ENSC 254

Summer 2018

Lab 6

An apology/breather lab: High-speed data exfiltration

Deadline: July 6th, 2018
Last revision: June 29, 2018

Karol Swietlicki
Simon Fraser University

1 Means available to you

Much like the entirety of Lab 4 was used towards analyzing an indicator, this section fulfills an indicator regarding doing research to guide a decision process. I personally don't believe it to be closely connected to the topic course, but it is a requirement that I gather data on the indicators.

The board that you have available has very many means of data transfer available to you.

You have a UART, the LEDs, the OLED screen, the audio jacks, the VGA connector, the PMOD slots, the extension connector and the Ethernet jack.

Perform research on the pros and cons of each of the following methods:

- Using the USB UART, a dedicated device for text transfer.
- Using a constructed vactrol and a microcontroller for extracting data from the changes in the light intensity of the LEDs.
- Displaying the data on the OLED screen and typing it into the computer by hand.
- Transferring it over the audio jacks into an audio recording and then decoding it into text on desktop.
- Attaching a PMOD to the FPGA and extracting the data out the board by creating a custom protocol
- Attaching an extension card to the board and connecting it to the computer and then extract the data using that.
- Using the Ethernet jack, writing in assembly the firmware for connecting the board over the network and transfer the data over HTTP.
- Using the VGA connector, write a display driver, learn the way the VGA standard works and create the image

Decide on a method you'd want to use. Write up your decision process. 200 words should be sufficent. Include at least three references.

2 Extracting data

Regardless of the means that you have determined to be optimal, we will need to use the USB UART to streamline the marking.

When a button is on the FPGA is pressed, write, one letter at a time, the string "Hello, world" to memory location 0xE0001030. This must then be displayed in a receiving application on your desktop computer.

The application on the computer that you will be using is called TeraTerm.

The USB UART runs at 115200 baud, uses 8-bit characters, won't use parity and uses a single bit time as the length of the stop bit.

To do this, follow this list of steps:

1. Connect the FPGA to the power, but don't connect the programming connector.
2. Attach a USB cable from the "UART" connector on the FPGA to any of the USB ports on the computer. Turn the FPGA on. Version 4.5 of the FPGA software should come up. You don't need to program the FPGA.
3. Open TeraTerm on the desktop computer. You can ignore the error about COM3 not being found.
4. Attempt to open a new connection. From the list of possible connections pick one that has the word USB in it.
5. If there are only COM ports listed and no USB port is anywhere to be seen, disconnect your FPGA from the USB port on the computer, reconnect it to a different one, shut down TeraTerm, reopen TeraTerm and retry the connection procedure. Do this a few times, until a USB UART does eventually appear in the list of ports available. There is no specific number under which the UART will appear, but it will clearly say "USB" on it. Yes, this is uncommonly silly. No, we don't know why you need to do this, it has been a fixture of the lab machines at least as far back as Fall 2017.
6. Once you connect to a USB UART, try typing something into the window. The text you are typing should appear back. This means that your wiring is correct and that the driver for the UART managed to pick up the FPGA correctly.
7. While keeping the USB UART connection in place, bring up your board as usual, including attaching a USB cable to the PROG connector and running the Xilinx SDK. You can write your code now.
8. In your assembly code prepare the UART component by setting up the registers as follows:
   (a) Into the mode register located at 0xE0001004 write the bit pattern which

will set up the transmitter to follow the convention that TeraTerm is expecting. (0x20)

(b) Into the baud rate generator located at 0xE0001018 write the CD value to start the baud rate setup. (You want to use 62, in decimal).

(c) Into the baud rate divider located at 0xE0001034 write the BDIV value to complete the setup of the baud rate (6, in decimal).

(d) Into the control register located at 0xE0001000 write the bit pattern which will enable and reset the UART. (0x117)

9. In your assembly code have a loop waiting for a button press.
10. When you detect a press, write the ASCII value for the capital letter H to memory location 0xE0001030.
11. A character for that letter should now appear on the desktop computer.
12. Celebrate being almost done with a lab that hopefully won't take forever!
13. Do that process for the other letters of the two words.

3 High-speed data extraction

Transfer the words "Hello, World" as quickly as possible, one hundred thousand times. This should run in under a minute. Use newlines and carriage returns (ASCII characters with decimal numbers 10 and 13) to separate each line. You can research the exact difference between the two characters on your own.

The only important consideration here is that you cannot overflow the UART's buffer. Your code should check for the transmit buffer being either nearly full or completely full. Should the buffer be in risk of overrunning, you need to wait it out.

It is your job to read the technical documentation of the UART to make sure that you don't overfeed it. Look up the Zynq Technical Reference Manual UG585. Section B.33 and Chapter 19 are all you need.

4 The deliverable

Include a short document in which you explain your reasoning for choosing the means of transfer.

Also include the code that, when run, outputs "Hello, World" 100000 times over the UART.

This lab was already cut down / rewritten 3 times in response to your feedback to labs 4 and 5. The content will stand no matter what and the deadline is pretty firm.