DIGITAL OSCILLOSCOPE DESIGN

Cemal Baran Erdoğan cbaran@sabanciuniv.edu

Electronics Engineering, Junior

Hamza Murat Yılmaz hmurat@sabanciuniv.edu

Electronics Engineering, Junior

Yaşar Gürbüz

Electronics Engineering

Ömer Ceylan

Electronics Engineering

Melik Yazıcı

Electronics Engineering

Abstract

The project aims to implement a digital oscilloscope based on Alter DE1-SoC board. This board was chosen since; its 12-bit ADC converter LTC2308 can work with a sample rate up to 500Kps, its VGA port can meet the needs of the project to display sampled signals and it has a low noise. Analog signal is converted to digital one through LTC2308 and register arrays store these sampled values. FPGA is the logic which operates stored values and VGA mode working with a 25MHz pixel clock draws waveforms and other provided parameters on the monitor.

Keywords:

ADC converter, FPGA logic, VGA output, Signal storage, Frequency and Voltage calculation

1. Introduction

This chapter covers general information about the project. It is divided into three parts. First parts is about research question and aim of the project. Second part covers brief explanation over background and literature search. The last part is about our findings and results.

1.1 Research Question

Oscilloscope is one of the main tools of an engineer. It exist in almost every laboratory and used in a lot of projects. Even undergraduate students are using them every week at their laboratory lessons, but there is one problem about oscilloscopes. They are expensive. In order to fix that problem our research question is "Can we build a cheaper but still functional digital oscilloscope?" for this purpose we have chosen a FPGA board. Aim of the project is building a functional digital oscilloscope over Altera DE1-SoC board by just using logic.

1.2 Background

Altera DE1-SoC board has all the required hardware modules for this project. It has an ADC with 12-bit data and 500Ksps sample rate, which can help us to convert width range of analog signal into digital. It includes a FPGA with 50MHz clock, so it can do the required calculations without creating a delay[1]. VGA part can display the results by drawing pictures by pixels. Switches and Keys can be used to control the system. LEDs and seven segment LEDs are useful tools to test and debug the system. So DE1-SoC board is a good option to complete this design.

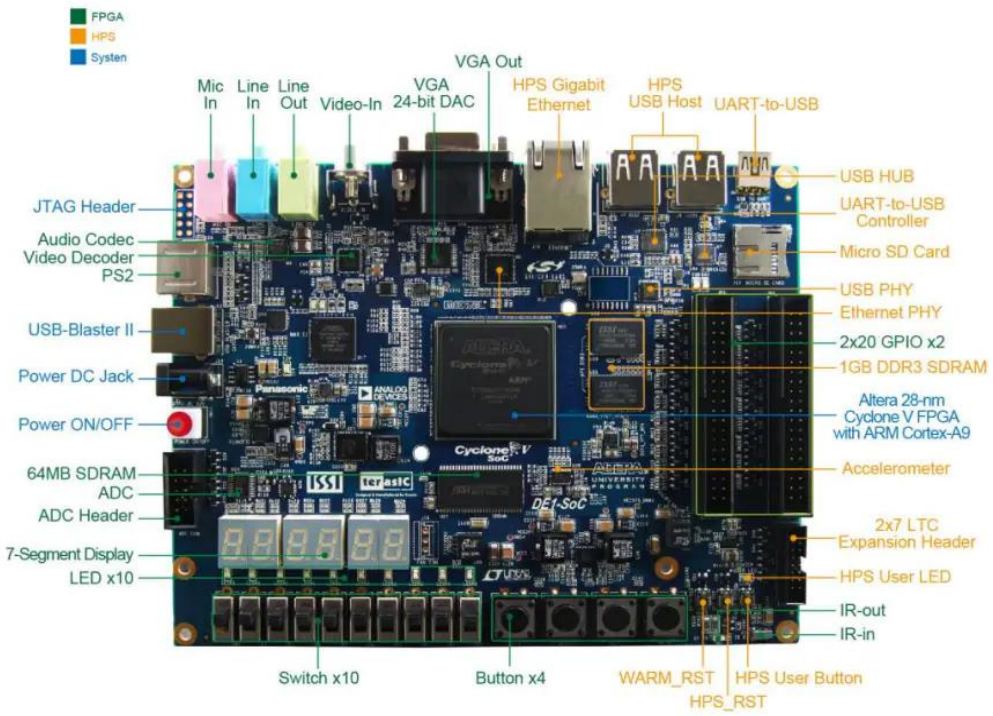

Figure 1: Altera DE1-SoC board

[1] Index of /downloads/cd-rom/de1-soc/, , accessed August 06, 2019, http://download.terasic.com/downloads/cd-rom/de1-soc/.

