People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdès

Institute of Electrical and Electronic Engineering
Department of Electronics

Project Report Presented in Partial Fulfilment of

the Requirements of the Degree of

'MASTER'

In Electrical and Electronic Engineering

Title:

Digital oscilloscope design using System on Chip

Presented By:

- **BENAZARA Sid Ahmed**
- **MOHAMMEDI Mahdi**

Supervisor:

Dr METIDJI Brahim

Registration Number:……../2021

Dedications

I, mahdi mohammedi , dedicate this humble work to my father and mother who are my stars and the beacons of my hope. to my sister who never fail to show their utmost support and always watch over me.

to my friends B.Nadjib ,R.ismail ,UGC family and B.Sidahmed who have stood by my side when I most needed help and supported me.

I am blessed to have you involved in my life and hope for your continuous patronage. Thank you very much.

"Mahdi"

I, Benazara Sidahmed , dedicate this humble work to my father , mother ,my brothers and my lovely sister.

"sidahmed"

Acknowledgement

We would like to first and foremost thank Allah Almighty who gave us the strength and the courage allowing us to finish this modest work.

We also thank the members of the jury for agreeing to judge and to evaluate our work and for sharing their valuable remarks with us and We are very thankful for all our teachers for their efforts that helped us to accumulate our modest knowledge and improve our skill set while studying at IGEE

Thank you.

Abstract

In this project we design a single channel 50 MHz -10 V to +10 V digital oscilloscope, with liquid crystal display displaying using Video Graphics Array, the oscilloscope is built on a board holding a System on Chip ,the board is zedboard and the chip is Zync7000 System on Chip. The System on Chip is a an Fpga connected to a anARM Microprocessor.

The Fpga is responsible for the sampling and data acquisition and it is programmed using Verilog language using special design platform called Vivado. The ARM microprocessor is programmed to display the signal and control the overall project using C language , the software development kit used to program the Microprocessor is SDK VIVADO.

Table of contents

Dedications I

Acknowledgement II

Abstract III

Tables of contents..... ……IV

List of figures VIII

List of tables X

List of abbreviations..... XI

Chapter I

Oscilloscope Fundamentals

1.1 Introduction 2

1.2 The Oscilloscope.....2

1.3 Types of Oscilloscopes 2

1.3.1 Digital Storage Oscilloscopes 3

1.3.2 Digital Phosphor Oscilloscopes 4

1.3.3 Digital Sampling Oscilloscopes 4

1.4 Oscilloscope Triggering.....5

1.5 Sampling theorem 6

1.6 conclusion 6

Chapter II

Materials and Software

2.1 Introduction.....8

2.2 Hardwar materials.....8

2.2.1 ZedBoard: 8

2.2.2 AD 9288:.....11

2.3 Video Graphics Array 13

2.3.1 VGA timing specification:.....13

2.3.2 VGA pin functions.....15

2.4 Software materials 16

2.4.1. Vivado.....16

2.4.1.1 Overview:.....16

2.4.1.2Features.....17

2.4.1.3Components 17

2.4.1.4Vivado Simulator.....18

2.4.1.5Vivado IP Integrator…….18

2.4.1.6VivadoTclStore……..……19

2.4.2 Intellectual Property.....20

2.4.3 Verilog: 20

2.4.4 VHDL Vs Verilog:.....20

2.4.5 SDK Vivado:.....21

2.5 Summary 22

Chapter III

Design & Implementation

3.1 Introduction.....……23

3.2 Input signal.....……23

3.3 ADC 24

3.4 Trigger Block .....…….25.

3.4.1 Reader Module simulation.....27

3.5 Display Block.....29

3.5.2 GRID Image ……..……..……..……..……...32
3.5.3 VGA Driver simulation.....34
3.6 Processing system Block: 35
3.7 Displaying testing 36
3.8 conclusion 37

