CSE 490/590 - Project 1 - Interacting with the Basys3 FPGA Board
(with contribution from Haohua Feng and Alexander Bohosian)

What you need:

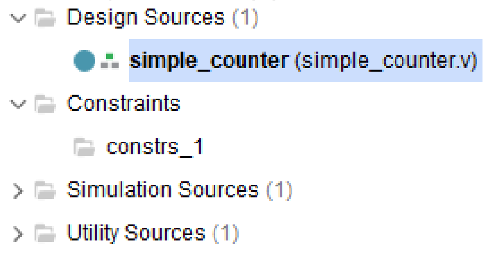
1. design sources:
   - ■ working .v file(s), which is your project
   - ■ test your design by running simulation, make sure it is working as expected.
2. Basys3 FPGA board
3. Bitstream of your design

   More information at: https://digilent.com/reference/programmable-logic/basys-3/start

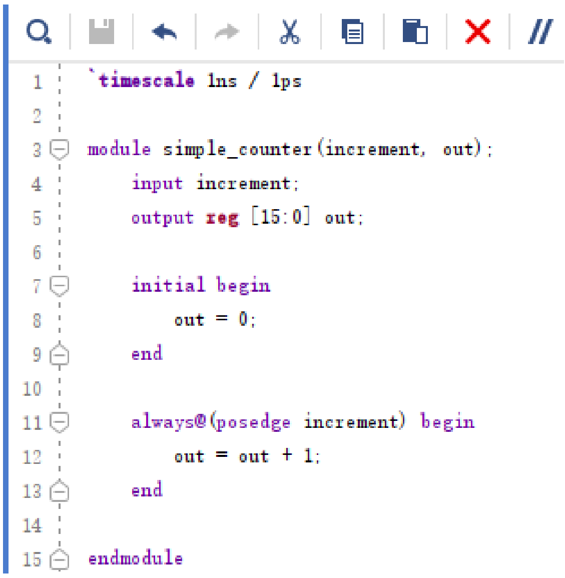
What to do (with demonstration):

- For demonstration, a simple counter is used. It has 1 input and 1 output. Each time the counter receives a positive edge signal it will increment by 1 and output the counter value.

1. Prepare your .v file(s)



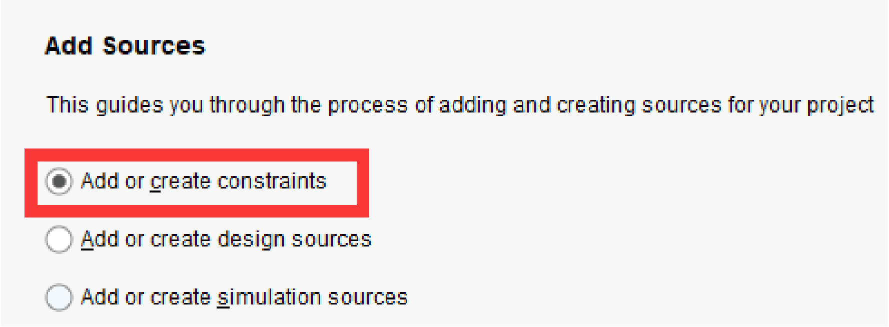
```
∨ Design Sources (1)
      simple_counter (simple_counter.v)
∨ Constraints
      constrs_1
> Simulation Sources (1)
> Utility Sources (1)
```



```
`timescale 1ns / 1ps

module simple_counter(increment, out);
    input increment;
    output reg [15:0] out;

    initial begin
        out = 0;
    end

    always@(posedge increment) begin
        out = out + 1;
    end

endmodule
```

2. Create a .xdc constraints file

- The constraints file is used to inform Vivado what physical pins are on the FPGA board that we will be using or connecting to in relation to code that you wrote to describe the behavior of the FPGA. (What is a Constraints File? - Digilent Reference)
  - ■ There are two common ways to create the constrains files:
    - ◆ Create the .xdc file by add sources, then edit by your own.



