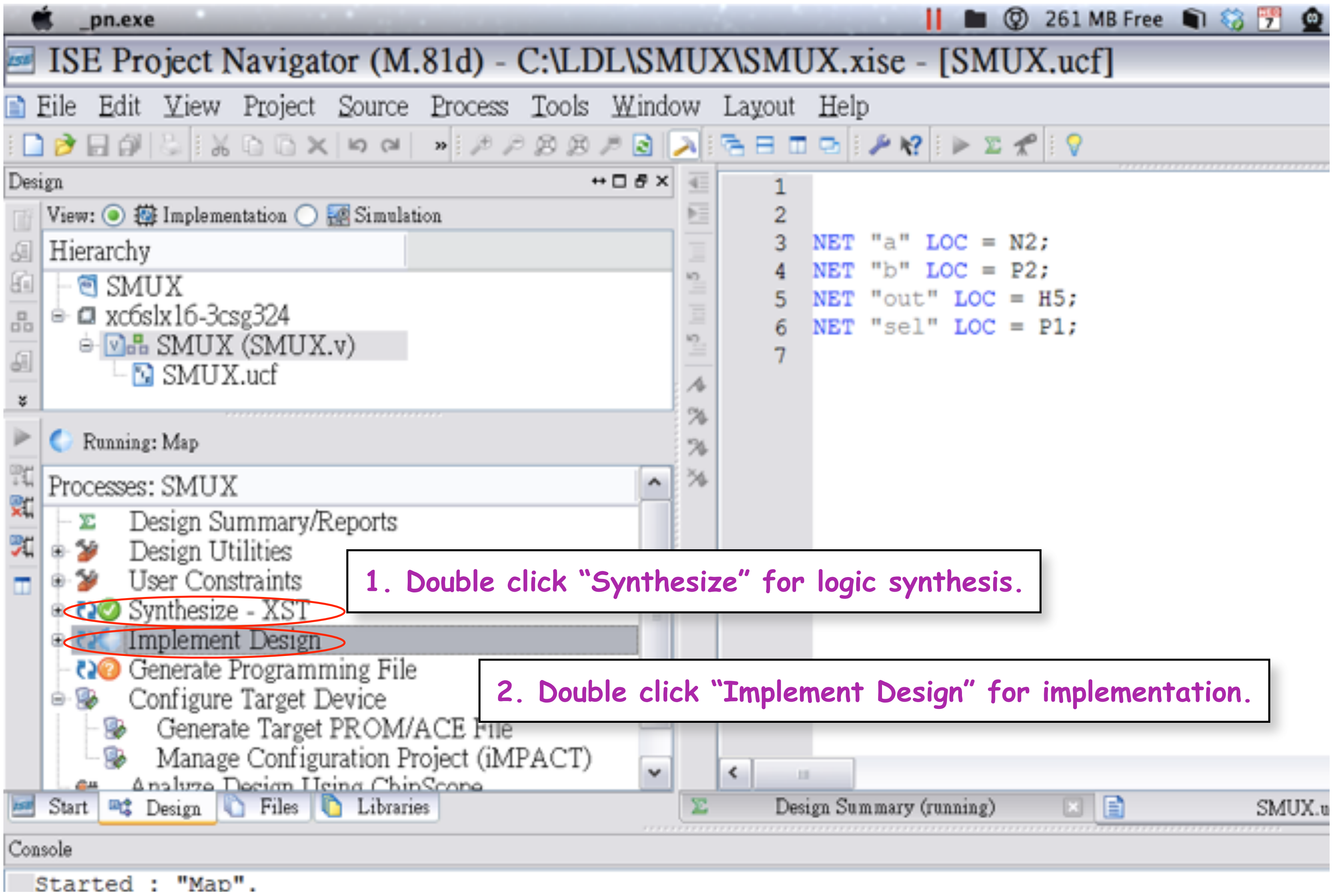
```
Preparing to edit SMUX.ucf...

Started : "Launching ISE Text Editor to edit SMUX.ucf".
```

You can use the file to assign your I/O pins also

Laboratory for Reliable Computing

Synthesize and Implementation


_pn.exe
261 MB Free
ISE Project Navigator (M.81d) - C:\LDL\SMUX\SMUX.xise - [SMUX.ucf]
File Edit View Project Source Process Tools Window Layout Help
Design
View: Implementation Simulation
Hierarchy
SMUX
xc6slx16-3csg324
SMUX (SMUX.v)
SMUX.ucf
Running: Map
Processes: SMUX
Design Summary/Reports
Design Utilities
User Constraints
Synthesize - XST
Implement Design
Generate Programming File
Configure Target Device
Generate Target PROM/ACE File
Manage Configuration Project (iMPACT)
Analyze Design Using ChipScope
Start
Design
Files
Libraries
Design Summary (running)
SMUX.u
Console