Chapter IV

Final results

4.1 Introduction.....39
4.2 conclusion 41

Conclusion and Future Work 42
References.....43

List of figures

Figure 1. 1: Analog oscilloscopes VS digital oscilloscopes 3
Figure 1. 2: The serial-processing architecture of a digital storage oscilloscope (DSO) 4
Figure 1. 3: The parallel-processing architecture of a digital phosphor oscilloscope (DPO... 4
Figure 1. 4: Positive and negative slope triggering 6
Figure 2. 1: Zed board 8
Figure 2. 2: Zed board Block diagram 10
Figure 2. 1: AD9288 Circuit and Block Diagram……11
Figure 2. 4: Normal Operation, Same Clock Channel Timing 12
Figure 2. 5: DE-15 Connector 13
Figure 2. 6: VGA timing specification map 14
Figure 2. 7: Vivado Graphical User Interface 16
Figure 2. 8: Vivado simulator 17
Figure 2. 9: Vivado IP integrator 18
Figure 2. 10: Vivado TCL Store 18
Figure 2. 11: Verilog Vs VHDL 21
Figure 2. 12: SDK Vivado GUI ……:::::: 22
Figure 3. 1: digital storage oscilloscope system key blocks diagram 24
Figure 3. 2: proposed front end circuit 24
Figure 3. 3 ADC and Zedboard interfacing 25
Figure 3. 4: Trigger Block 26
Figure 3. 5: RTL reading & writting process 27
Figure 3. 6: reader_0 RTL 27
Figure 3. 7: Reader Module Simulation 28
Figure 3. 8: Display Block Diagram 29
Figure 3. 9: screen_0 RTL 30
Figure 3. 10: Color_0 RTL……31
Figure 3. 11: Vga_0 RTL 32
Figure 3. 12: the background image of our oscilloscop 32
Figure 3. 13: VGA Driver Simulation 33
Figure 3. 14: The main program flowchart 34
Figure 3. 15: Displaying of generated triangular signal 35

Figure 4. 1: Mini function generator 35
Figure 4. 2: Testing Oscilloscope Circuit 35
Figure 4. 3: Testing with low frequency 36
Figure 4. 4: Testing with high frequency 36

List of tables

Table 2. 1: AD9288 Operational Modes 12
Table 2. 2: Horizontal timing (line) 14
Table 2. 3: Vertical timing (frame) 14
Table 2. 4: ZedBoard DE-15 pins description..... 16
Table 3. 1: ADC and Zedboard interfacing ..... ……..25

List of abbreviations

ADC	Analog to Digital Converter
DC	Direct Current
AC	Alternative Current
DSO	Digital Storage Oscilloscope
DPO	Digital Phosphor Oscilloscope
VGA	Video Graphics Array
LCD	Liquid Crystal Display
CPU	Central Processing Unit
PS	Processing System
IBM	International Business Machines Corporation
HS	Horizontal synchronous
VS	Vertical synchronous
HDL	Hardware Description Language
ISIM	Integrated Science Instrument Module
ISE	Integrated Synthesis Environment
IDE	Integrated Design Environment
IP	*Intellectual Property*
LRM	Language Reference Manual
RTL	Register Transfer Level
VHDL	VHSIC Hardware Description Language
PL	Programmable Logic
SoC	System on Chip
PMOD	Peripheral Module
GPIO	General Purpose Input Output

General Introduction

An ocean wave, earthquake, sonic boom, explosion, sound through air, or the natural frequency of a moving body all move in the shape of a sine wave. Our physical cosmos is filled with energy, vibrating particles, and other unseen forces. Part of the light.

The fundamental Frequency of a particle, part wave, can be seen as color. Sensors are useful in a variety of situations. Turn these pressures into electrical signals that may be observed and studied with an electrical signal analyzer

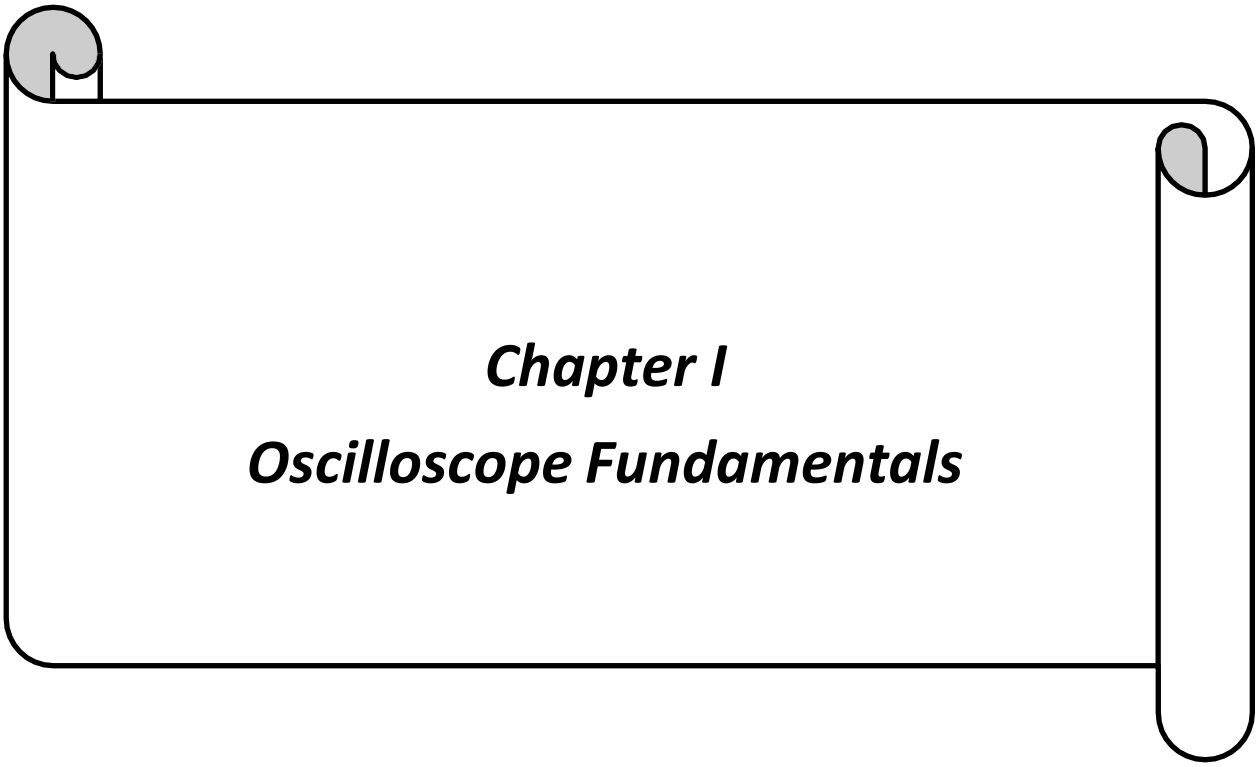
Scientists, engineers, technicians, educators, and others can use oscilloscopes to "see" events that change over time.

