KU LEUVEN

FACULTY OF ENGINEERING TECHNOLOGY
CAMPUS GROEP T LEUVEN

Lab #5
Digital Design Concepts

Final Project – Hexadecimal Editor in Verilog

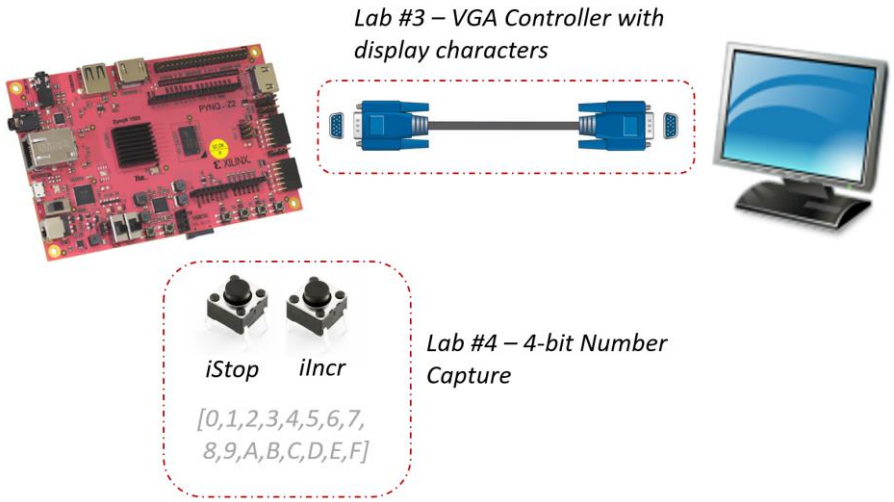
Josep Balasch, Yuri Cauwerts, Gil Vranken, Martin Andries

Academic year 2021-2022

Content

Introduction

In Lab #5 of Digital Design Concepts, we are going to develop the **final project** of the sessions: a hexadecimal editor. For this, you will need to integrate and extend the designs from Lab #3 (VGA controller that displays characters on the monitor) and Lab #4 (4-bit number capture module). The basic concept is simple: a sequence of numbers from 0 to 15, pressed using the push buttons, will be displayed on the monitor.



The first part of the handout describes the basic functionality of the final assignment. It explains and shows how to **integrate** the 4-bit Number Capture Module with the VGA controller with display characters (block design diagram). For the assignment to work, each number pressed using the *iIncr* and *iStop* buttons should write to the **ScreenBuffer** memory the address of the number pressed. Numbers 0 to 9 will map to characters 0 to 9, while numbers 10-15 will map to characters A to F. The sequence of hexadecimal characters corresponding to the numbers pressed will be displayed on the monitor, starting at position 0.

The second part of the handout describes **additional improvements and features** to the hexadecimal editor. This part is not mandatory, but you can use it to strengthen the submission and get extra points. You can add, for instance, extra functionality such as: resetting the whole screen to a black state on the press of *iRst*, incorporating a blinking cursor, adding special characters (spaces, line breaks, delete characters), or even configuring screen colours.

IMPORTANT ! This final project needs to be submitted for evaluation and presented during Lab #6 (the last session of the course).

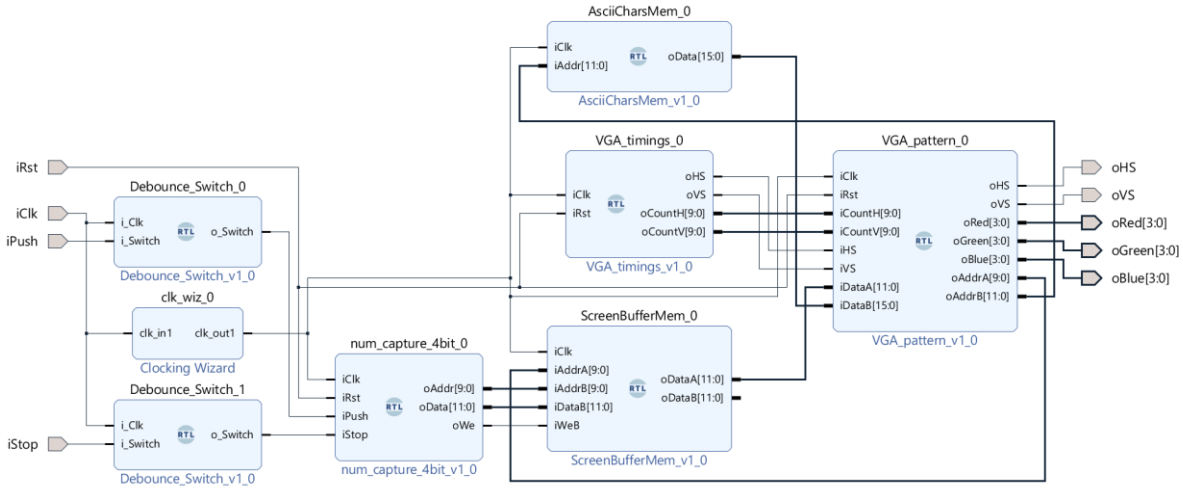
The deadline for submission is 2 days before Lab #6.