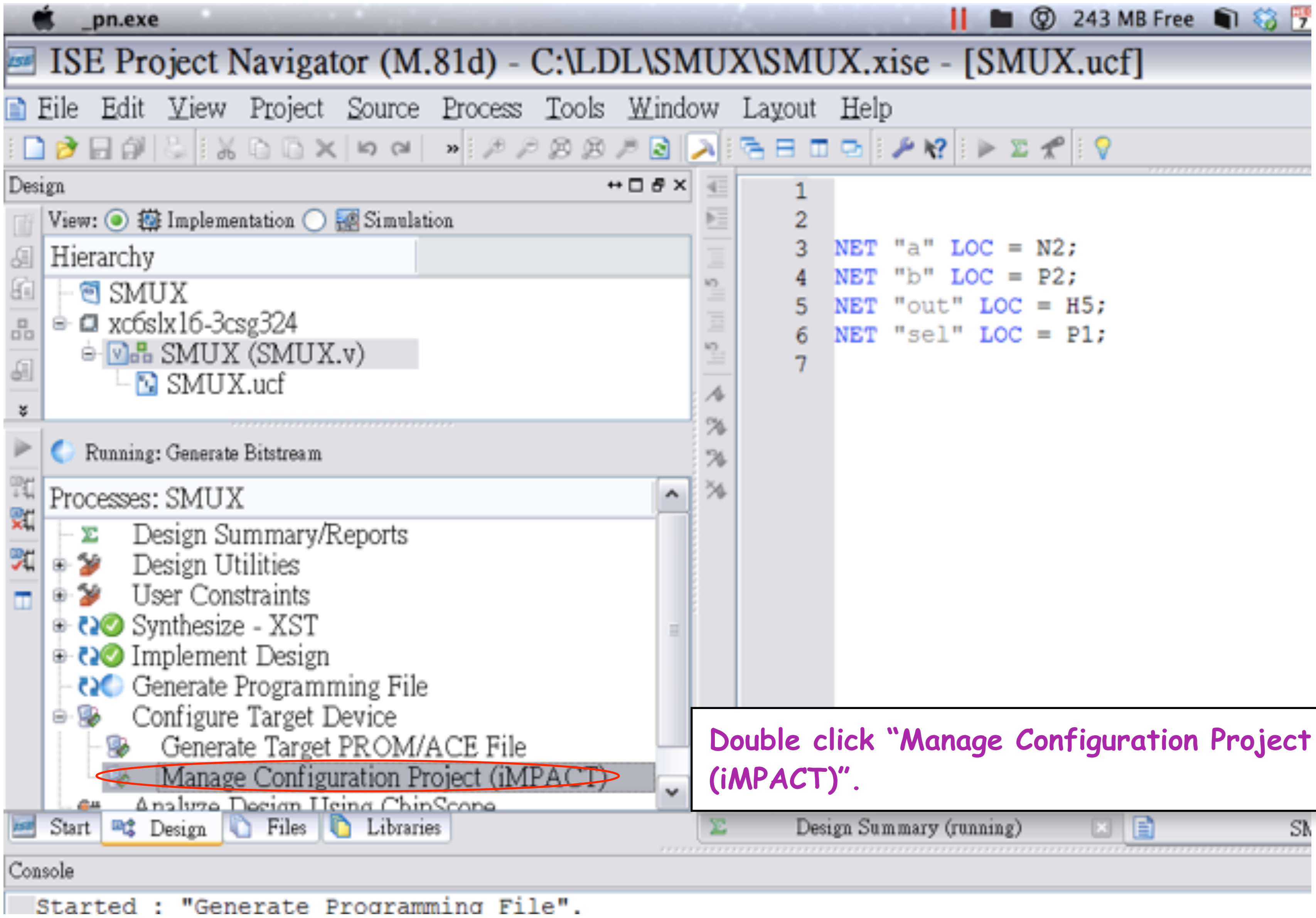
```
NET "a" LOC = N2;
NET "b" LOC = P2;
NET "out" LOC = H5;
NET "sel" LOC = P1;
```

```
Started : "Map".
```

1. Double click "Synthesize" for logic synthesis.

2. Double click "Implement Design" for implementation.

Download Bit File for Emulation

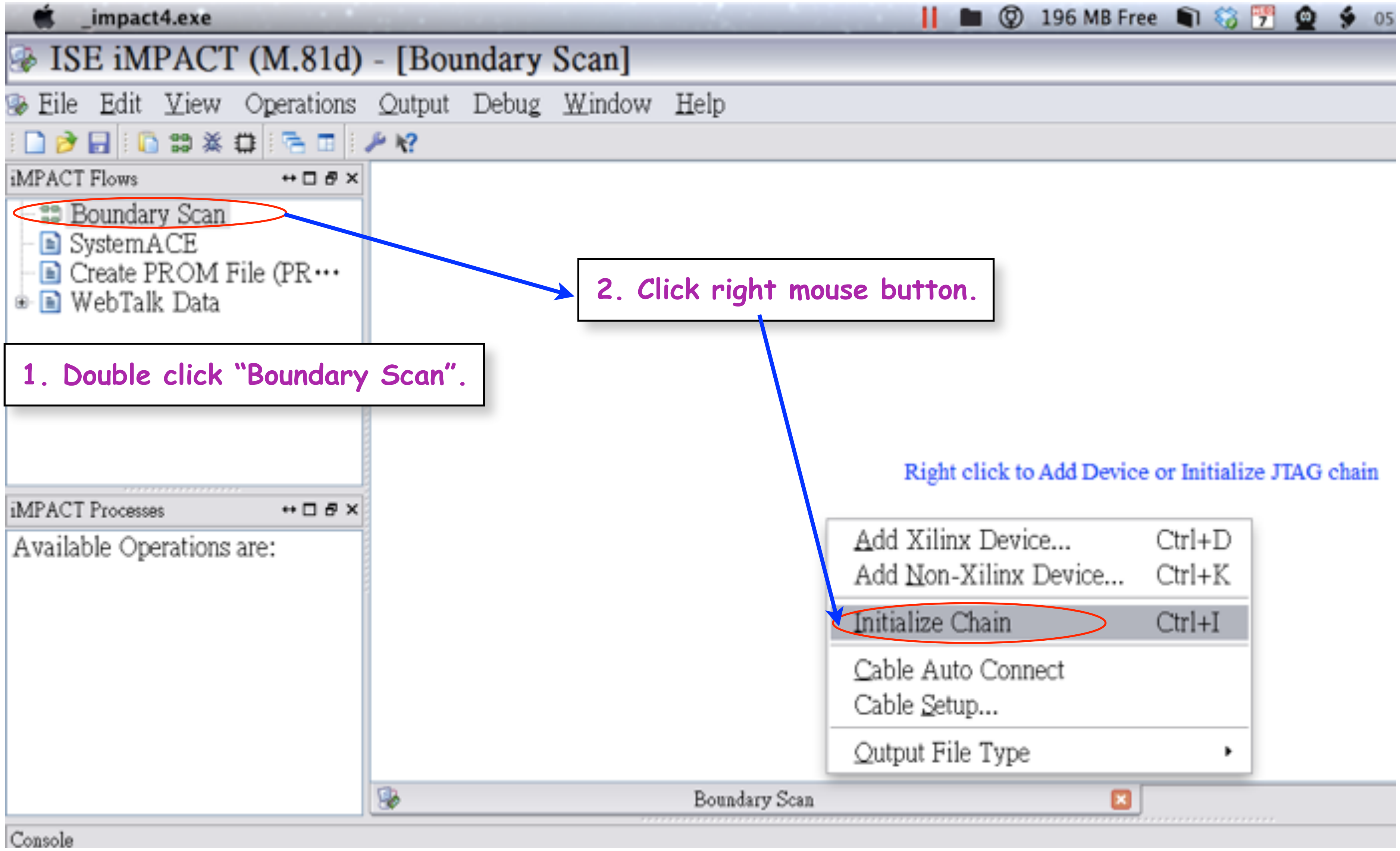


ISE Project Navigator (M.81d) - C:\LDL\SMUX\SMUX.xise - [SMUX.ucf]

```
NET "a" LOC = N2;
NET "b" LOC = P2;
NET "out" LOC = H5;
NET "sel" LOC = P1;
```

```
Started : "Generate Programming File".
```