Oscilloscopes are essential instruments for anyone involved in the design, manufacture, or maintenance of electronic equipment. Engineers need the greatest tools available to handle their measuring difficulties quickly and correctly in today's fast-paced world. Oscilloscopes, as the engineer's eyes, are essential for tackling today's demanding measuring requirements. An oscilloscope's utility is not confined to the field of electronics. An oscilloscope can measure a wide range of events with the right sensor. A sensor is a device that responds to physical stimuli such as sound, mechanical tension, pressure, light, or heat by generating an electrical signal. A microphone is a device that detects sound and turns it into an electrical signal. Everyone from scientists to television repair specialists uses oscilloscopes.

An oscilloscope is used by an automobile engineer to compare analog data from sensors to serial data from the engine control unit. An oscilloscope is used by a medical researcher to measure brain waves.

The options are limitless. The principles provided in this primer will give you an excellent foundation for learning oscilloscope basics, and the glossary at the end of the book will help you grasp them better.

Unfamiliar terminology will be defined in this primer. This primer is excellent because of the vocabulary and multiple-choice written exercises on oscilloscope theory and controls. help in the classroom No prior understanding of mathematics or electronics is required.



Chapter I

Oscilloscope Fundamentals

1.1 Introduction

Electric engineers and technicians must observe and study the evolution of electric signals over time in order to come up with the best solutions and designs for their technical problems, and they must have the best tools at their disposal to solve their measurement challenges quickly and accurately. The oscilloscope is a critical tool for solving today's demanding measurement difficulties and is a must-have for anybody designing, as well as designing and repairing electronic equipment. An oscilloscope can measure a wide range of phenomena with the right sensor.

1.2 The Oscilloscope

An oscilloscope is a device that displays a graph of an electrical signal. The graph depicts how signals change over time in most applications, with the vertical (Y) axis representing voltage and the horizontal (X) axis representing time. The display's intensity or brightness is commonly referred to as the Z axis.

This basic graph can reveal a lot about a signal, including:

- A signal's timing and voltage values an oscillating signal's frequency
- The signal represents the “moving parts” of a circuit.
- The frequency with which a specific portion of the signal occurs in comparison to other parts of the signal
- Whether or not a faulty component is causing the signal to be distorted
- How much of a signal is direct current (DC) or alternating current (AC).
- How much of the signal is noise and whether the noise is changing with time? [1]

1.3 Types of Oscilloscopes

Analog and digital oscilloscopes are the two main categories of oscilloscopes. Digital oscilloscopes work with discrete binary numbers that represent voltage samples, whereas analogue oscilloscopes work with continuous voltages. each type has unique properties that may make it more or less suitable for various uses

a digital oscilloscope converts the measured voltage into a digital value using an analogue-to-digital converter (ADC). It obtains the waveform as a series of samples and stores them until it has a sufficient number of samples to describe the waveform. The digital oscilloscope then re-assembles the waveform for display on the screen. (Refer to Figure 5) Digital oscilloscopes are divided into two types: digital storage oscilloscopes and digital

storage oscilloscopes. Digital phosphor oscilloscopes and digital sampling oscilloscopes are two types of digital oscilloscopes. [1]

