

Lab 6

Objective: Running hardware design on an FPGA board

1 Overview

In this lab, you will practice synthesizing, downloading, and testing a hardware design on the Basys3 FPGA board. You will be using the same board for your project.

2 Find your groupmates

Split your 4-person project group into *two* 2-person subgroups. You will be working as 2-person group in this lab. At the end, all 4 of you have to complete this lab to borrow the FPGA board for use in your project.

3 Getting the files

Download the files for this lab from

<http://www.eee.hku.hk/~elec3342/fa22/handout/elec3342lab6.tar.gz>

4 Getting the board for this Lab

Take one FPGA board from your TA. You will return this board at Checkoff 1.

5 Creating the project

Create a new project in Vivado, and:

1. Select **Basys3** as the target board.
2. Add `rtl/top.vhd`, `rtl btn_debounce.vhd` and `rtl/lock.vhd` to the project as *design source*.
3. Add `xdc/basys3.xdc` to the project as *constraints*.

The FSM lock is implemented in `rtl/lock.vhd`:

1. There are 3 input buttons that are labeled `btn1`, `btn2` and `btn3` respectively.
2. An output signal `locked` is asserted when the locking mechanism is engaged.
3. On hardware reset, the FSM resets to the locked state. If the user inputs the correct *secret* sequence of buttons with 4 numbers: $1 \rightarrow 3 \rightarrow 2 \rightarrow 1$, the FSM will transfer to unlocked state.
4. Once the lock is unlocked, it remains unlocked until it is locked again by the `clr` signal.
5. The lock continues to read input button until the last 4 numbers are the correct secret sequence. There is no error state.

5.1 Getting the connection right The input and output signals are connected to physical components on the Basys3 board as follows:

Direction	Name	Connected to	Description
Input	btn1	Physical Button BTNL	'1' if button 1 is pressed, '0' otherwise
	btn2	Physical Button BTNC	'1' if button 2 is pressed, '0' otherwise
	btn3	Physical Button BTNR	'1' if button 3 is pressed, '0' otherwise
	clr	Physical Button BTNU	'1' to reset the flip flops to initial value
Output	locked	Physical LED LDO	'1' if the lock is locked, '0' otherwise

The constraints file `xdc/basys3.xdc` defines the connection between the module pins in VHDL and the physical pins on the FPGA board. This file is necessary for the implementation to run successfully.

5.2 Check Yourself

Open the constraints file (`basys3.xdc`) from within Vivado.

- Can you locate the lines that specify how the internal VHDL signals are connected to their physical locations on the board?
- What is the name of the physical connection for the signal `locked`?
- Look on the Basys3 board physically. Can you see the markings on the board that correspond to the buttons and the LEDs?

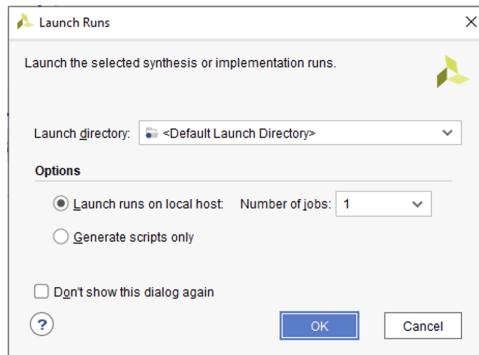
6 Running Synthesis, Implementation, and Bitstream Generation

6.1 Running Synthesis

1. Click **Run Synthesis** in the **Flow Navigator** window:



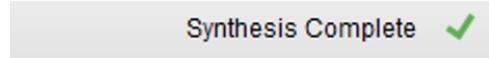
2. Click **OK** to launch the Synthesis:



3. You can find the status in the top-right corner of the Vivado:



4. Wait until the synthesis is done successfully, and you should see:

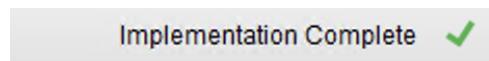


6.2 Running Implementation

1. Click **Run Implementation** in the **Flow Navigator** window:



2. Similarly, click **OK** to launch the Implementation. And check the status in the top-right corner of the Vivado.
3. Wait until the implementation is done successfully, and you should see:

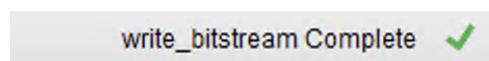


6.3 Generating Bitstream

1. Click **Generate Bitstream** in the **Flow Navigator** window:



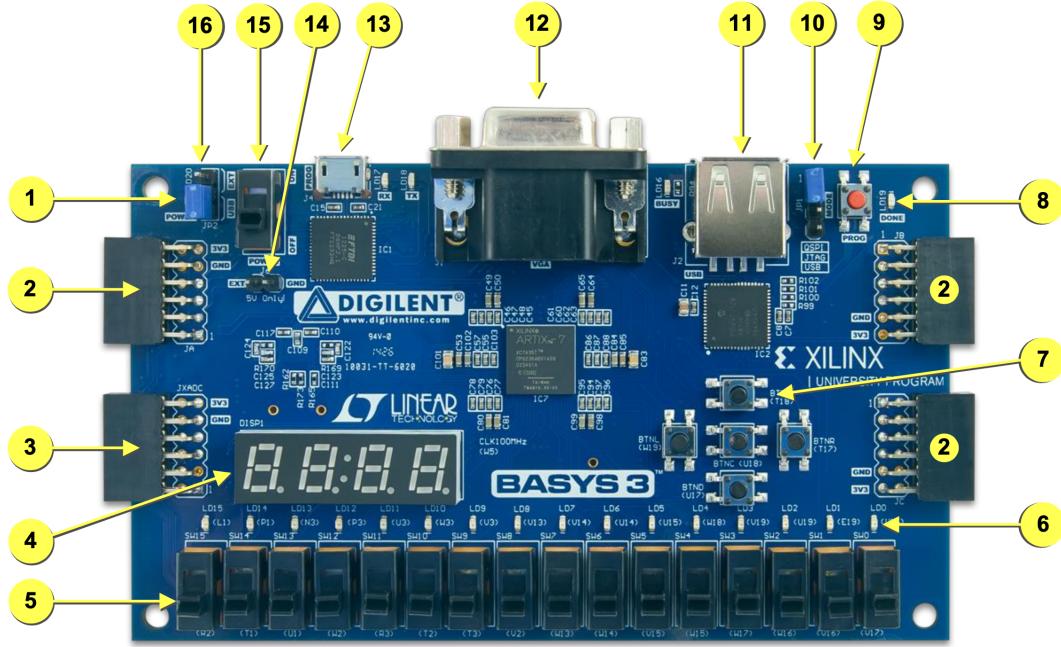
2. Similarly, click **OK** and wait until the bitstream is generated successfully:



3. You should find the bitstream file <your_project_name>.runs/impl_1/top.bit in the project directory.

7 Connecting the Basys3 board

7.1 Checking the Basys3 board The components on Basys3 are shown in the following figure:



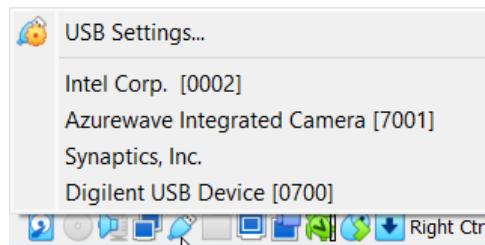
1. Make sure the power jumper(callout 16) is on **USB** so the board will be powered by the USB cable.
2. Make sure the mode jumper(callout 10) is on **JTAG** so you can program the FPGA from Vivado. (The mode jumper in the figure is on **QSPI** which is not correct.)

7.2 Connecting to your PC

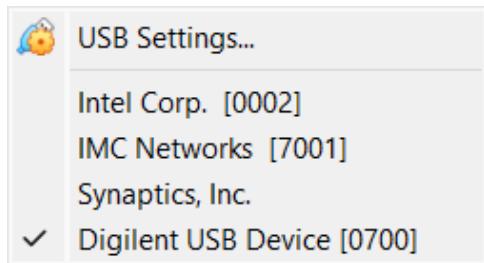
1. Connect the USB cable to the USB port(callout 13) on the Basys3 board. And connect the other end of the USB cable to your PC.
2. Turn power switch(callout 15) to **ON**. The power LED(callout 1) should light up.