Add Sources

This guides you through the process of adding and creating sources for your project

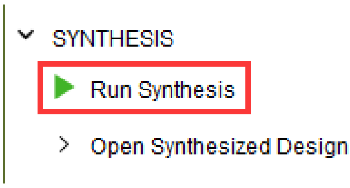
● Add or create constraints

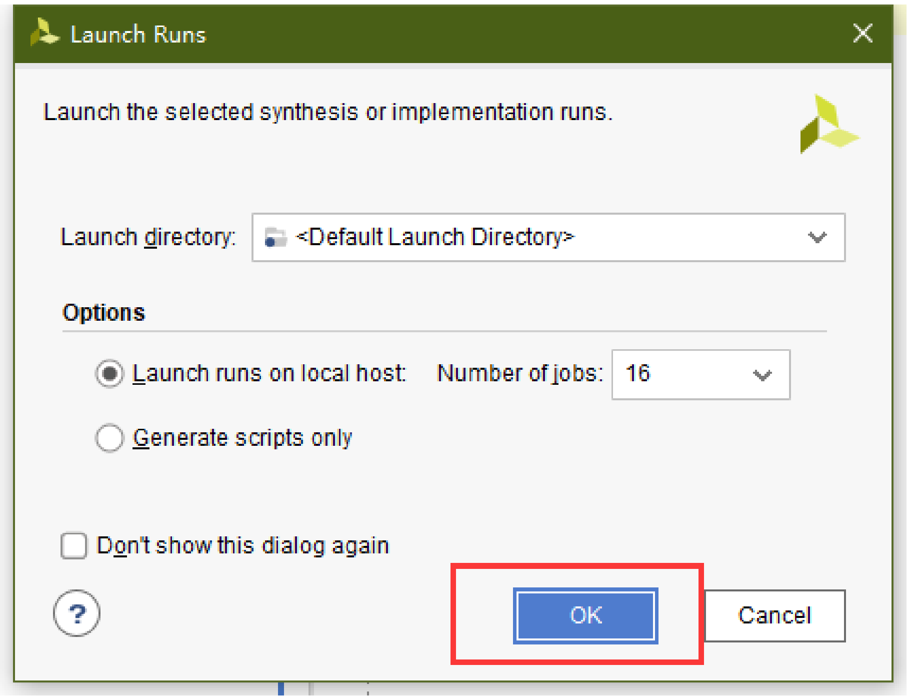
○ Add or create design sources

○ Add or create simulation sources

- Or a simpler way, see below.

a) Set your target .v file as top, then run “Synthesis” in the flow navigator window or in the drop list of the “Flow” tab on the top.





Launch the selected synthesis or implementation runs.

Launch directory: <Default Launch Directory>

Options

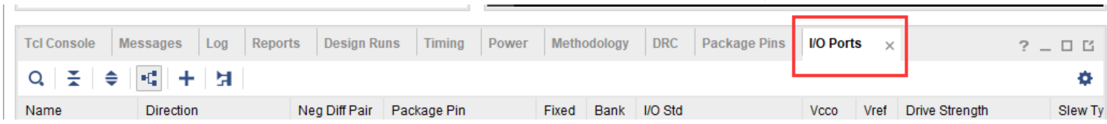
● Launch runs on local host: Number of jobs: 16

○ Generate scripts only

□ Don't show this dialog again

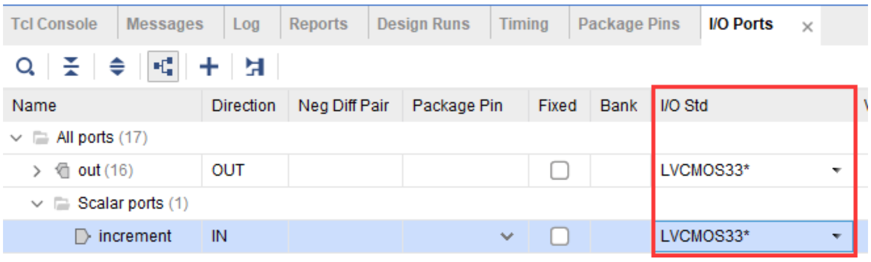
You can just use the default setting and click on “OK”. Wait for it to be finished.

b) Create a constraints file.
In the “I/O Ports” interface. Search “I/O Ports” in search bar if you cannot find it.



