FPGA Emulation

黃元豪
Yuan-Hao Huang

國立清華大學電機工程學系
Department of Electrical Engineering
National Tsing-Hua University

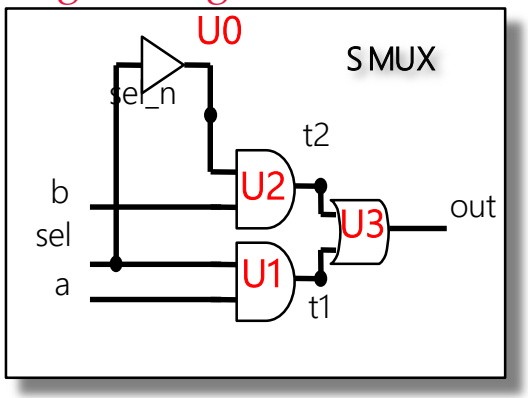
Verilog HDL Utilization Scenario I

- Simulation and verification of logic circuits on PC.

Specification

2-to-1 Multiplexer

Logic Design

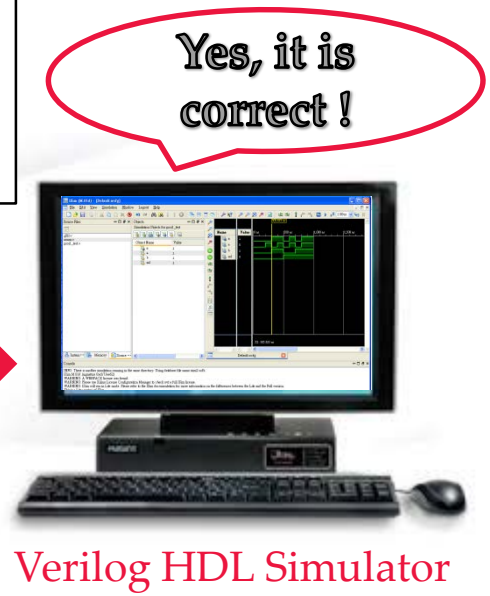


Verilog HDL Coding

```
module SMUX(out, a, b, sel);

output out;
input a,b,sel;
wire sel_n,t1,t2;

not U0(sel_n,sel);
and U1(t1,a,sel);
and U2(t2,b,sel_n);
or  U3(out,t1,t2);

endmodule
```

Test Pattern

a, b, sel
000
001
010
…
111



Verilog HDL Simulator

- Imaging what happens when the circuit is as complex as a CPU or MP3 player processor.

Verilog HDL Utilization Scenario II

- Design and Implementation of logic circuits in FPGA (IC)

Specification

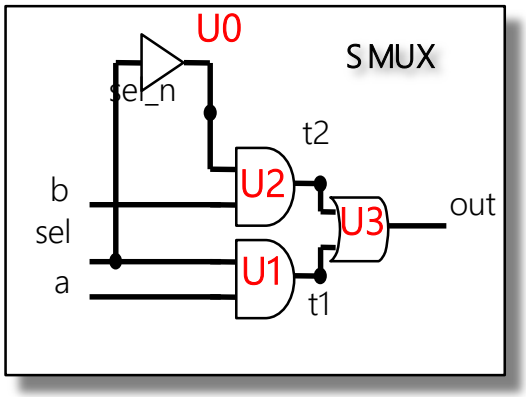
2-to-1 Multiplexer

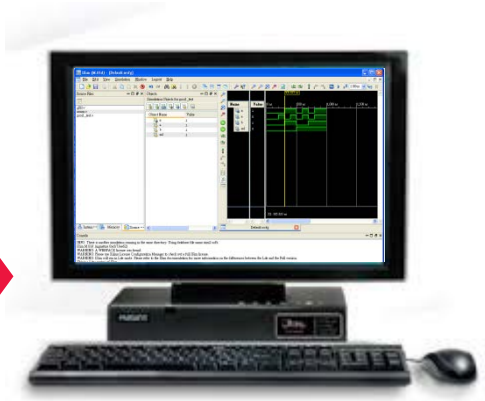
Verilog HDL Coding

```
module SMUX(out, a, b, sel);

output out;
input a,b,sel;
wire sel_n,t1,t2;

 assign out = sel ? a:b;

endmodule
```

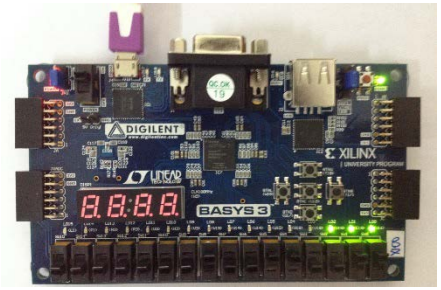
Verilog HDL Logic **Synthesizer**

Synthesized Logic Circuits





FPGA **Implement Design** and **Programming**



Now, it can work !

Basys3 FPGA Board

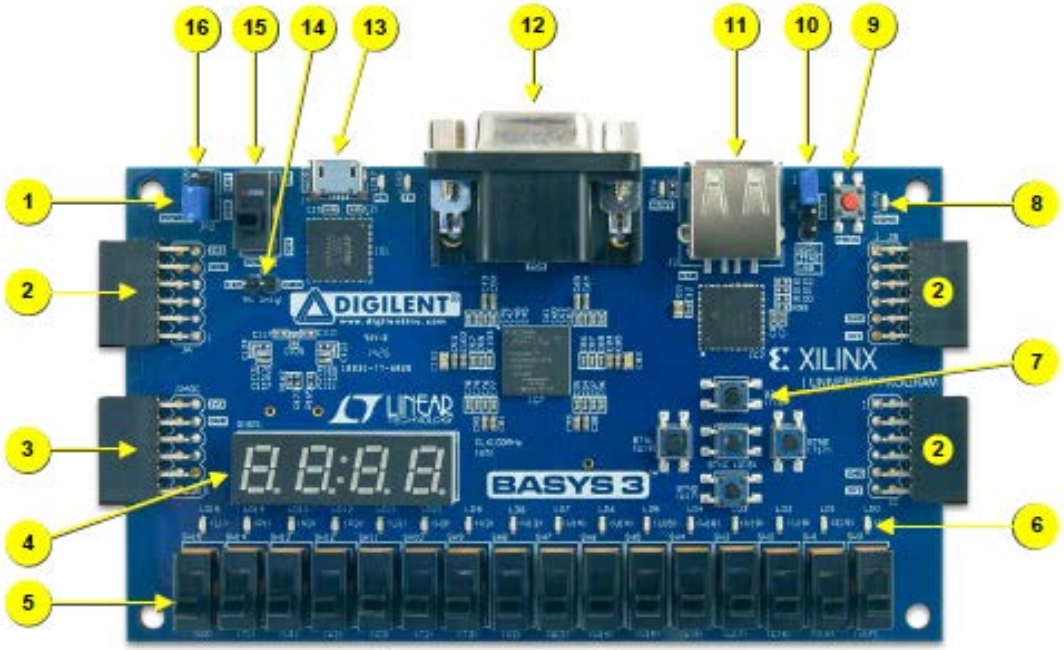
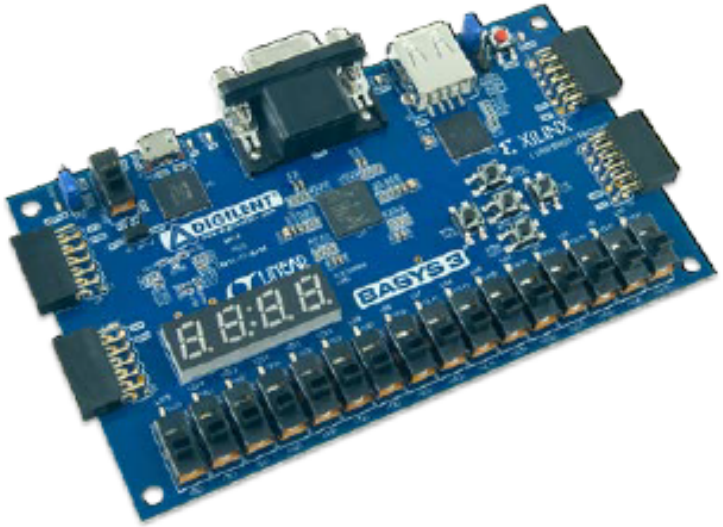


Figure 1. Basys3 FPGA board with callouts.

Callout	Component Description	Callout	Component Description
1	Power good LED	9	FPGA configuration reset button
2	Pmod connector(s)	10	Programming mode jumper
3	Analog signal Pmod connector (XADC)	11	USB host connector
4	Four digit 7-segment display	12	VGA connector
5	Slide switches (16)	13	Shared UART/ JTAG USB port
6	LEDs (16)	14	External power connector
7	Pushbuttons (5)	15	Power Switch
8	FPGA programming done LED	16	Power Select Jumper

Table 1. Basys3 Callouts and component descriptions.

EVS6 FPGA Board Specification



- 33,280 logic cells in 5200 slices (each slice contains four 6-input LUTs and 8 flip-flops)
- 1,800 Kbits of fast block RAM
- Five clock management tiles, each with a phase-locked loop (PLL)
- 90 DSP slices
- Internal clock speeds exceeding 450MHz
- On-chip analog-to-digital converter (XADC)

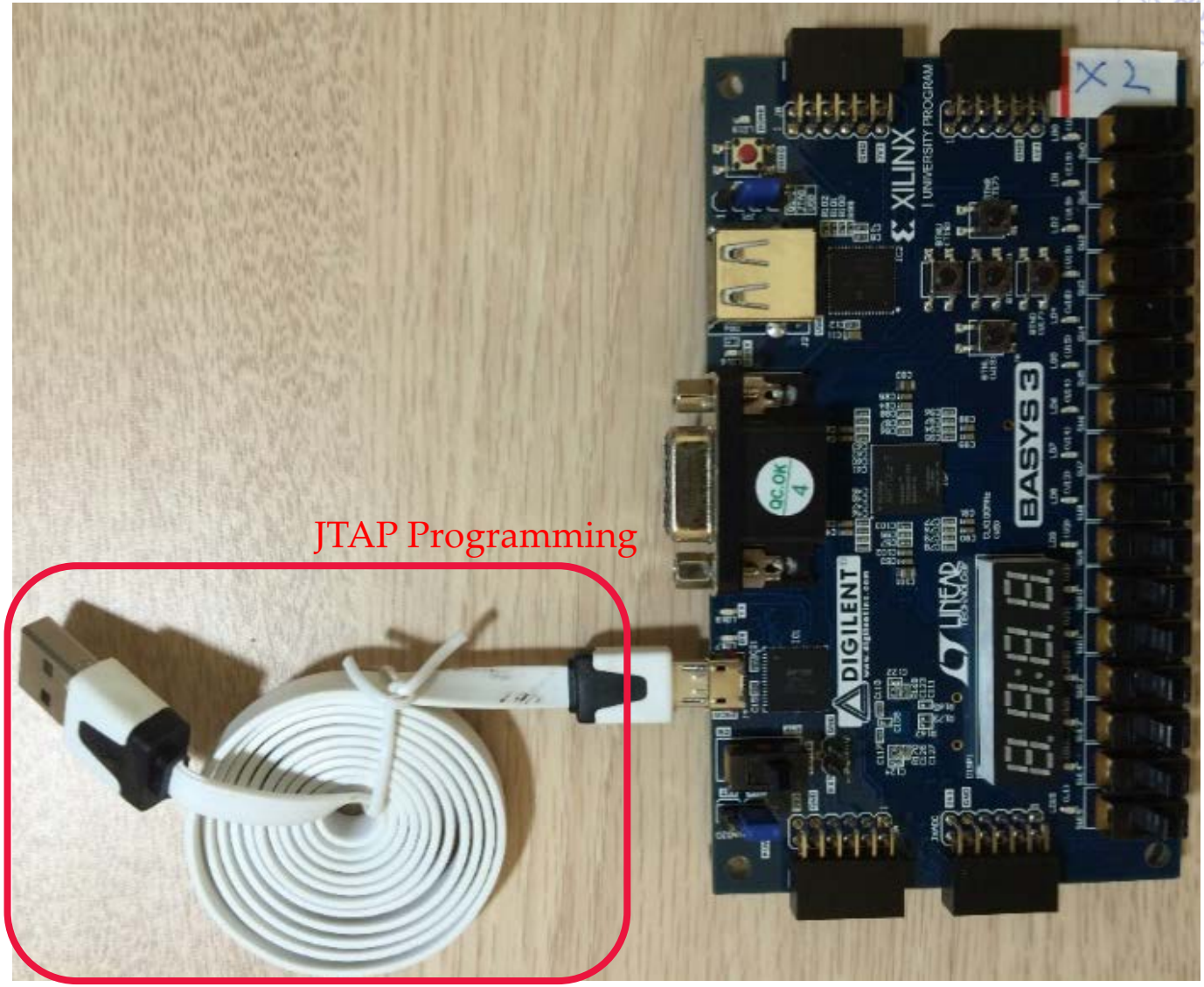
- 16 user switches
- 4-digit 7-segment display
- 12-bit VGA output
- Digilent USB-JTAG port for FPGA programming and communication

- 16 user LEDs
- Three Pmod connectors
- USB-UART Bridge
- USB HID Host for mice, keyboards and memory sticks

- 5 user pushbuttons
- Pmod for XADC signals
- Serial Flash

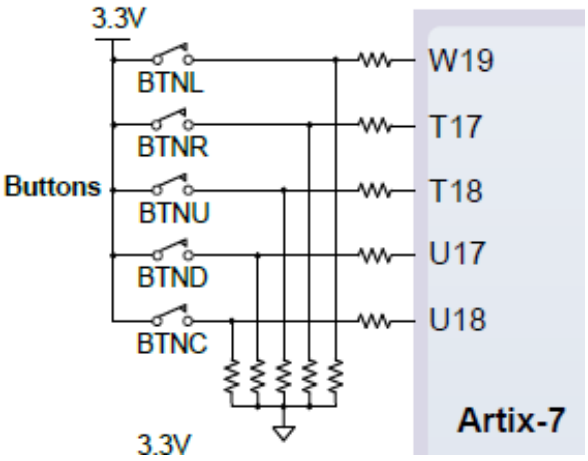
The complete FPGA board document (Basys3_rm.pdf) is on the course website.