More information is provided in the last part of the handout.

1 Final Assignment – Hexadecimal Editor

The goal of the final assignment is to build a hexadecimal editor. You need to add the FSM project developed in Lab #4 (number capture module) to the design developed during Lab #3. The idea is that the FSM module will connect to Port B of the ScreenBufferMem, in order to update which characters are displayed on the screen and at which position.

Your final block design should look similar to the one shown below. Note that you need to modify the interfaces of the *num_capture_4bit* module, so that it can interface with Port B of the ScreenBufferMem via write enable, address and data.



In order to complete the final project in a gradual manner, we recommend to follow these steps:

- Start by re-writing the state transition diagram of your *num_capture_4bit* module. In contrast to Lab #4, you will need to define output signals other than oLeds. For this, you may need to add extra states. Do not start coding before you know how to integrate the desired functionality.

- Once the transition diagram is ready, edit the num_capture module to obtain the functionality of the hexadecimal editor. Instead of outputting the pressed number to the LEDs, you will have to write the address of the character to be printed to the monitor to the ScreenBuffer memory. Note that you will need a register to keep track on the position characters written in the screen (the value oAddr).

In case of errors, you should make use of testbenches to test the functionality of your modules or even whole design.

1.1 (optional – extra points) Extensions to Final Assignment

Once your base final assignment is completed, you can further extend the *num_capture* module to add extra functionalities to the editor. The most important one, is to enable the **iRst** to clear the screen. In other words: when pressing iRst, your monitor should return to the initial blank state with no characters shown.

Other features you can integrate are, for instance, using the remaining push button (alone or in combination with two DIP switches of the Pynq-Z2 labeled SW0 and SW1) to add functionalities to the editor. Example:

- If BTN3=pressed, SW1 = ON and SW0 = ON → add a SPACE character
- If BTN3= pressed, SW1 = ON and SW0 = OFF → move to beginning of next line (line break)
- If BTN3= pressed, SW1 = OFF and SW0 = ON → delete last character
- etc.

You could also create some additional logic that shows a blinking cursor to the current position in the screen. Or find a way to change the color or the characters or the background color of the monitor.

This part is completely left open for you to add whichever feature you think can strengthen your project and give an added value (a.k.a. extra points) to your design. In case you opt to for this, make sure you briefly document all extra features added to your design on a README file, located inside the root Vivado project directory that you submit.

1.2 Final Project Submission

To submit the final assignment, simply create a .zip file of your Vivado project folder and upload it in Toledo. Only one submission per group is required.

Important: Before submitting the .zip file, check that the project folder contains **all source and simulation files**. It happens often that some files are not imported in the project, and this gives errors when evaluating it.

The deadline for submission is set to 2 days (48 hours) before Lab #6.
Example: if your group has Lab #6 on December 9th at 9h00, you need to submit your project the latest on December 7th at 9h00.

1.3 Final Project Presentation

During Lab #6, you will present the project to your lab instructors. This will take approximately 20 minutes. The final project presentation is part of the evaluation for your final lab session grade, along with the project code submission.

In the first half of the presentation, you will give a demo of your project using your Pynq-Z2 and a monitor. If you have added extra features, make sure to showcase them to the lab instructor during this part.

In the second half of the presentation, the lab instructor will ask questions about your project and code. Questions about the project may include design decisions or high-level architecture of your modules. It is advised that you bring the state transition diagrams you have developed during Labs #4 and #5. Questions about the code are meant to test your insight on Verilog.

The schedule for Lab #6 (which group needs to come at which hour for the presentation) will be arranged during Lab #5.