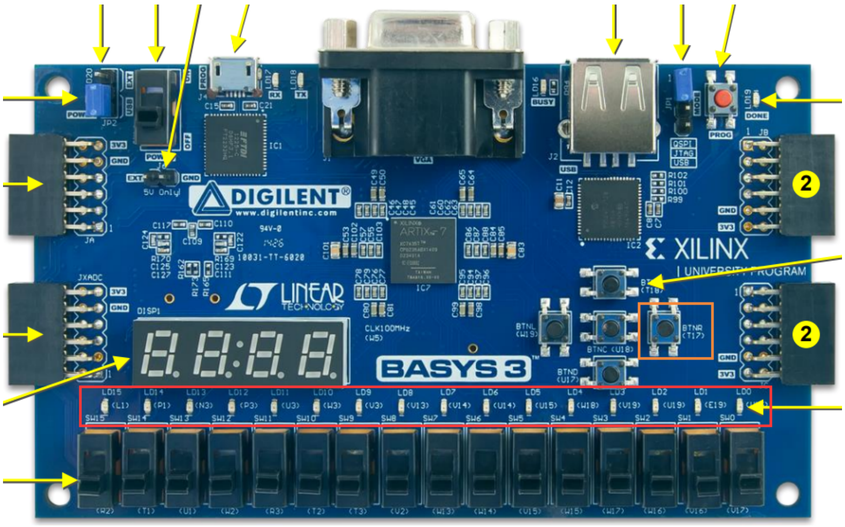
c) Assign pins for the output and inputs of your module

- Set the “I/O Std” of all the ports to “LVCMOS33”

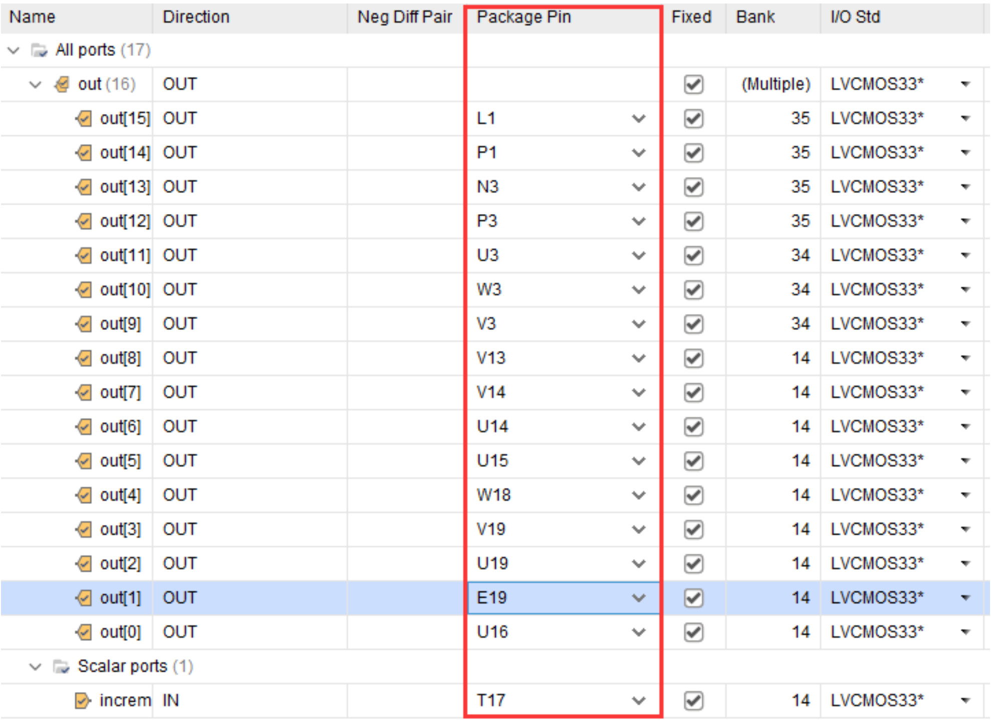


Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std
All ports (17)						
out (16)	OUT			□		LVCMOS33*
Scalar ports (1)						
increment	IN			□		LVCMOS33*

