



FPGA Emulation

Hsi-Pin Ma

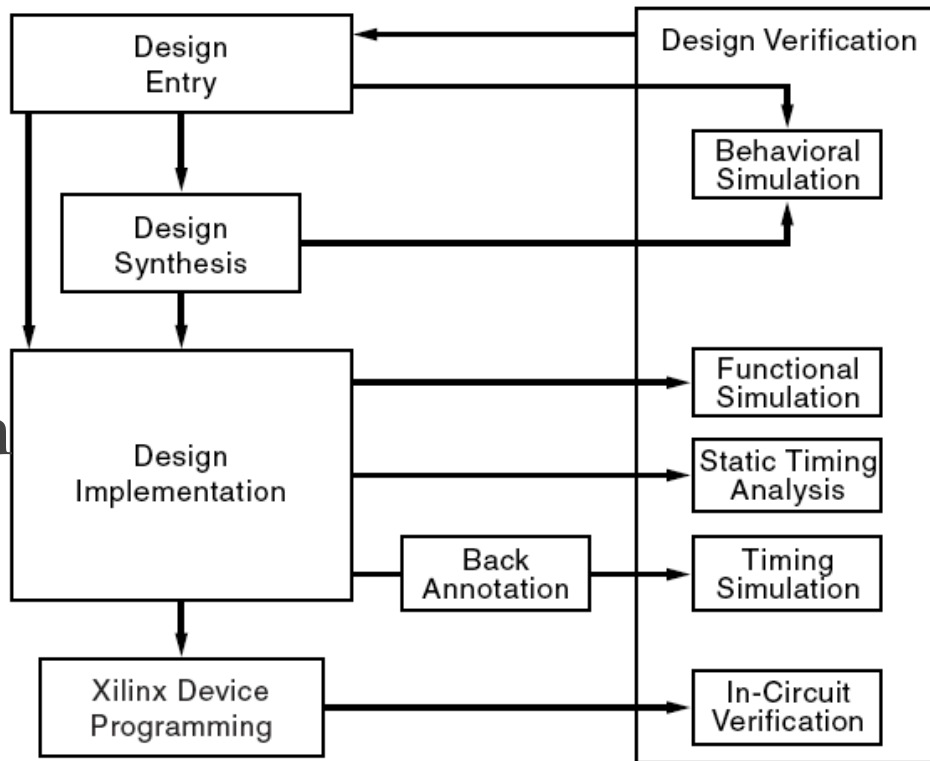
<http://lms.nthu.edu.tw/course/24953>

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



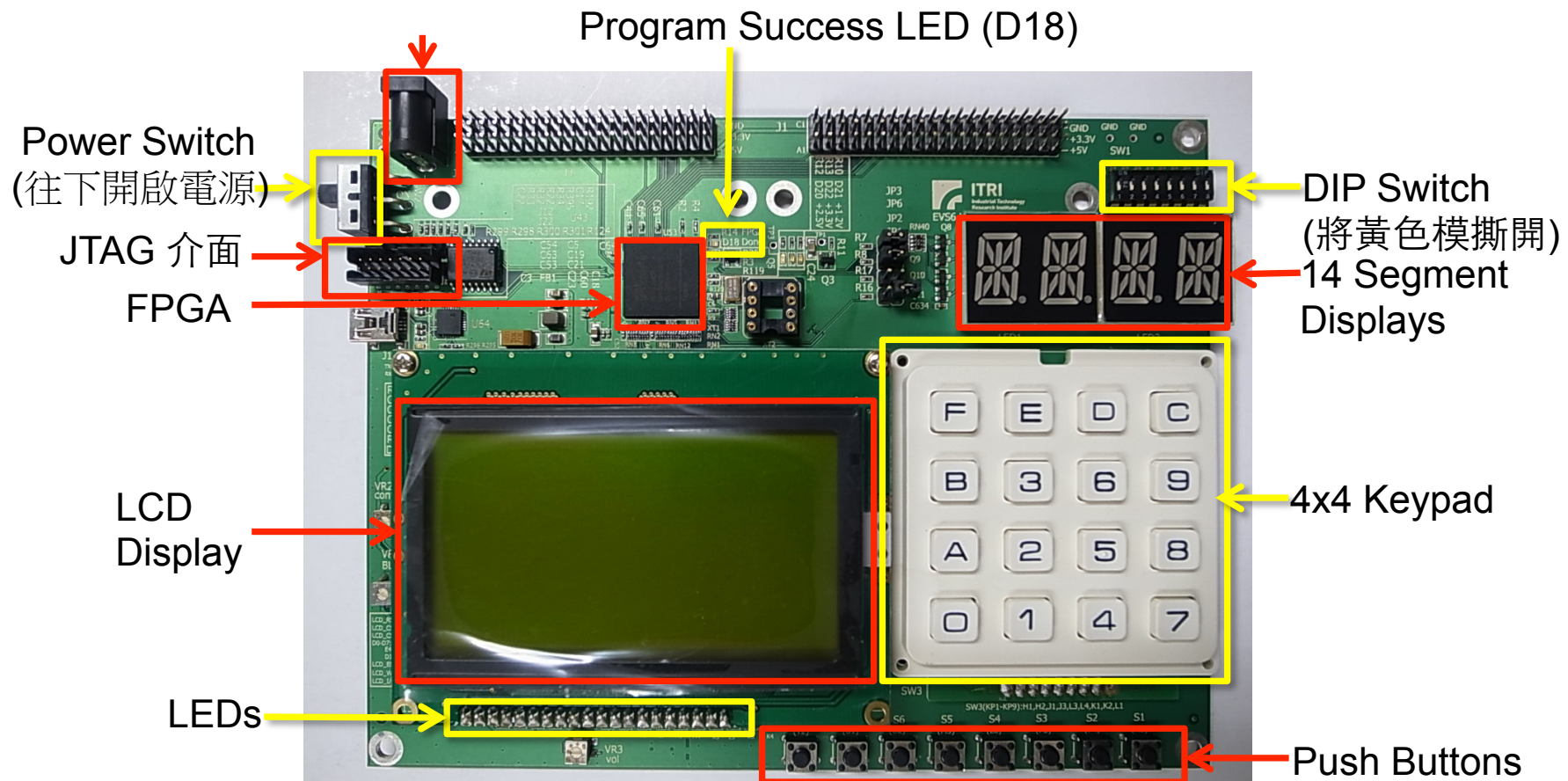
Important Notes

- **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

Xilinx XC6SLX16 CS324 Demo Board

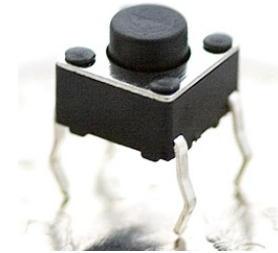
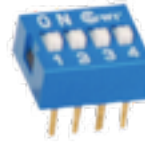


I/O Devices

- Clock generation: 40MHz oscillator

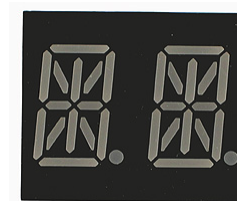
- Input devices

- 8 DIP switches
- 8 push buttons
- 1 4x4 keypad



- Output devices

- 16 user LEDs
- 4 14-segment display
- 1 128x64 LCD display
- Audio interface

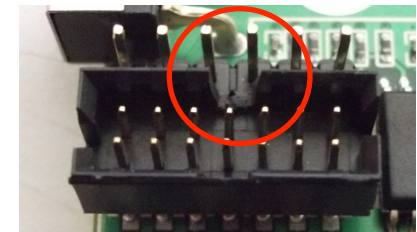
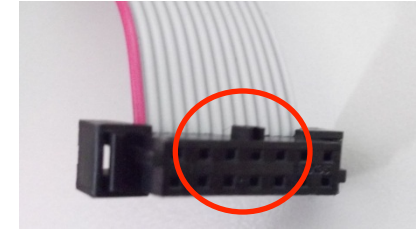


Input / Output Connections

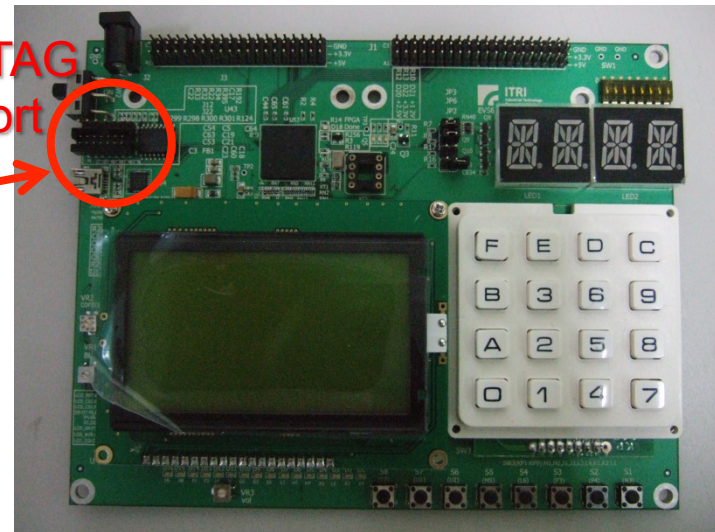
- LEDs are pre-wired such that
 - A **HIGH** signal from the FPGA to turn it on
- Push buttons and DP switches are pre-wired such that
 - The corresponding input is tied to **LOW** when a push button is pressed or a DIP switch is turned on.