Hardware languages works differently than software languages. Parallel programing provides multi-tasking. Verilog is used to program FPGAs. We learned Verilog and used Quartus interface to program our FPGA.

1.3 Result

We have implemented digital oscilloscope over DE1-SoC board. It has multiple functionalities as trigger adjustment, horizontal position adjustment, run&stop, AC&DC mode, voltage cursor, reset and channel selection. User can adjust the required settings by using switches and buttons. User can also display the user guide to learn the functionalities of the buttons and switches.

Input signal is displayed at the monitor with essential data. At AC mode frequency and peak to peak voltage value given, at DC mode voltage value given.

Input signal given to 2x5 ADC header is converted by ADC converter at 625Ksps sample rate. 12-bit ADC data is stored into register array and parallel calculations made according to 12-bit value. Results are send to VGA DAC and displayed on the monitor.

2. ADC

First stage of this project is converting analog input signal to digital signal. This chapter will include step by step explanation of how we achieved to get correct ADC data measurements.

2.1 LTC2308

ADC converter used in DE1-SoC board is LTC2308. It has low noise, 8 channel, 12 bit data and 500Ksps sample rate. Its output data clock can operate up to 40MHz. Its pins can measure the voltages between 0 and 4.096 V[2]. One channel will be chosen according to input value and converted to 12bit data according to clock frequency.

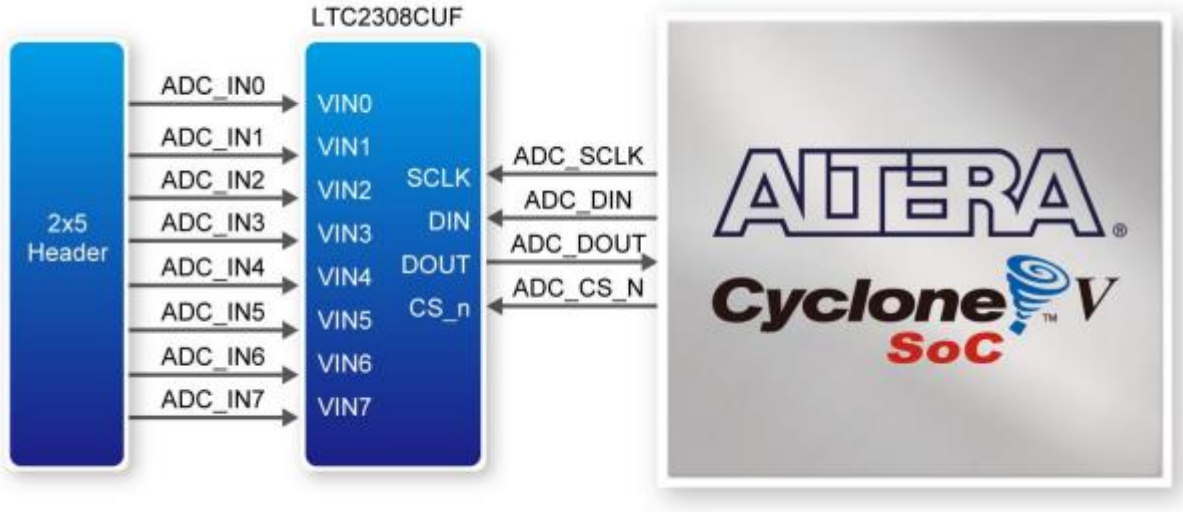

Figure 2: ADC connections with FPGA

[2] Index of /downloads/cd-rom/de1-soc/, , accessed August 06, 2019, http://download.terasic.com/downloads/cd-rom/de1-soc/.

2.2 Clock Frequency

LTC2308 simply works with 500Ksps sample rate but this can be changed by the user. Increasing sample rate can provide us more data and we can achieve better results with more data.

Table at Figure 3 shows the minimal timing values and Figure 4 shows the working procedure of ADC converter. We can minimize the time required to increase the sampling rate. If we give 40MHz as output data clock, we can increase the sampling rate to 625Ksps.