- Take a look on the FPGA board and decide what pins will be mapped to wires in your design



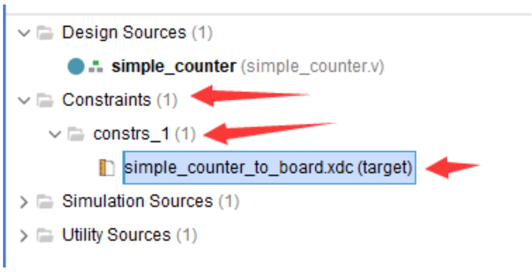
For demonstration, I am using the 16 leds to represent the 16-bit binary output of my simple counter. (0: off, 1: on) And button <T17> as the input signal.



Name	Direction	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std
All ports (17)						
out (16)	OUT			☑	(Multiple)	LVCMOS33*
out[15]	OUT		L1	☑	35	LVCMOS33*
out[14]	OUT		P1	☑	35	LVCMOS33*
out[13]	OUT		N3	☑	35	LVCMOS33*
out[12]	OUT		P3	☑	35	LVCMOS33*
out[11]	OUT		U3	☑	34	LVCMOS33*
out[10]	OUT		W3	☑	34	LVCMOS33*
out[9]	OUT		V3	☑	34	LVCMOS33*
out[8]	OUT		V13	☑	14	LVCMOS33*
out[7]	OUT		V14	☑	14	LVCMOS33*
out[6]	OUT		U14	☑	14	LVCMOS33*
out[5]	OUT		U15	☑	14	LVCMOS33*
out[4]	OUT		W18	☑	14	LVCMOS33*
out[3]	OUT		V19	☑	14	LVCMOS33*
out[2]	OUT		U19	☑	14	LVCMOS33*
out[1]	OUT		E19	☑	14	LVCMOS33*
out[0]	OUT		U16	☑	14	LVCMOS33*
Scalar ports (1)						
increm	IN		T17	☑	14	LVCMOS33*

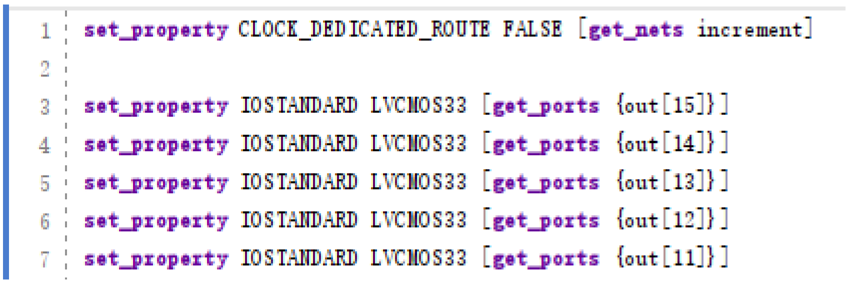
This will generate the .xdc file. Ctrl + S to name and save the .xdc file.

d) Open the .xdc file just created



- Design Sources (1)
  - simple_counter (simple_counter.v)
- Constraints (1)
  - constrs_1 (1)
    - simple_counter_to_board.xdc (target)
- Simulation Sources (1)
- Utility Sources (1)

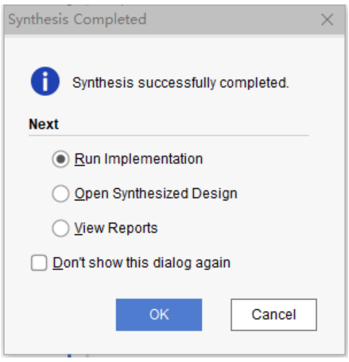
Add “set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets increment]” to it and save it. We are not using the build-in clock to run the simple counter, it is based on manual input signal “increment”.



```
set_property CLOCK_DEDICATED_ROUTE FALSE [get_nets increment]

set_property IOSTANDARD LVCMOS33 [get_ports {out[15]}]
set_property IOSTANDARD LVCMOS33 [get_ports {out[14]}]
set_property IOSTANDARD LVCMOS33 [get_ports {out[13]}]
set_property IOSTANDARD LVCMOS33 [get_ports {out[12]}]
set_property IOSTANDARD LVCMOS33 [get_ports {out[11]}]
```

The constraint files is created

e) Run Implementation once the Synthesis is successfully completed.