USB / JTAG Download Cable

注意：凹槽要對到卡榫



JTAG
port



Test Your FPGA Board (1/3)

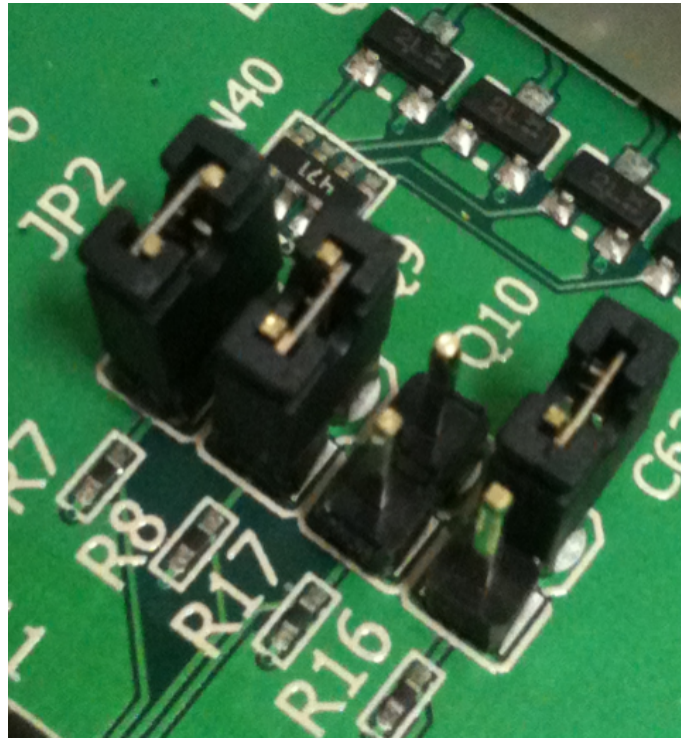
- Connect the demo board to PC and also the power supply

- 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

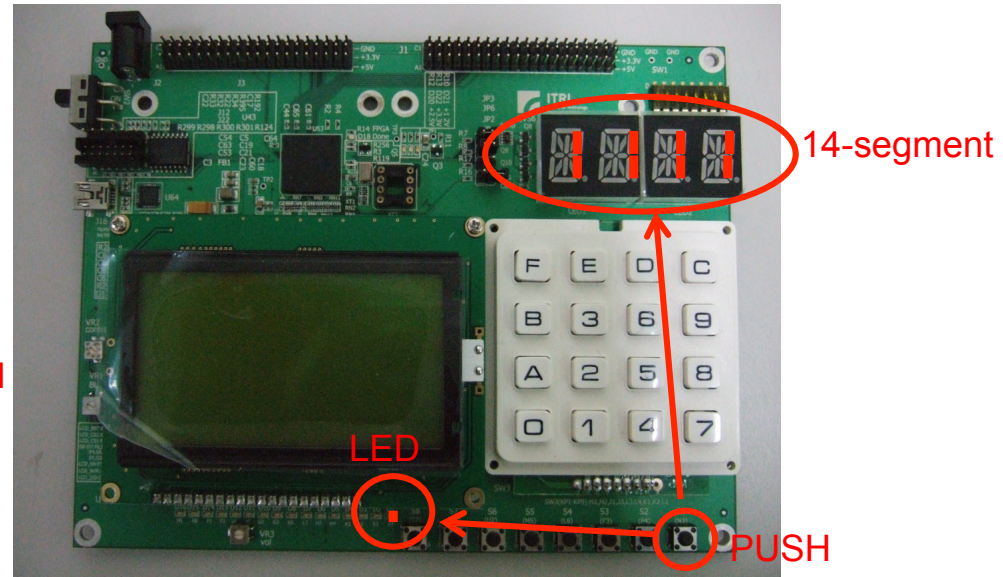
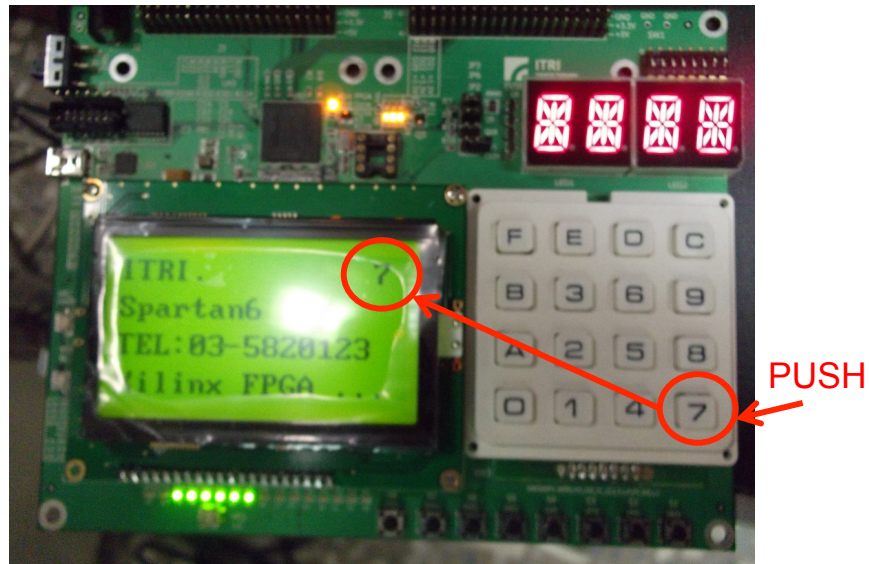


Test Your FPGA Board (2/3)

- Don't move or remove the jumpers



Test Your FPGA Board (3/3)



FPGA Emulation Using Xilinx ISE

Open New Project (2/2)

New Project Wizard ✕

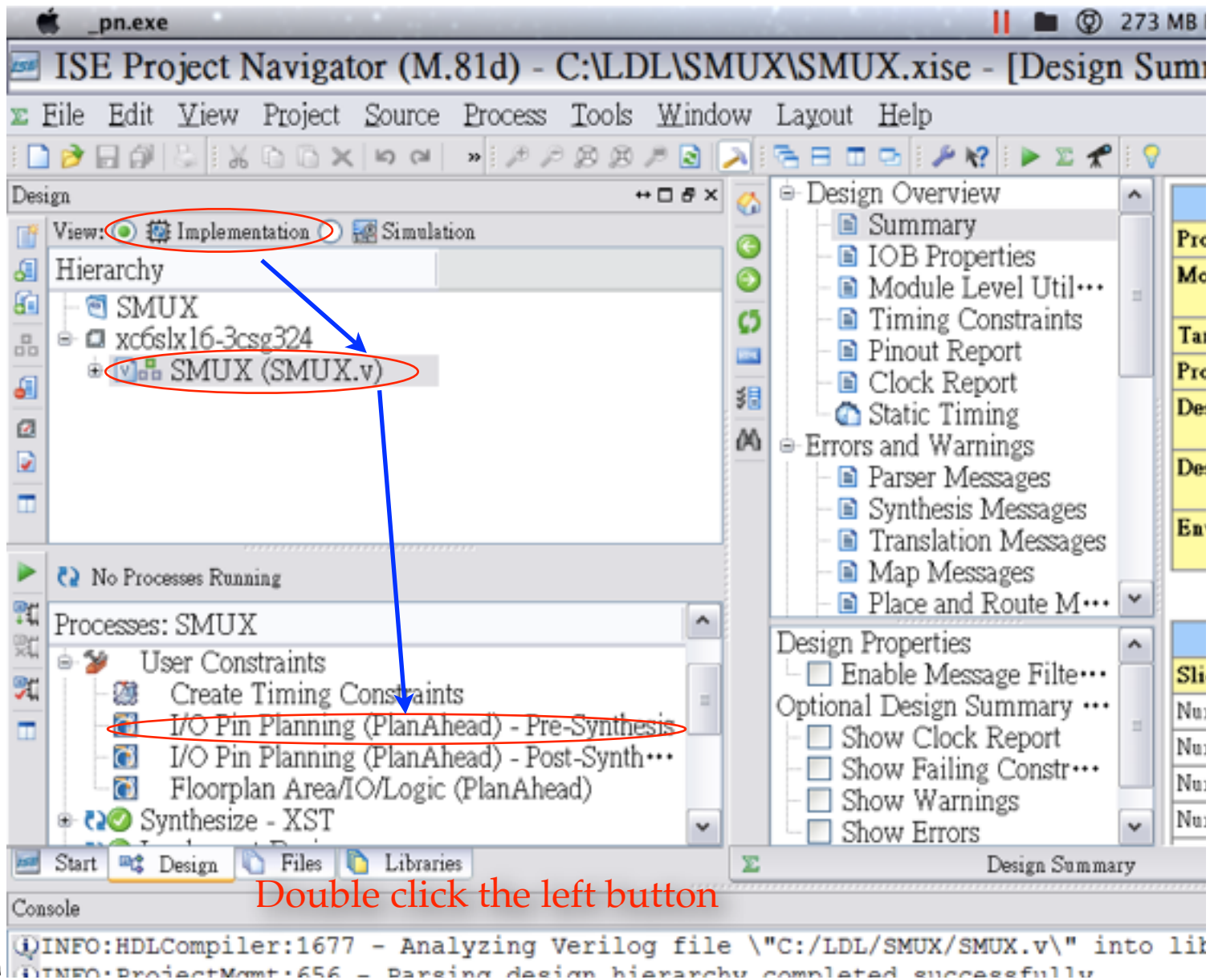
Project Settings
Specify device and project properties.

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	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93

[More Info](#) [< Back](#) [Next >](#) [Cancel](#)

Back to Implementation



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

File Edit View Project Source Process Tools Window Layout Help

Design Overview

- Summary
- IOB Properties
- Module Level Util...
- Timing Constraints
- Pinout Report
- Clock Report
- Static Timing

Errors and Warnings

- Parser Messages
- Synthesis Messages
- Translation Messages
- Map Messages
- Place and Route M...

Design Properties

- Enable Message Filte...

Optional Design Summary ...

