Lab 6

Date: 11/11/2021 **lab duration: 10:00-13:00**

Plan of the day:

[1] To learn how to program VGA display on the university monitor.

[2] Complete the lab tasks by building applications to display different colours on screen as required.

[3] Be familiar with the VGA colour codes in preparation not just for this lab but also for assignment 2.

[3] I plan to spend 15 mins to read the lab 6 guidelines, regarding the knowledge about the Export Hardware step of the Vivado and basic VGA hardware code, another 15 mins about VGA colours in different Gpio VGA_COLOUR regions via the FPGA board. 30 mins to complete the introductory task 1 and test it by connecting FPGA with the monitor and check the VGA colour display. Then with 1 hour to complete task 2 and testing. The final hour is for task 3 and its testing.

Chronological Lab record (24-hour format):

[10:11]

Read introduction paragraph of the lab 6 guide

Opened the Vivado project for lab 6.

I tried to export hardware here but failed, I consulted a Ta and he told me to read the first section of the lab 6 guideline first.

I read the first two pages of the lab guide, I discovered that the problem was that the hardware did not generate bitstream at the moment.

Clicked on Open block Design under the IP integrator in the Flow Navigator panel to see the

block design of the MicroBlaze IP.

[10:21]

In the project manager of the vivado window, double clicked on VGA_Hardware Verilog code in the Sources window.

Read through page 3, 4 of the lab guide[1], understand the VGA Hardware code.

[10:51]

Clicked on Flow in the menu bar and selected Generate Bitstream.

Exported Hardware and Launched SDK.

Programmed the FPGA.

Connected the FPGA board to the monitor with the VGA cable and selected the alternative monitor display VGA source using the "source" button at the front of the monitor.

I did not really understand the description on the lab 6 guidance here, because the monitors in the Kelvin suite laboratory don't really have obvious source setting buttons, and the touch panels are nameless, I have to try every single buttons to switch the source to VGA input from the FPGA.

So I pressed the fourth button on the monitor, and then the third button -> first button -> third button to switch the source to VGA.

Now I see the required pattern showing 12 regions on the monitor, as described in the lab 6 guide.

Figure 1 in the appendix shows the FPGA VGA connection in the Testing phase.

[11:14] Start Task 2

Created a new application called ColourTheWorld.

Created neccessary source and header files main.c, gpio_init.c, gpio_init.c containing the declaraion and definitions for VGA_REGION, VGA_COLOUR, SLIDE_SWITCHES

Called the initGpio() function in main() and declared two u16 variables slideSwitchIn and region and set them to 0.

My plan is to read the input from the slide switches and write them into these two variables, which will be pass to the VGA Gpios for colour and region settings.

In a while(1) loop, use the right 12 slide switches to set the VGA_COLOUR and the 4 leftmost slide switches to set the VGA_REGION.

Programmed the FPGA, set up Run Configuration and run the application.

The model code in the lab was really helpful, especially the switch() for selecting different VGA regions.

Tested the functions of the application on the monitor.

Figure 2 shows the resulting monitor display for the colourtheworld application in the testing phase. With the region set to 3 in binary and colour set to green(000011110000) using the slide switches

[12:07]

Created a new application named ColourTheWorldAdv.

My plan to accomplish the expected function is to implement an interrupter that has a duration of 0.2 seconds, meaning that if I want the region being displayed to change every second, I need the interrupter to set new colour and region every 5 interrruptCounter counts. An initial VGA colour can be set arbitrarily, and it shall increment by certain value to produce a different colour each time the interrupt sets the next region.

Wrote codes to display a letter G with moving coloured squared on the VGA display, with changing colours.

Chose a suitable interruption timer such that the coloured square moves every one seconds.

Programmed FPGA and set up run configuration before running the application.

Modified the code to display letter O on the monitor with moving squares, this was done by simply modifying the order of regions in the interrupter cycle.

Figure 3 shows parts of the resulting monitor display for the colourtheworldadv application in the testing phase. The application is tested by looking at the changing monitor display, showing a moving letter G and O.

Summary:

All three tasks were successfully completed today. The lab file not being able to export hardware troubled me some time, however, I still managed to read the instructions regarding the hardware file and generating bitstream and completed task 1 in the first hour as planned. Task 2 was nice and easy, I completed in just 30 mins, faster than my original plan. The testing including input different colour and regions with slide switches took another 20 mins. Task 3 was also easy to me on the basis of task 2, it took some time to implement the hwTimerISR() to cycle through the square display in the order I want. I spent one hour on it and testing it on the lab monitor.

Reference:

[1] Prof Tughrul Arslan. (2021)' Laboratory 6 – VGA Applications'. *Engineering Software 3*. The University of Edinburgh.