SYMBOL	PARAMETER	CONDITIONS		MIN	TYP	MAX	UNITS
$f_{SMPL(MAX)}$	Maximum Sampling Frequency		●			500	kHz
f_{SCK}	Shift Clock Frequency		●			40	MHz
t_{WHCONV}	CONVST High Time	(Note 9)	●	20			ns
t_{HD}	Hold Time SDI After SCK↑		●	2.5			ns
t_{SUDI}	Setup Time SDI Valid Before SCK↑		●	0			ns
t_{WHCLK}	SCK High Time	$f_{SCK} = f_{SCK(MAX)}$	●	10			ns
t_{WLCLK}	SCK Low Time	$f_{SCK} = f_{SCK(MAX)}$	●	10			ns
$t_{WLCONVST}$	CONVST Low Time During Data Transfer	(Note 9)	●	410			ns
$t_{HCONVST}$	Hold Time CONVST Low After Last SCK↓	(Note 9)	●	20			ns
t_{CONV}	Conversion Time		●		1.3	1.6	µs
t_{ACQ}	Acquisition Time	7th SCK↑ to CONVST↑ (Note 9)	●	240			ns
$t_{REFWAKE}$	REFCOMP Wakeup Time (Note 12)	$C_{REFCOMP}$ = 10µF, C_{REF} = 2.2µF			200		ms
t_{dDO}	SDO Data Valid After SCK↓	C_L = 25pF (Note 9)	●		10.8	12.5	ns
t_{hDO}	SDO Hold Time After SCK↓	C_L = 25pF	●	4			ns
t_{en}	SDO Valid After CONVST↓	C_L = 25pF	●		11	15	ns
t_{dis}	Bus Relinquish Time	C_L = 25pF	●		11	15	ns
t_r	SDO Rise Time	C_L = 25pF			4		ns
t_f	SDO Fall Time	C_L = 25pF			4		ns
t_{CYC}	Total Cycle Time				2		µs

Figure 3

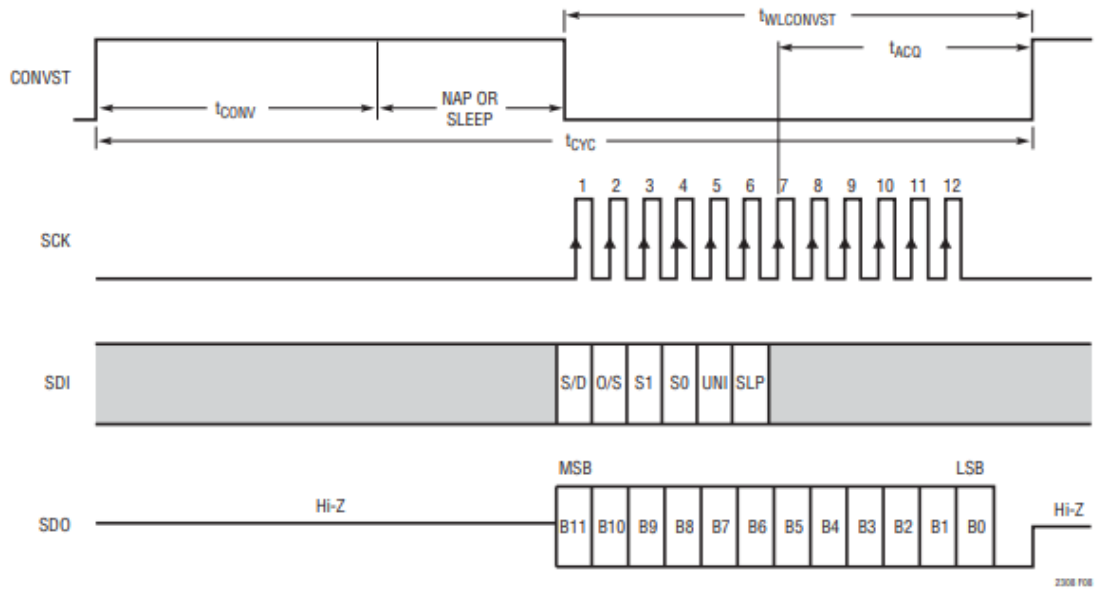

Figure 4

2.3 Data Conversion and Test

First stage of our design is converting the analog data into digital. For this we used Verilog code provided by Altera. This code measures the ADC data and displays it into Nios II window. Our plan is displaying it on a VGA monitor, so we didn't use unnecessary parts of the code for us. Our FPGA provides us 50MHz clock but we need 40MHz clock for our ADC converter and 25MHz clock for VGA part, so PLL module provided us 40MHz and 25MHz clocks. But there was a problem at 25MHz clock, therefore we created it with an always loop.