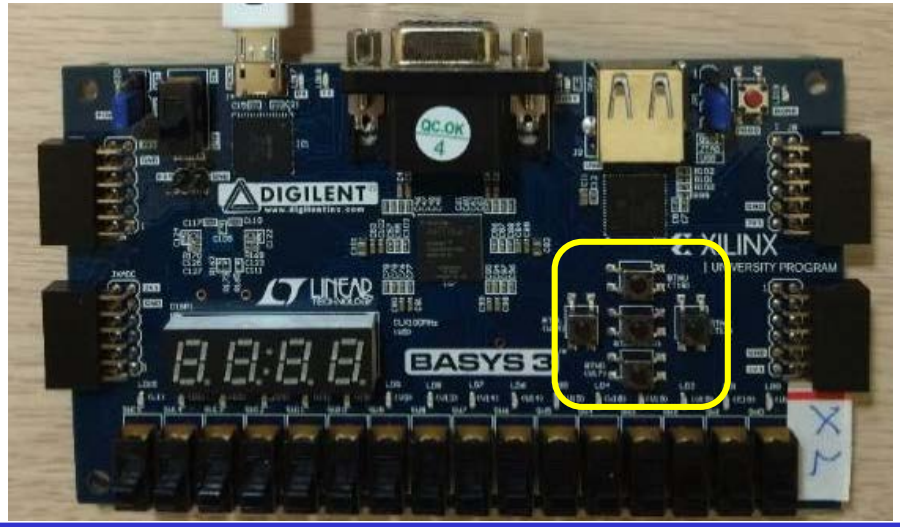
JTAG Programming



Specific I/Os – Push Buttons

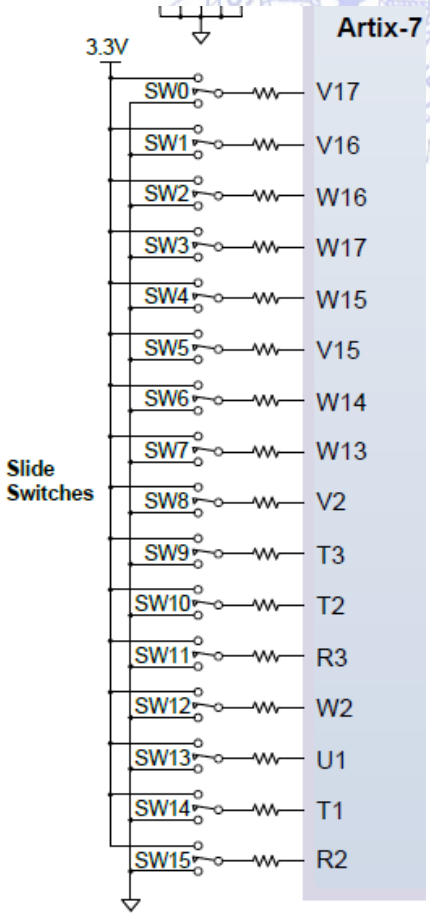
```
set_property PACKAGE_PIN W19 [get_ports {BTNL}]
set_property IOSTANDARD LVCMOS33 [get_ports {BTNL}]
set_property PACKAGE_PIN T17 [get_ports {BTNR}]
set_property IOSTANDARD LVCMOS33 [get_ports {BTNR}]
set_property PACKAGE_PIN T18 [get_ports {BTNU}]
set_property IOSTANDARD LVCMOS33 [get_ports {BTNU}]
set_property PACKAGE_PIN U17 [get_ports {BTND}]
set_property IOSTANDARD LVCMOS33 [get_ports {BTND}]
set_property PACKAGE_PIN U18 [get_ports {BTNC}]
set_property IOSTANDARD LVCMOS33 [get_ports {BTNC}]
```



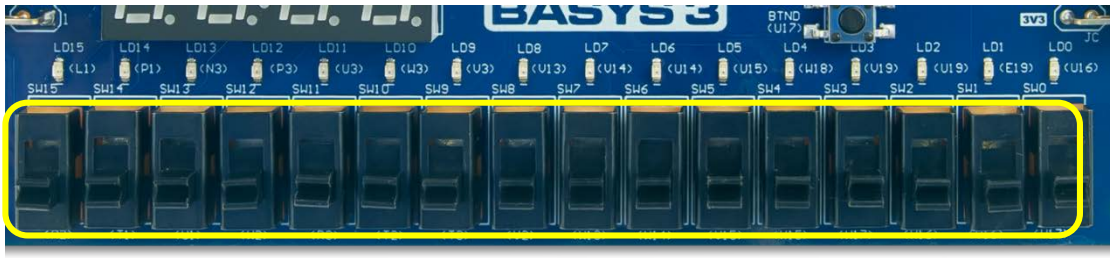


Specific I/Os – DIP Switches

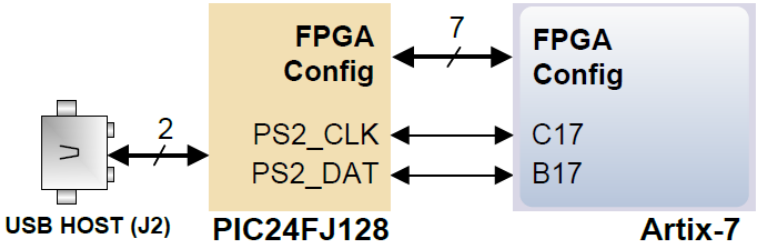
```
set_property PACKAGE_PIN V17 [get_ports {SW0}]
set_property IOSTANDARD LVCMOS33 [get_ports {SW0}]
set_property PACKAGE_PIN V16 [get_ports {SW1}]
set_property IOSTANDARD LVCMOS33 [get_ports {SW1}]
set_property PACKAGE_PIN W16 [get_ports {SW2}]
set_property IOSTANDARD LVCMOS33 [get_ports {SW2}]
set_property PACKAGE_PIN W17 [get_ports {SW3}]
set_property IOSTANDARD LVCMOS33 [get_ports {SW3}]
set_property PACKAGE_PIN W15 [get_ports {SW4}]
set_property IOSTANDARD LVCMOS33 [get_ports {SW4}]
```

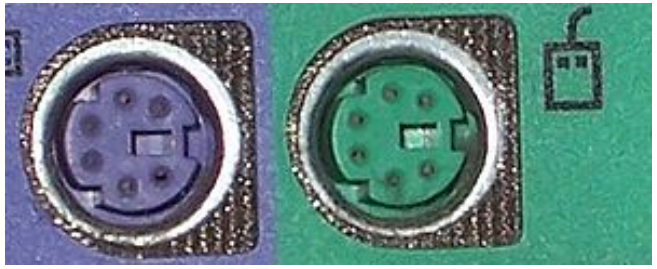


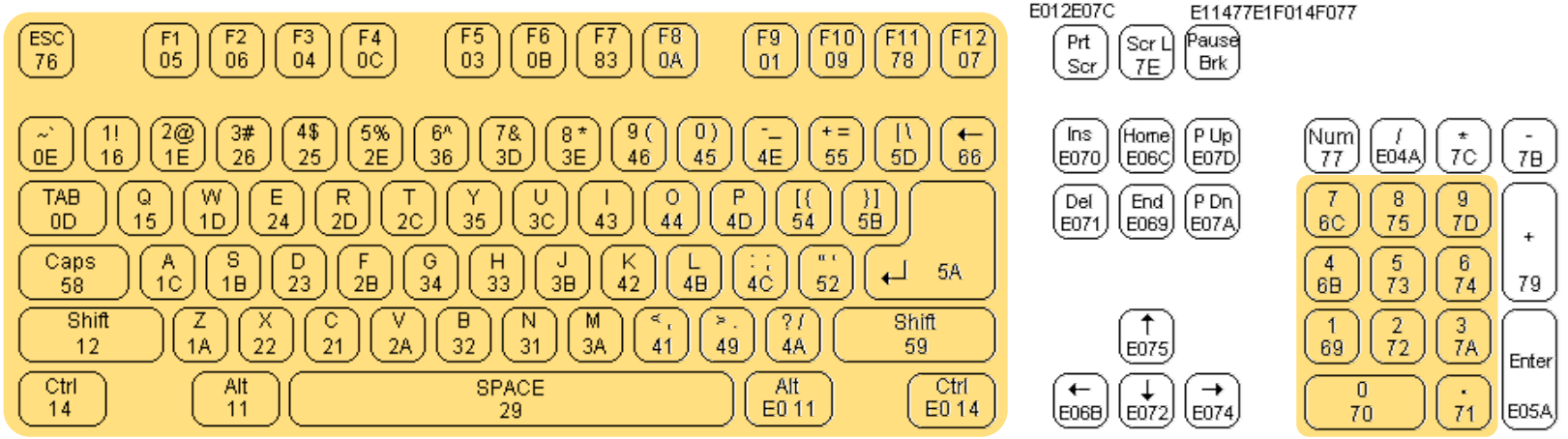

```
set_property PACKAGE_PIN T1 [get_ports {SW14}]
set_property IOSTANDARD LVCMOS33 [get_ports {SW14}]
set_property PACKAGE_PIN R2 [get_ports {SW15}]
set_property IOSTANDARD LVCMOS33 [get_ports {SW15}]
```



Keyboard

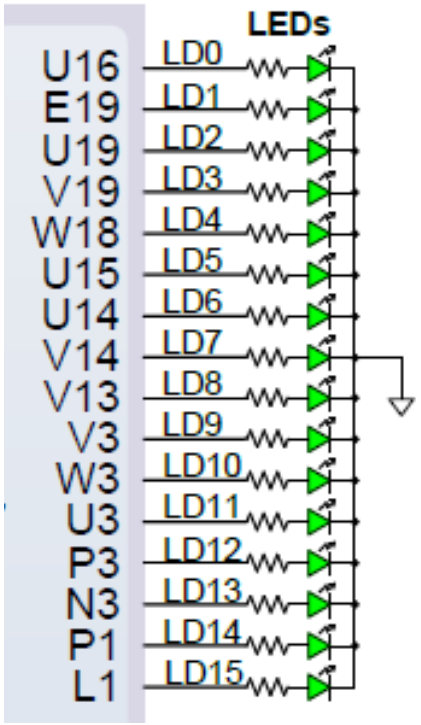

FPGA
Config
7
FPGA
Config
2
PS2_CLK
PS2_DAT
C17
B17
USB HOST (J2)
PIC24FJ128
Artix-7




ESC 76
F1 05
F2 06
F3 04
F4 0C
F5 03
F6 0B
F7 83
F8 0A
F9 01
F10 09
F11 78
F12 07
~` 0E
1! 16
2@ 1E
3# 26
4$ 25
5% 2E
6^ 36
7& 3D
8* 3E
9(46
0) 45
-_ 4E
+= 55
|\ 5D
66
TAB 0D
Q 15
W 1D
E 24
R 2D
T 2C
Y 35
U 3C
I 43
O 44
P 4D
[{ 54
}] 5B
Caps 58
A 1C
S 1B
D 23
F 2B
G 34
H 33
J 3B
K 42
L 4B
:; 4C
"' 52
5A
Shift 12
Z 1A
X 22
C 21
V 2A
B 32
N 31
M 3A
<, 41
>. 49
?/ 4A
Shift 59
Ctrl 14
Alt 11
SPACE 29
Alt E0 11
Ctrl E0 14
E012E07C
E11477E1F014F077
Prt Scr
Scr L 7E
Pause Brk
Ins E070
Home E06C
P Up E07D
Del E071
End E069
P Dn E07A
E075
E06B
E072
E074
Num 77
/ E04A
* 7C
- 7B
7 6C
8 75
9 7D
+ 79
4 6B
5 73
6 74
1 69
2 72
3 7A
Enter E05A
0 70
. 71

Specific I/Os - LED

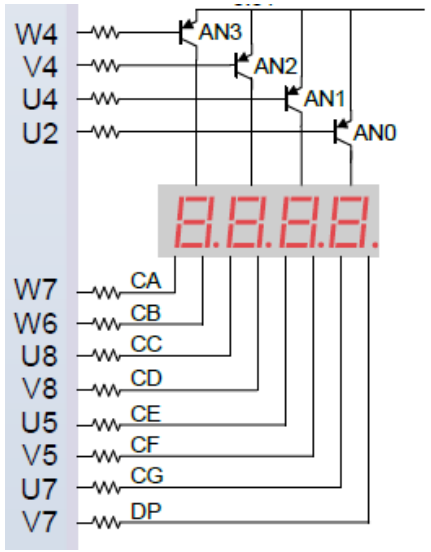
```
set_property PACKAGE_PIN U16 [get_ports {LD0}]
set_property IOSTANDARD LVCMOS33 [get_ports {LD0}]
set_property PACKAGE_PIN E19 [get_ports {LD1}]
set_property IOSTANDARD LVCMOS33 [get_ports {LD1}]
set_property PACKAGE_PIN U19 [get_ports {LD2}]
set_property IOSTANDARD LVCMOS33 [get_ports {LD2}]
set_property PACKAGE_PIN W18 [get_ports {LD3}]
set_property IOSTANDARD LVCMOS33 [get_ports {LD3}]
set_property PACKAGE_PIN U15 [get_ports {LD4}]
set_property IOSTANDARD LVCMOS33 [get_ports {LD4}]
```