Synthesis Completed

Synthesis successfully completed.

Next

● Run Implementation

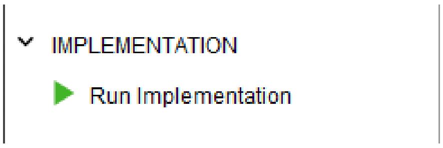
○ Open Synthesized Design

○ View Reports

□ Don't show this dialog again

OK Cancel

or



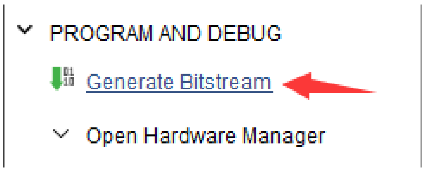
IMPLEMENTATION

Run Implementation

Similar as the previous step, you can just use the default setting for “Launch Runs”. Wait for it to be finished.

3. Generate the bitstream

- Everytime you modify the .v files or I/O ports, you need to rerun the Synthesis and Implementations to update it.



PROGRAM AND DEBUG

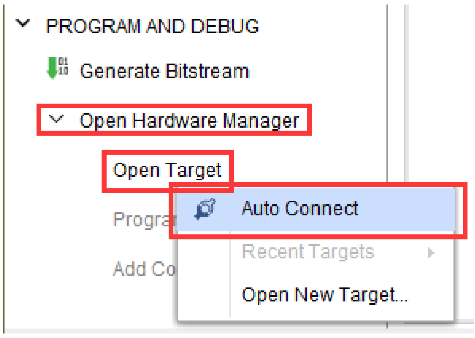
Generate Bitstream

Open Hardware Manager

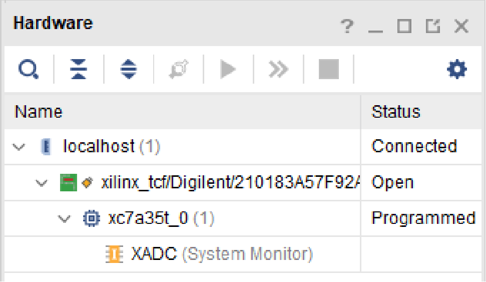
4. Connect the FPGA board to your computer and power it on

- Make sure your are using the proper micro-usb cable, however Vivado *might* be able to recognize the board if you are using a generic charging cable.

Go to “Open Hardware Manager” then click on “Open Target”, select “auto connect”

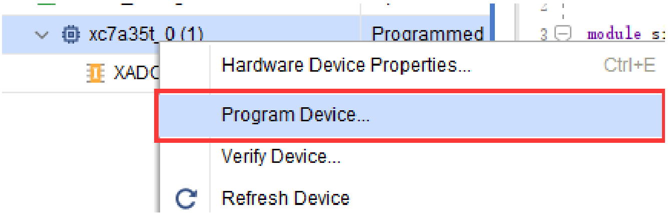


You should to see the “Hardware” window shows up with FPGA board listed in.

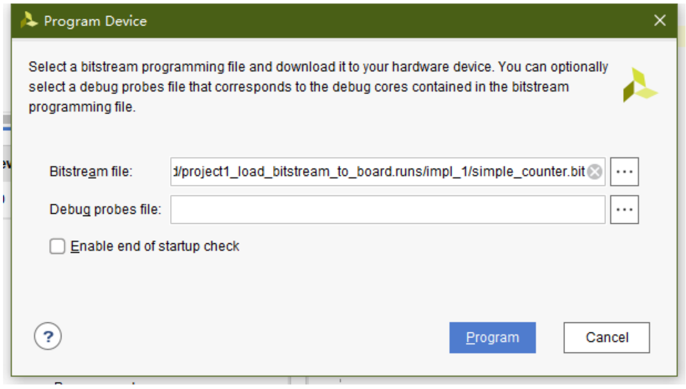


Name	Status
localhost (1)	Connected
xilinx_tcf/Digilent/210183A57F92/	Open
xc7a35t_0 (1)	Programmed
XADC (System Monitor)	

Right click on it and select “Program Device...”



