EE314 DIGITAL ELECTRONICS LABORATORY

SPRING 2017-2018 TERM PROJECT: AN FPGA BASED OSCILLOSCOPE

Introduction:

This document contains the project definition of the EE314 laboratory. Here are the important points about the project:

• Note that this is not a weekend project. Start working on it now. If you would like to test your designs you can use the equipment in the EE314 lab in working hours unless there is a laboratory session proceeding. During weekends, laboratory will be closed. However after last experiment FPGA development boards will be lent to groups (probably 1 board for 3 groups).

• The aim of this project work is to make you more familiar with some subjects you were introduced in digital electronics and logic design class. However, you may need to do some research and study extra material to accomplish the task. This will be a good step for 4th year graduation projects.

• The project groups will contain at most 2 students. Although it is not recommended, you may do your project alone. So, determine your project partner as soon as possible. It is not necessary that your lab partner and project partner is the same person.

• You are free and encouraged to use your own ideas. Although your design approach is not limited, the systems are supposed to be economical.

• All assistants are responsible for the project. Primary contact mechanism with the assistants is via email.

•No early demonstration will be allowed (apart from the crucial reasons, such as Erasmus, foreign student, etc.).

Important Dates:

-14th May: Proposal Report

-2nd -3rd June: Project Demonstrations

-4th June 17:00: Final Report Submissions

Report Format

Proposal Report: The aim of the proposal report is for you to start your research early on so that you can have a solid idea about the project. This report will contain preliminary work on your project. A good report should include your proposed way to solve the problem, the equipment required for the solution, some block diagrams of the overall system and any additional info (circuit schematics, mathematical calculations etc.) you see fit. Maximum page limit for the preliminary report is 4 pages (Times New Roman, 10 point font). Longer reports will be rejected. It is crucial that you determine your project partner, and do some brain storming to come out with solutions well before the preliminary report deadline. Your report is supposed to include state diagram and timing of the projects. You should finish the conceptual design of your project. Moreover, you should implement and simulate at least one measurement algorithm (frequency/amplitude/rms voltage…). Note that your algorithm can change after submitting your proposal report.

You have to upload your proposal report in pdf format to ODTUCLASS until 14th May, 23:59. Late submissions will not be accepted.

Final Report: The final report should be in the IEEE double column paper format (please check the IEEE paper format) and it should not exceed 10 pages in total, any more pages will decrease your grade. The formatting is one of the most important parts of the project. If the final report is not in the IEEE paper format, the project will not be graded and you will get zero from the whole project. Any formatting mistake (such as no figure captions, not referral to the figure in your main text, etc.) will result in grade deduction. You have to upload your proposal report in pdf format to ODTUCLASS until 4th of January, 17:00. Late submissions will not be accepted. Your report should include the following items:

- Theoretical background and literature research
- Design methodology and mathematical analysis of the subsystems
- State diagram and timing diagram of the project
- Simulation results (with Modelsim)*
- Simulation results verifying that your subsystems and overall system is working properly. (with Modelsim)*
- Experimental results
- Comparison of the experimental results with the simulation results and mathematical calculations and explanation of any discrepancies.

*Note that, the most important part of the report is the simulation result and you have to present your simulation result performed in **ModelSim**. Required testbench codes are needed to be written by you and submitted with your report.

You should upload your verilog and testbench codes as seperate files.

Grading:

-Proposal Report: 10 pts

-Project Demonstrations: 50 pts

-Final Report: 40 pts

Project Definition

In this term project, you need to design a FPGA based oscilloscope that can visualize a given single frequency signal with different waveforms (sinusoidal, square or triangular wave) on VGA screen and measure the parameters given in Table 1. Moreover your oscilloscope should satisfy the specifications given below:

- Your screen should be renewed once in one second.
- Your oscilloscope should have two modes: AC and DC mode. In DC mode you will visualize the entire signal given from the signal generator. In AC mode, you will subtract the DC value of the signal and then visualize it on the VGA screen. You can use switches on the FPGA development board to make transition between DC to AC mode.
- Your signal on the screen should start from the mean value of the signal given. You should also have a reference line on the screen which shows the ground level.
- Your oscilloscope should have adjustable time/div and volt/div. We should be able to arrange those using push buttons.
- Your oscilloscope should have an autoscale button. Pushing autoscale button your system should arrange time/div and volt/div values according to the peak-to-peak value and the frequency of the applied signal.

Table 1: The required measurements that the designed oscilloscop can achieve

Parameter	Max-Min values	Resolution
Frequency	DC-20 KHz	10 Hz
V_{pp}	0 to 5 Volt	20 mV
V_{rms}	0 to 5 Volt	20 mV
V_{offset}	0 to 5 Volt	20 mV

The input signal will be given from a signal generator (you may prefer designing a simple oscillator to work at home/dormitory easily.). The signal will be quantized by the ADC (Analog to Digital Converter) which is embedded on our FPGA development board. You should transmit the quantized data and store it in a RAM (Random-Access Memory). After storing the data, you can implement required computation and visualization on VGA screen. In Figure 1, schematic diagram of the project is given.