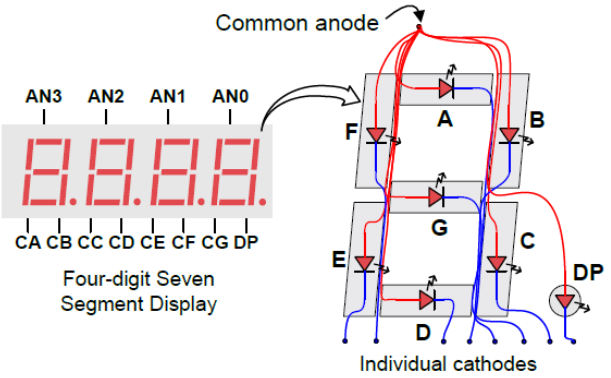



```
set_property PACKAGE_PIN P1 [get_ports {LD14}]
set_property IOSTANDARD LVCMOS33 [get_ports {LD14}]
set_property PACKAGE_PIN L1 [get_ports {LD15}]
set_property IOSTANDARD LVCMOS33 [get_ports {LD15}]
```

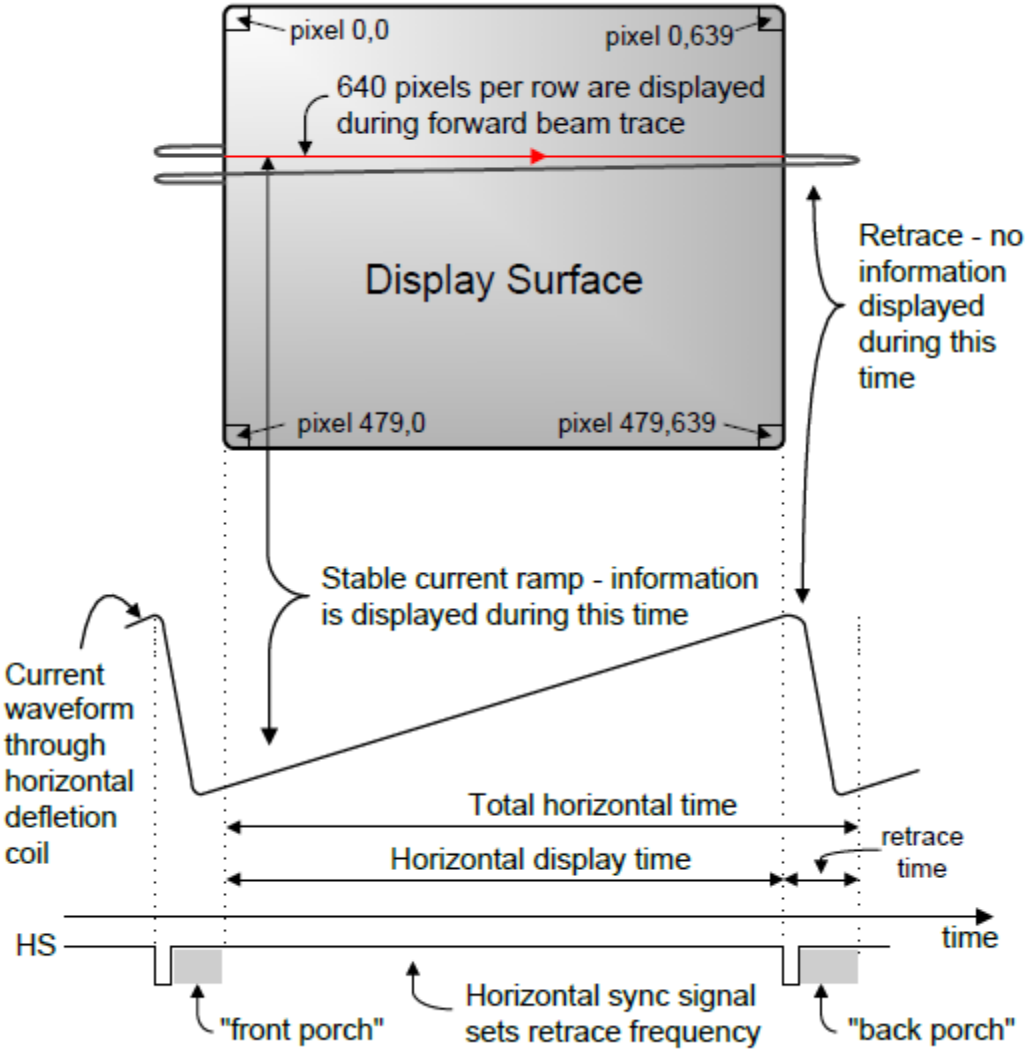
Specific I/Os – Segment Displays

```
set_property PACKAGE_PIN W4 [get_ports {AN3}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN3}]
set_property PACKAGE_PIN V4 [get_ports {AN2}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN2}]
set_property PACKAGE_PIN U4 [get_ports {AN1}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN1}]
set_property PACKAGE_PIN U2 [get_ports {AN0}]
set_property IOSTANDARD LVCMOS33 [get_ports {AN0}]
set_property PACKAGE_PIN W7 [get_ports {CA}]
set_property IOSTANDARD LVCMOS33 [get_ports {CA}]
                              ⋮
set_property PACKAGE_PIN V7 [get_ports {DP}]
set_property IOSTANDARD LVCMOS33 [get_ports {DP}]
```



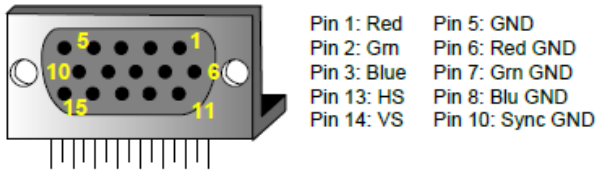


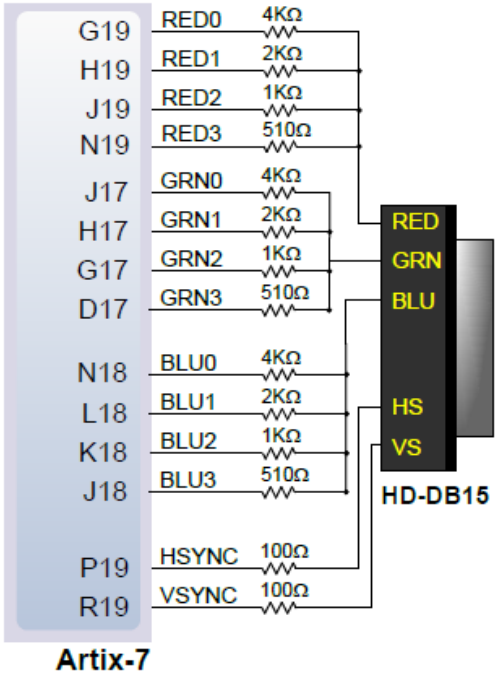
Specific I/Os – VGA Control



Specific I/Os – VGA Control

```
set_property PACKAGE_PIN G19 [get_ports {RED0}]
set_property IOSTANDARD LVCMOS33 [get_ports {RED0}]
set_property PACKAGE_PIN H19 [get_ports {RED1}]
set_property IOSTANDARD LVCMOS33 [get_ports {RED1}]
set_property PACKAGE_PIN J19 [get_ports {RED2}]
set_property IOSTANDARD LVCMOS33 [get_ports {RED2}]
set_property PACKAGE_PIN N19 [get_ports {RED3}]
set_property IOSTANDARD LVCMOS33 [get_ports {RED3}]
                    ⋮
                   GRN
                    ⋮
                   BLU

set_property PACKAGE_PIN P19 [get_ports {HSYNC}]
set_property IOSTANDARD LVCMOS33 [get_ports {HSYNC}]
set_property PACKAGE_PIN R19 [get_ports {VSYNC}]
set_property IOSTANDARD LVCMOS33 [get_ports {VSYNC}]
```





Audio

```
set_property PACKAGE_PIN J1 [get_ports {JA[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {JA[0]}]
set_property PACKAGE_PIN L2 [get_ports {JA[1]}]
Set_property IOSTANDARD LVCMOS33 [get_ports {JA[1]}]
set_property PACKAGE_PIN J2 [get_ports {JA[2]}]
set_property IOSTANDARD LVCMOS33 [get_ports {JA[2]}]
set_property PACKAGE_PIN G2 [get_ports {JA[3]}]
set_property IOSTANDARD LVCMOS33 [get_ports {JA[3]}]
set_property PACKAGE_PIN H1 [get_ports {JA[4]}]
set_property IOSTANDARD LVCMOS33 [get_ports {JA[4]}]
set_property PACKAGE_PIN K2[get_ports {JA[5]}]
set_property IOSTANDARD LVCMOS33 [get_ports {JA[5]}]
set_property PACKAGE_PIN H2 [get_ports {JA[6]}]
set_property IOSTANDARD LVCMOS33 [get_ports {JA[6]}]
set_property PACKAGE_PIN G3 [get_ports {JA[7]}]
set_property IOSTANDARD LVCMOS33 [get_ports {JA[7]}]
```

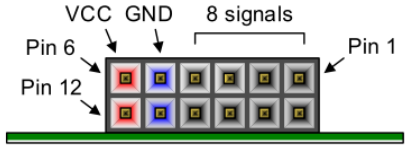


Figure 20. Pmod connectors; front view as loaded on PCB.

Pmod JA	Pmod JB	Pmod JC
JA1: J1	JB1: A14	JC1: K17
JA2: L2	JB2: A16	JC2: M18
JA3: J2	JB3: B15	JC3: N17
JA4: G2	JB4: B16	JC4: P18
JA7: H1	JB7: A15	JC7: L17
JA8: K2	JB8: A17	JC8: M19
JA9: H2	JB9: C15	JC9: P17
JA10: G3	JB10: C16	JC10: R18

Table 6. Basys3 Pmod pin assignm

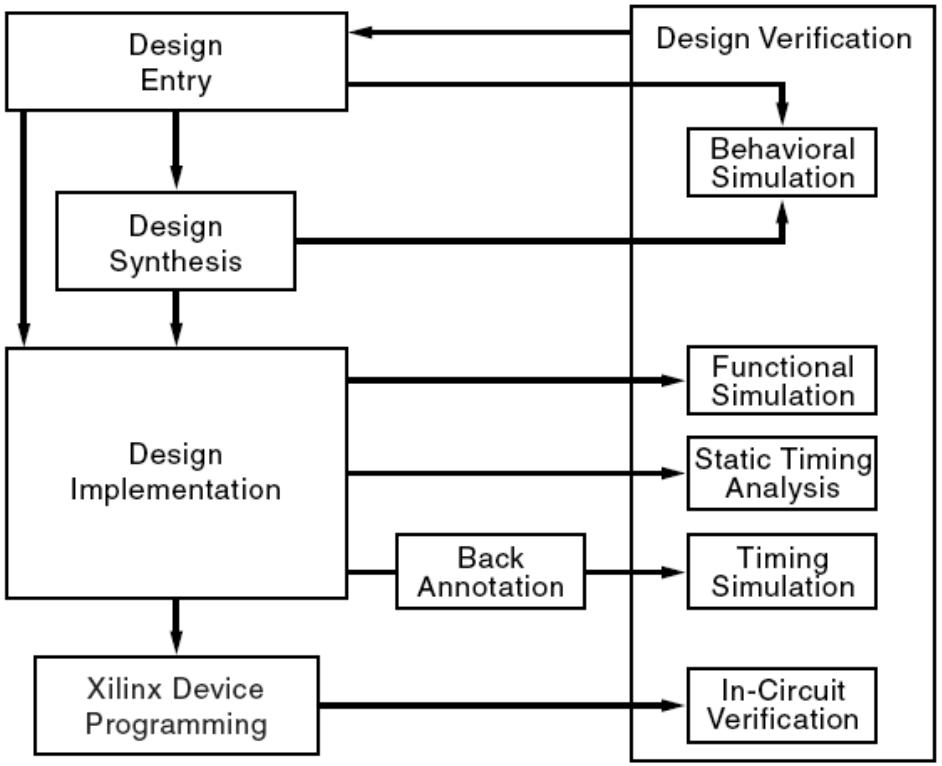
Clock

- External Clock with 100MHz frequency

```
set_property PACKAGE_PIN W5 [get_ports {CLK}]
set_property IOSTANDARD LVCMOS33 [get_ports {CLK}]
```