Select your bitstream file to program the FPGA board.

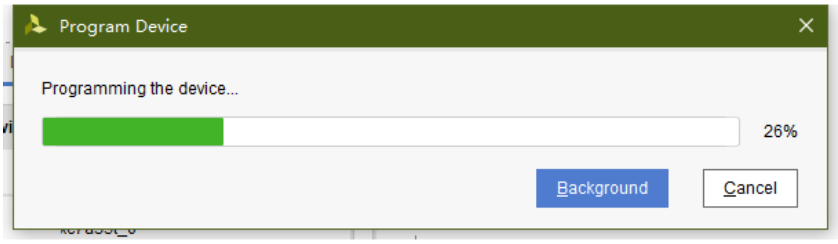

Program Device

Select a bitstream programming file and download it to your hardware device. You can optionally select a debug probes file that corresponds to the debug cores contained in the bitstream programming file.

Bitstream file: 1/project1_load_bitstream_to_board.runs/impl_1/simple_counter.bit

Debug probes file:

□ Enable end of startup check



Program Device

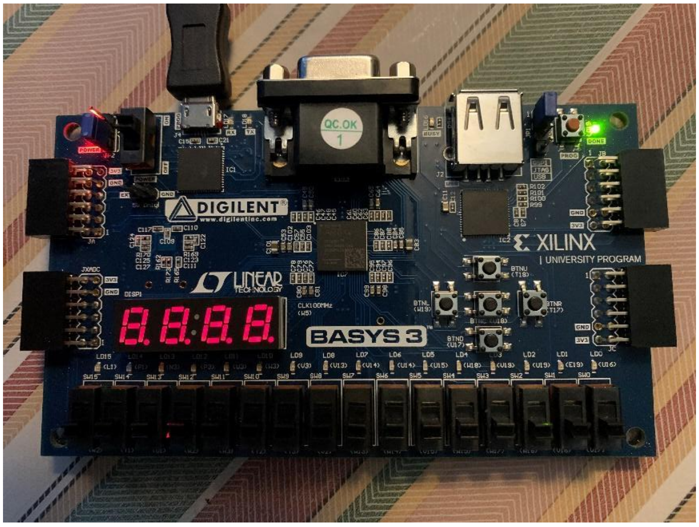
Programming the device...

Now, you have successfully loaded the bitstream to the board.

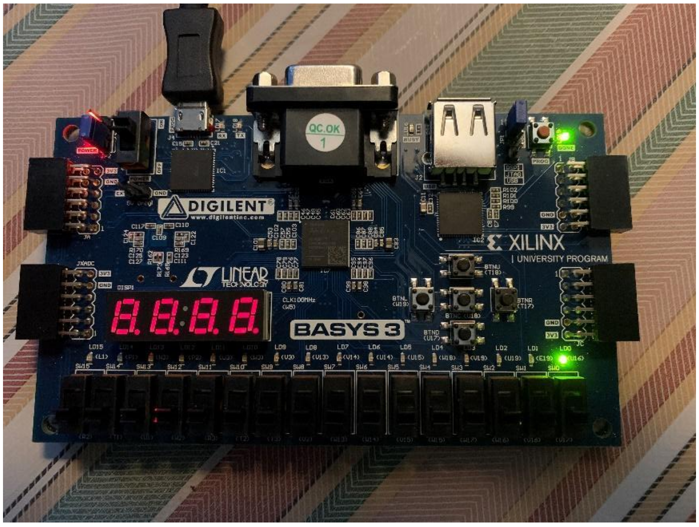
Note: If the device is powered off or the PROG button is pressed, the bitstream will be lost and you will need to program the board again in Vivado.