- Show Clock Report
- Show Failing Constr...
- Show Warnings
- Show Errors

Design Summary

View: Implementation Simulation

Hierarchy

- SMUX
 - xc6slx16-3csg324
 - SMUX (SMUX.v)

No Processes Running

Processes: SMUX

- User Constraints
 - Create Timing Constraints
 - I/O Pin Planning (PlanAhead) - Pre-Synthesis**
 - I/O Pin Planning (PlanAhead) - Post-Synth...
 - Floorplan Area/IO/Logic (PlanAhead)
- Synthesize - XST

Start Design Files Libraries

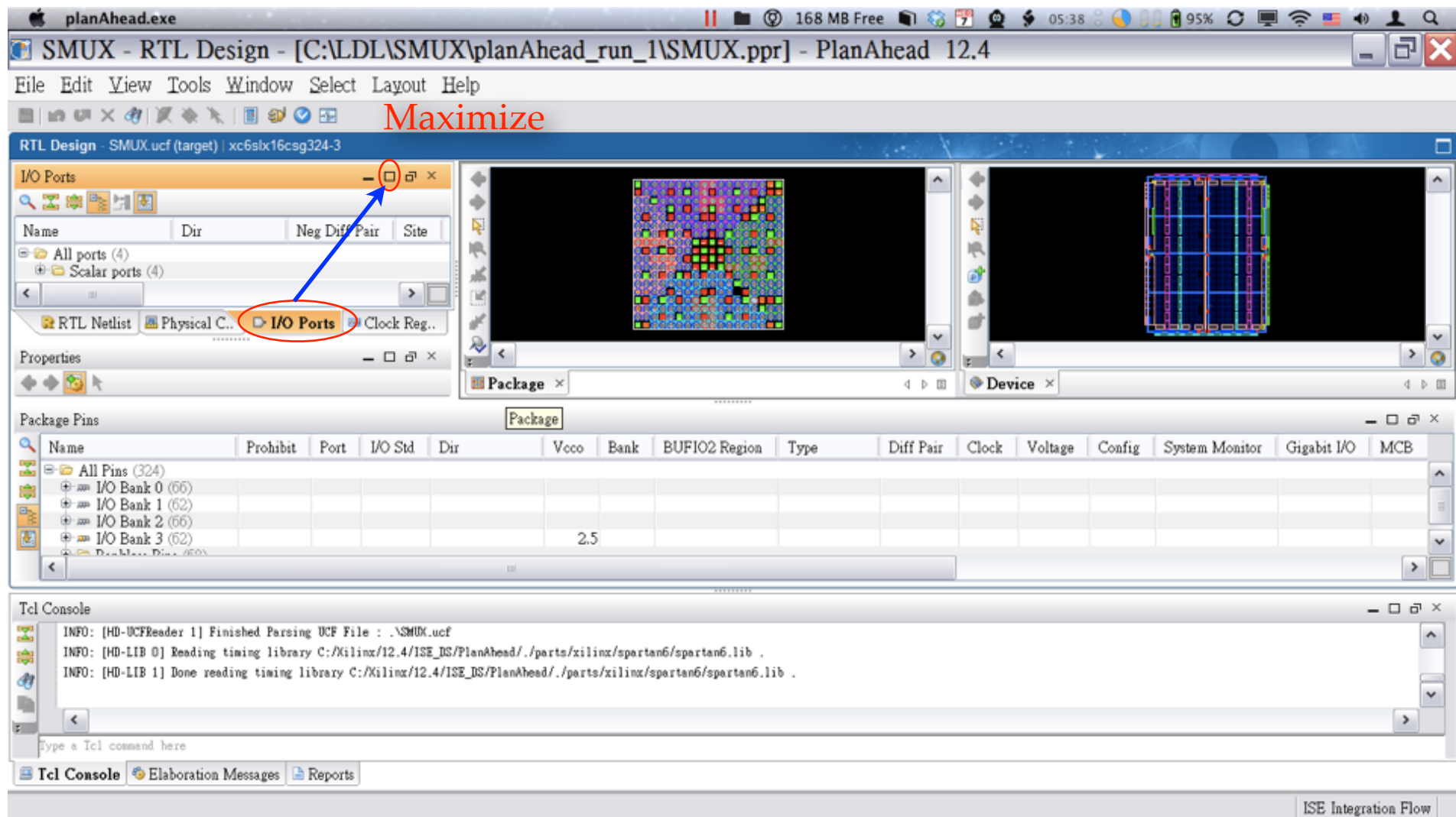
Console

INFO:HDLCompiler:1677 - Analyzing Verilog file \"C:/LDL/SMUX/SMUX.v\" into lib

INFO:ProjectMgmt:656 - Parsing design hierarchy completed successfully

Double click the left button

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



planAhead.exe 168 MB Free 05:38 95%

SMUX - RTL Design - [C:\LDL\SMUX\planAhead_run_1\SMUX.ppr] - PlanAhead 12.4

File Edit View Tools Window Select Layout Help

RTL Design - SMUX.ucf (target) | xc6slx16csg324-3

Maximize

I/O Ports

Name	Dir	Neg Diff Pair	Site
All ports (4)			
Scalar ports (4)			

RTL Netlist Physical C... I/O Ports Clock Reg..

Package Pins

Name	Prohibit	Port	I/O Std	Dir	Vcco	Bank	BUFIO2 Region	Type	Diff Pair	Clock	Voltage	Config	System Monitor	Gigabit I/O	MCB
All Pins (324)															
I/O Bank 0 (66)															
I/O Bank 1 (62)															
I/O Bank 2 (66)															
I/O Bank 3 (62)															
Bankless Pins (60)															

Tcl Console

```
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 .
```

Type a Tcl command here

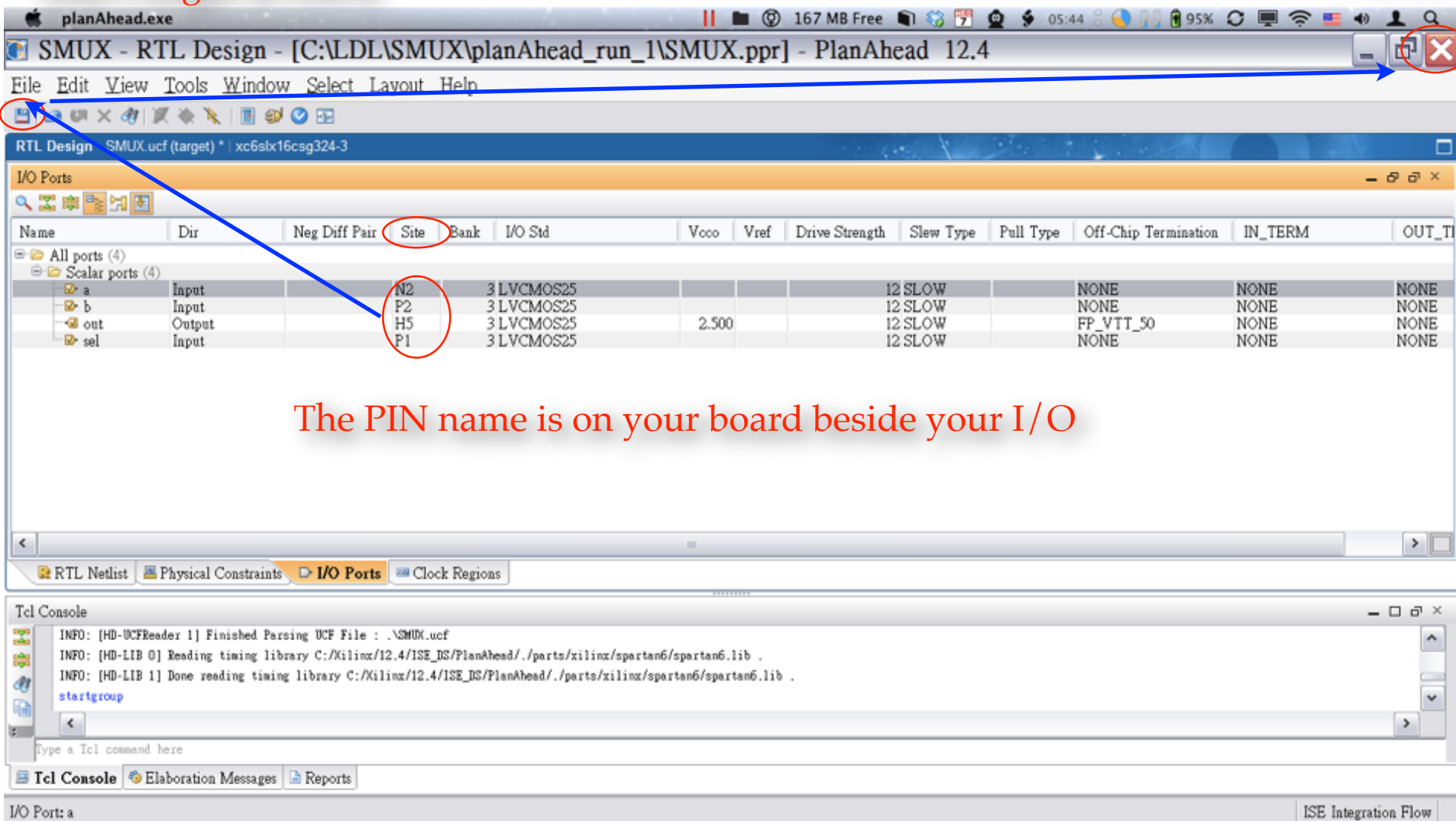
Tcl Console Elaboration Messages Reports

ISE Integration Flow

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

Don't forget to SAVE

close



planAhead.exe 167 MB Free 05:44 95%

SMUX - RTL Design - [C:\LDL\SMUX\planAhead_run_1\SMUX.ppr] - PlanAhead 12.4

File Edit View Tools Window Select Layout Help

RTL Design SMUX.ucf (target) * xc6slx16csg324-3

I/O Ports

Name	Dir	Neg Diff Pair	Site	Bank	I/O Std	Vcco	Vref	Drive Strength	Slew Type	Pull Type	Off-Chip Termination	IN_TERM	OUT_TERM
All ports (4)													
Scalar ports (4)													
a	Input		N2	3	LVC MOS25				12 SLOW		NONE	NONE	NONE
b	Input		P2	3	LVC MOS25				12 SLOW		NONE	NONE	NONE
out	Output		H5	3	LVC MOS25	2.500			12 SLOW		FP_VTT_50	NONE	NONE
sel	Input		P1	3	LVC MOS25				12 SLOW		NONE	NONE	NONE