Design Flow

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



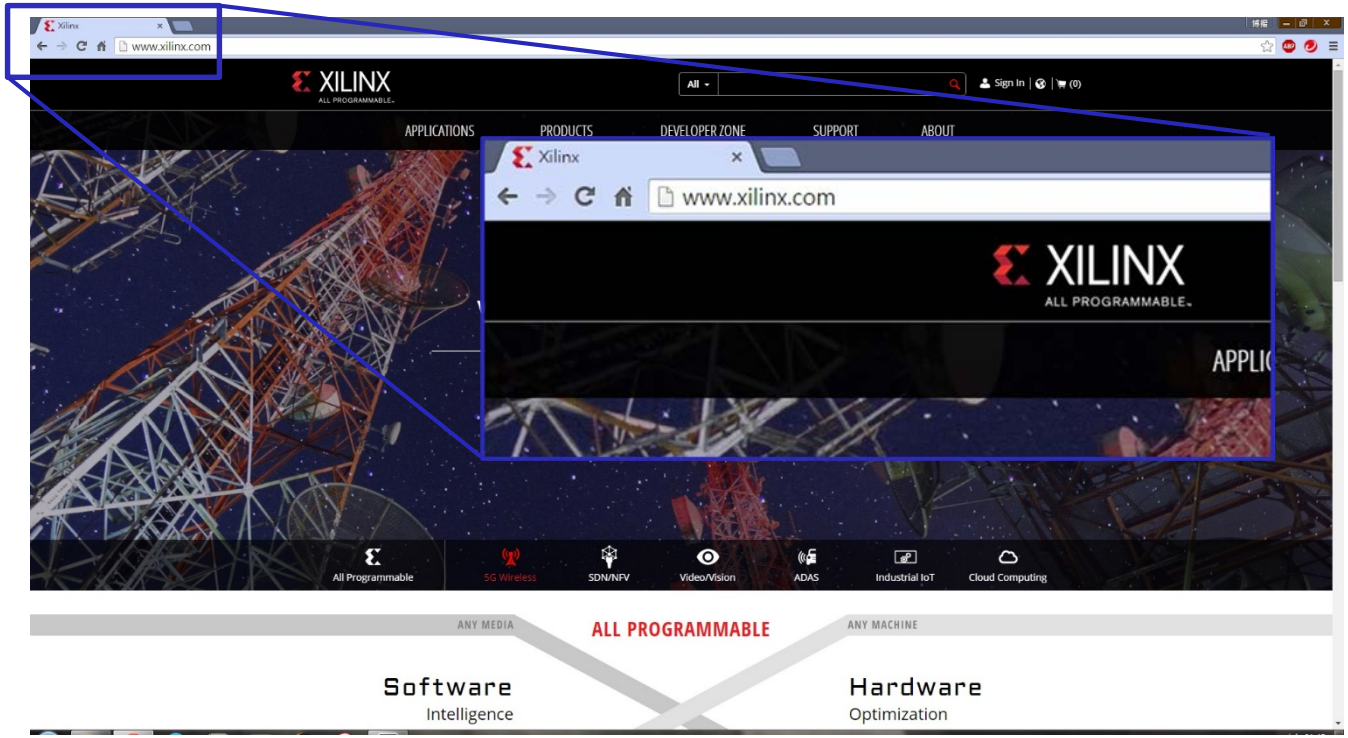
Important Notes

- **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

FPGA Emulation Using Xilinx Vivado

How to setup Xilinx ISE

Download link : **http://www.xilinx.com**



*** You can follow the installation guide (lab0.1 Vivado Installation.pdf) in the course website.**

FPGA Emulation Using Xilinx Vivado

Design Flow

- Design Source Preparation
  - Design modules (.v)
  - Design constraints (I/O pin assignments) (.xdc)
- Design Simulation
  - testbench (.v)
- Design Synthesis and Implementation

Create New Project (2/3)

Continue from previous unit slides

New Project

Default Part

Choose a default Xilinx part or board for your project. This can be changed later.

Select: Parts Boards

Filter

Product category: All

Family: Artix-7

Package: cpg236

Speed grade: -1

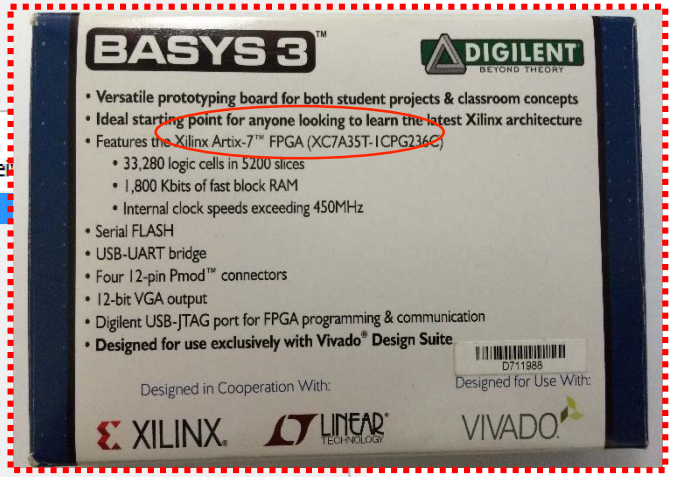
Temp grade: All Remaining

Reset All Filters

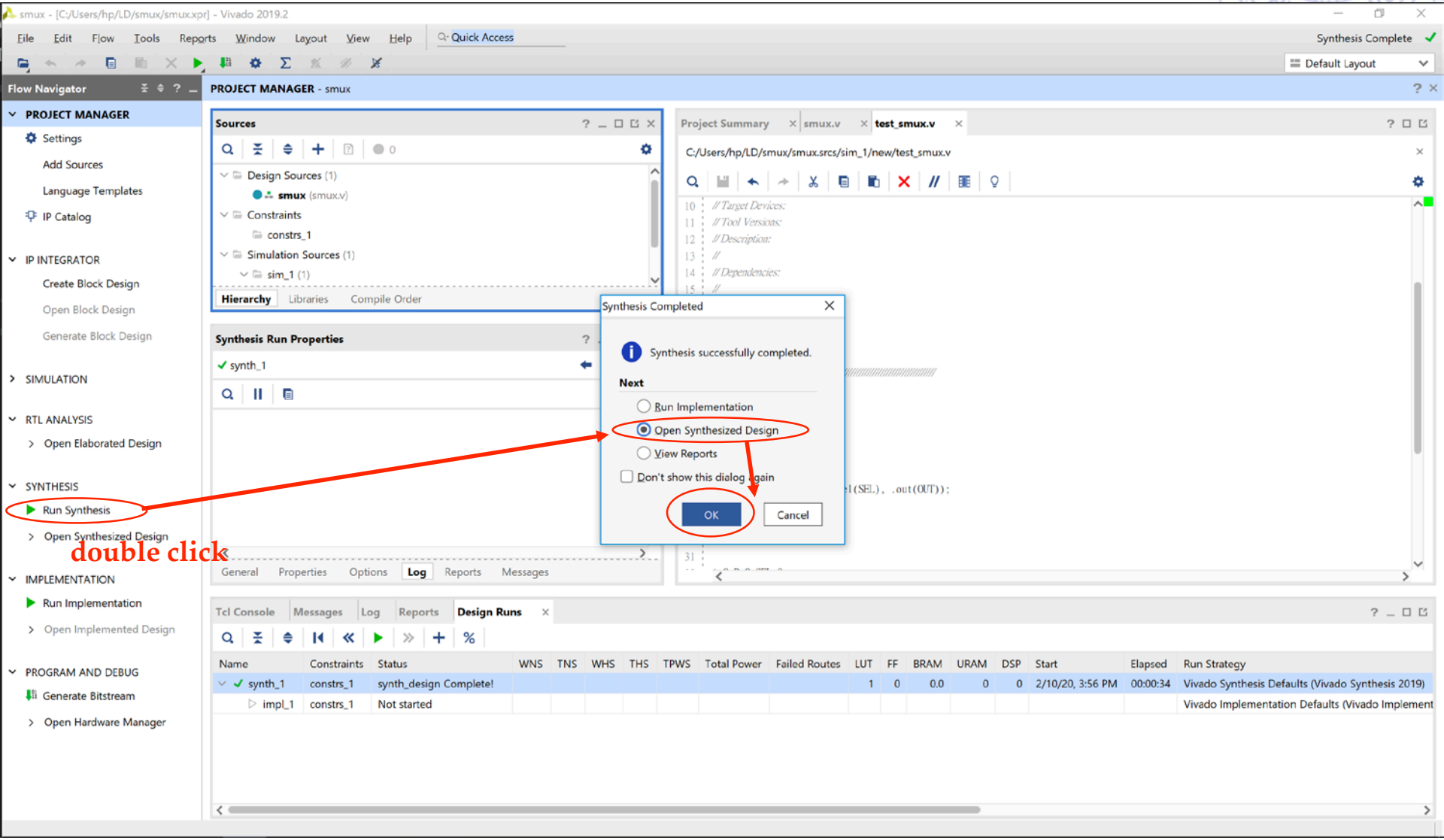
Search: xc7a35tcp (1 match)

Part	I/O Pin Count	Block RAMs	DSPs	FlipFlops	GTPE2 Transceivers	Gb Transce
xc7a35tcpg236-1	236	50	90	41600	2	2