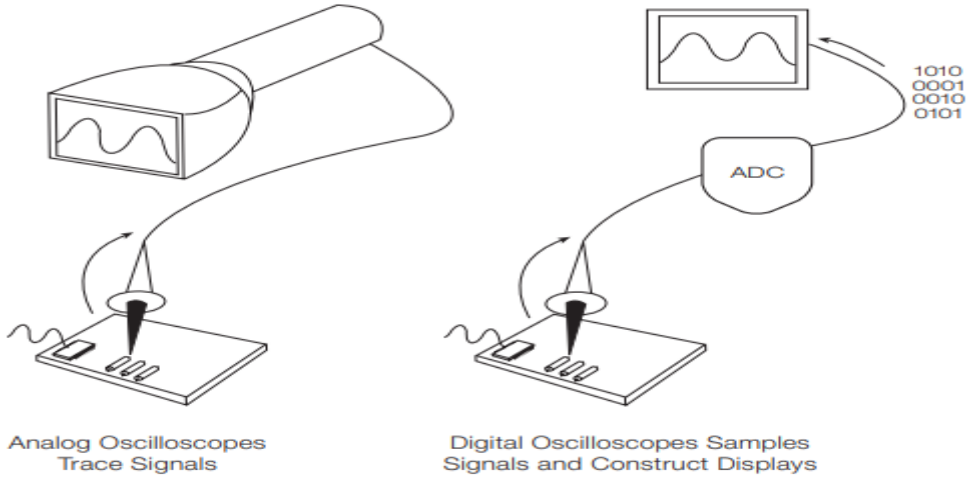


Figure 1. 1: Analog oscilloscopes VS digital oscilloscopes [1]

1.3.1 Digital Storage Oscilloscopes

A digital storage oscilloscope is a type of traditional digital oscilloscope (DSO). Instead of glowing phosphor, it usually uses a raster-type screen for display. DSOs (digital storage oscilloscopes) allow you to catch and view transients, which are events that happen just once. Because the waveform data is saved as a series of binary values in digital form, it may be analyzed, archived, printed, and otherwise processed within the oscilloscope or by an external computer The waveform does not have to be continuous; it can be presented even if the signal is no longer present. Digital storage oscilloscopes, unlike analogue oscilloscopes, offer permanent signal storage as well as sophisticated waveform processing. DSOs, on the other hand, often lack real-time intensity grading and so are unable to communicate various levels of intensity in a live signal. DSO subsystems are similar to analogue oscilloscope subsystems in certain ways. DSOs, on the other hand, include extra data-processing subsystems that collect and display data for the complete waveform. DSO uses a serial-processing architecture to record and display a signal on its screen. Following is a description of this serial-processing architecture. Architecture for Serial Processing The initial (input) stage of a DSO, like that of an analogue oscilloscope, is a vertical amplifier. At this step, you can modify the amplitude and position range using the vertical controller. Following that, the horizontal system's analogue to digital converter (ADC) samples the signal at discrete points in time and converts

the voltage at these times into digital values known as sample points. Digitizing a signal [1]

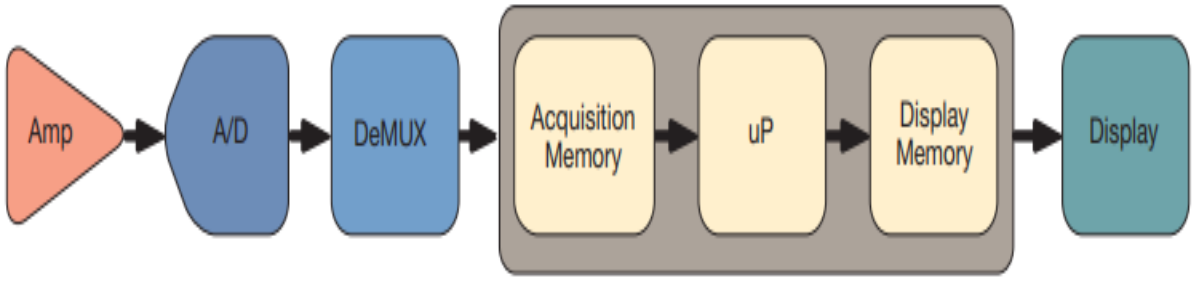


Figure 1. 2: The serial-processing architecture of a digital storage oscilloscope (DSO). [1]

1.3.2 Digital Phosphor Oscilloscopes

The digital phosphor oscilloscope (DPO) takes a fresh look at oscilloscope design. A DPO can use this architecture to provide unique acquisition and display capabilities for precisely reconstructing a signal. While a DSO captures, displays, and analyzes signals using a serial-processing architecture, a DPO does same using a parallel-processing architecture, as shown in figure7. The DPO design uses dedicated ASIC hardware to gather waveform images, resulting in faster waveform capture rates and better signal visualization. This performance raises the chances of seeing transient occurrences like runt pulses, glitches, and transition mistakes that occur in digital systems. Following is a description of this parallel-processing architecture.