RTL Netlist Physical Constraints I/O Ports Clock Regions

Tcl Console

```
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
```

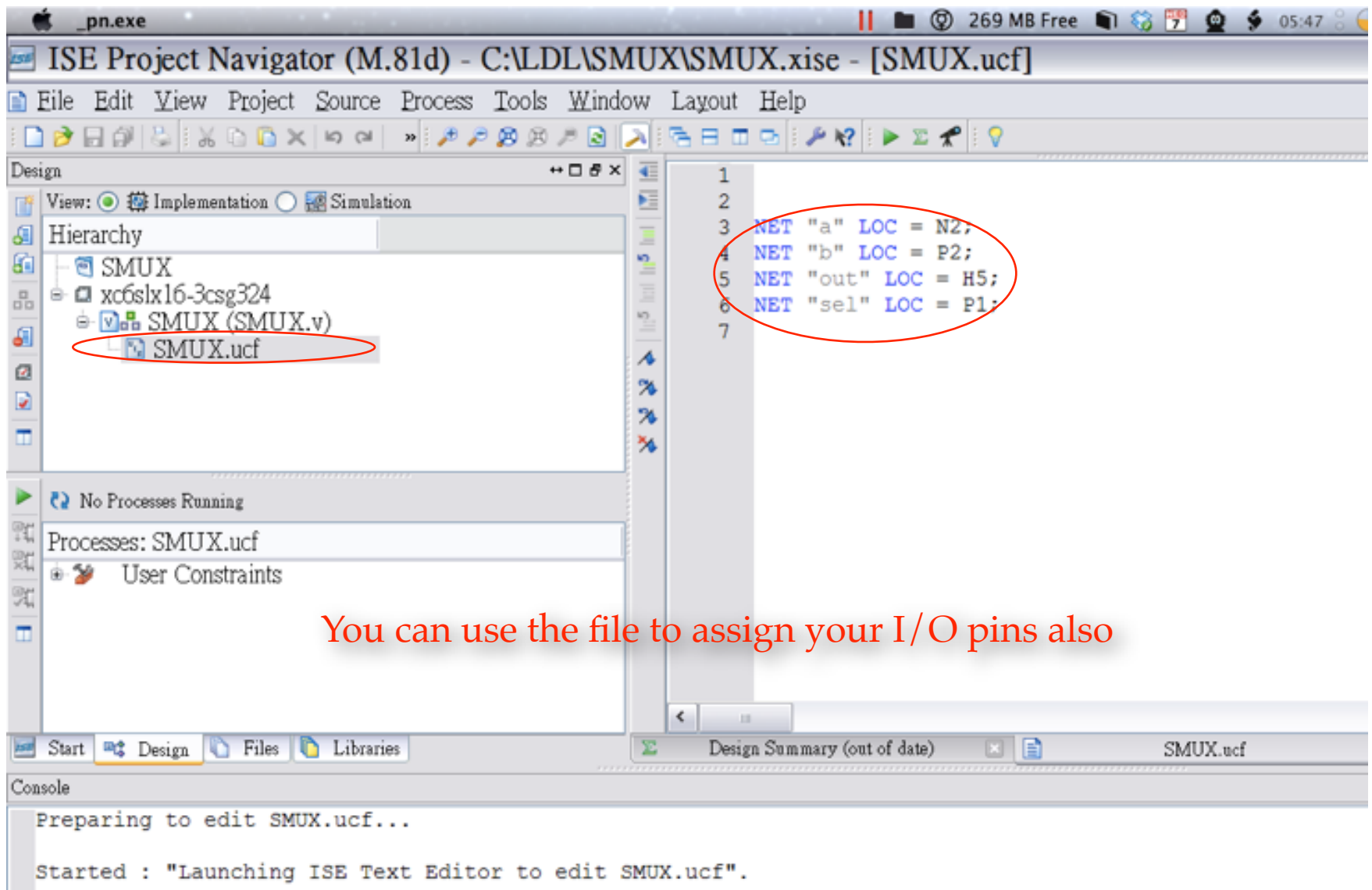
Type a Tcl command here

Tcl Console Elaboration Messages Reports

I/O Port: a ISE Integration Flow

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

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



The screenshot shows the ISE Project Navigator interface. The Design window displays the Hierarchy tree with the SMUX.ucf file selected. The SMUX.ucf file is circled in red. The SMUX.ucf file content is shown in the editor window, with the following lines circled in red:

```
1  
2  
3 NET "a" LOC = N2;  
4 NET "b" LOC = P2;  
5 NET "out" LOC = H5;  
6 NET "sel" LOC = P1;  
7
```

The console window shows the following output:

```
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

Synthesize and Implementation

The screenshot shows the Xilinx ISE Project Navigator interface. The main window displays the project hierarchy on the left and the logic editor on the right. The logic editor contains the following code:

```
1  
2  
3 NET "a" LOC = N2;  
4 NET "b" LOC = P2;  
5 NET "out" LOC = H5;  
6 NET "sel" LOC = P1;  
7
```

The "Processes: SMUX" list on the left shows the following steps:

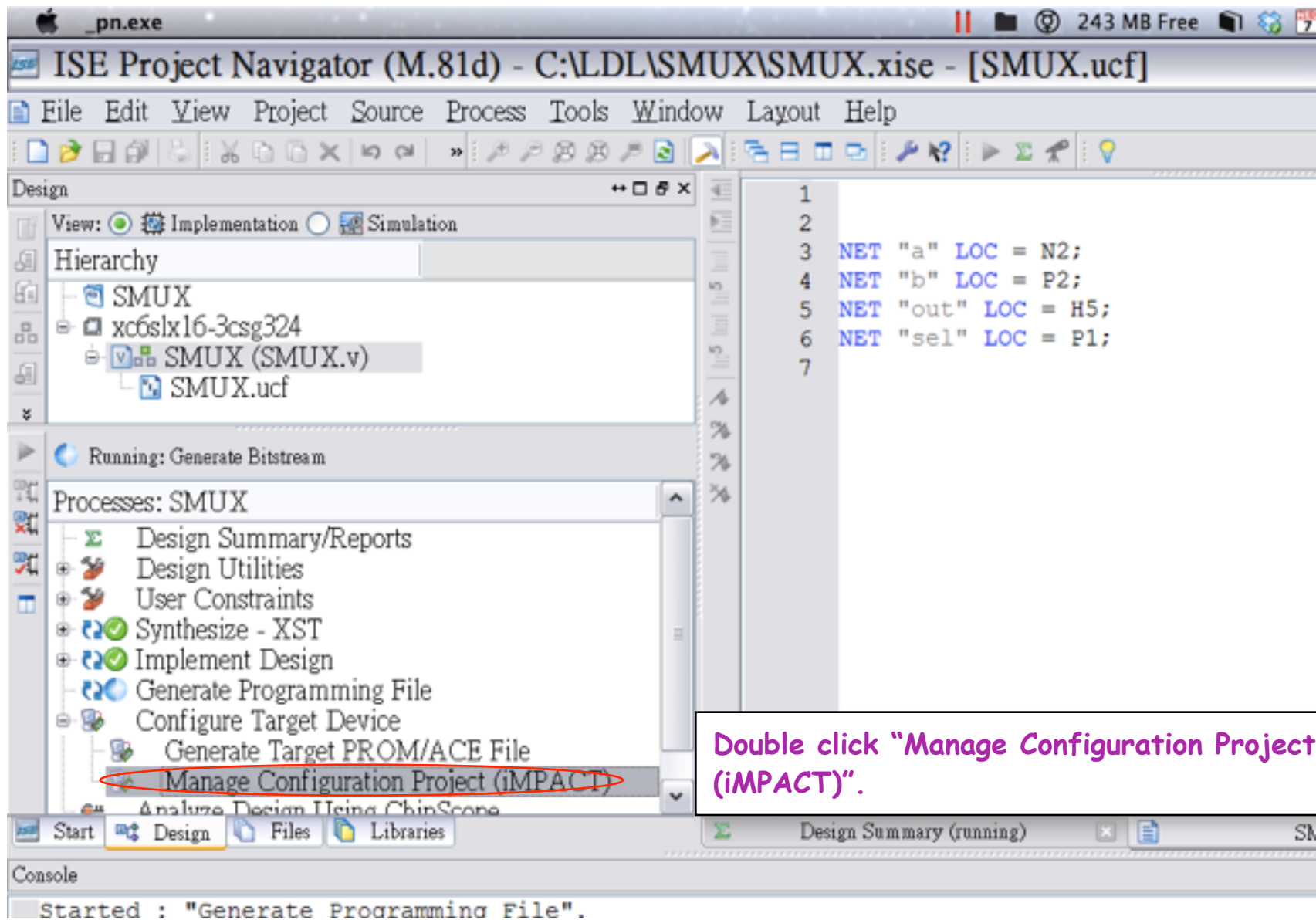
- Design Summary/Reports
- Design Utilities
- User Constraints
- Synthesize - XST** (highlighted with a red oval)
- Implement Design** (highlighted with a red oval)
- Generate Programming File
- Configure Target Device
- Generate Target PROM/ACE File
- Manage Configuration Project (iMPACT)
- Analyze Design Using ChipScope

Two callout boxes provide instructions:

1. Double click "Synthesize" for logic synthesis.
2. Double click "Implement Design" for implementation.

The console at the bottom shows the message: "Started : "Map"."

Download Bit File for Emulation



The screenshot shows the Xilinx ISE Project Navigator interface. The main window displays the project hierarchy on the left and the configuration editor on the right. The configuration editor shows the following code:

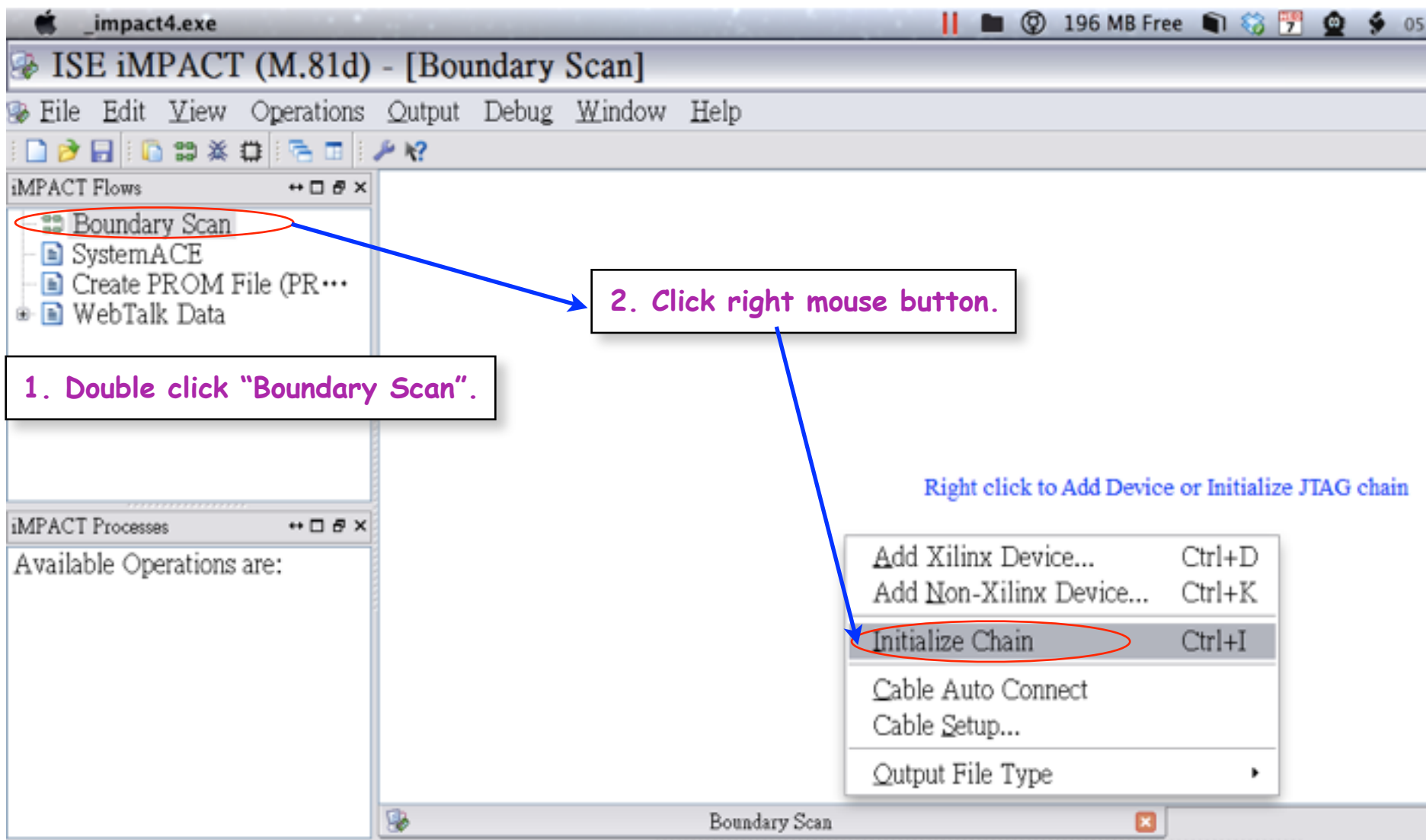
```
1  
2  
3 NET "a" LOC = N2;  
4 NET "b" LOC = P2;  
5 NET "out" LOC = H5;  
6 NET "sel" LOC = P1;  
7
```

The 'Processes: SMUX' pane on the left shows the following steps:

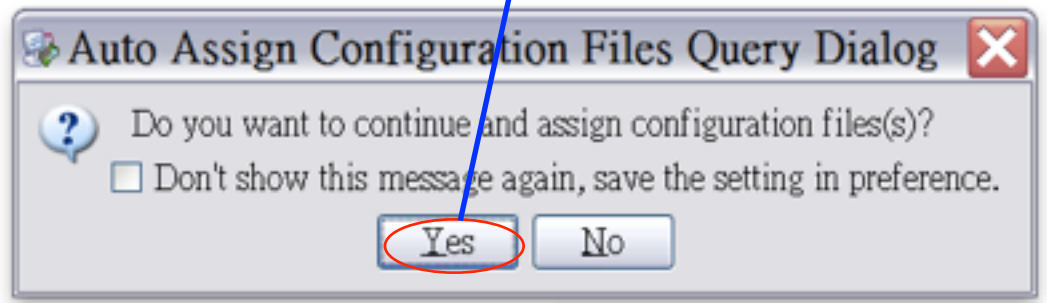
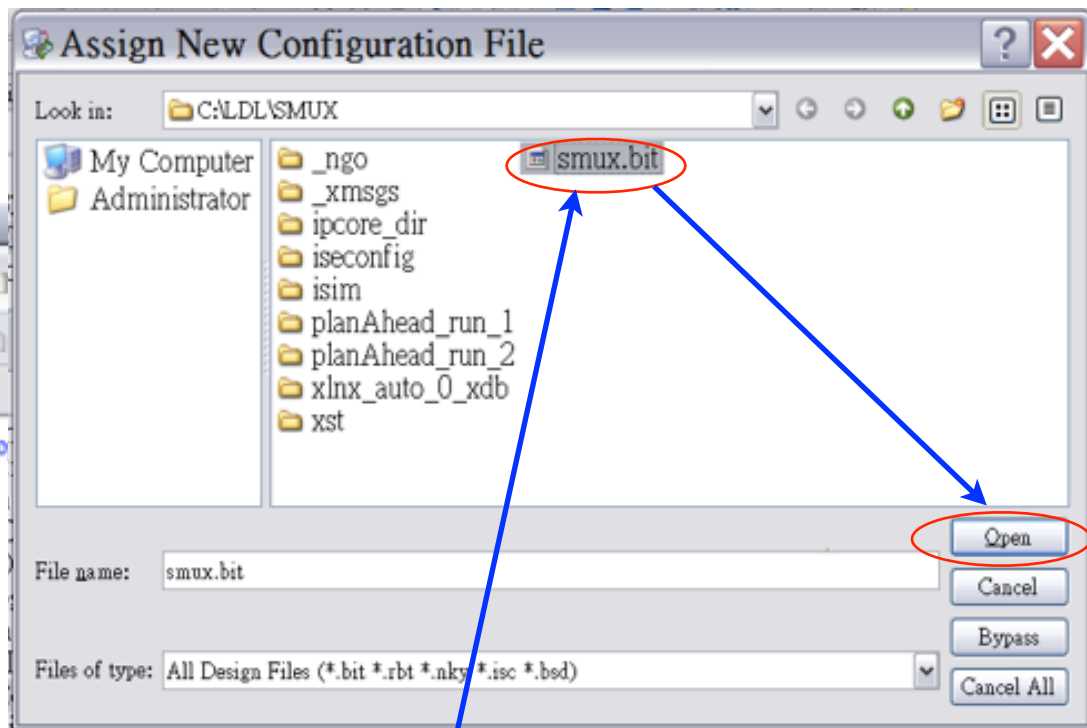
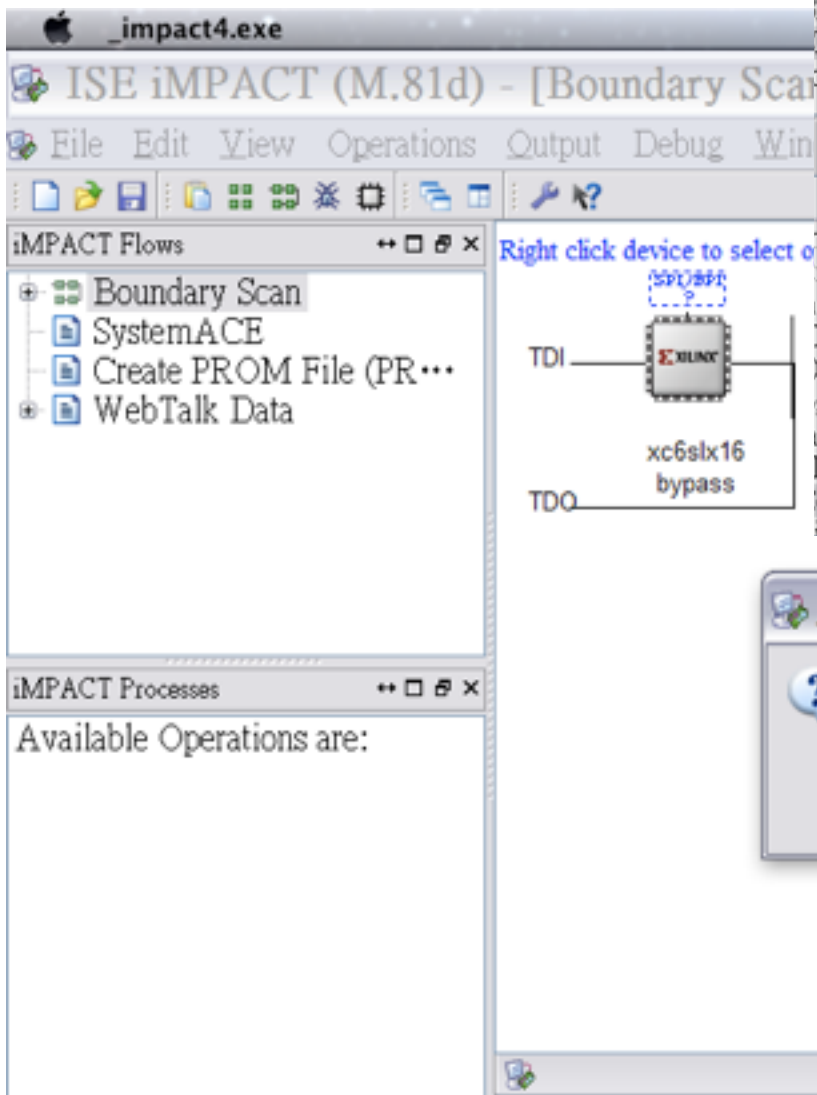
- Design Summary/Reports
- Design Utilities
- User Constraints
- Synthesize - XST
- Implement Design
- Generate Programming File
- Configure Target Device
- Generate Target PROM/ACE File
- Manage Configuration Project (iMPACT)** (highlighted with a red circle)
- Analyze Design Using ChipScope

A callout box with a black border and purple text points to the 'Manage Configuration Project (iMPACT)' step, stating: "Double click 'Manage Configuration Project (iMPACT)'".

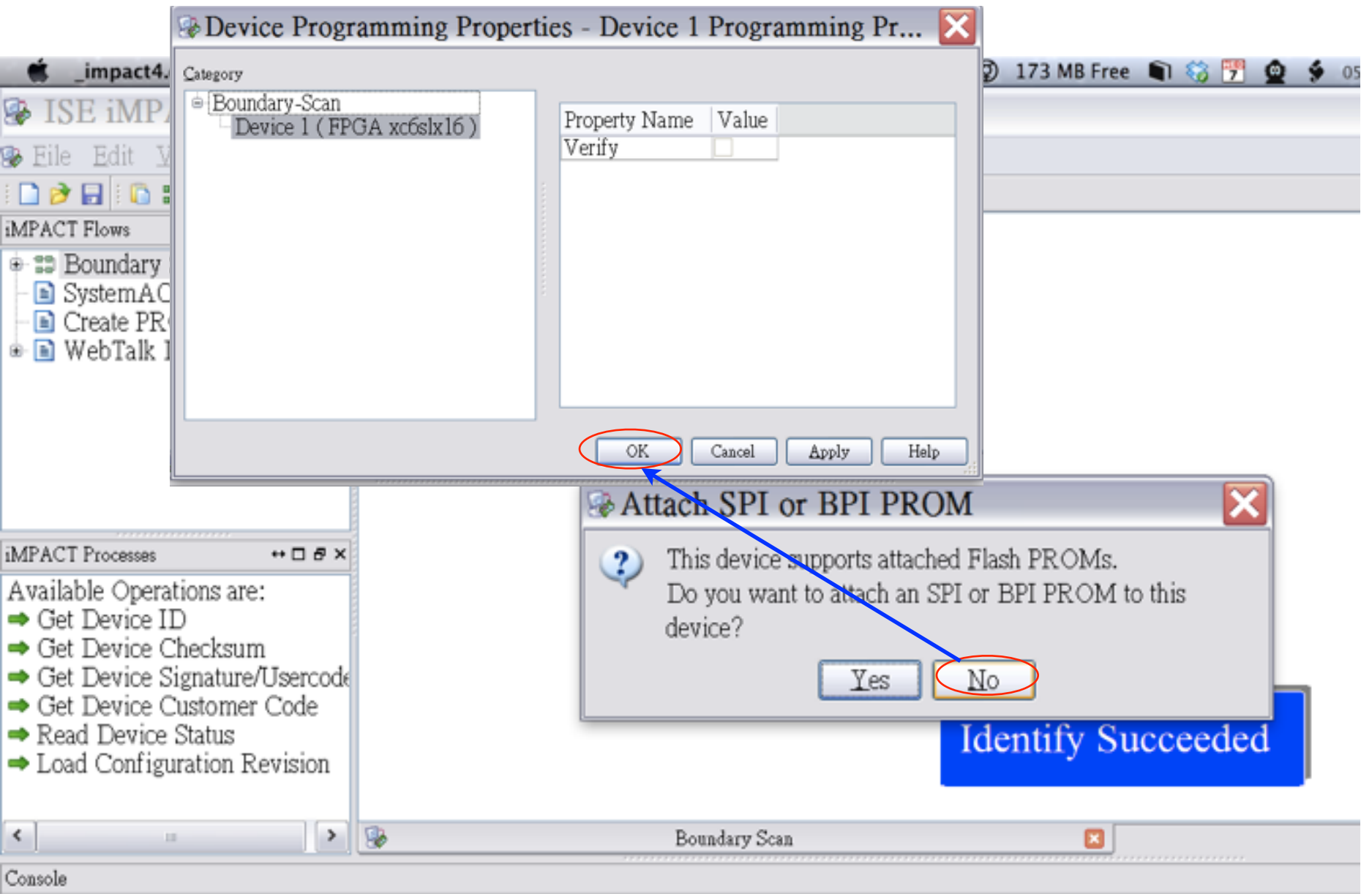
The console at the bottom shows the message: "Started : 'Generate Programming File'".



Assign Configuration File



Identify Succeeded



The screenshot displays the Xilinx ISE software interface. The main window is titled "Device Programming Properties - Device 1 Programming Pr...". The "Category" tree on the left shows "Boundary-Scan" expanded to "Device 1 (FPGA xc6slx16)". The "Property Name" and "Value" table contains one entry: "Verify" with an unchecked checkbox.

Property Name	Value
Verify	<input type="checkbox"/>

At the bottom of the "Device Programming Properties" dialog, the "OK" button is circled in red. A blue arrow points from this "OK" button to the "No" button in the "Attach SPI or BPI PROM" dialog box below it.

The "Attach SPI or BPI PROM" dialog box contains the following text:

?

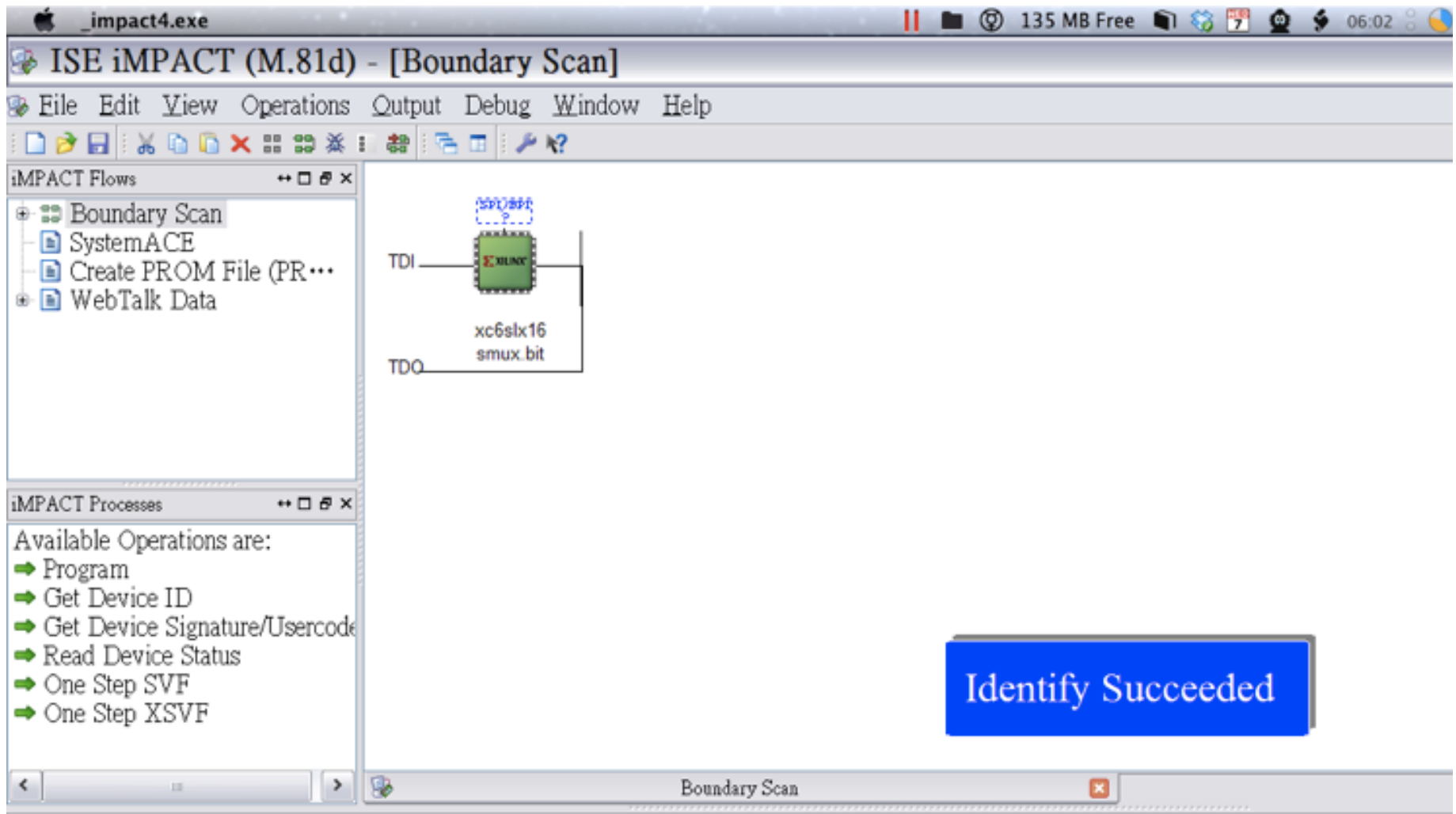
This device supports attached Flash PROMs.
Do you want to attach an SPI or BPI PROM to this device?

Yes No

The "No" button in this dialog is also circled in red. A blue banner at the bottom right of the dialog area contains the text "Identify Succeeded".

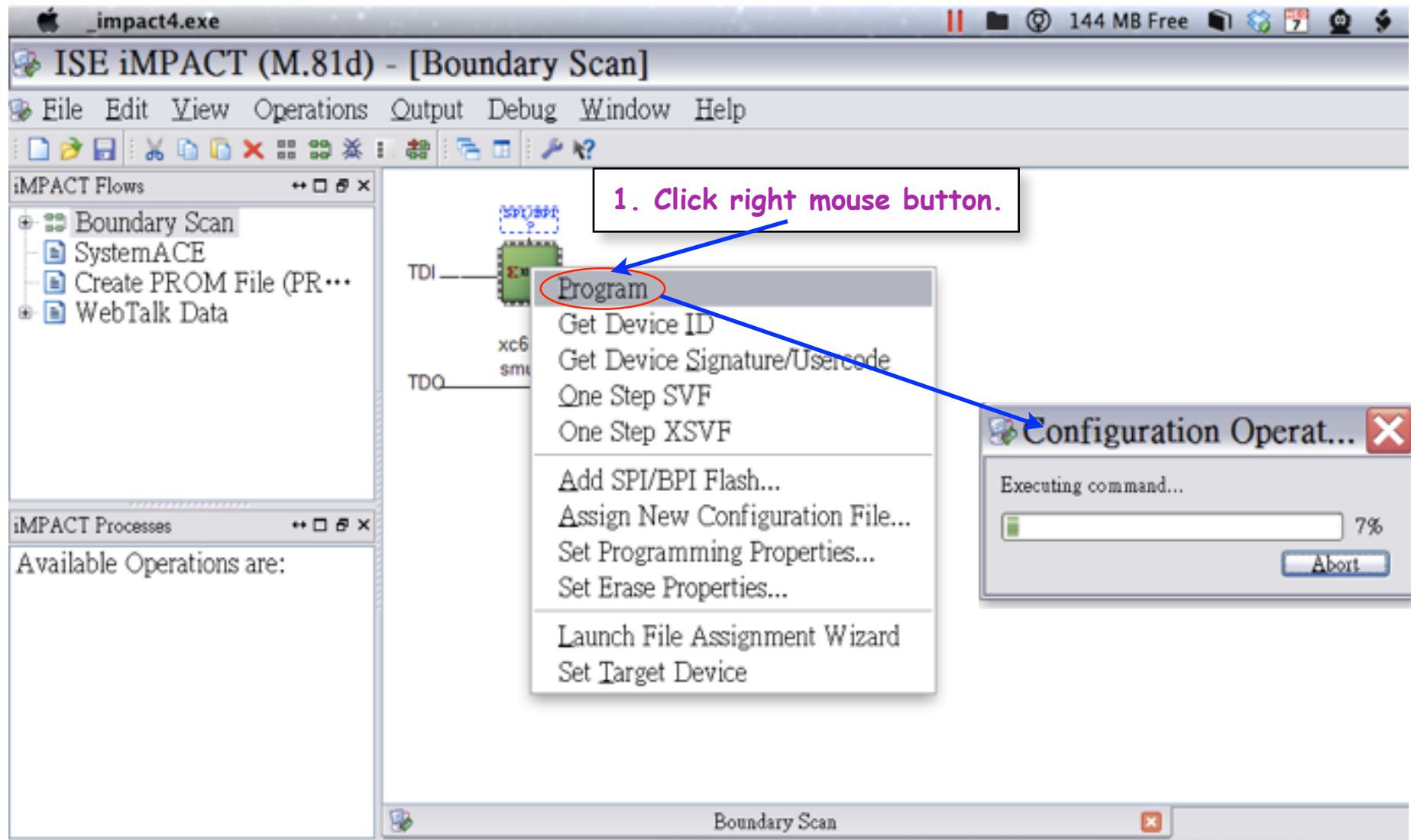
The background interface includes the "IMPACT Flows" tree with "Boundary", "SystemAC", "Create PR...", and "WebTalk I" visible. The "IMPACT Processes" window lists available operations such as "Get Device ID", "Get Device Checksum", "Get Device Signature/Usercode", "Get Device Customer Code", "Read Device Status", and "Load Configuration Revision". The status bar at the bottom shows "Boundary Scan" and a "Console" window.