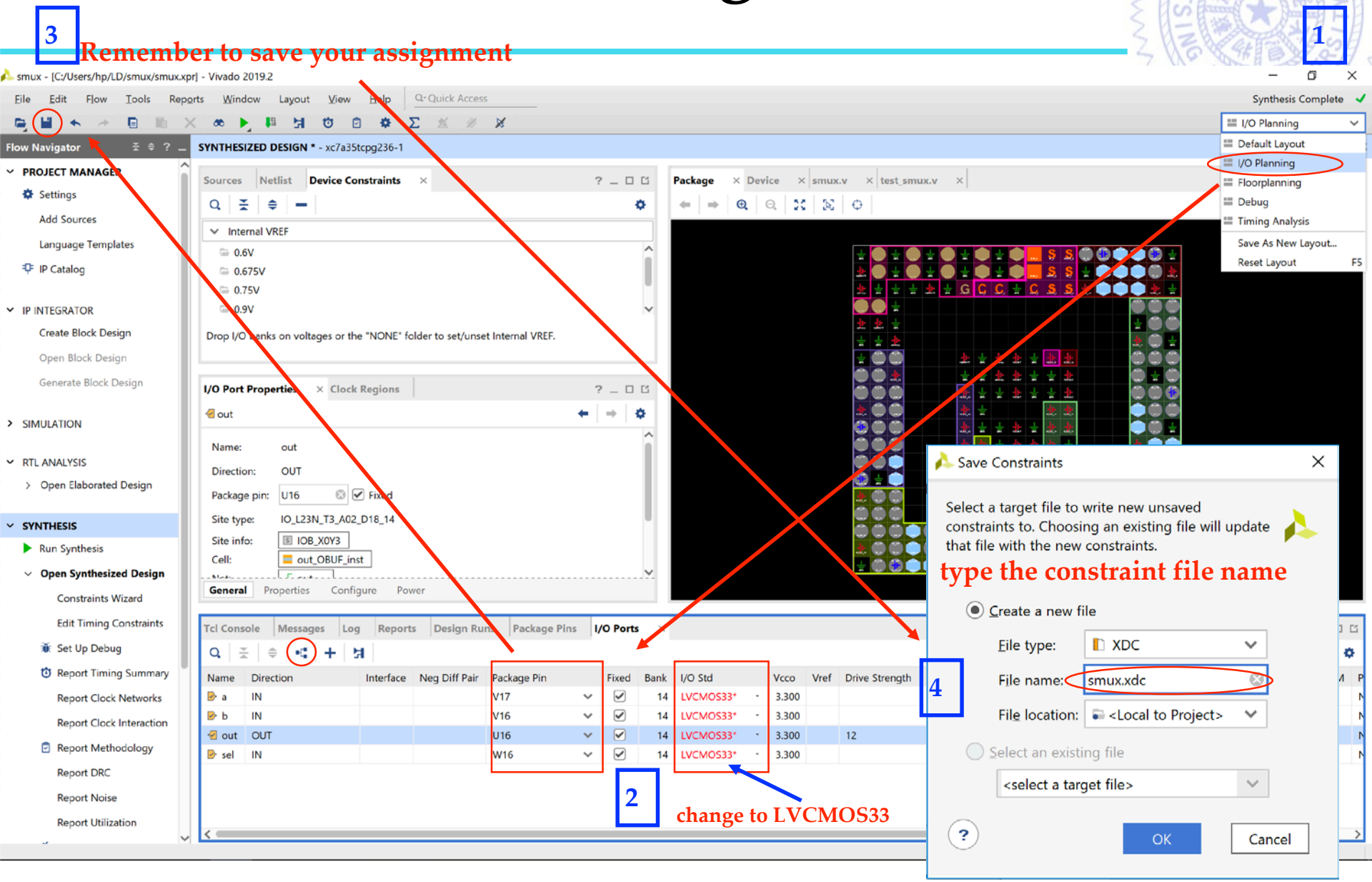
? < Back Next > Finish Cancel



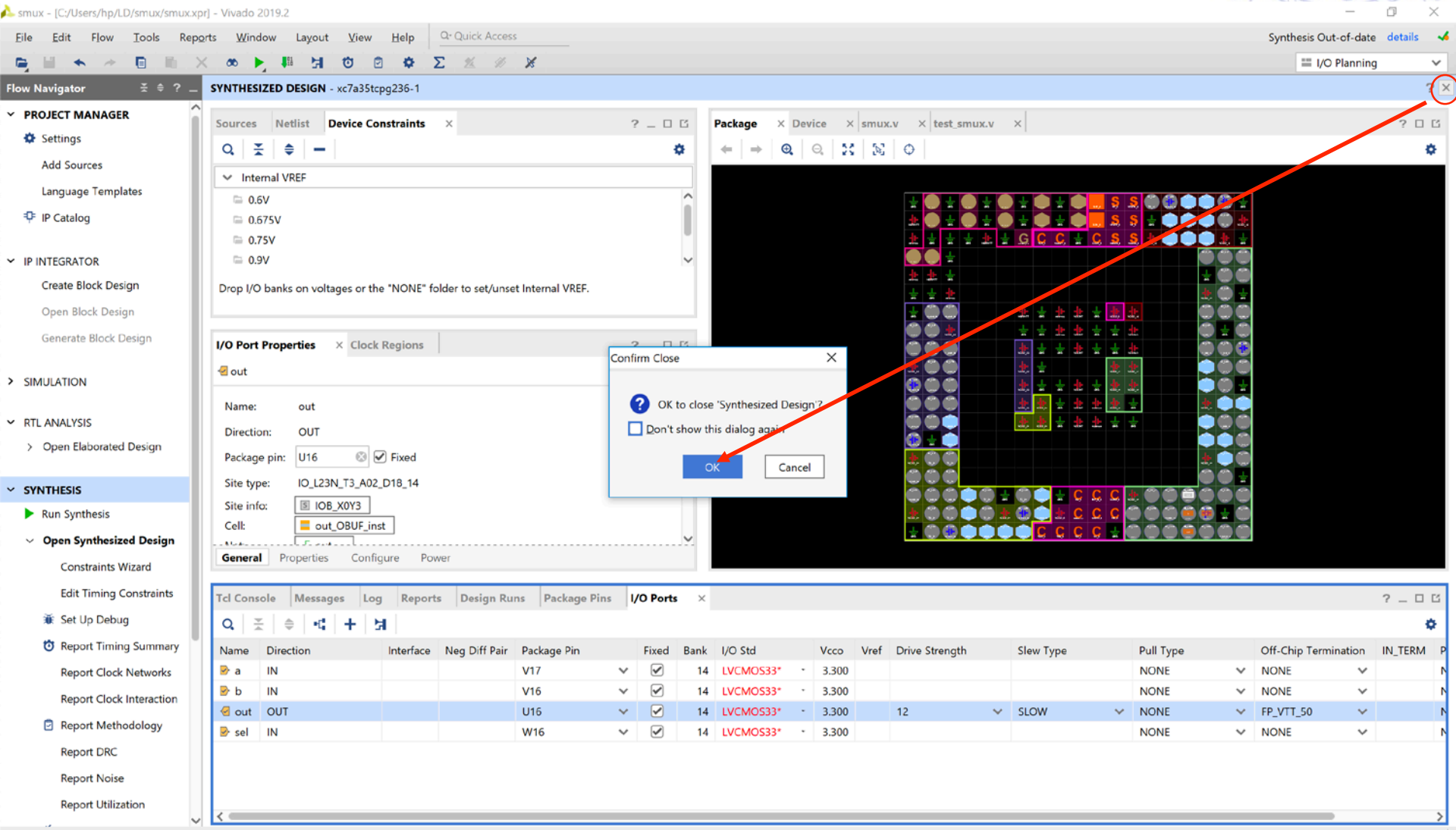
I/O Pins Assignment (1/4)



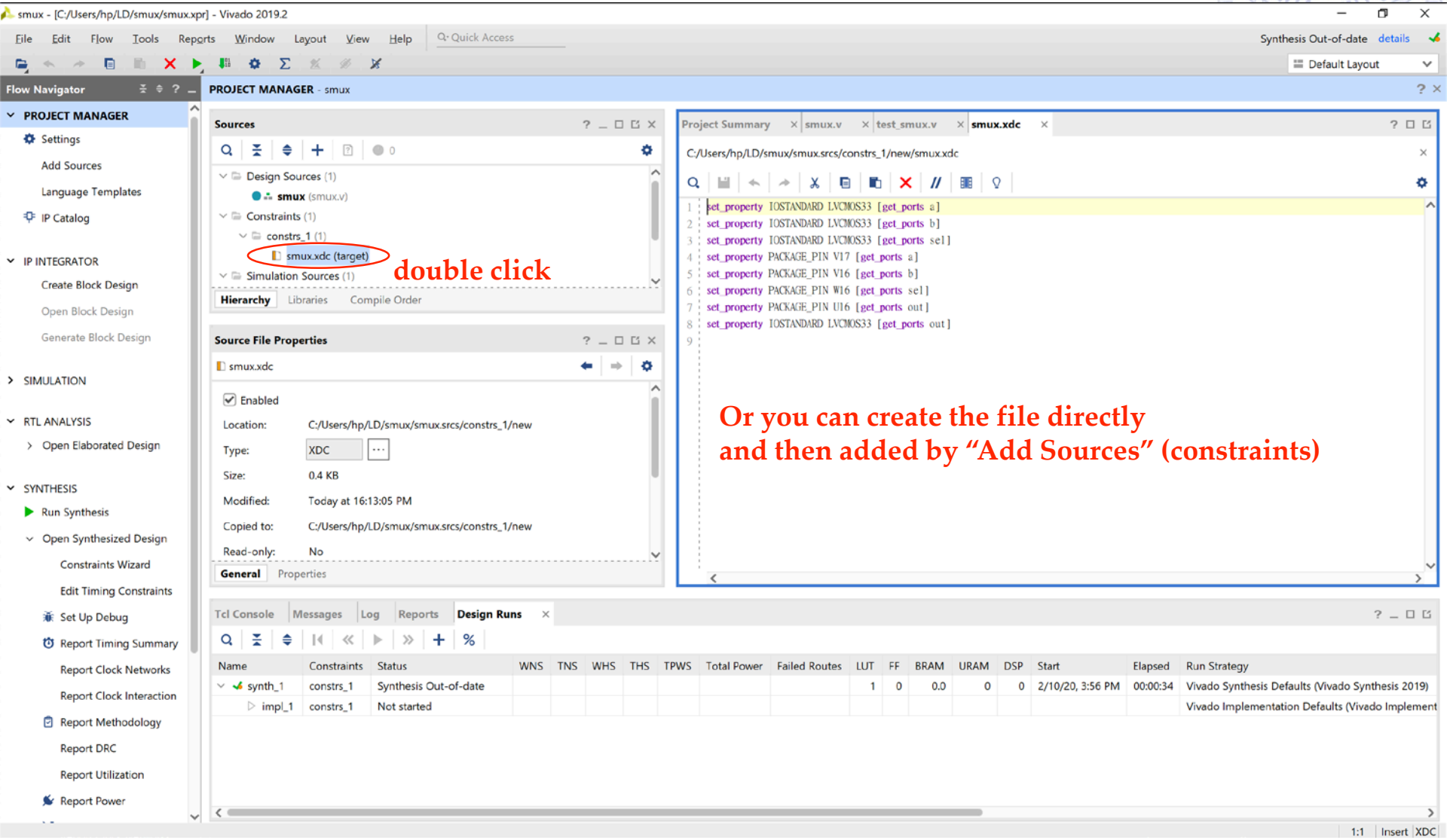
I/O Pins Assignment (2/4)



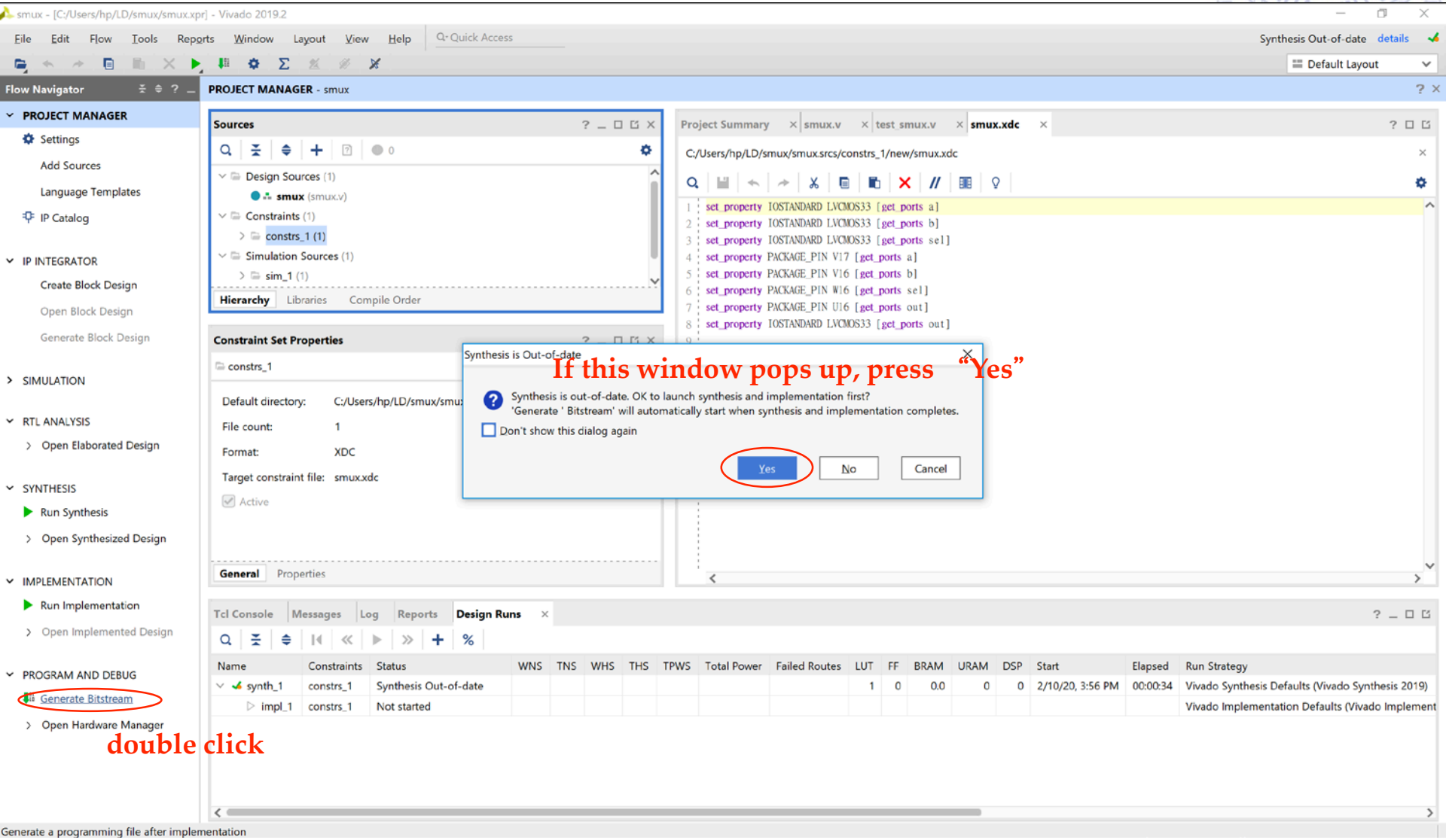
I/O Pins Assignment (3/4)



I/O Pins Assignment (4/4)

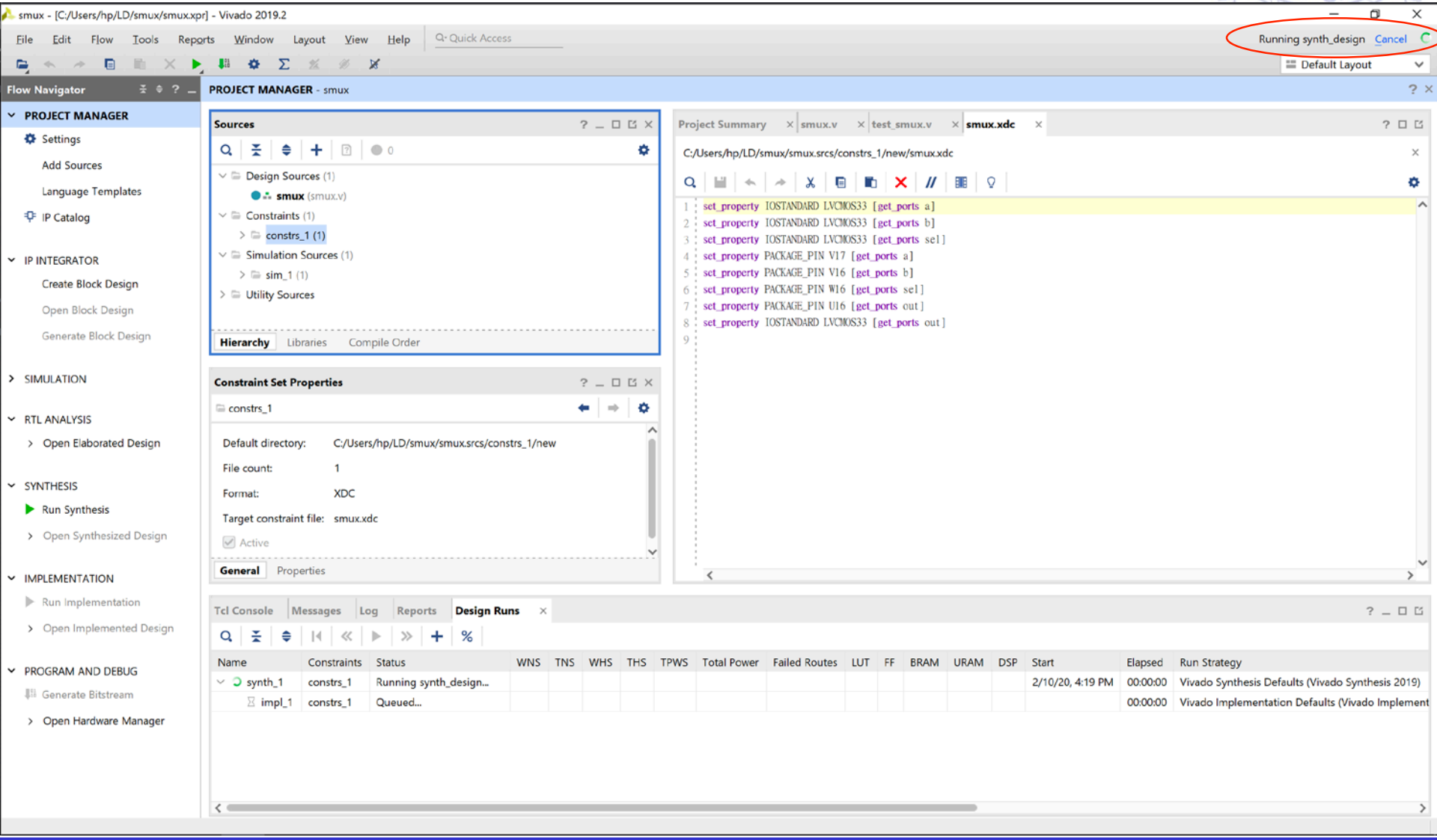


Synthesis and Implementation (1/7)