7.3 (Optional) Forwarding USB connection to virtual machine For those who are running Vivado in VirtualBox, you need to forward the USB port from the host to the virtual machine:

1. On a running virtual machine, right-click on the USB port icon at right-bottom corner. You should see *Digilent USB Device*:



2. Click on the *Digilent USB Device* and wait until a tick appears on the left:



- Now the Basys3 is connected to the virtual machine. You can double-check it by running `lsusb` in the terminal:

```
student@vm-elec3342:~$ lsusb
Bus 001 Device 003: ID 0403:6010 Future Technology Devices International, Ltd FT2232C/D/H Dual UART/FIFO IC
Bus 001 Device 002: ID 80ee:0021 VirtualBox USB Tablet
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

The first device shown in the figure is the Basys3 board.

.....

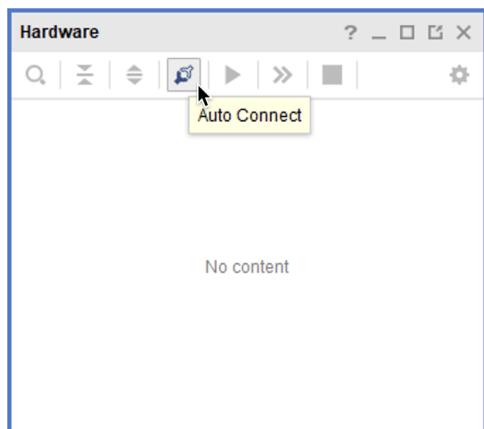
8 Downloading Bitstream

Keep your Basys3 board connected to your PC. And:

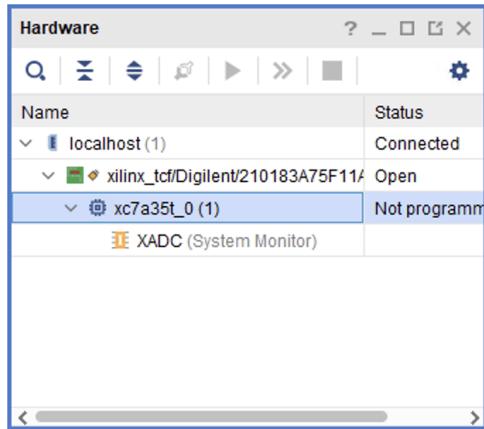
- Click **Open Hardware Manager** in the **Flow Navigator** window:



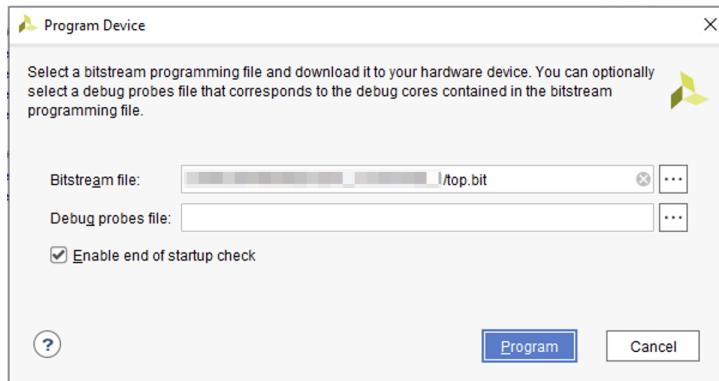
- Click **Auto Connect** in the **Hardware** window:



- You should see a device **xc7a35t_0** is connected:



4. Right-click on the device and click **Program Device**.
5. Vivado will automatically locate generated bitstream file. Now just click **Program** to start programming:



6. The program done LED(callout 8) will light up when the programming is done.
-

9 Testing, Modify, & more Test

9.1 Testing default design Once you have programmed the FPGA as indicated by the done LED (callout 8), your FSM will be running and waiting for your button press. Press the buttons(callout 7) on the board to unlock the lock. Try different combinations to see how your lock behaves. You should see the LED (callout 6) on the board lights up accordingly.

9.2 Modify & Test

Now, go back to step 5.1.

- Modify the constraint file `basys3.xdc` such that when the lock is unlocked, it will light up a different LED, e.g. LD5.
- Rerun the synthesis and implementation steps.
- Download the NEW bitstream to the Basys3 board.
- Test your new design and check if it is behaving as you expected.

10 Checkoff 1

Demonstrate to a TA that you can:

1. Show your TA what changes you have made to the constraint file.
 2. Download the bitstream to the Basys3 board.
 3. Demonstrate on the board how to unlock the lock and lock it again.
 4. Return the board to your TA
-

11 Regroup & Checkout Equipment

Regroup after BOTH subgroups of your Project Group have completed Checkoff 1 of this lab. Before you leave the lab do the following:

- Find your TA to get 1 SET of equipment: FPGA + Microphone.
 - Note the serial number of the set.
 - From Moodle, select “Project Equipment Checkout”
 - Enter the *serial number* in the assignment.

After you have submitted the assignment, you can take the equipment home with you for use in your project. You MUST return the equipment at the end of the semester IN WORKING CONDITION before you will receive your project grade for this course. So please take good care of them.