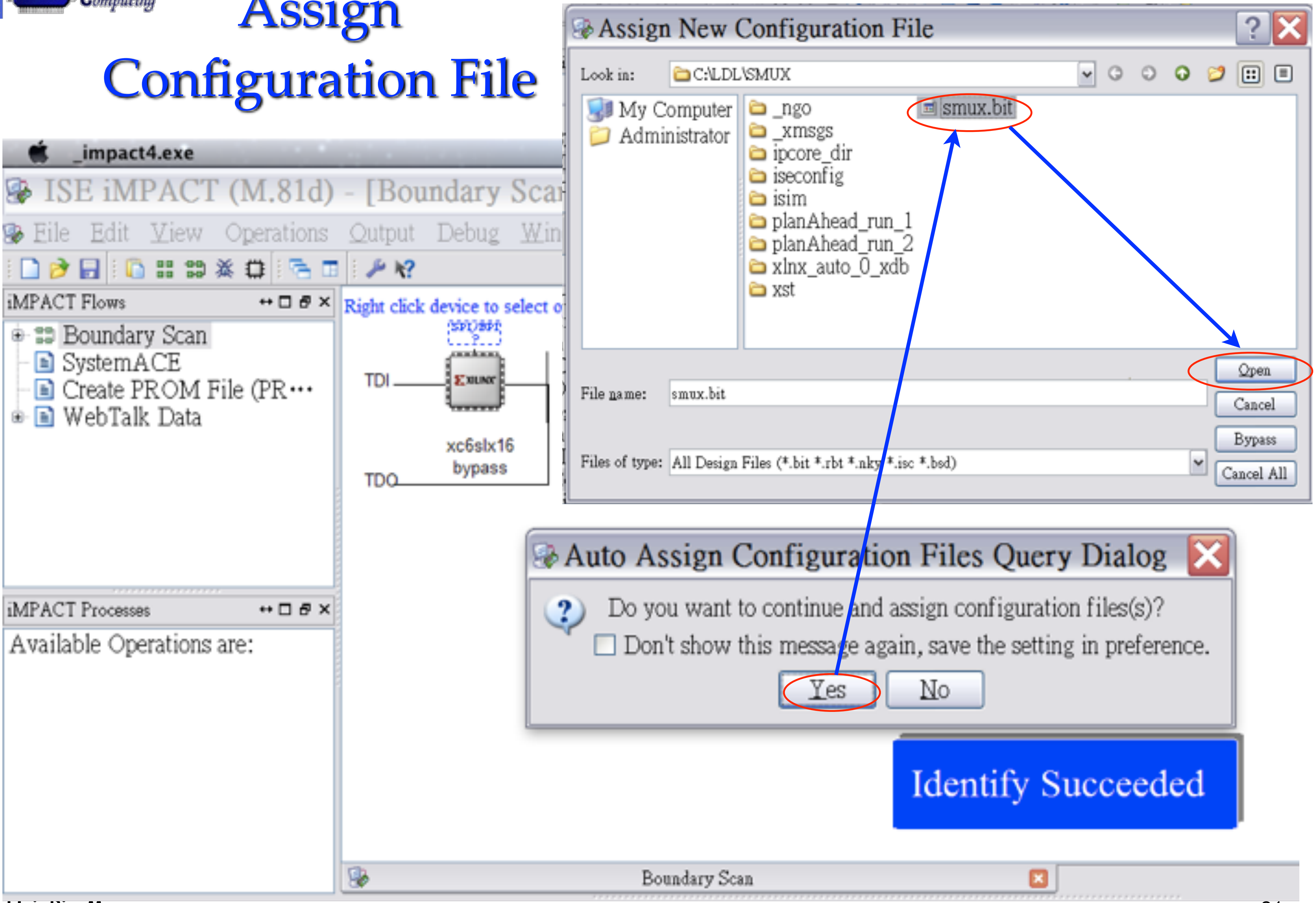
Double click "Manage Configuration Project (iMPACT)".