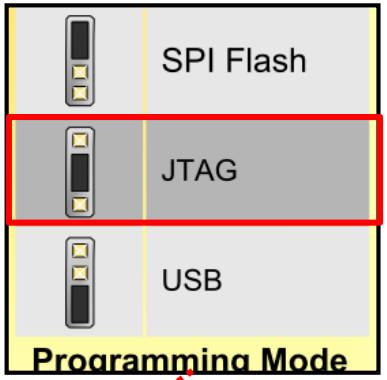
Synthesis and Implementation (2/7)

implementation progress information

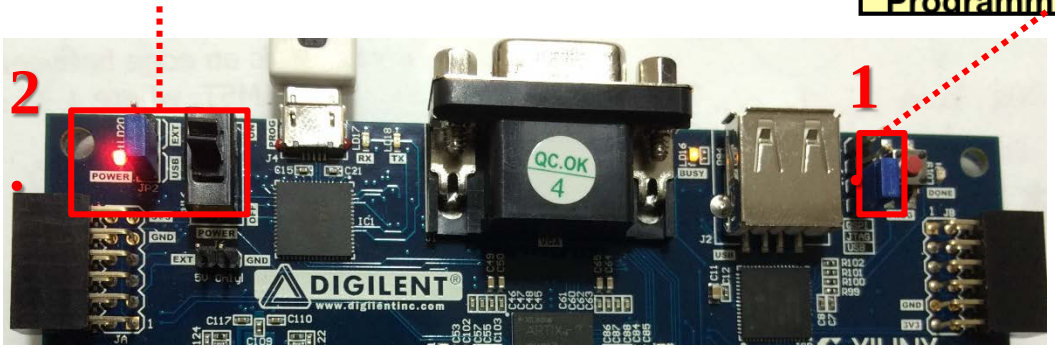


Power-on and JTAG-Mode

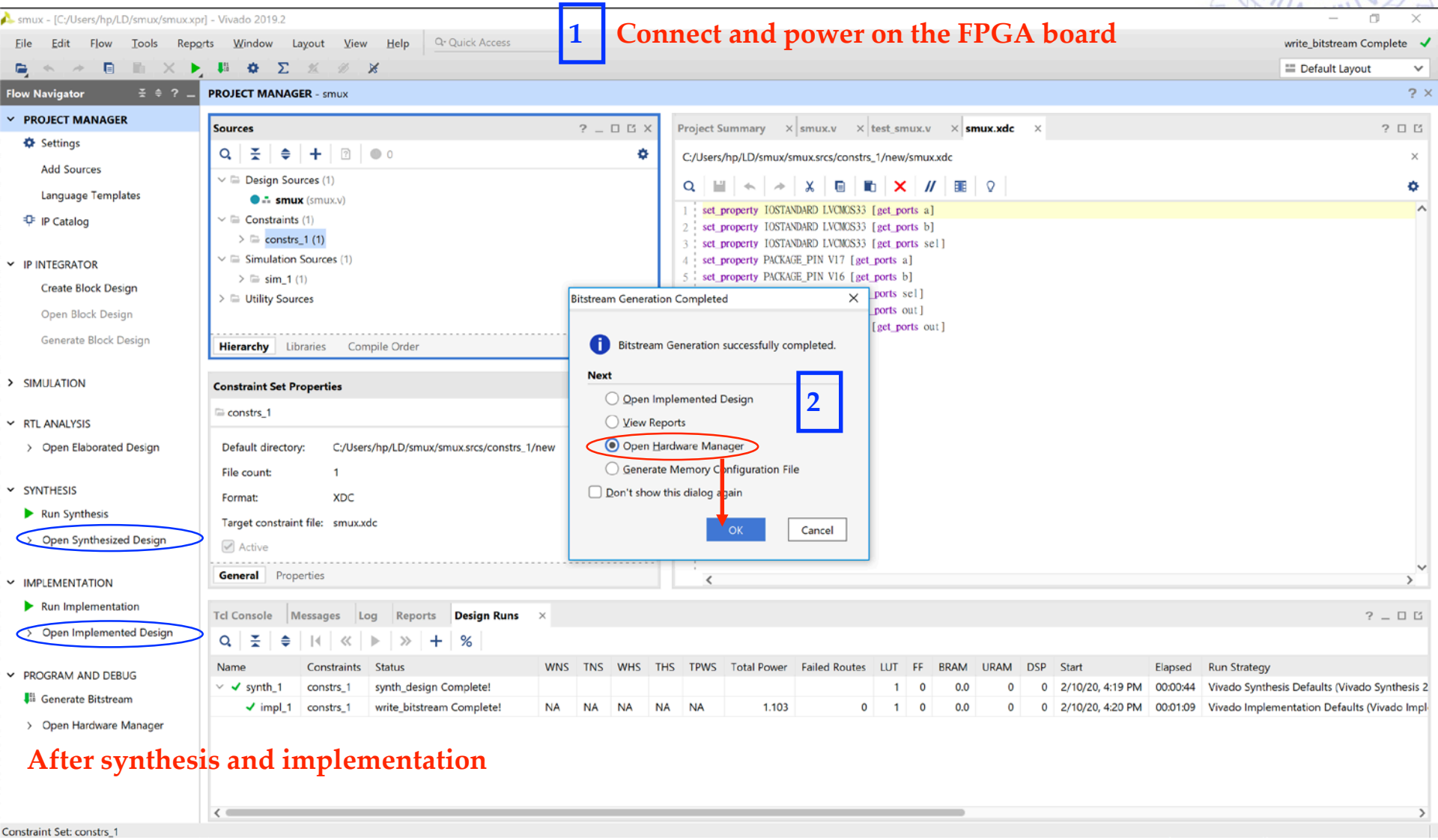
Step 1:
Change blue jumper to [**JTAG**] mode



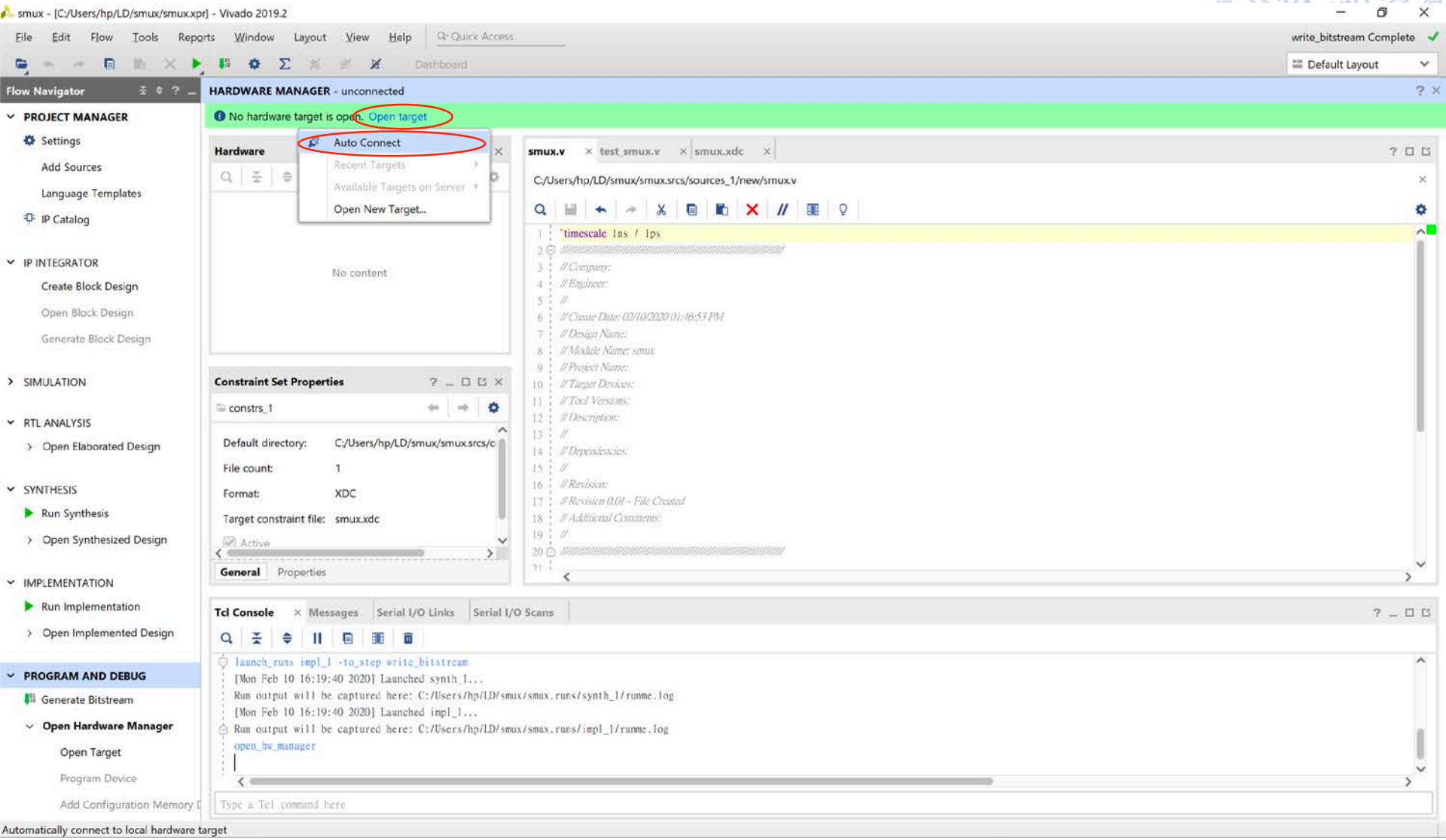
Step 2:
Turn the power switch to [**ON**]



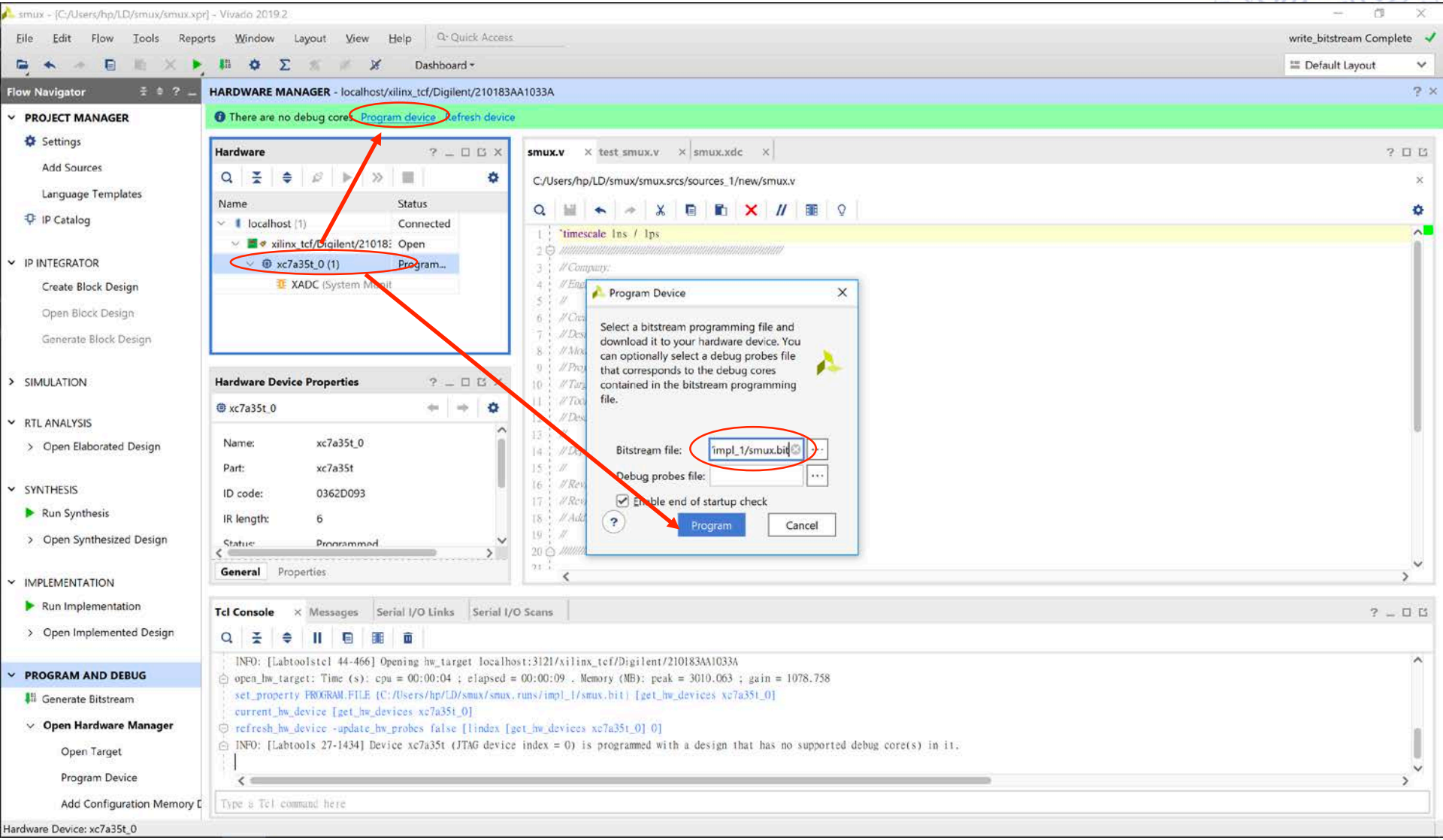
Synthesis and Implementation (3/7)