Our data conversion code has 3 stages. First measuring starts, ADC measures the data given at header. After the measurement conversion starts, so we wait until 12 bit data to be created. Finally we store the raw ADC data into the memory for further usage.

We wanted to increase the sampling rate up to 625Ksps, thus we created a manual reset to restart the ADC converter after 1.6 us[3].

We preferred to use single ended channel configuration to measure the analog input. We assigned switch[2:0] for channel configuration.

After we manage to compile ADC converter code, we continued to test phase. We assigned the measured ADC value to the LEDs and chose channel 0 for the test phase. After that we gave DC voltage to the system and LEDs started to light according to given DC voltage. This was a good start but LEDs were hard to observe and there were just 10 LEDs, therefore we decided to use seven segment LEDs to make a better observation. We created a module to convert 12 bit binary data into decimal and find the value at every digit. We assigned digits into seven segment LEDs. Again we gave a DC voltage and this time we obtained the voltage value at seven segment LEDs. Our result was much more clear and correct. We changed DC value and our design reacted fast and accurate. When we changed the input from DC to AC, all LEDs started to light because the input was changing very fast and LEDs have to react, so this was also a good result for us.

[3] Hanchen Jin, Digital Scope Implemented on Altera DE1-SoC, report, Electrical and Computer Engineering, Cornell University (2016).

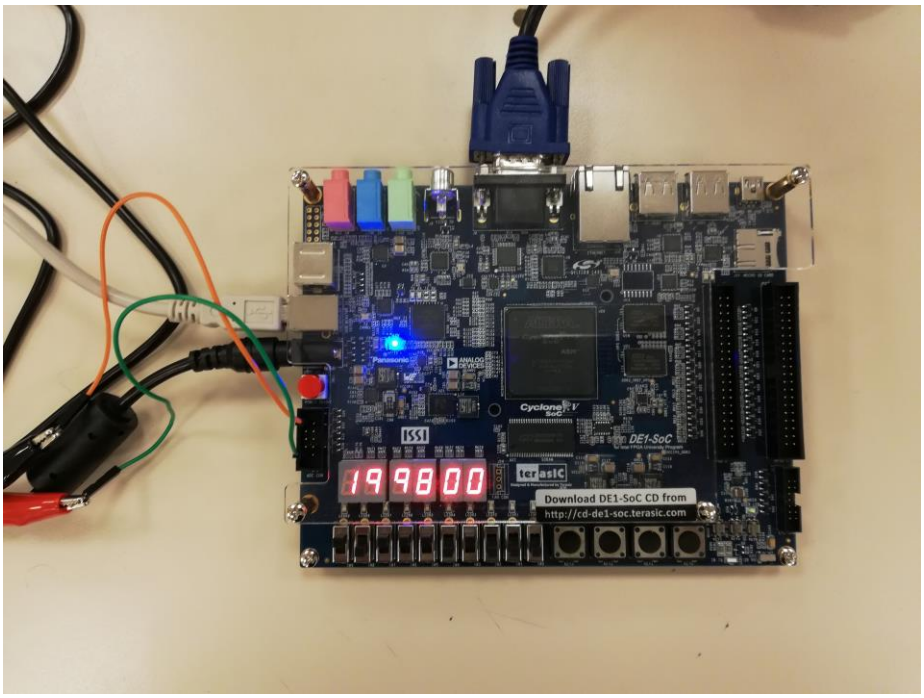

Figure 5: ADC test, voltage value displayed at seven segment LEDs

Tests proved the accuracy of our ADC converter. We completed conversion part successfully, thus we moved on to VGA part to display the signal on a monitor.

3. VGA Display

After sampling process had been completed successfully, VGA port of the DE1-SoC board was used to control what is displayed at each pixel of the monitor. We must store only a few pixel values at memory because most of the pixels are blank in our design. In addition to basic display strategies, calculations related to trigger functions, position adjustments will be covered in this section.