Appendix

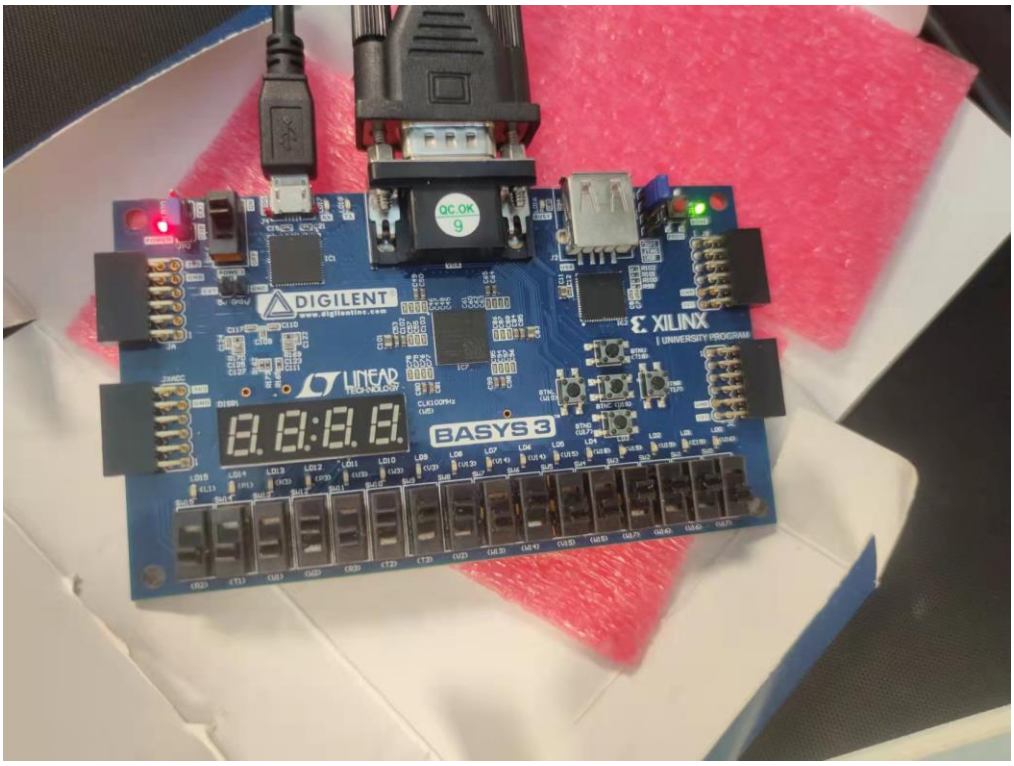

Figure 1: FPGA VGA connection in the Testing phase

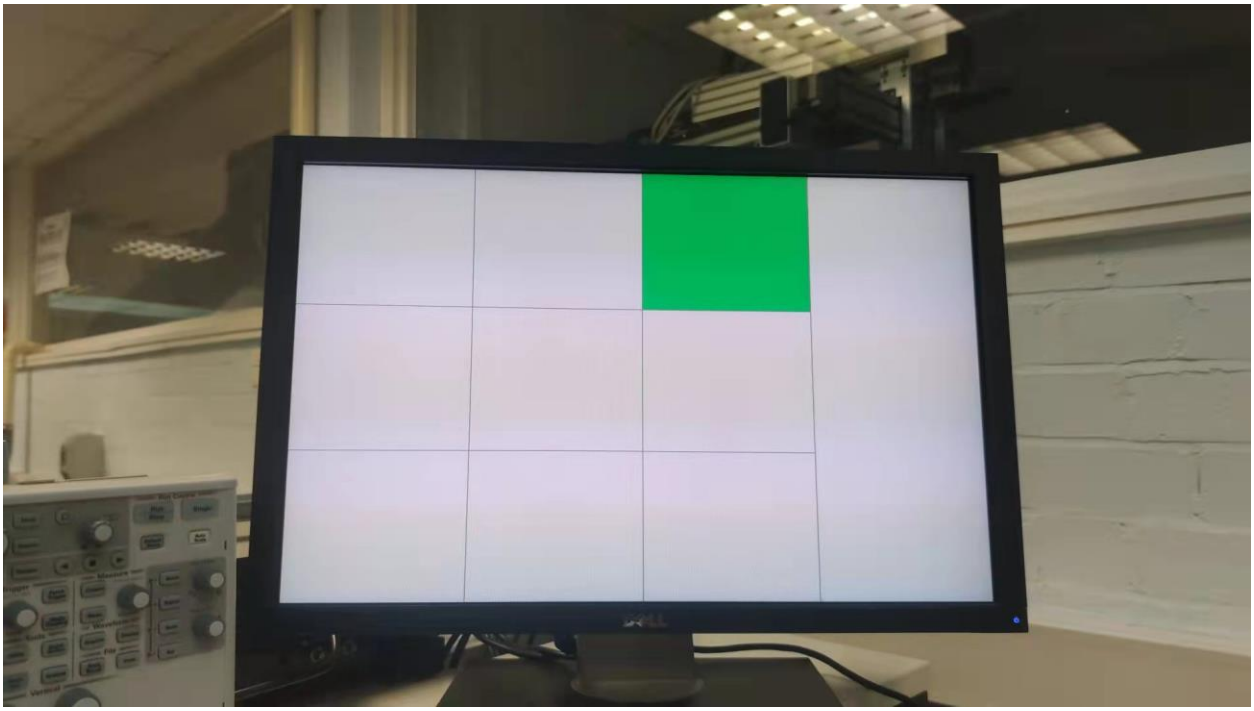

Figure 2 shows the resulting monitor display for the colourtheworld application in the testing phase.