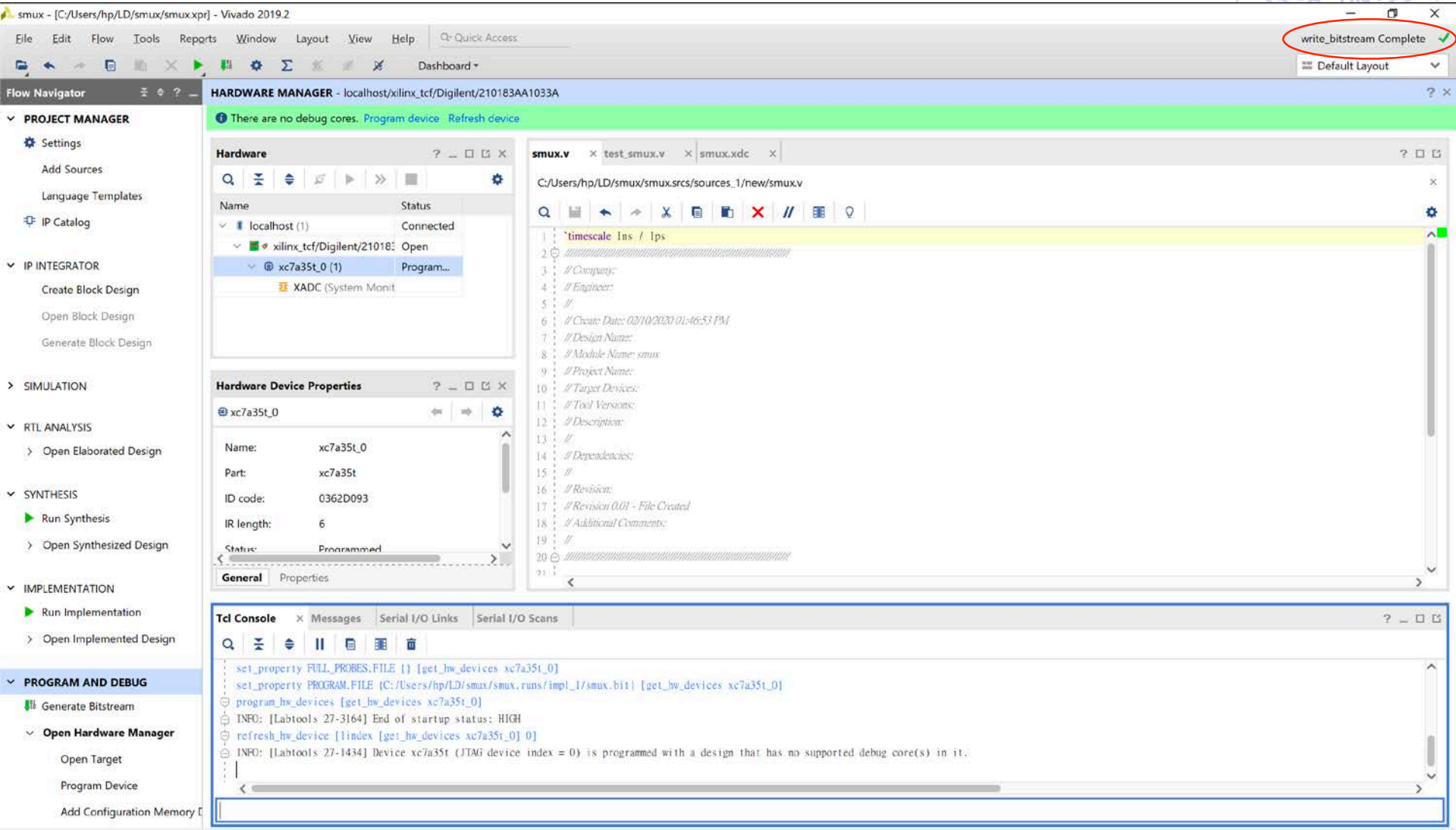
Synthesis and Implementation (4/7)



Synthesis and Implementation (5/7)

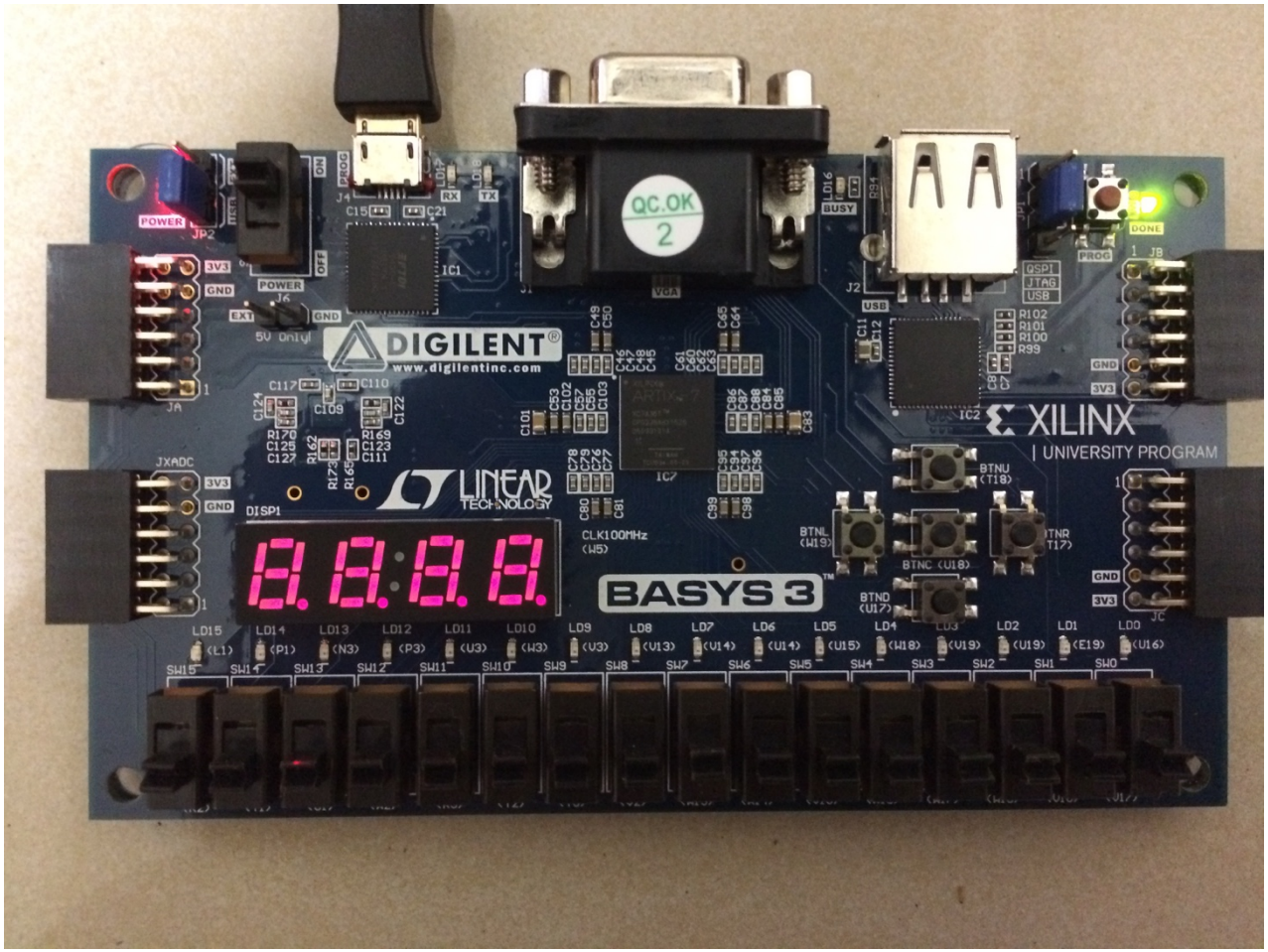


Synthesis and Implementation (6/7)