3.1 Specifications of Timing for VGA

In our project, a screen with 640x480 pixels is used to display one picture and images were updated with a speed of 60 Hz. As the datasheet of VGA of DE1-Soc board states, pixel numbers and updating rate are the factors that affect the clock controlling display part of the project. As a result of suggested calculations by datasheet, 25 MHz pixel clock was preferred for this part.[4]

(a)

VGA mode		Horizontal Timing Spec				
Configuration	Resolution(HxV)	a(us)	b(us)	c(us)	d(us)	Pixel clock(MHz)
VGA(60Hz)	640x480	3.8	1.9	25.4	0.6	25

(b)

Configuration	Resolution(HxV)	a(lines)	b(lines)	c(lines)	d(lines)	Pixel clock(MHz)
VGA(60Hz)	640x480	2	33	480	10	25

Figure 6 VGA Timing Specification

[4] Index of /downloads/cd-rom/de1-soc/, , accessed August 06, 2019, http://download.terasic.com/downloads/cd-rom/de1-soc/.

3.2 Basic VGA Project to Test Displaying

Before starting to work on VGA part for our project, a basic VGA project was implemented in order to increase our ability to control VGA modules to generate HSYNC and VSYNS signals. The objective of this test project is dividing the screen as eight identical parts horizontally and displaying different colors at each part. The clock in this project was formed using PLL. Then, the coordinate pixels for x and y directions were stored in registers. Consequently, we could create a project which outputs colored bars on the monitor as figure below shows.

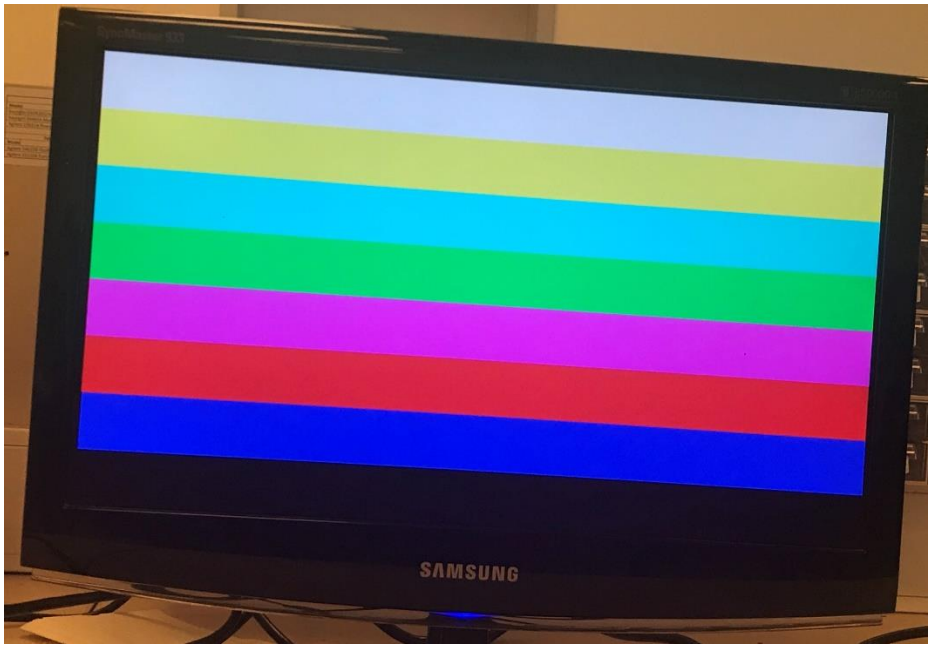

Figure 7: Colored Bar VGA Test

3.3 Display of Static Images

We needed to make a plan of which pixels are left from which parts. According to our design, we put static notes such as title and measuring range together on an area, which is at the bottom of the monitor, formed with 640x50 pixels. In order to display VGA output on this area, we need a font library should be built including 26 letters and some other necessary signals, which takes much time.[5] Because our notes do not change and are not able to be controlled by user, we do not have to go from this way. It is more efficient to build a picture with 640x50 pixels and then display them. For this reason, a bmp file including static notes were formed through Paint program. After that, we needed to convert this image into VGA pixels and MATLAB was used for this aim and a mif file was obtained. Finally, pixel values of these notes were stored in ROM files.