Demonstration of the "Counter" running on the Basys 3 FPGA Board:

Test on board by pressing button T17:

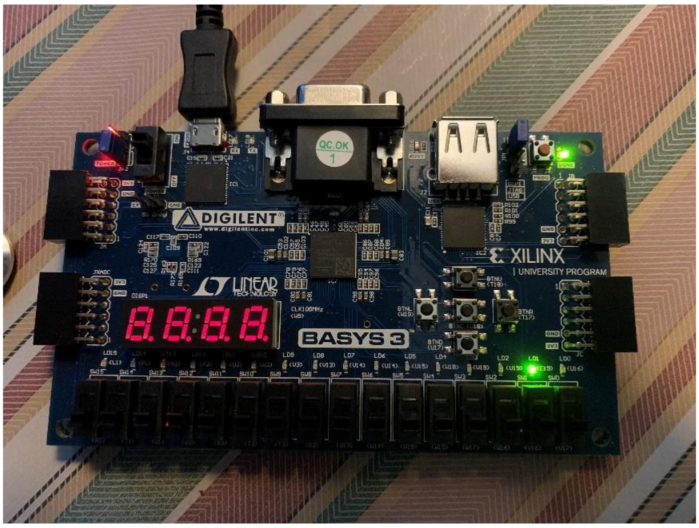


counter: N/A 0 presses



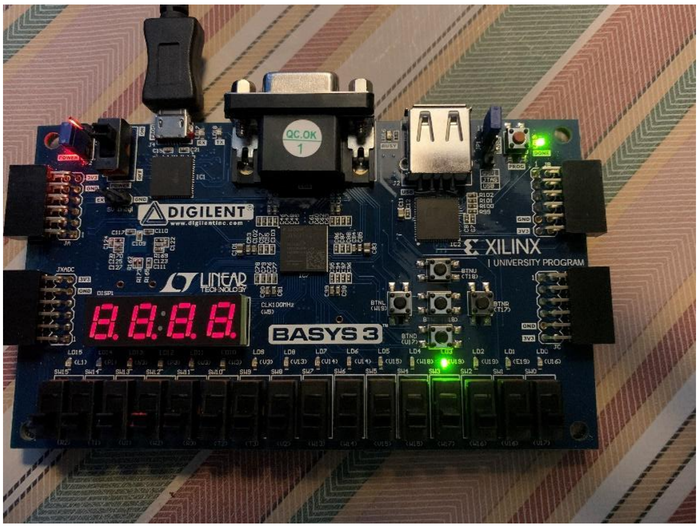
counter: 1 1 press

CSE 490/590 - Project 1 - Interacting with the Basys3 FPGA Board



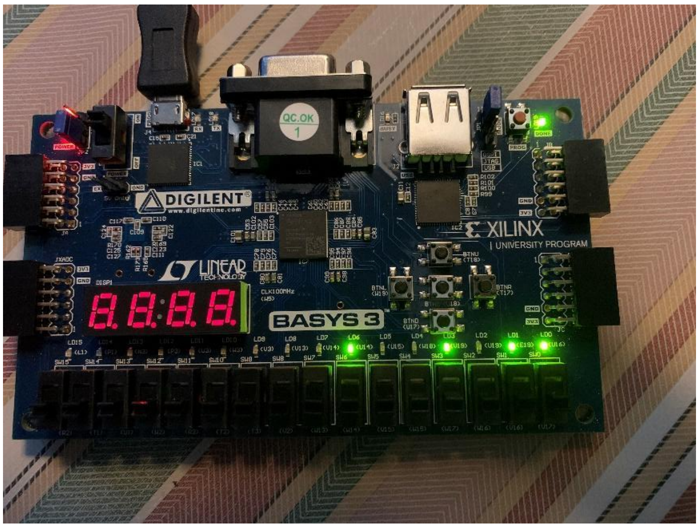
counter: 2   2 presses

……



counter: 8   8 presses

……



counter: 75   75 presses