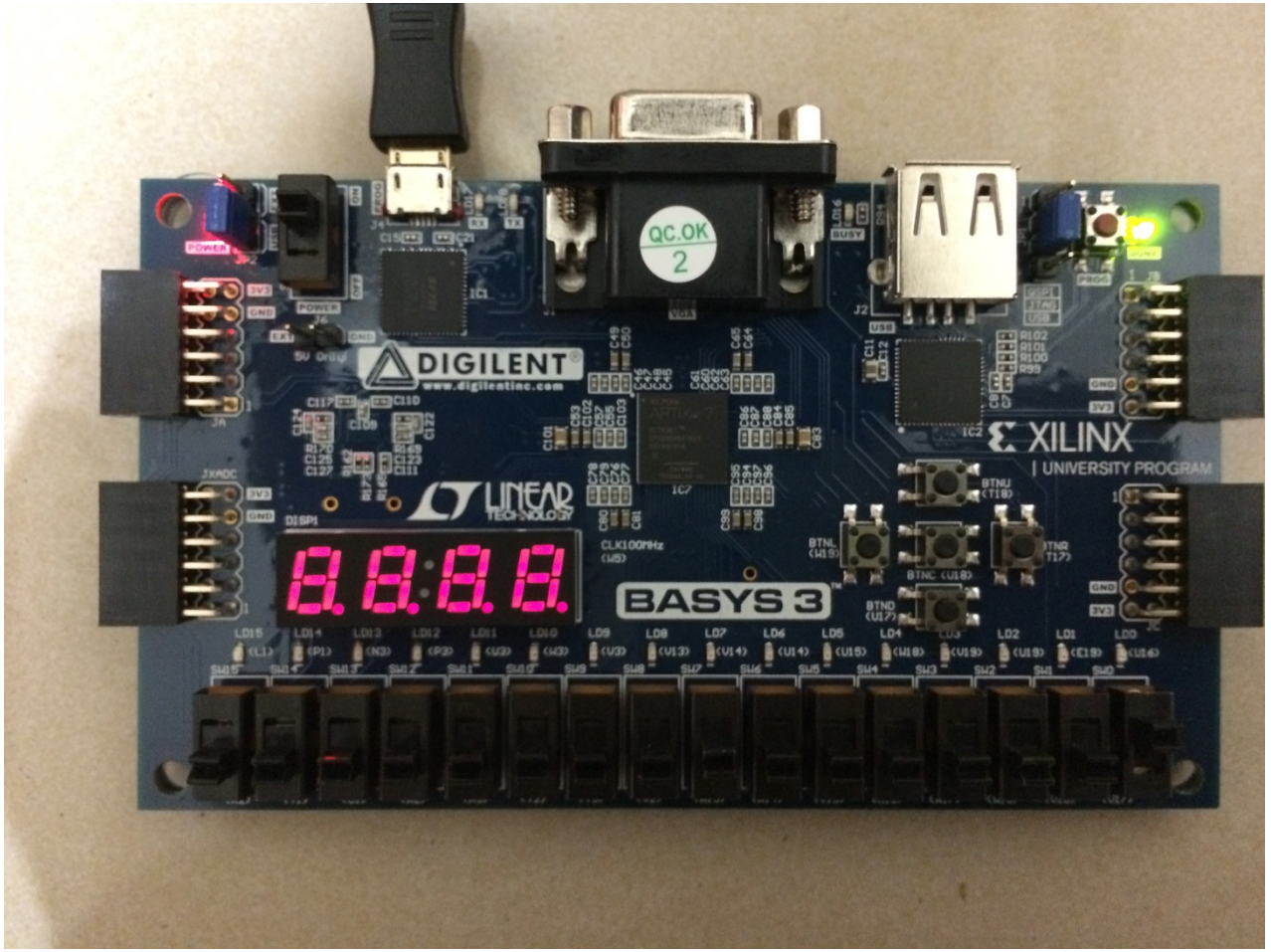
Result

sel = 0
a = 0
b = 0



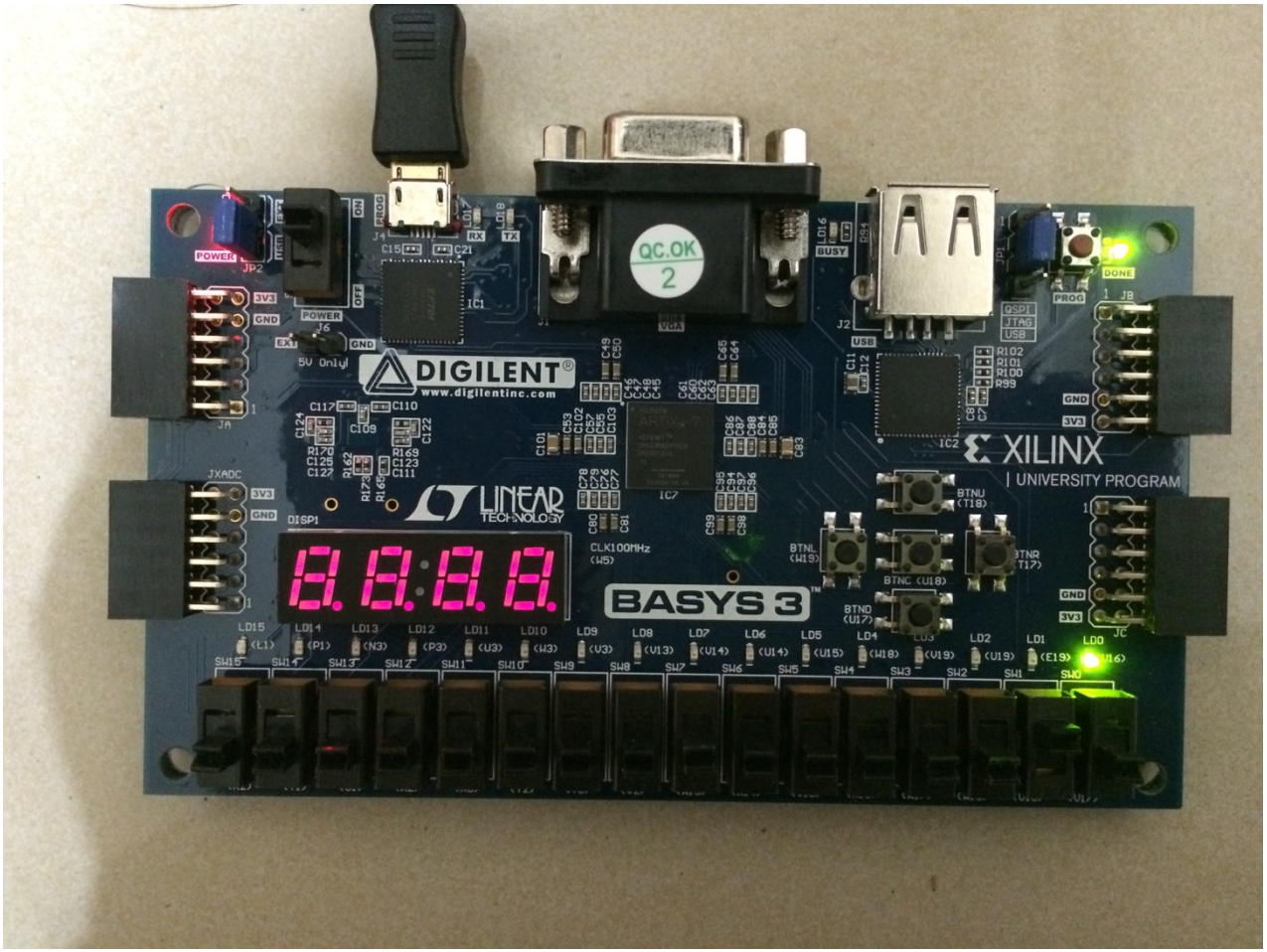
Result

sel = 0
a = 1
b = 0



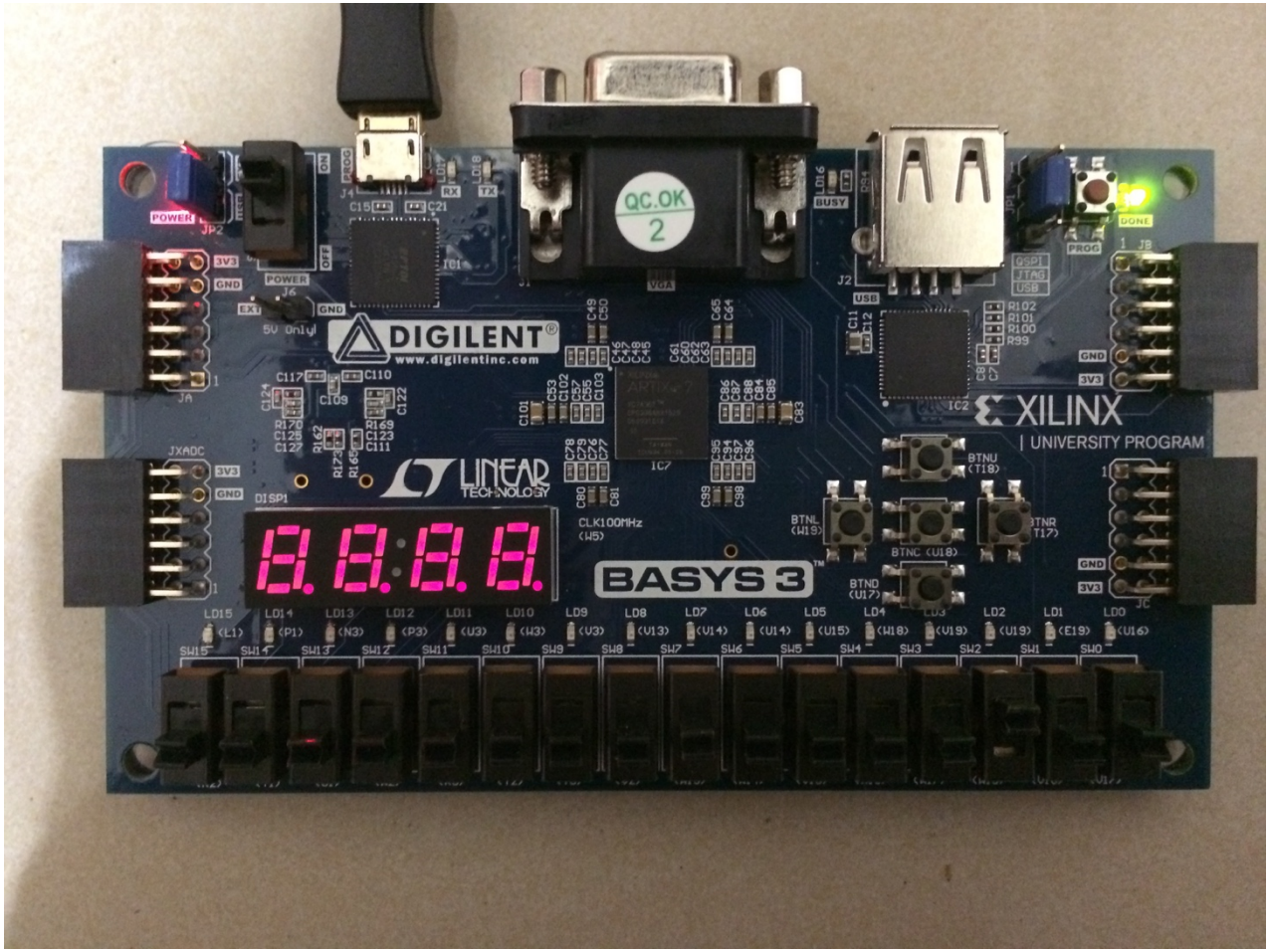
Result

sel = 0
a = 0
b = 1



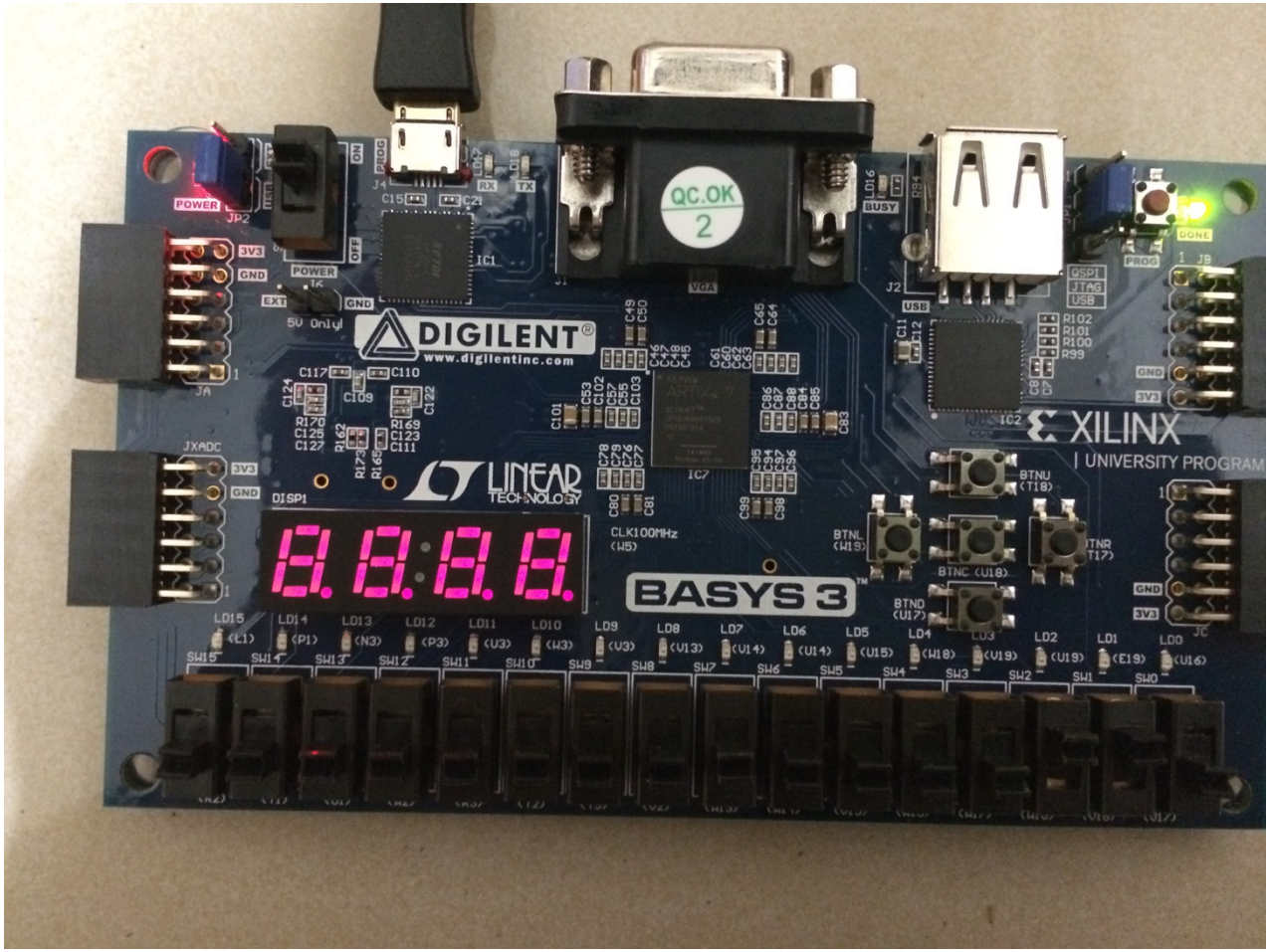
Result

sel = 1
a = 0
b = 0



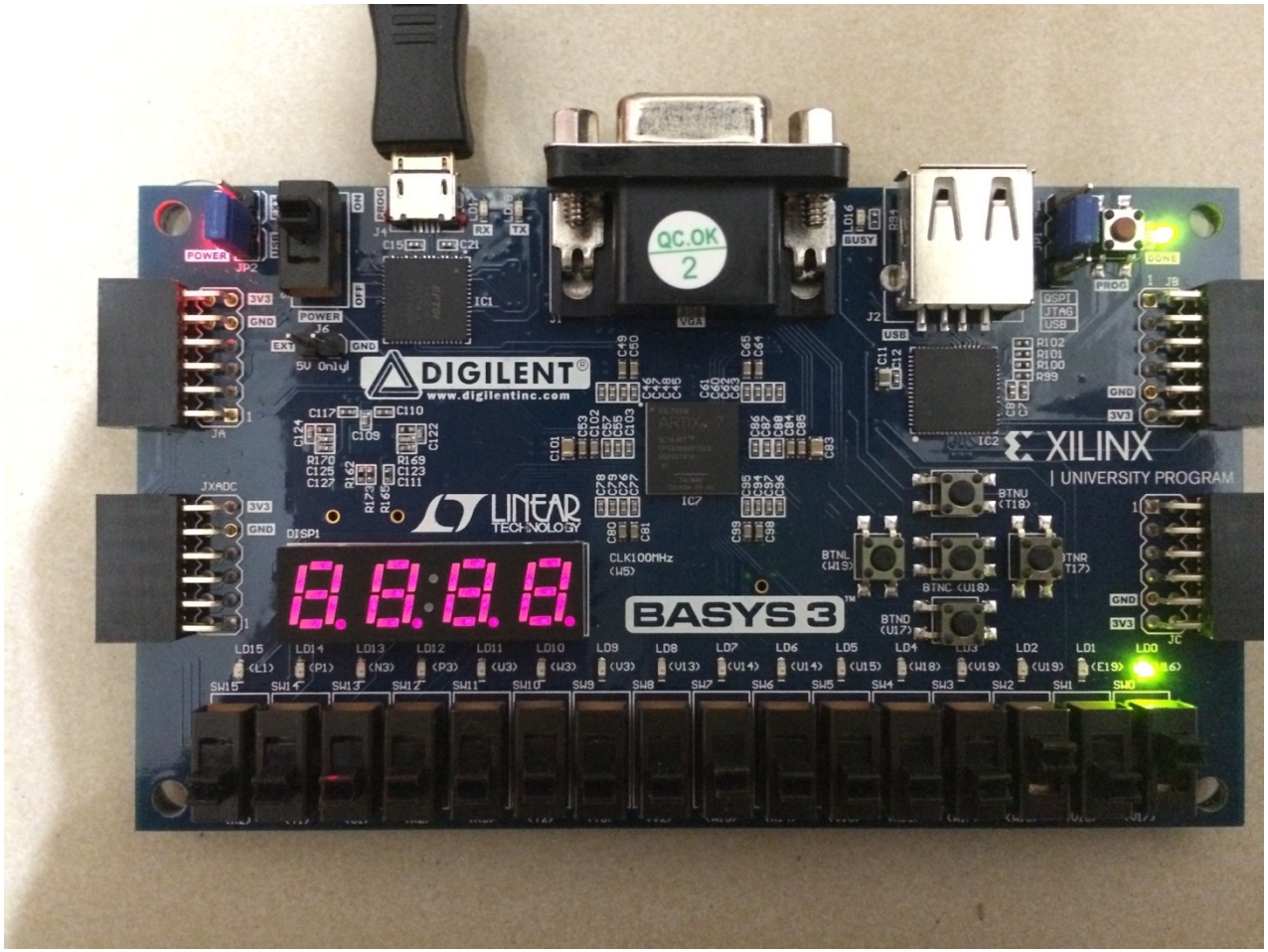
Result

sel = 1
a = 0
b = 1



Result

sel = 1
a = 1
b = 0



Verilog RTL Code Examples

```
`timescale 1ns / 1ps

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

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 @(a or b or sel)
  out = (sel) ? a : b ;

endmodule
```