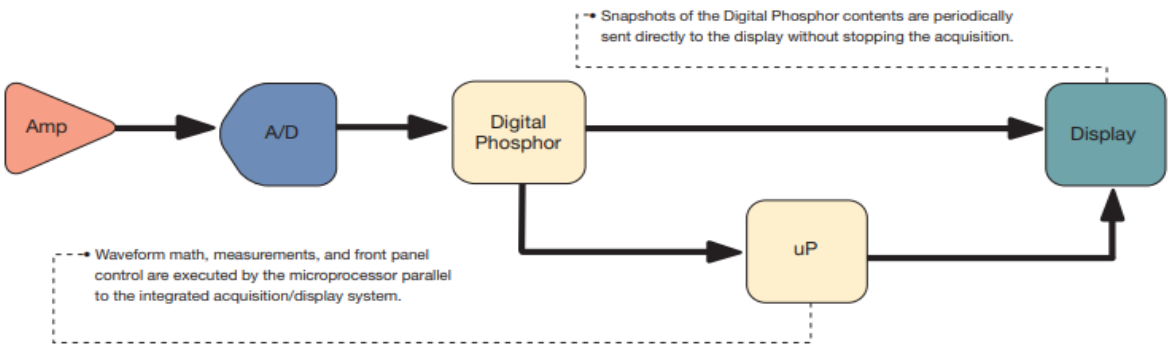


Figure 1. 3: The parallel-processing architecture of a digital phosphor oscilloscope (DPO) [1]

1.3.3 Digital Sampling Oscilloscopes

The oscilloscope may not be able to acquire enough samples in one sweep while measuring high-frequency signals. A digital sampling oscilloscope is useful for recording signals with frequency components that are substantially higher than the sample rate of the oscilloscope. This oscilloscope is capable of measuring signals of up to an order of magnitude

faster than any other oscilloscope. It can achieve bandwidth and high-speed timing ten times higher than conventional oscilloscopes for repeating signals. With bandwidths up to 80 GHz, sequential equivalent-time sampling oscilloscopes are available.

1.4 Oscilloscope Triggering

Continuous capture is the oscilloscope's preferred operational mode since it allows for the immediate display of signal changes. Triggering becomes more crucial with continuous capture. Without proper triggering, a stable signal display cannot happen.

Each frame's triggering is performed by "pinning down" one point on the curve. The trigger point is the location where the pinning occurs. The triggering requirements are defined by three parameters [2]:

1. the pre-triggering percentage,
2. the trigger level
3. the slope.

The pre-triggering percentage identifies the imaginary vertical line on which the trigger point lies. The default value is a pre-triggering percentage of 50%. This default value corresponds to a vertical line that is exactly half way from the left edge of the display area to the right edge of the display area. With a pre-triggering of 50%, exactly one-half of the displayed curve lies to the left of the trigger point. This default value is appropriate for most situations, so you won’t need to change it often. The trigger level is the voltage of the trigger point. If the signal display is to be stable, the trigger level must remain within the excursion range of the signal. Otherwise we are asking for the impossible: to trigger at a level never reached by the signal. The pre-triggering percentage and the trigger level together identify the trigger point. The pre-triggering percentage is essentially the x coordinate, and the trigger level is the y coordinate of the trigger point. The third parameter, the slope, is a descriptor, not a number. This descriptor has one of two values: positive slope or negative slope. If a positive slope is specified, it means that the curve must have a positive slope while passing through the trigger point. Otherwise, the curve must have a negative slope while passing through the trigger point [2].

Following Figure shows you how the trigger slope and level settings determine how a waveform is displayed.

