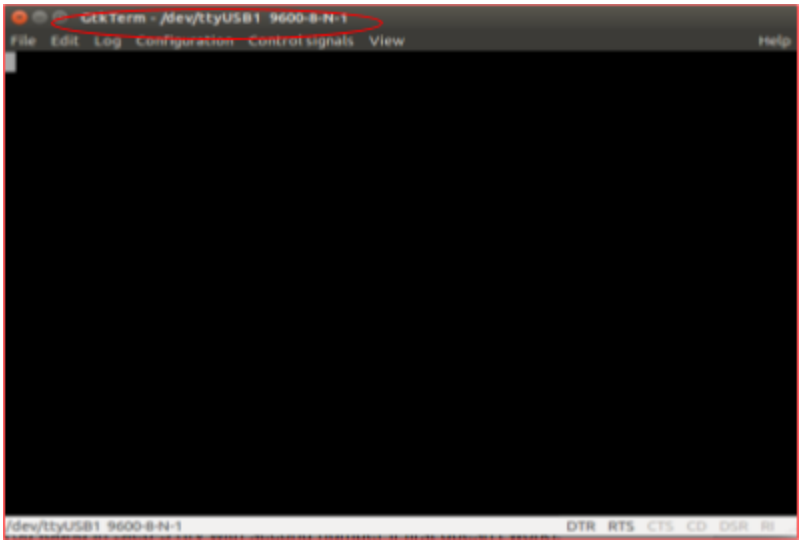
**Instructions to use Serial Terminal interface for COL216 lab assignment 4**

This document explains how to write the content to the FPGA BRAM using serial terminal interface and how to read out the memory contents using the same.

1) Import all the files and IP given along with your Datapath code to make a vivado project.
2) The released code contains the top level testbench named “Lab4_TB.vhd”. Instantiate your Datapath top module in this file. The sample connections are already given.
3) Generate the bit file with this testbench as top module.
4) While generating the bit file, initialize the instruction/data memory in datapath with the test instruction, you want to test your code with.
5) Make sure to add enough instructions for each operation.
6) This Testbench has an additional memory in which you can store control signals to derive your data path through UART.
7) Connect your Basys3 board to the PC and power it on. Download the generated bit file to the FPGA.
8) The Test bench code contains a UART receiver module which can receive data from PC through the same USB cable connected for downloading the bit file.
9) Press U18(BTNC) to reset the FPGA device after programming the bit file. Make sure to keep SW0, SW1, SW2 in off position while resetting.
10) We will demonstrate the rest procedure using GtkTerm tool, but you can use any Serial port based terminal for sending your binary file containing control signals for testing.
11) Install GtkTerm using sudo apt-get install gtkterm. It is already installed on the lab machines.
12) Open terminal and type gtkterm to open gtkterm GUI.
13) Now, you need to find out which USB port of the PC reflects the serial communication link. For this, you need to do the below:
    a) Type dmesg | tail and look for last message similar to “usb 1-1.4: FTDI USB Serial Device converter now attached to **ttyUSB1**”.
    b) This indicates that the serial port used is named **ttyUSB1**. Look for the particular port in your machine.
14) Now, go to the Configuration tab of GtkTerm and select Port to configure the serial port settings.
    - ❏ Set Port as /dev/**ttyUSB1** (i.e. the particular port you got in message).
    - ❏ Set Baud as 9600, Parity as none, Bits as 8, Stopbits as 1, and Flow control as none.