Identify



The screenshot displays the ISE iMPACT (M.81d) - [Boundary Scan] window. The title bar shows the application name and system tray icons including a battery level indicator, network status, and the time 06:02. The menu bar includes File, Edit, View, Operations, Output, Debug, Window, and Help. The toolbar contains various icons for file operations and debugging. The main workspace shows a diagram of an xc6slx16 smux.bit device with TDI and TDO connections. A blue box with the text "Identify Succeeded" is overlaid on the right side of the workspace. The iMPACT Flows pane on the left lists Boundary Scan, SystemACE, Create PROM File (PR...), and WebTalk Data. The iMPACT Processes pane on the left lists available operations: Program, Get Device ID, Get Device Signature/Usercode, Read Device Status, One Step SVF, and One Step XSVF. The status bar at the bottom shows the current operation as Boundary Scan.

Program the Device (1/2)



The screenshot shows the ISE iMPACT (M.81d) - [Boundary Scan] application window. The main workspace displays a circuit diagram with a device component labeled "xc6 smt". A right-click context menu is open over this component, with the "Program" option highlighted. A blue arrow points from the "Program" option to a "Configuration Operat..." dialog box. The dialog box shows "Executing command..." with a progress bar at 7% and an "Abort" button. The "iMPACT Flows" pane on the left lists "Boundary Scan", "SystemACE", "Create PROM File (PR...", and "WebTalk Data". The "iMPACT Processes" pane at the bottom left shows "Available Operations are:". The status bar at the bottom indicates "Boundary Scan".

1. Click right mouse button.

Program

- Get Device ID
- Get Device Signature/Usercode
- One Step SVF
- One Step XSVF
- Add SPI/BPI Flash...
- Assign New Configuration File...
- Set Programming Properties...
- Set Erase Properties...
- Launch File Assignment Wizard
- Set Target Device

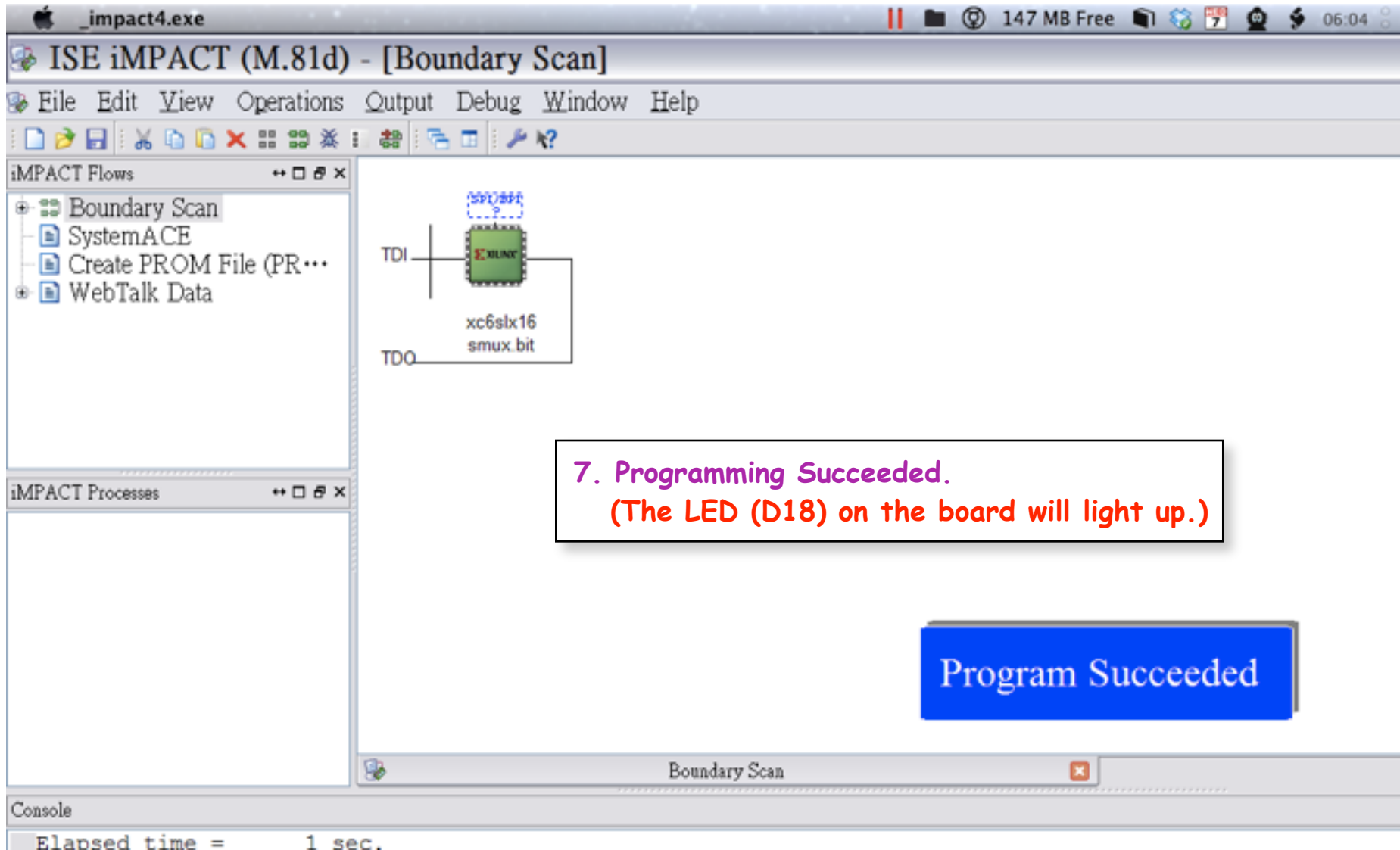
Configuration Operat... X

Executing command...

7%

Abort

Program the Device (2/2)



The screenshot shows the ISE iMPACT software interface. The title bar indicates the application is running on a Mac and is titled "ISE iMPACT (M.81d) - [Boundary Scan]". The menu bar includes File, Edit, View, Operations, Output, Debug, Window, and Help. The main workspace displays a diagram of an "xc6slx16 smux.bit" device with TDI and TDO pins. A "Boundary Scan" window is open, showing a list of flows: Boundary Scan, SystemACE, Create PROM File (PR...), and WebTalk Data. The "iMPACT Processes" window is empty. A blue button labeled "Program Succeeded" is visible. The console at the bottom shows "Elapsed time = 1 sec." and a "Boundary Scan" window with a close button.

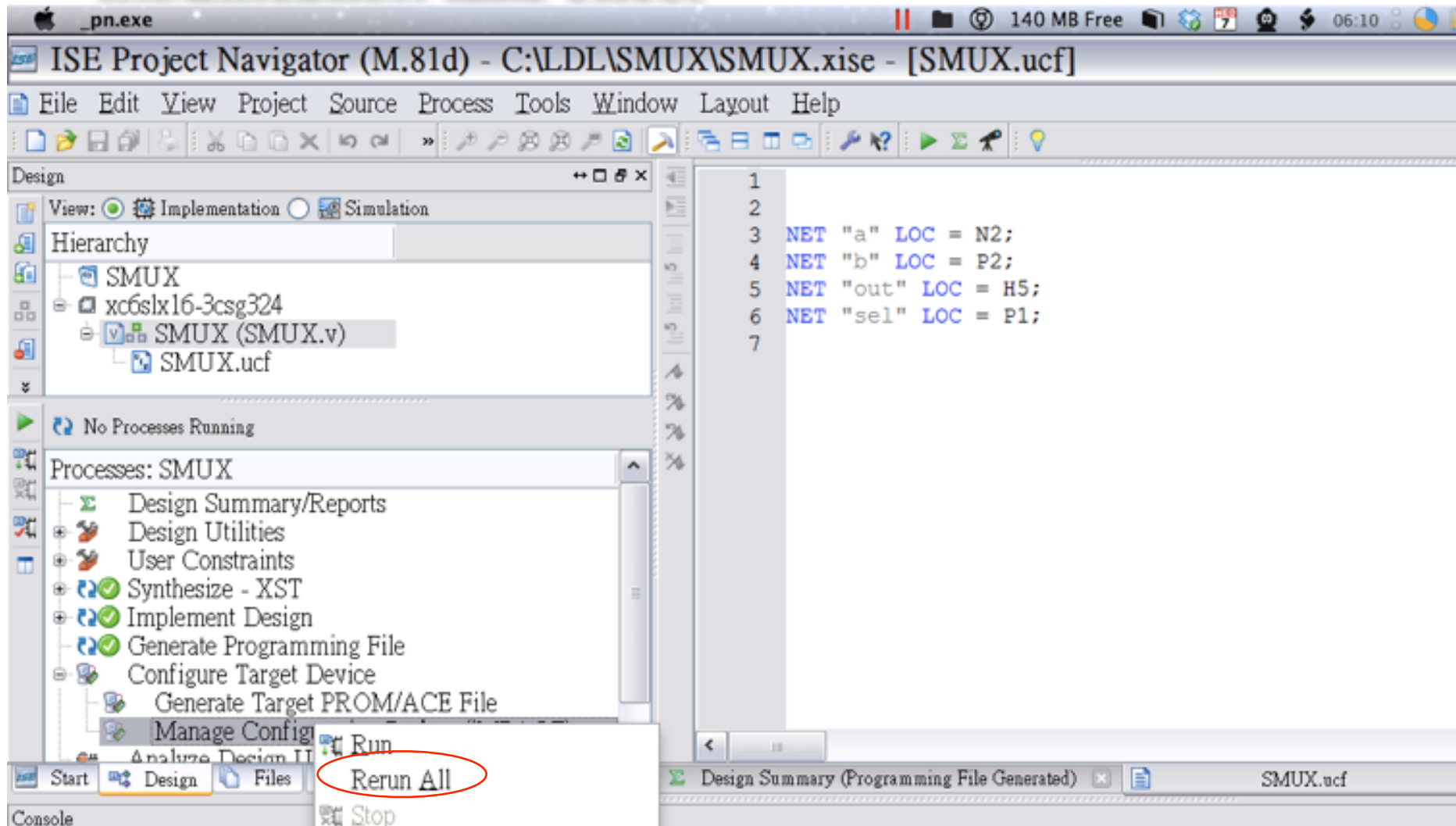
7. Programming Succeeded.
(The LED (D18) on the board will light up.)

Program Succeeded

Elapsed time = 1 sec.

Some Notes (1/2)

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



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 function  
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
```