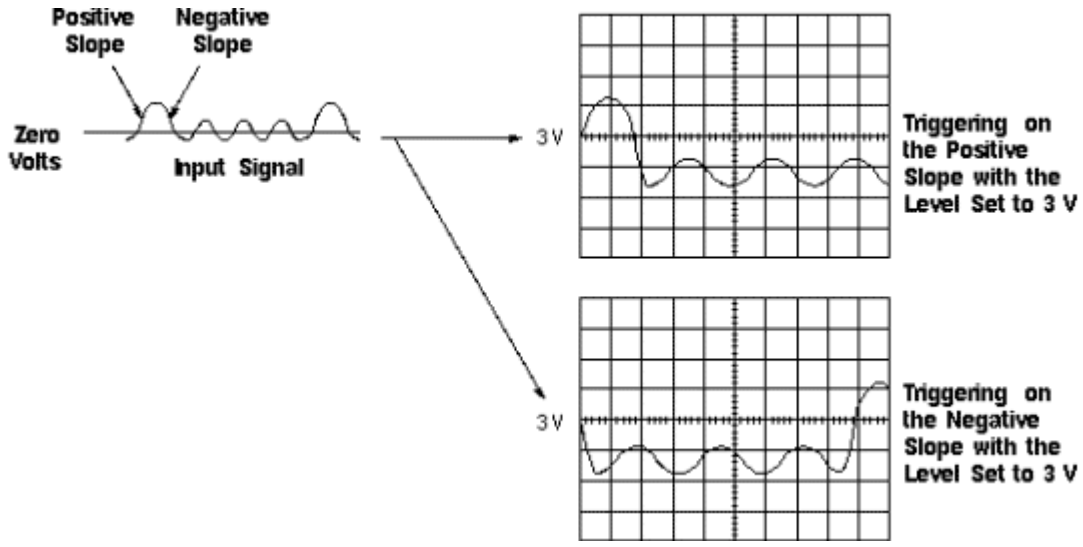


Figure 1. 4: Positive and negative slope triggering [3]

1.5 Sampling theorem:

The sampling theorem specifies the minimum-sampling rate at which a continuous-time signal needs to be uniformly sampled so that the original signal can be completely recovered or reconstructed by these samples alone [4].

If a continuous time signal contains no frequency components higher than W Hz, then it can be completely determined by uniform samples taken at a rate fs samples per second where

Fs >2W

The minimum sampling rate allowed by the sampling theorem (fs = 2W) is called the Nyquist rate [4].

1.6 Conclusion:

Hopefully, after going through the entire chapter, we now know what is the importance of the oscilloscope. That being said, we wish to conclude here by hoping that we were able to provide you enough information regarding the device and what each of the functions of the device is.

Chapter II

Materials and Software

2.1 Introduction

The oscilloscope used in this project is a simple digital oscilloscope with a VGA connector for LCD display. It consists of an FPGA coupled to a chip-based ARM CPU. The data from an ADC will be received by this chip, which will then show the waveform on an LCD.

2.2 Hardwar materials:

2.2.1 ZedBoard:

The Xilinx Zynq-7000 Extensible Processing Platform is the basis for the ZedBoard, an evaluation and development board. The Zynq-7000 EPP combines a twin Corex-A9 Processing System (PS) with 85,000 Series-7 Programmable Logic (PL) cells, making it suitable for a wide range of applications. The ZedBoard's extensive set of on-board peripherals and extension options make it a suitable platform for new designers [5].

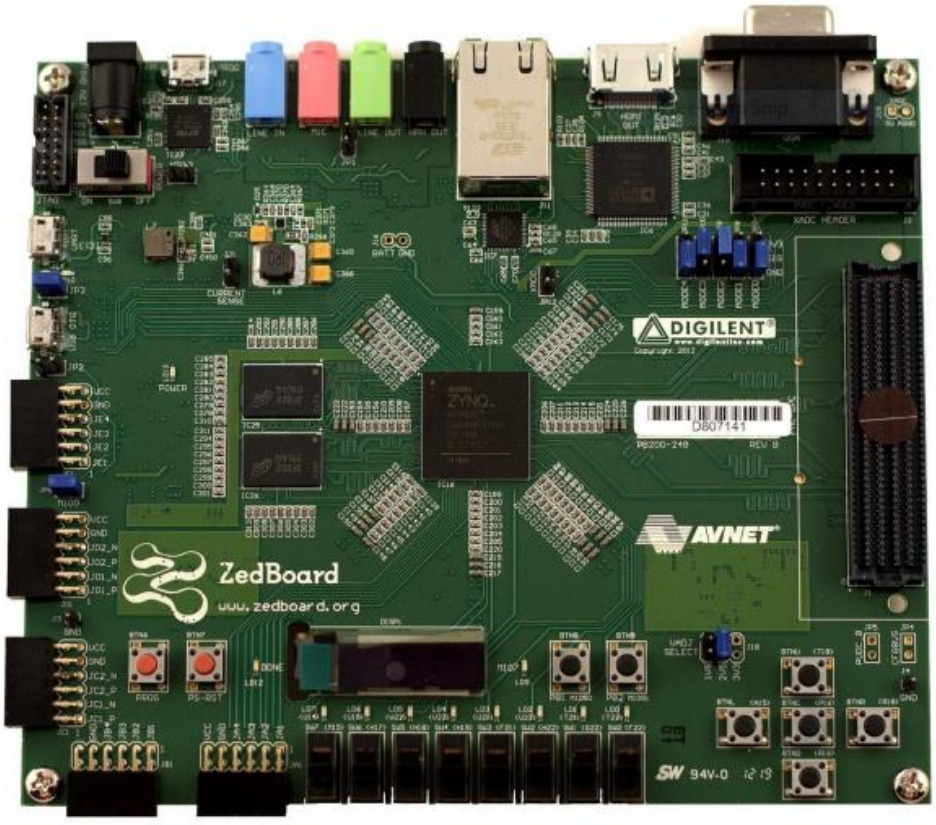


Figure 2. 1: Zed board [5]