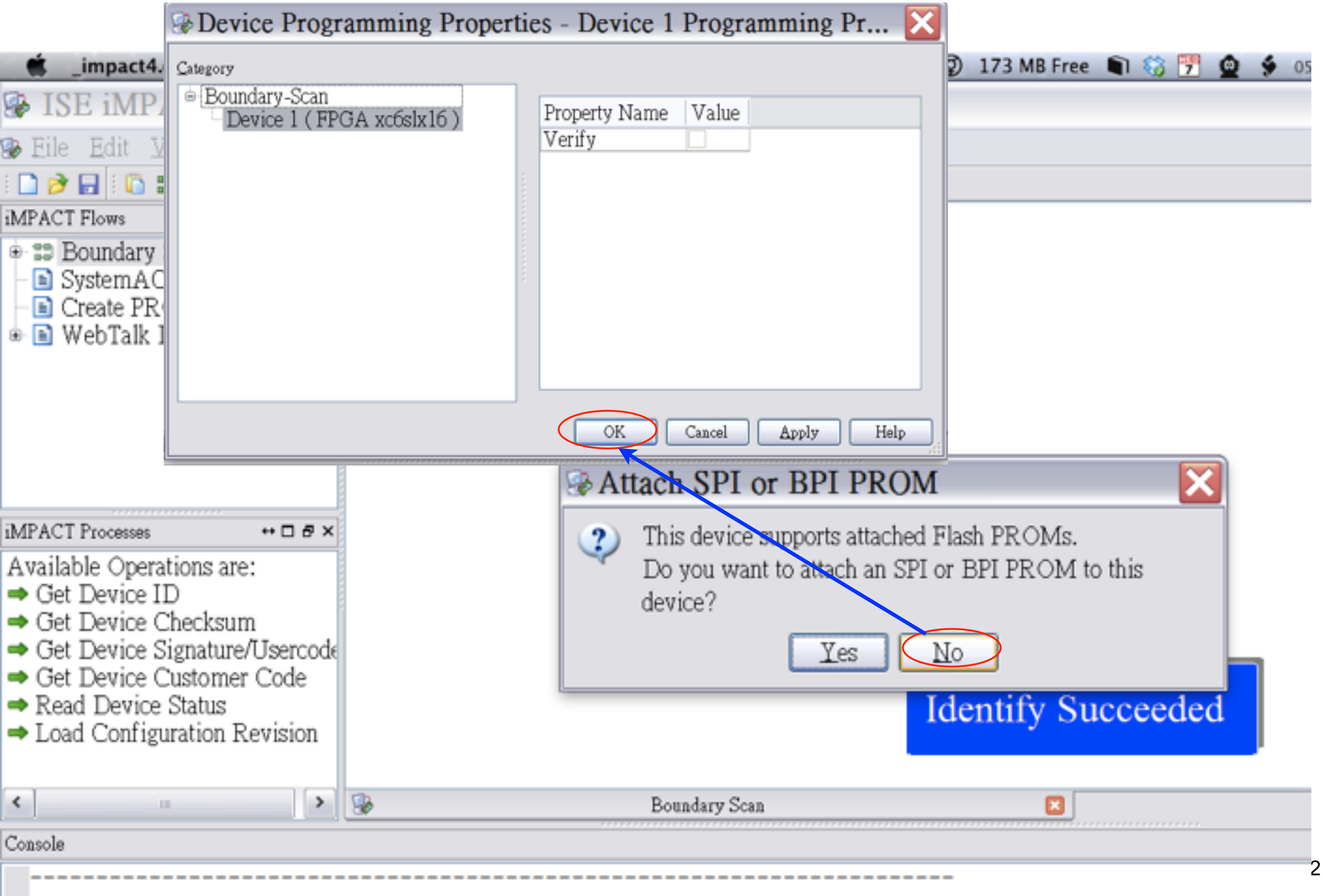

_impact4.exe
196 MB Free
ISE iMPACT (M.81d) - [Boundary Scan]
File Edit View Operations Output Debug Window Help
iMPACT Flows
Boundary Scan
SystemACE
Create PROM File (PR…
WebTalk Data
iMPACT Processes
Available Operations are:
Right click to Add Device or Initialize JTAG chain
Add Xilinx Device... Ctrl+D
Add Non-Xilinx Device... Ctrl+K
Initialize Chain Ctrl+I
Cable Auto Connect
Cable Setup...
Output File Type
Boundary Scan
Console

1. Double click "Boundary Scan".

2. Click right mouse button.

Assign Configuration File


_impact4.exe
ISE iMPACT (M.81d) - [Boundary Scan
File Edit View Operations Output Debug Win
iMPACT Flows
Boundary Scan
SystemACE
Create PROM File (PR…
WebTalk Data
iMPACT Processes
Available Operations are:
Right click device to select o
SPI/BPI
TDI
TDO
xc6slx16
bypass
Boundary Scan
Assign New Configuration File
Look in: C:\LDL\SMUX
My Computer
Administrator
_ngo
_xmsgs
ipcore_dir
iseconfig
isim
planAhead_run_1
planAhead_run_2
xlnx_auto_0_xdb
xst
smux.bit
File name: smux.bit
Files of type: All Design Files (*.bit *.rbt *.nky *.isc *.bsd)
Open
Cancel
Bypass
Cancel All
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
Identify Succeeded



Device Programming Properties - Device 1 Programming Pr...

Property Name	Value
Verify	☐

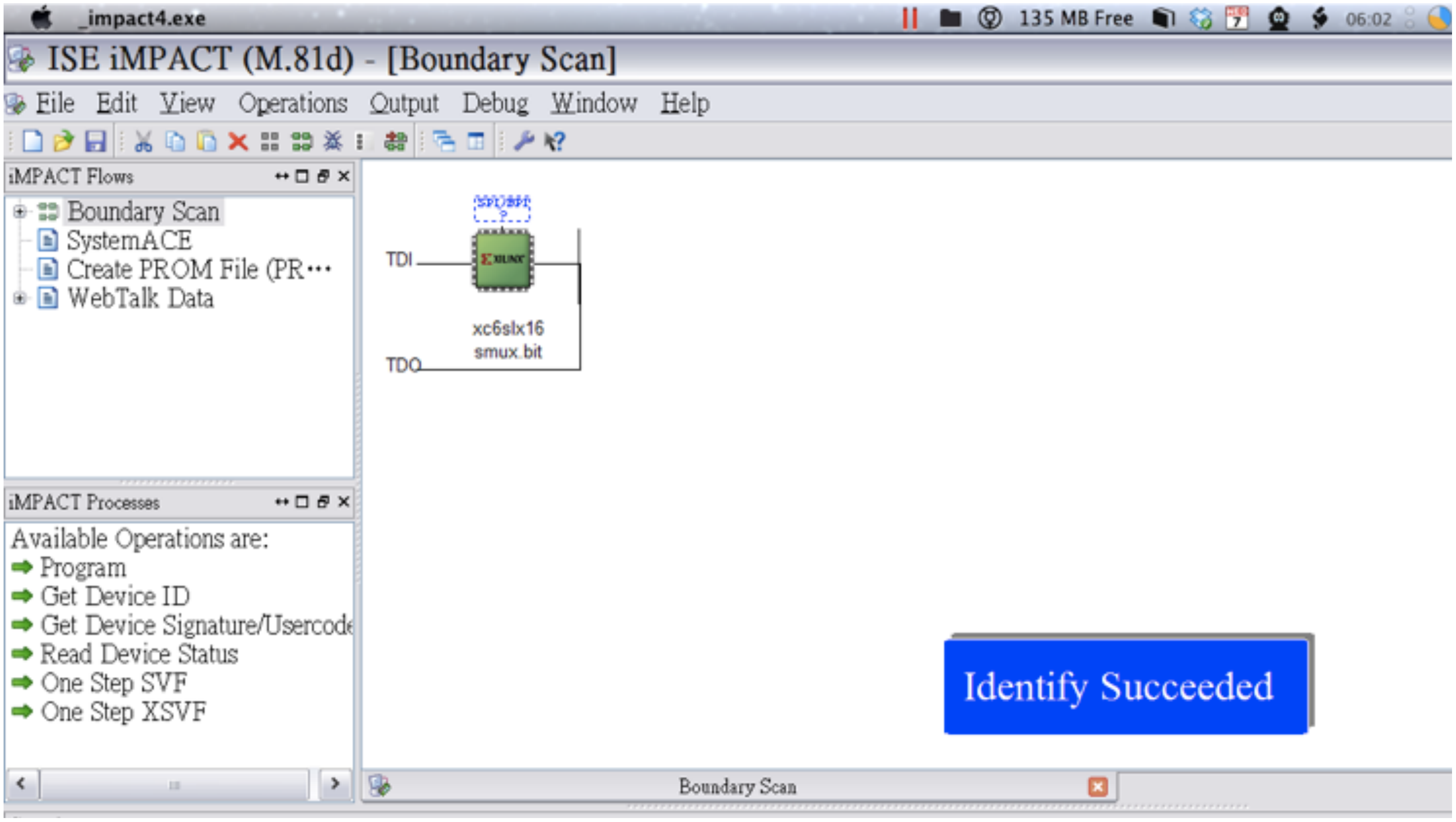
Attach SPI or BPI PROM

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

Yes [✓ No]