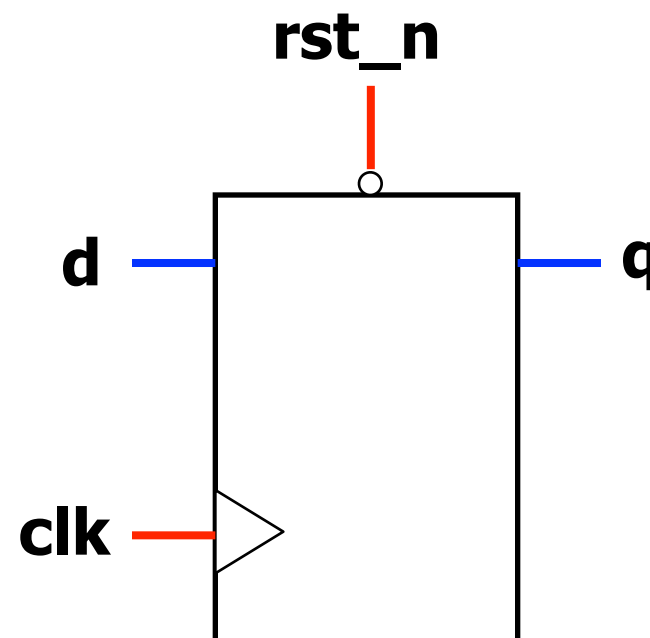
```
always @*
```

```
    out = (sel) ? a : b ;
```

```
endmodule
```