3.4 Display of DC Waveform

Displaying DC signals on the screen is not a challenging job if we can get true ADC values and store them as registers. The working principle of this part is about catching same value at y-axis and

[5] Hanchen Jin, Digital Scope Implemented on Altera DE1-SoC, report, Electrical and Computer Engineering, Cornell University (2016).

output into register and making white this axis. As a result, we could get errors lower than %1. Figure shows one of our measurements that we saw 1.924 V on the screen when a 1.92 V DC signal was given as input.

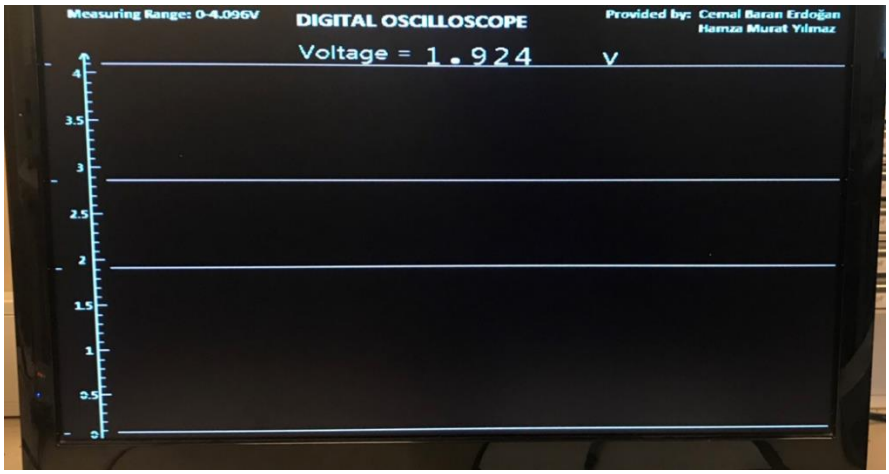

Figure 8: Display of DC signal on our monitor

3.5 Display of AC Waveform

Similar to drawing strategy for DC signals, register arrays coming from sampling process is used to draw AC signals on the monitor. The only difference is a piece of code was written that counted x coordinates in due to the change at voltage value of AC signals with respect to time. As we did in DC displaying part; when the data in register array and value at y-axis are same, we gave a white output at this pixel. Since we have 640x480 pixels screen, consequently we had 640 colored pixels. However, the display of the waveform was not clear enough. The reason and solution for this problem was discussed at next chapters.