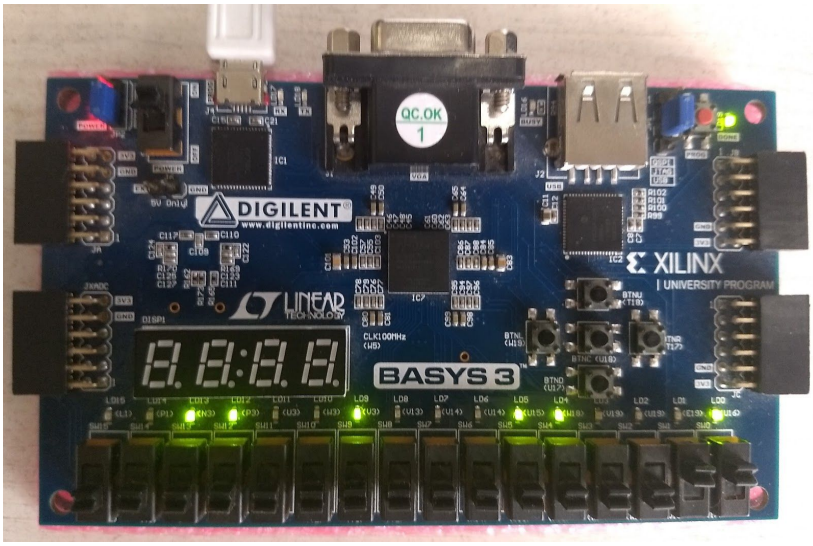

Click OK button once the settings are done. (NOTE: The title bar should reflect the new configuration)



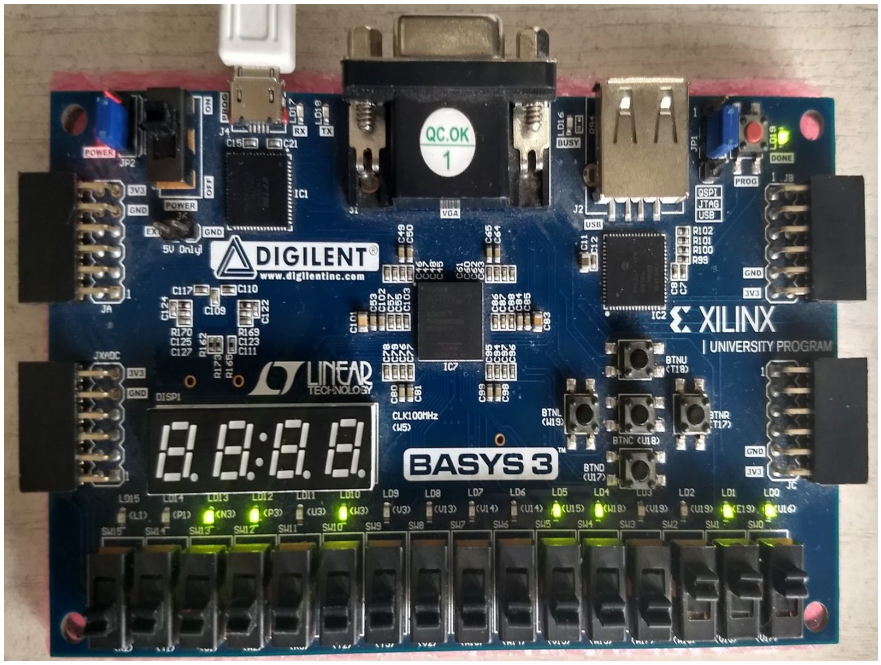

15) Click on **Configuration->Local echo** tab to see whatever you type on gtkterm terminal.
16) Now you need to send a binary file containing set of control signals using GtkTerm. You can create this file using any binary file editor. We will suggest using online tool "https://hexed.it/". You can enter your data and then export it to save to your local drive.
17) Turn on the UART receiver on the board by making turning on switch 0 (SW0) on board.
18) Now, to send the file to the Testbench memory, click on **File->Send raw file** tab and select the file you have created. Note that while transferring the file, the LED LD18 with label TX blinks which indicate that the file is being written to the memory.
19) Testbench code reads the control signals from its memory and display the output on the LEDs for your checking.
20) Test bench reads the contents of the memory in step wise manner i.e. reading the next address on every press of button BTND.
21) By default the contents of address 0 are being read.
22) Each address in the memory stores 32 bits of data. To display first 16 bits (i.e. from 0 to 15) of the data set SW1 and SW2 to "10" position. To display last 16 bits (i.e. from 16 to 31) of the data set SW1 and SW2 to "11" position. For example: address 0 contains:

| | | | |
|---|---|---|---|
| 0x34 | 0x33 | 0x32 | 0x31 |

LD0 - LD7 will display 0x31, LD8 - LD15 will display 0x32 (when the switches 1 and 2 are in position “10”)




LD0 - LD7 will display 0x33, LD8 - LD15 will display 0x34 (when the switches 1 and 2 are in position “11”)




23) To reset the memory, in case required, press button BTNC.