Available Operations are:
- Get Device ID
- Get Device Checksum
- Get Device Signature/Usercode
- Get Device Customer Code
- Read Device Status
- Load Configuration Revision

Identify

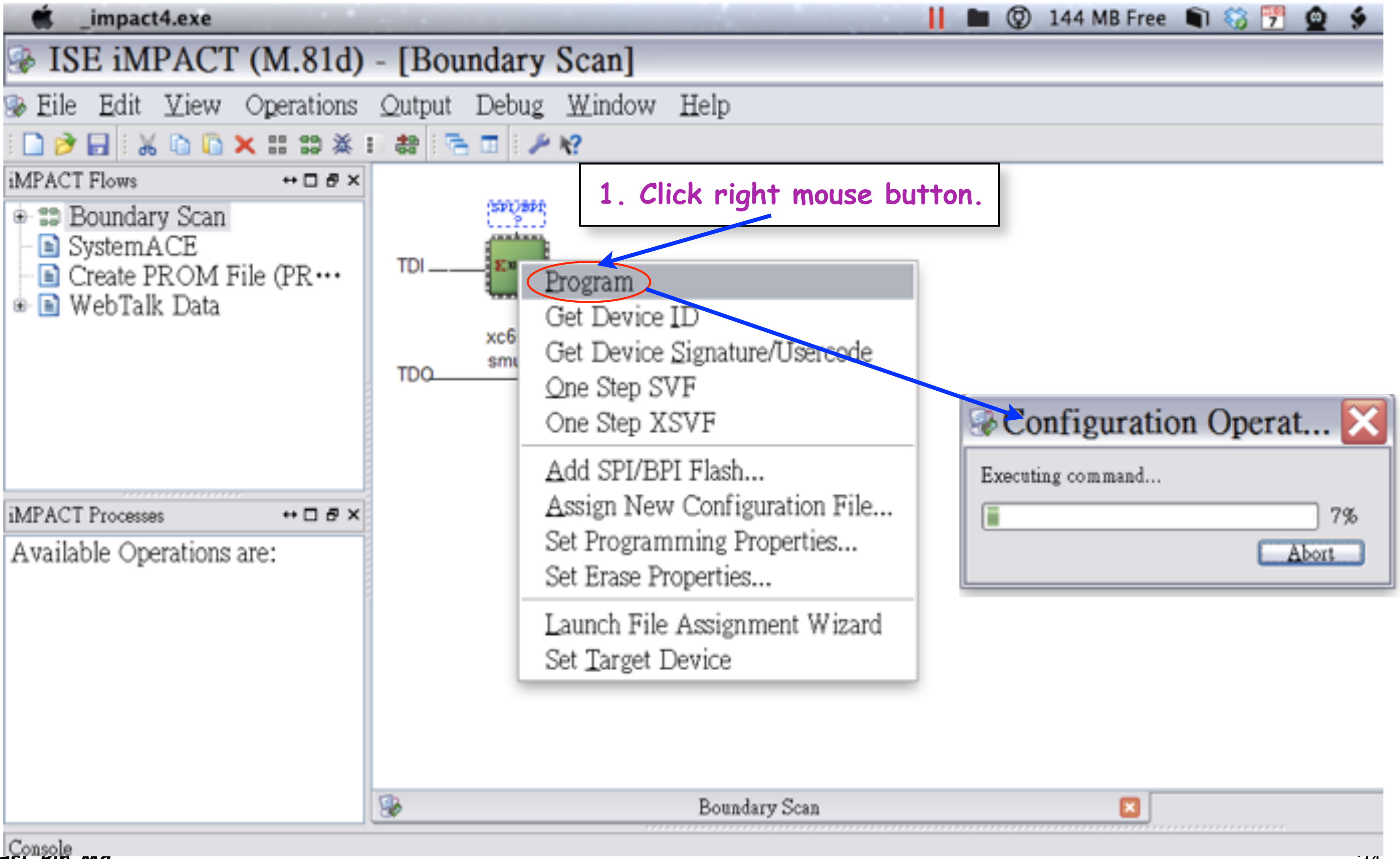


iMPACT Processes

Available Operations are:
- Program
- Get Device ID
- Get Device Signature/Usercode
- Read Device Status
- One Step SVF
- One Step XSVF