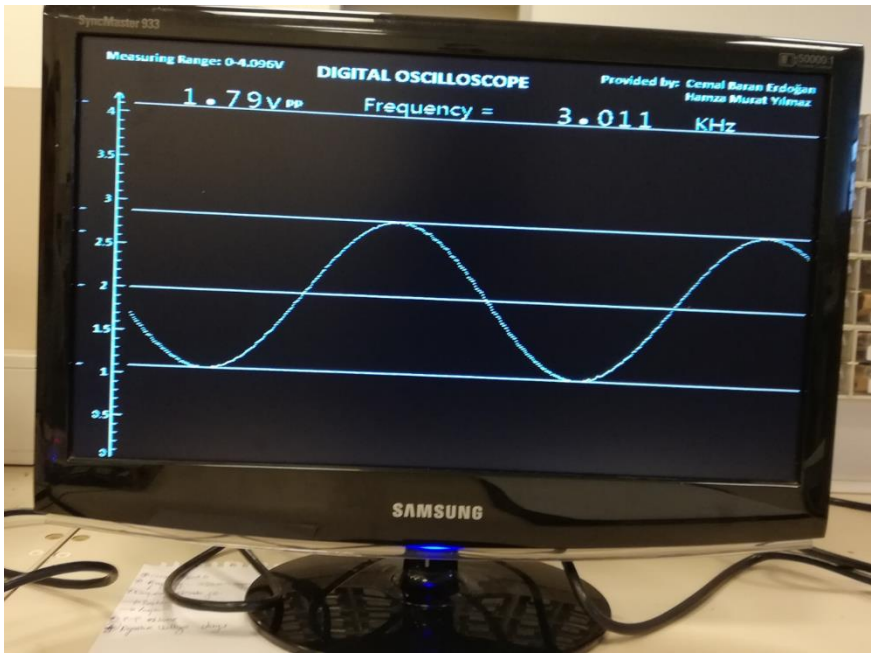

Figure 9: Display of AC signal on the screen

3.6 Signal Triggering

Before we did not add trigger adjustments to the project, the image of AC waveforms had been messy. It is because the time for displaying ADC value at the screen is not equal to update rate of register array storing sampled signals.[6] Thus, trigger function plays an important role for presenting this project with high quality. The working process of triggering starts with the detection of current mode. If the signal is DC, the state machine for triggering will do nothing but; if we have an AC signal, we will continue with the process of storing data in the memory. Then, if the signal is equal to trigger value, state machine continues with next stage. Furthermore; if the clock is at positive edge and 640 samples are took, state machine for triggering comes to the end. However; if any of these cases are not provided, next stage cannot be reached.

Moreover, a piece code is formed in order to allow users to make some adjustment about trigger using one switch and two keys. When the switch6 is on, trigger voltage can be seen on the monitor. Also, users can change trigger voltage by using Key2 and Key3 when this switch is on.

3.7 Horizontal Position Adjustment

Trigger function allowed us to observe more stable waveforms, whereas we had still little bit unclear images in the case of higher frequencies than 12-15 kHz. This is because, there were many cycles on the monitor at these frequencies. That is why, a module that gives us a chance to make some adjustments on signal horizontally through switches and keys was built. Like trigger settings, users can do necessary changes about horizontal position of the signal and get rid of unstable images.

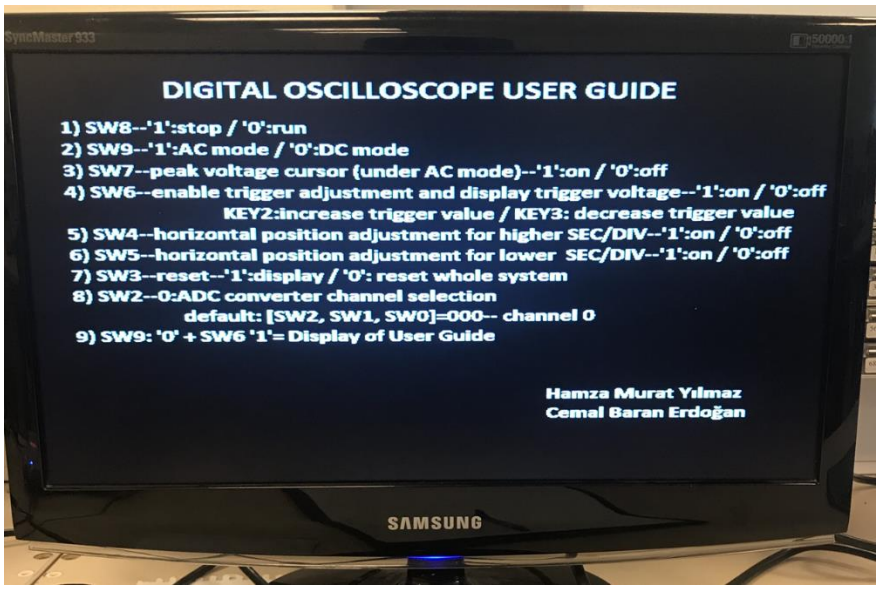

Figure 10 User Guide of Digital Oscilloscope

[6] Hanchen Jin, Digital Scope Implemented on Altera DE1-SoC, report, Electrical and Computer Engineering, Cornell University (2016).

4. Data

We want to make our oscilloscope functional, therefore we wanted to gather as much data as we can. For DC mode we displayed voltage value and for AC mode we displayed peak to peak voltage and frequency value to the user. When we gave more value to the user, user can analyze the signal better and this makes our oscilloscope more functional.

4.1 DC Voltage

Oscilloscope must provide voltage value, when DC input is given. 12 bit ADC data directly gives us the voltage value, thus we designed our system to display this value to the user. We created mif files for "Voltage=", numbers from 0 to 9, "v" and dot[7]. We created space for 3 digits after the dot because our input can vary from 0 to 4.096. We used binary o decimal module and converted 12 bit ADC data into decimals and obtained the numbers at every digit. After that we called the correct file and displayed the voltage value.

4.2 AC Peak to Peak Voltage

When AC input is given, we cannot directly display the voltage value, because voltage changes instantly. But we can create cursors to show user the peak values at the coordinate system and display the peak to peak voltage value. Our design stores voltage values on an array, thus we created template maximum and minimum values. Then we compared these values with the converted values and replaced them if larger or smaller values appear[8].