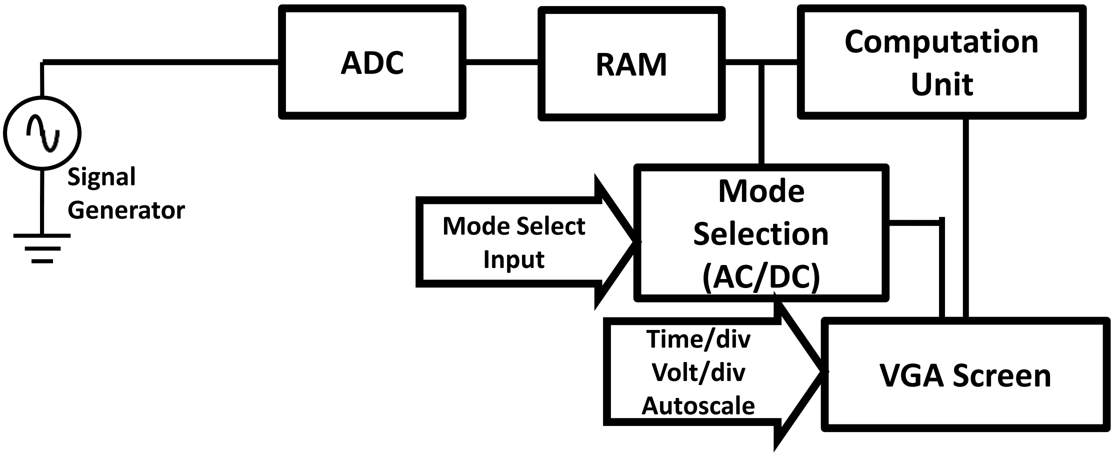


Figure 1: The block diagram of the project

The project sub-blocks are explained in below:

ADC: You are supposed to use the ADC embedded on the FPGA development board. You should design the ADC driver to quantize and store the signal properly. Detailed information about the ADC embedded on FPGA development board can be accessed from the links below:

1) DE-1 SoC User Manual

2) https://class.ee.washington.edu/271/hauck2/de1/adc/ADC_Tutorial.pdf

RAM: The data quantized by the ADC should be stored in a RAM. Since ADC gives a serial output, you may first need to convert serial data to parallel data and then store it. Since data that comes first should be deleted first, you can use FIFO (First-in first-out) structure. For further details about FIFO structure, you should do your research.

Computation unit: In this unit there will be mainly calculations to determine frequency, peak-to-peak voltage, waveform, rms voltage, and offset voltage. You should develop your own algorithms to measure those parameters. This part of the project requires both creativity and analysis ability. Please do not use too complicated algorithms that are not feasible to implement in FPGA with Verilog. Try to make your algorithms as simple, original, and efficient as possible.

VGA Screen: Your computation results and waveform should be visualized on a VGA screen. Detailed information about VGA drive unit will be supplied in appendix.

Bonuses

- Obtain a 2-channel oscilloscope and visualize them in one screen with separate measurements. Triggering with 1st or 2nd channel should be possible if you are designing a 2 channel oscilloscope your oscilloscope should also have XY mode property. (10 pts)

Further reading material with some source codes:

1) https://people.ece.cornell.edu/land/courses/eceprojectsland/STUDENTPROJ/2015to2016/hj424/hj424_report_201605191237.pdf

Appendix: VGA Interface

VGA is a widely used standard in video industry for the transmission of video signals from a computer or microprocessor into a monitor or TV. Each 640x480 image is called a 'frame' and each frame contains 480 lines which are made up of 640 pixels.

The monitor starts displaying each frame by beginning from the first line and then the first pixel of this line. In each line, the display order is from left to right; and each frame is written in an order from top to bottom. So, your first pixel is always at the top left corner, while the last pixel at the bottom right.

You will need to generate an image buffer with at least 640x480 = 307200 bits to store each line and frame in order to form a coherent image; however you will also need to adjust two synchronization signals called HSync (Horizontal Synchronization) and VSync (Vertical Synchronization) in order to see a video. These signals tell the monitor when a line or frame is finished, and the monitor should start from the next line or frame.

As shown in Figure 2, VGA interface is actually very simple, and you will only need to make 3 connections, namely R-G-B. For example, for a white pixel all three inputs should be high, and for a black pixel the inputs should be low. The FPGA cards in the laboratory already have a VGA output port with color outputs, so you will only need to supply the R-G-B data digitally to the VGA port. Necessary pins for these assignments can be found in the user manual. (http://www.terasic.com.tw/cgibin/page/archive_download.pl?Language=English&No=836&FID=eac30a7aaacf5187a4ace0d613cd4676)

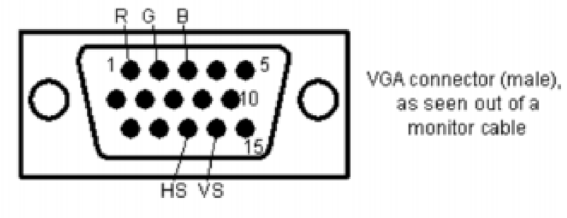


Figure 2: VGA interface.