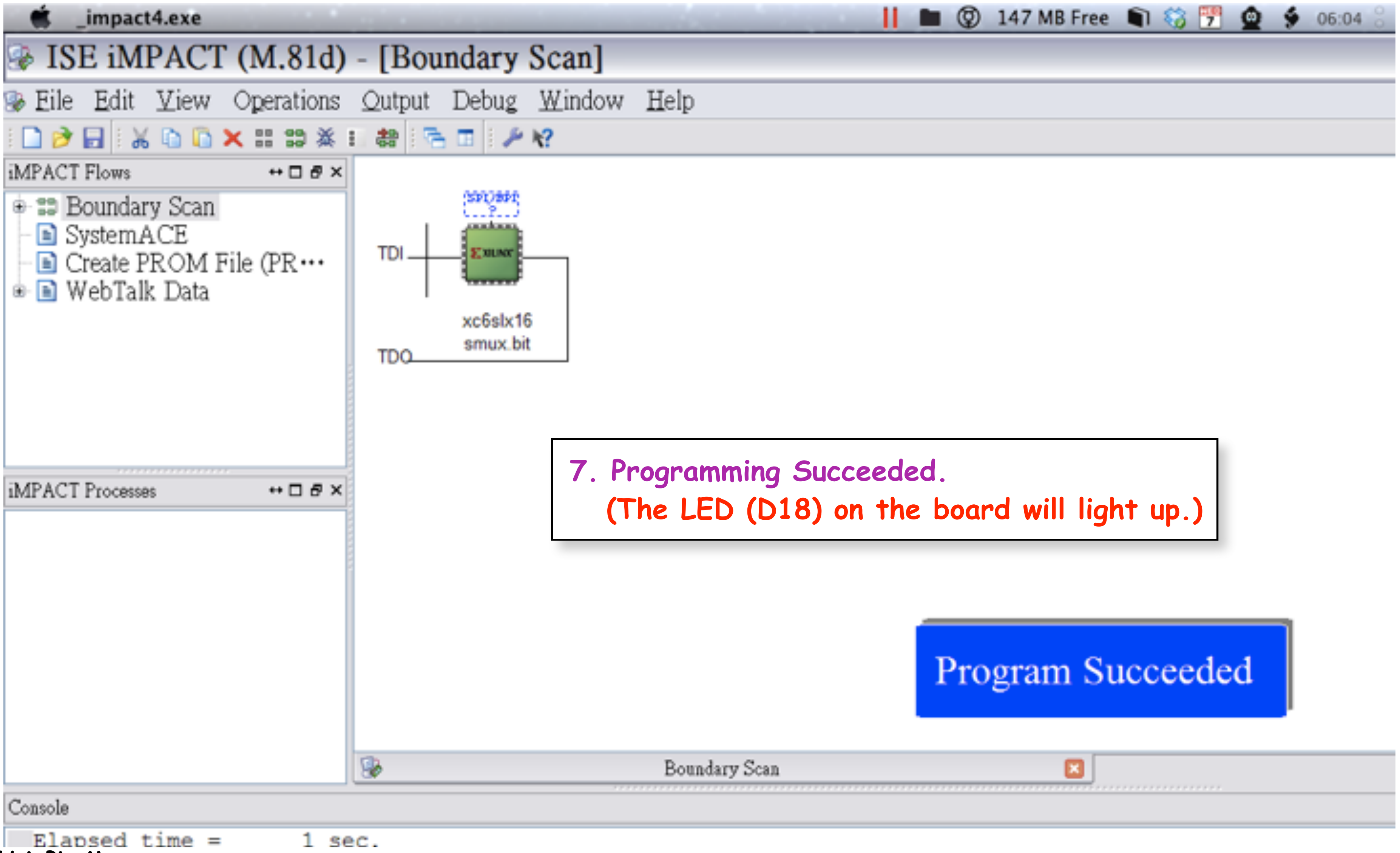
Identify Succeeded

Program the Device (1/2)



Laboratory for Reliable Computing

Program the Device (2/2)



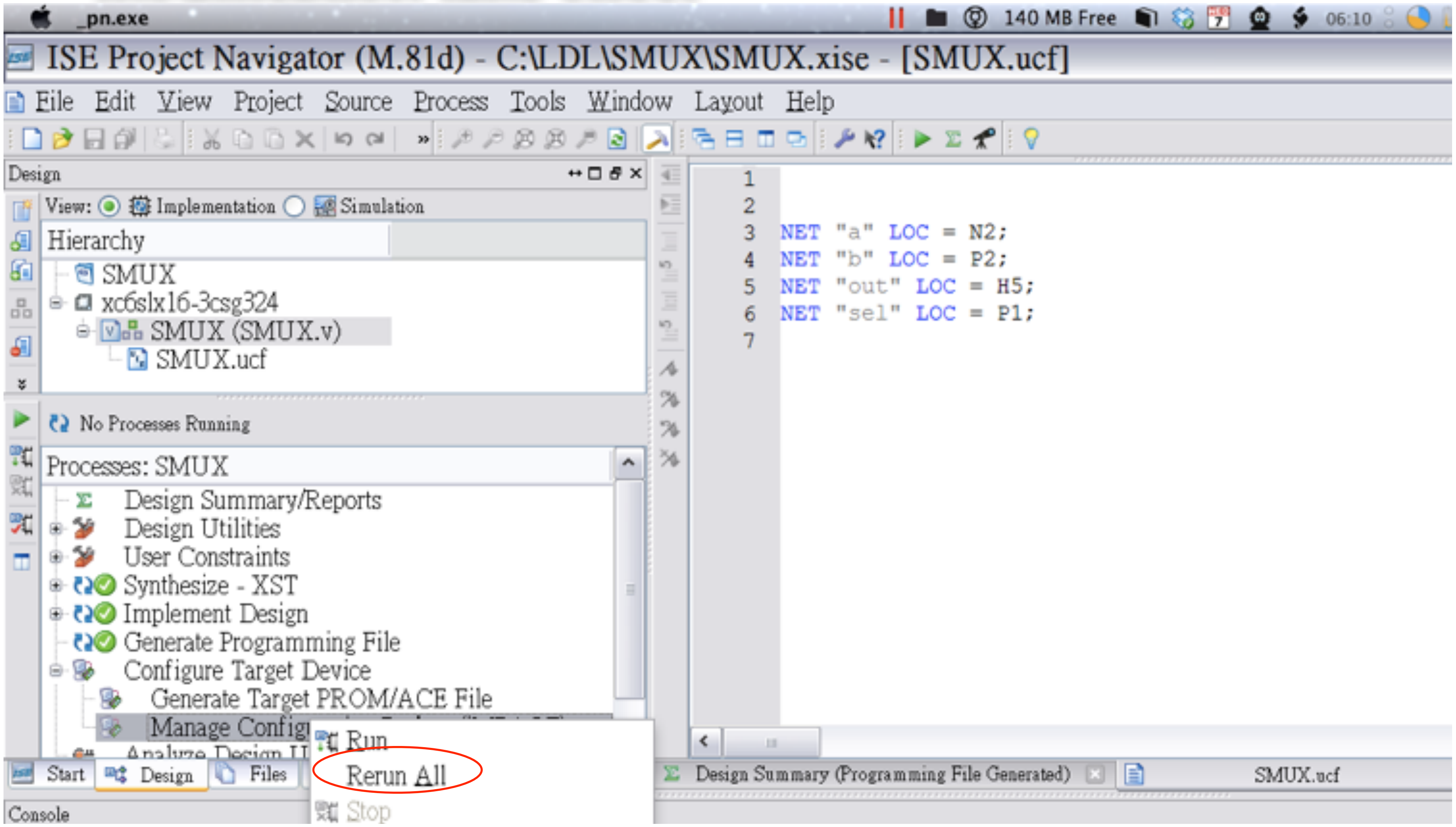
Program Succeeded

```
Elapsed time =      1 sec.
```

7. Programming Succeeded.
   (The LED (D18) on the board will light up.)

Some Notes (1/2)

- In any stage, you can 'Rerun All' to let your modification take effect



ISE Project Navigator (M.81d) - C:\LDL\SMUX\SMUX.xise - [SMUX.ucf]

```
NET "a" LOC = N2;
NET "b" LOC = P2;
NET "out" LOC = H5;
NET "sel" LOC = P1;
```