We created cursors to show maximum and minimum values on the coordinate system as a line and we assigned these cursors to a switch. Than we subtracted minimum value from the maximum value and obtained the peak to peak voltage value. We displayed this value on the monitor and assigned it to a switch.

4.3 AC Frequency Calculation & Display

Another important parameter for AC signals is frequency. We have a simple algorithm which uses trigger value and ADC converters 40MHz frequency to calculate the frequency of the signal.

Frequency calculation algorithm doesn't work at the DC mode, it is only active when AC mode is chosen. We start comparing every sample to the trigger voltage value. If first sample is bigger than trigger value, we wait until sample is smaller than trigger value. When sample is smaller, we start a counter which counts the samples. After a while sample becomes bigger than the trigger but we keep counting until sample is smaller again. At the end counter gives us the number of how many sample is taken from one wave. To calculate the frequency, we divide 40MHz to the counter value and the result gives us the frequency of the input signal.

Divided value is in bits, therefore we used binary to decimal module to convert frequency value and obtain the digits. We display frequency value on the monitor and seven segment LEDs. Our display

[7] Hanchen Jin, Digital Scope Implemented on Altera DE1-SoC, report, Electrical and Computer Engineering, Cornell University (2016).

[8] Hanchen Jin, Digital Scope Implemented on Altera DE1-SoC, report, Electrical and Computer Engineering, Cornell University (2016).

algorithm counts the digits of the frequency value and assigns the proper unit. We created mif files for "Frequency=" and units. We also used the number files and gave enough pixel space to display them on the monitor.

5. Reset Function

Sometimes, the system of the all project can stop or new data cannot be reflected on the monitor because of the wrong inputs or infinite loops at state machines. Especially; when users change current mode from AC to DC, this problem occurs. The reason for this is that state machine of VGA display works based on trigger voltage value on AC mode unlike DC mode and so this difference can sometimes create a confusion for our system. To fix these types of problems, a reset function controlled by a switch is implemented. This is a type of function which finish and then start whole state machine and so provides a proper work of the design again.

6. Discussion and Conclusion

6.1 Summary and Evaluation of Results

In this project, a digital oscilloscope was implemented using DE1-SoC board. ADC converter of board was programmed to sample analog signal and this sampled signal was processed through FPGA logic. Then, VGA port displayed all stored data on monitor. After implementation of project had been completed; we made observations on accuracy of parameters, clearness of waveforms and application of functionalities by giving different DC and AC signals as input to the system. According to our observations, functions of the projects such as trigger voltage settings, peak voltage cursor, reset and horizontal position adjustments work properly. Also, waveforms of signals can be seen clearly as much as we aimed after trigger function was implemented. Moreover, DC signals could be measured under one percent error. Similarly, measurements of AC signals having frequencies lower than 10-15 kHz give errors under one percent. On the contrary, we observed around %2-3 errors for AC signals with higher frequencies. It is probably stemmed from limited sample rate of ADC converter we used.

6.2 Our Gains from This Project

Before the project was started, we increased our knowledge about logic sense and hardware description languages theoretically. Also, we used Verilog language, FPGA sense and DE1-Soc board throughout the project and so we could increase our ability about these concepts. Additionally, we practiced MATLAB and some other software programs to convert files. Besides technical gains, we also experienced how a project process goes on and so this project presented us gains about some important concepts such as time management, problem solution and teamwork ability.

6.3 Future Research Directions

ADC converter of DE1-SoC board limited the accuracy of results of AC signals with high frequencies. For this reason, a different ADC converter having higher sample rate can be used in order to decrease error percentage of measurement results. Furthermore, another input can be added to this system and made a design to show phase difference between different inputs.

References

1) Jin, Hanchen. *Digital Scope Implemented on Altera DE1-SoC*. Report. Electrical and Computer Engineering, Cornell University. 2016. 1-36.

2)Adam Walter, Hart. (n.d.). Digital Oscilloscope and High Speed Analog Conversion Card For The

DE2 Development Kit

https://people.ece.cornell.edu/land/courses/eceprojectsland/STUDENTPROJ/2008to2009

3) Index of /downloads/cd-rom/de1-soc/. Accessed August 06, 2019. http://download.terasic.com/downloads/cd-rom/de1-soc/.