HSync and VSync: HSync and VSync are necessary in order to tell the monitor to 'start' or 'stop' writing a line or frame. You will need to build the necessary digital blocks in order to correctly form these two signals. These blocks are basically counters with some modifications and are very easy to implement in Verilog. You can see the horizontal and vertical synchronization signals in Figure 3 with the corresponding timing in Table 2.

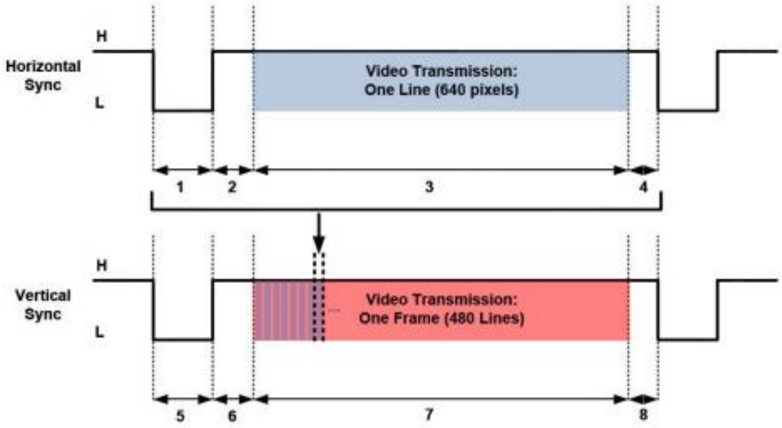
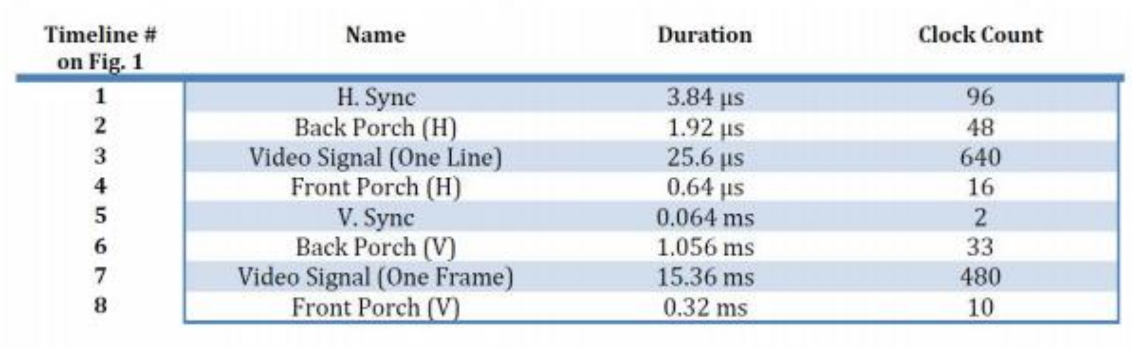


Figure 3: HS and VS.



Timeline # on Fig. 1	Name	Duration	Clock Count
1	H. Sync	3.84 µs	96
2	Back Porch (H)	1.92 µs	48
3	Video Signal (One Line)	25.6 µs	640
4	Front Porch (H)	0.64 µs	16
5	V. Sync	0.064 ms	2
6	Back Porch (V)	1.056 ms	33
7	Video Signal (One Frame)	15.36 ms	480
8	Front Porch (V)	0.32 ms	10

Table 2: Timing.

By observing Figure 3 and Table 2, we can understand that the HSync signal is used to synchronize one line in a frame, while VSync is used to synchronize each frame. Basically, when HSync or VSync is low, the monitor understands that it needs to switch from one line or frame to the next. Back and front porch are idle stages where the monitor is getting ready to write the next pixel or line. They also include 8 pixel and line over scan or ‘border’ pixel/lines outside our standard view of the monitor.

IMPORTANT NOTE: The video input signals (R, G, B) of a VGA monitor should be off (or black) during H. or V. Sync stages, and front/back porch stages. The video input signals should only be active during an active video transmission stage, which are highlighted in Figure 3.

In order to construct these HSync and VSync signals and to achieve transmission of each line/pixel, you will need a 25 MHz clock signal. This will also mean that each pixel will be transmitted at 25 MHz to the monitor during active video stages.

Internal clock information about ALTERA can be found under the Clock Circuitry part of the user manual.

http://www.epanorama.net/documents/pc/vga_timing.html
http://martin.hinner.info/vga/640x480_60.html