Some Notes (2/2)

- Sometimes, the database of the design will be corrupted, and any changes will not take effect or your board behaves weird.
  - Open a new project with fresh source files.
- Look into the 'Errors' or 'Warnings' windows to debug your design.
- If you finish your lab at dorm and want to bring it to the lab for demo
  - DO NOT copy the entire directory to the lab and use the same directory for demo
    - Just copy the .v and .ucf files to the lab is sufficient.
    - Use 'New Project' in the lab and open the existing source files to re-implement your design for demo

Verilog RTL Code Examples

```
`timescale 1ns / 1ps
//***************************************************************************
// Filename       : SMUX.v
// Author         : Hsi-Pin Ma
// Function       : multiplexer
// Last Modified : Date: 2007-02-16 21:16:17 +0800 (Fri, 16 Feb 2007)
// Revision       : Revision: 1
// Copyright (c), Laboratory for Reliable Computing (LaRC), EE, NTHU
// All rights reserved
//***************************************************************************

module SMUX(
  out, // multiplexer output
  a, // multiplexer input a
  b, // multiplexer input b
  sel // selection control signal
);
output out; // multiplexer output
input a,b; // two inputs to be selected
input sel; // selection control signal

// multiplexer funtion
assign out = (sel) ? a : b ;

endmodule
```

```
`timescale 1ns / 1ps
//*************************************************************************
// Filename       : SMUX.v
// Author        : Hsi-Pin Ma
// Function       : multiplexer
// Last Modified : Date: 2007-02-16 21:16:17 +0800 (Fri, 16 Feb 2007)
// Revision       : Revision: 1
// Copyright (c), Laboratory for Reliable Computing (LaRC), EE, NTHU
// All rights reserved
//*************************************************************************

module SMUX(
  out, // multiplexer output
  a, // multiplexer input a
  b, // multiplexer input b
  sel // selection control signal
);
output out; // multiplexer output
input a,b; // two inputs to be selected
input sel; // selection control signal

reg out; // multiplexer output

// multiplexer funtion
always @*
  out = (sel) ? a : b ;

endmodule
```

Laboratory for Reliable Computing

D-type Flip Flop

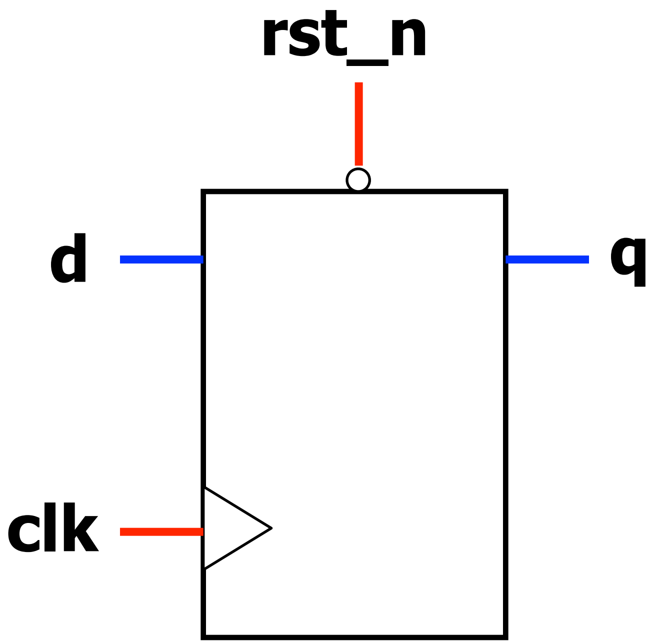
```
module dff(
 q,  // output
 d,  // input
 clk,  // global clock
 rst_n // active low reset
);

output q;  // output
input d;  // input
input clk;  // global clock
input rst_n;  // active low reset

reg q; // output (in always block)

always @(posedge clk or negedge rst_n)
 if (~rst_n)
   q<=0;
 else
   q<=d;

endmodule
```



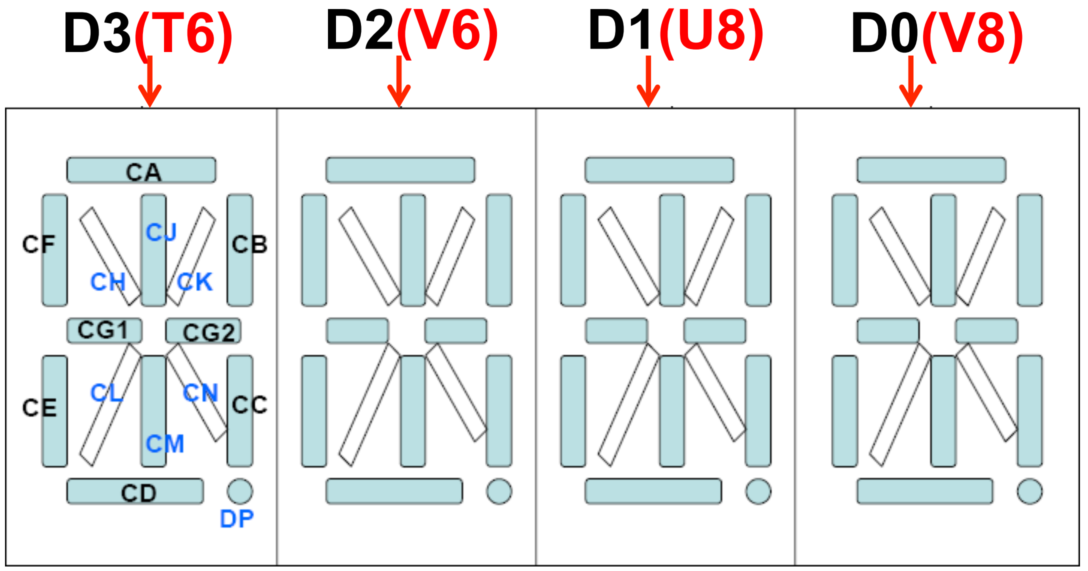
Laboratory for Reliable Computing

Declaration Syntax of Verilog Wire / Registers

- wire <range> ? <name> <,<name>>*;
- reg <range> ? <name> <,<name>>*;
- Example
  - reg a;
  - wire a;
  - reg [5:2] b,c;
  - wire [3:0] b,c;

14-Segment Display (1/4)