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
```

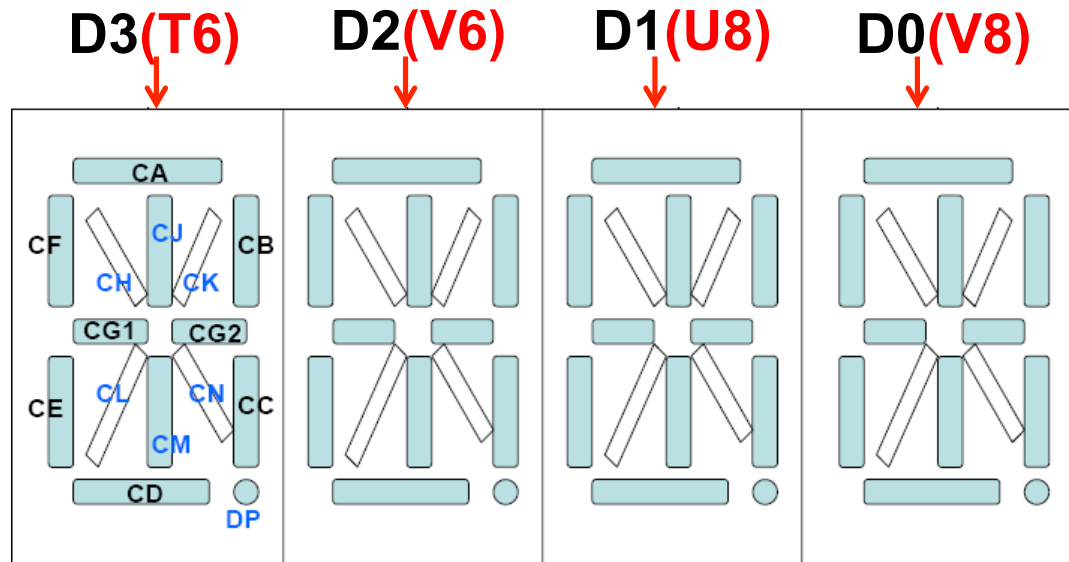


Declaration Syntax of Verilog Wire/ Registers

- `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)

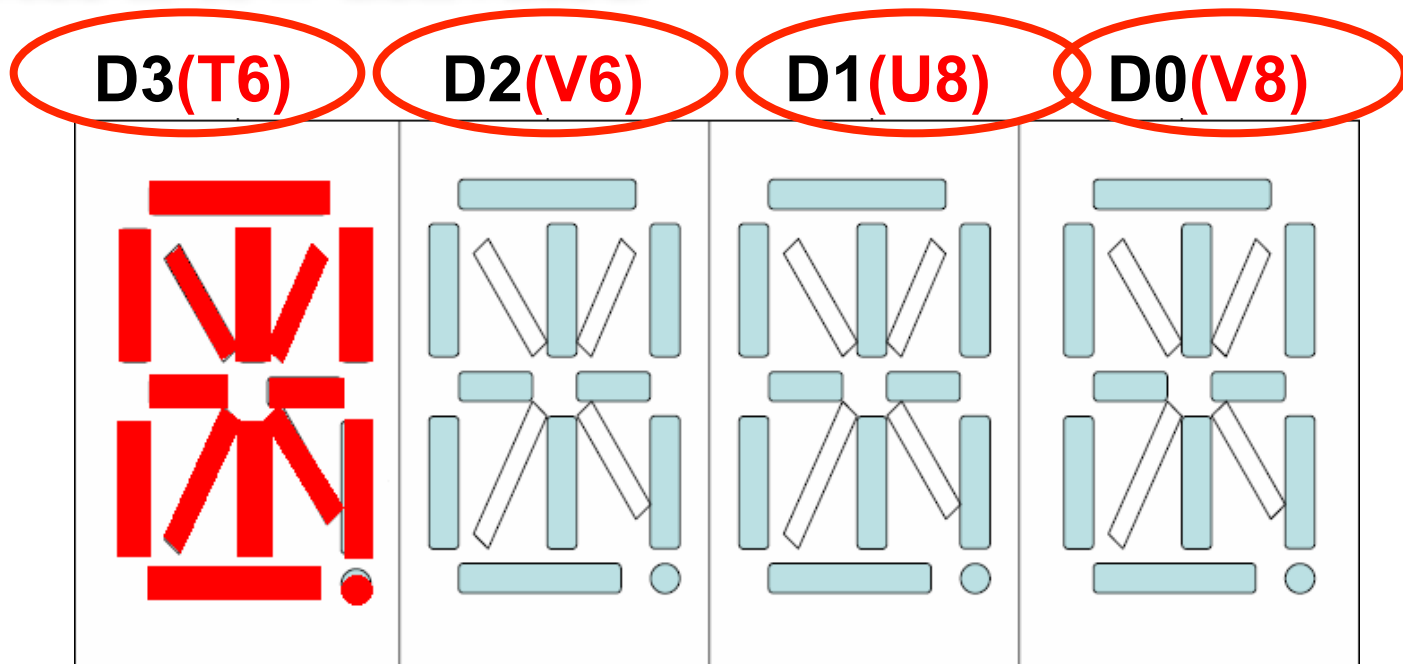
- The cathodes of similar segments on all four displays are connected to the same FPGA pin



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)

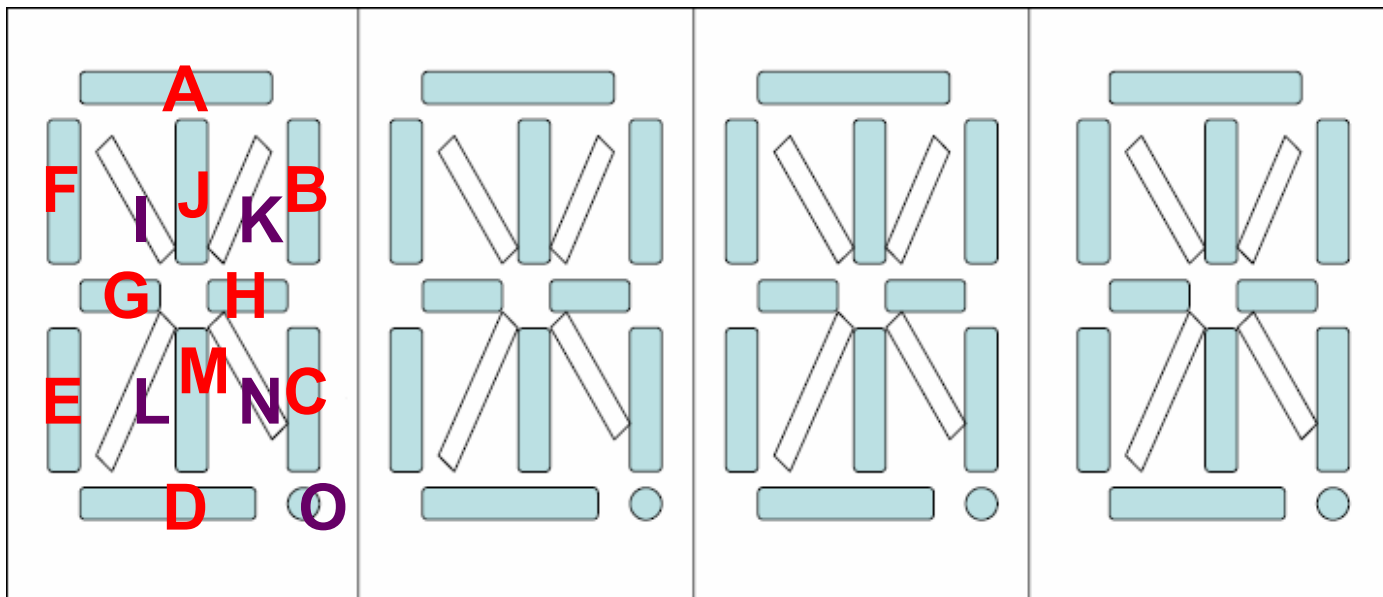
- 15 pins to control each 14-segment display
 - Including the point
- 4 pins to choose which 14-seg to display
- Device is low activated



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