Features [5]**:**

The features provided by the ZedBoard consist of:

- Xilinx® XC7Z020-1CSG484CES EPP

- Primary configuration = QSPI Flash
- Auxiliary configuration options
  - Cascaded JTAG
  - SD Card

- Memory
  - 512 MB DDR3 (128M x 32)
  - 256 Mb QSPI Flash
- Interfaces
  - USB-JTAG Programming using Digilent SMT1-equivalent circuit
    - Accesses PL JTAG
    - PS JTAG pins connected through PS Pmod
    - 10/100/1G Ethernet
    - USB OTG 2.0 o SD Card
    - USB 2.0 FS USB-UART bridge
    - Five Digilent Pmod compatible headers (2x6) (1 PS, 4 PL)
    - One LPC FMC
    - One AMS Header
    - Two Reset Buttons (1 PS, 1 PL)
    - Seven Push Buttons (2 PS, 5 PL)
    - Eight dip/slide switches (PL)
    - Nine User LEDs (1 PS, 8 PL)
    - DONE LED (PL)
- On-board Oscillators
  - 33.333 MHz (PS)
  - 100 MHz (PL)
- Display/Audio
  - HDMI Output
  - VGA (12-bit Color)
  - 128x32 OLED Display
  - Audio Line-in, Line-out, headphone, microphone
- Power
  - On/Off Switch
  - 12V 5A AC/DC regulator

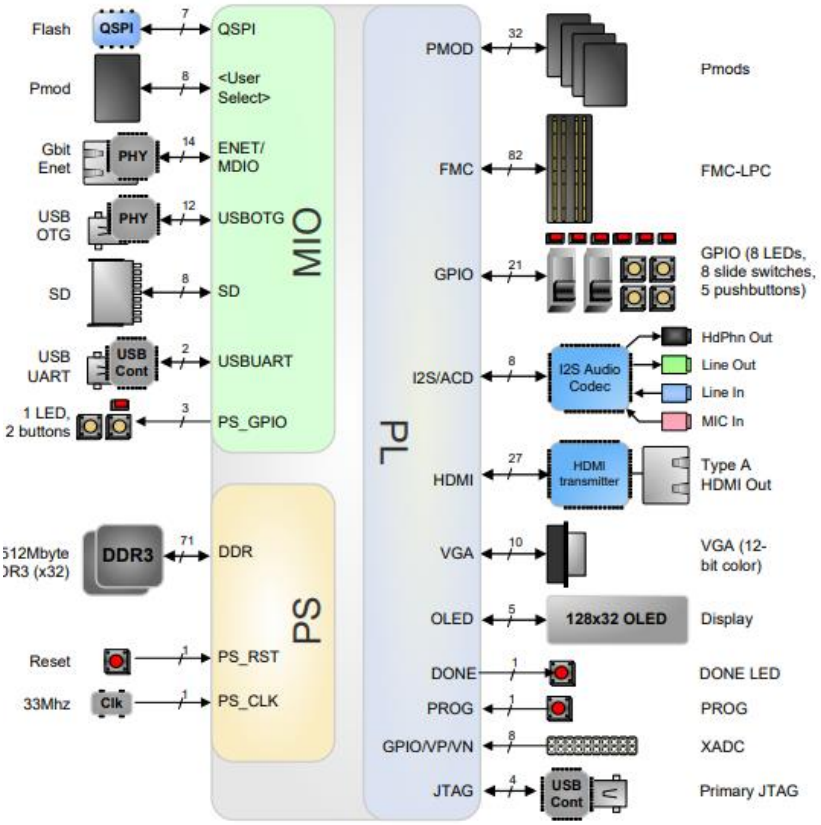


Figure 2. 2: Zed board Block diagram [5]

2.2.2 AD 9288:

It is dual 8-bit monolithic sampling analogue to digital converter, it can operate at 100