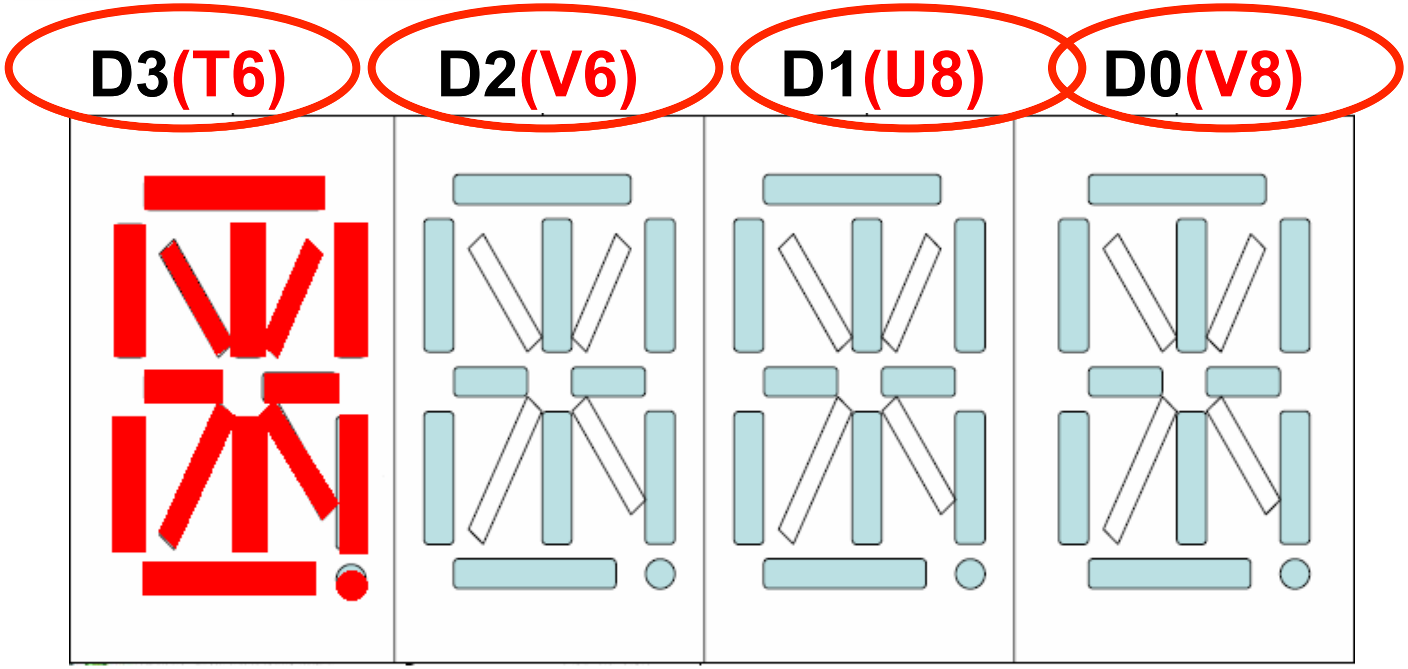
- The cathodes of similar segments on all four displays are connected to the same FPGA pin



symbol	CA	CB	CC	CD	CE	CF	CG1	CG2
FPGA pin	P6	N4	V5	T5	U7	R3	N5	R5
symbol	CH	CJ	CK	CL	CM	CN	DP	
FPGA pin	T3	T4	V4	V7	R7	T7	U5	

14-Segment Display (2/4)

- 15 pins to control each 14-segment display
  - Including the point
- 4 pins to choose which 14-seg to display
- Device is low activated



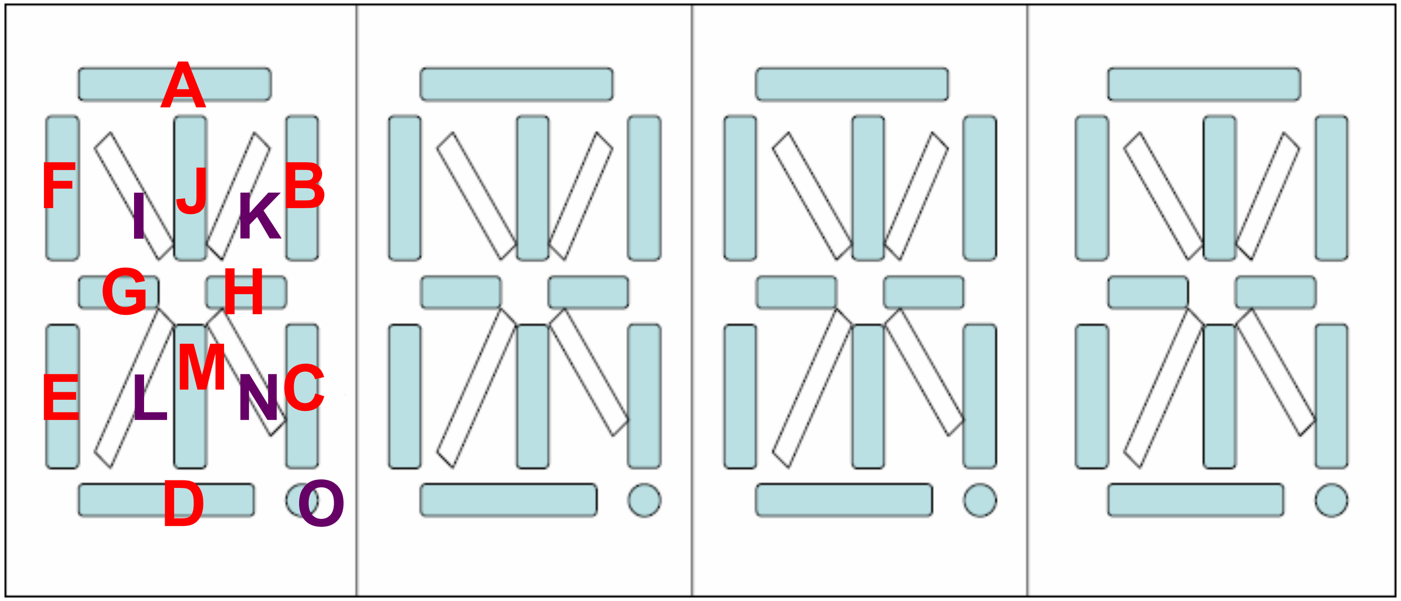
NTHU EE

Laboratory for Reliable Computing

14-Segment Display (3/4)

- `Input:  0101 0000 0011 1111 111`
  `Output:      ABCD EFGH IJKL MNO`

What is the response?



14-Segment Display (4/4)

- `Input:  0101 0000 0011 1111 111`
  `Output:      ABCD EFGH IJKL MNO`