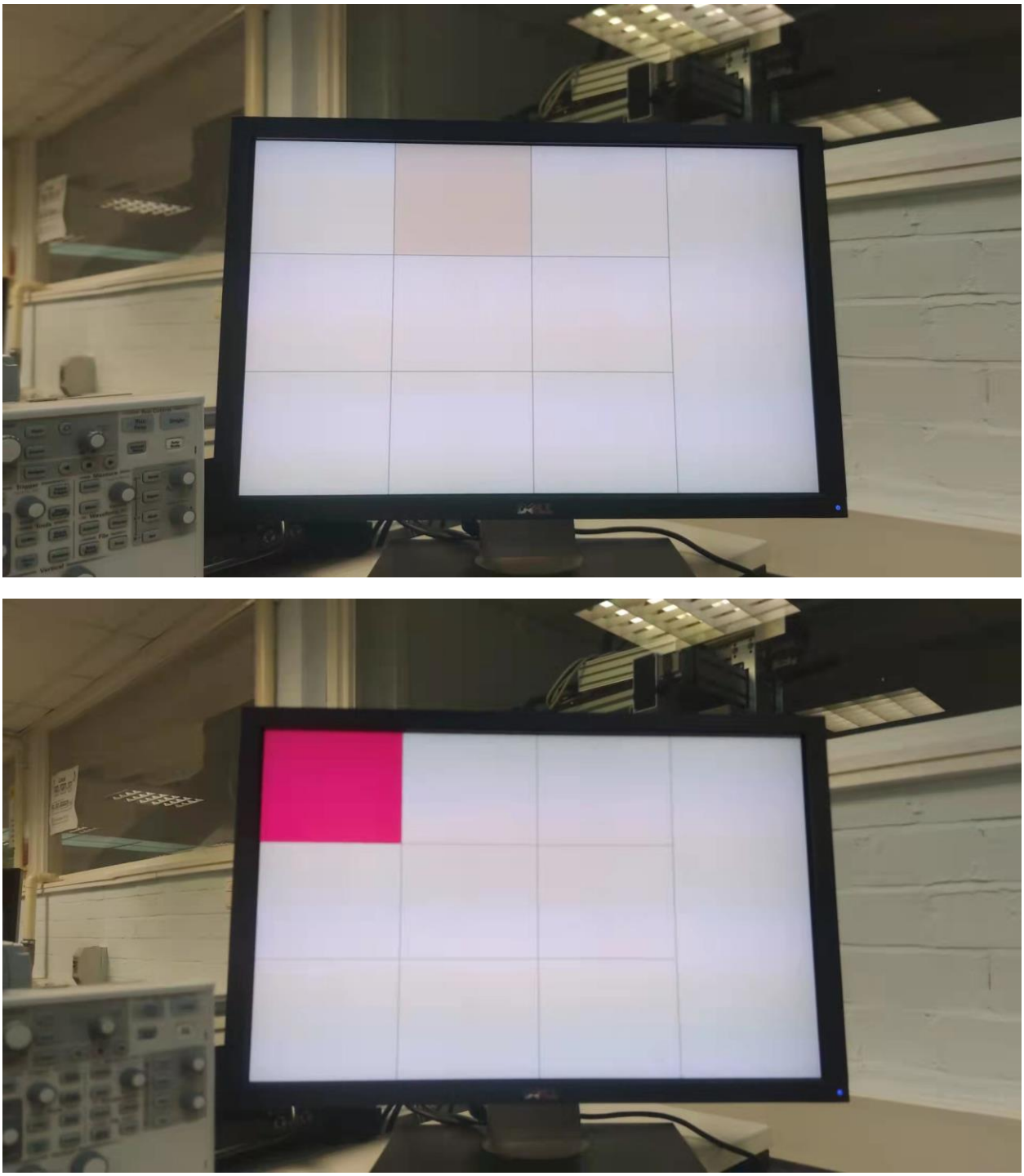

Figure 3 shows parts of the resulting monitor display for the colourtheworldadv application in the testing phase, the square moved from region 2 to region 1.