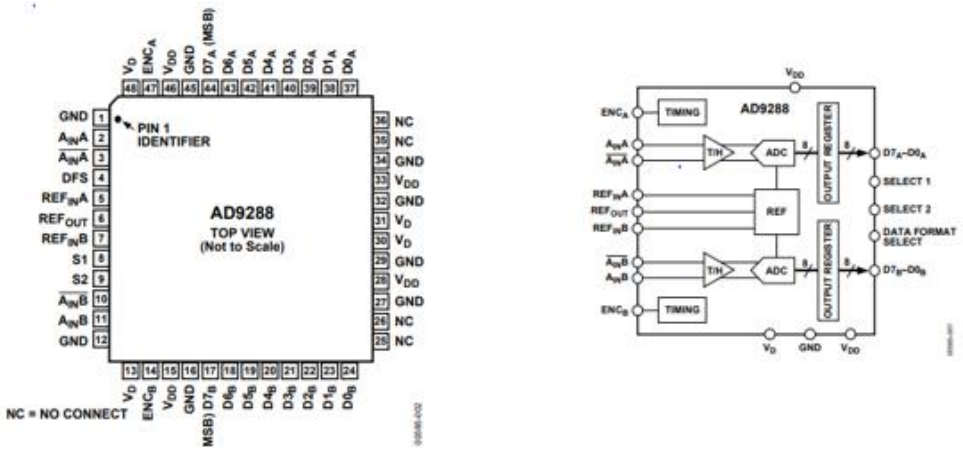


Figure 2. 3: AD9288 Circuit and Block Diagram[6]

- The ADCs are clocked by ENCA ENCB pins for each ADC,
- D7 D0 parallel digital output for each channel A, B.
- the analogue input signal is received threw the pair of pins Ain, the nominal input range is 1.024 V p-p centered at VD × 0.3,
- A stable and accurate 1.25 V voltage reference is built into the AD9288 (REFOUT), In normal operation, the internal reference is used by strapping Pins 5 (REFINA) and 7 (REFINB) to Pin 6 (REFOUT),
- Data Format Select: Offset binary output available if set low. Twos complement output available if set high, for the adc output it is calculated according to this formula

Digital output=1.25/255x Digital Input

- 2 pins S1 and S2 can be used in a variety of operational modes. These choices allow the user to put both channels in standby mode, excluding the reference, or simply the B channel. The output buffers and clock inputs are placed in high impedance states in both modes. The user can also skew the B channel output data by 1/2 of a clock cycle using the opposite option. Enabling the data align permits Channel B output data to be available at the rising edge of Clock A if two clocks are fed to the AD9288 that are 180° out of phase. If both channels have the same Encode clock and the data align pin is enabled, the output data from Channel B will be 180 degrees out of phase with Channel A. If the same Encode clock is provided to both channels and the data align pin is disabled, both outputs are delivered on the same rising edge of the clock.

Table 2. 1: AD9288 Operational Modes[6]

S0	S1	Option
0	0	Standby Both Channels A and B
0	1	Standby channel B only
1	0	Normal operation(Data Align Disabled)
1	1	Data Align Enabled

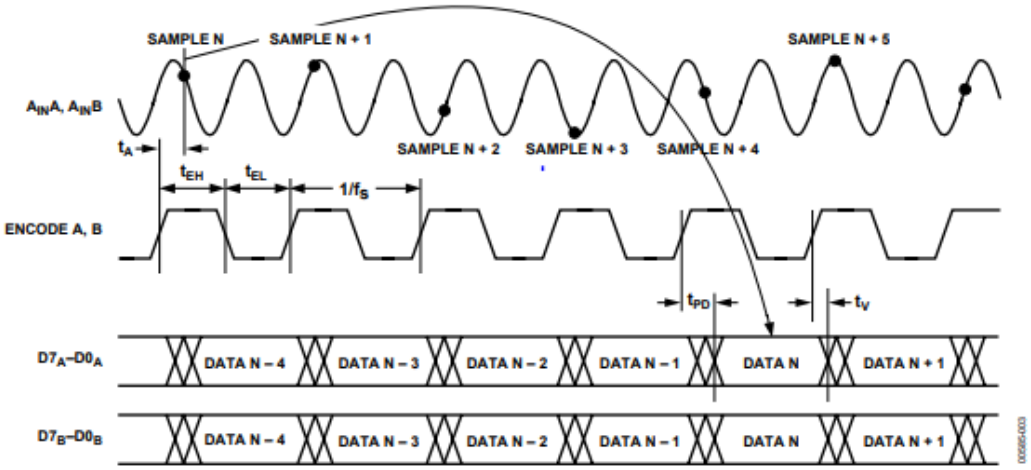


Figure 2. 4: Normal Operation, Same Clock Channel Timing[6]

As figure 2.4 shows both outputs are delivered on the same rising edge of the clock when the data align is disabled. Every rising edge + tpd a new data sample is ready to be read, but reading data after every falling edge is more suitable.

2.3 Video Graphics Array

Video Graphics Array is the abbreviation for Video Graphics Array. It originally referred to the display hardware that was initially introduced with the IBM computer in 1987. It now mainly refers to the analog computer display standards the DE-15 Connector (often known as VGA connector) or the 640x480 resolution itself, due to widespread use.

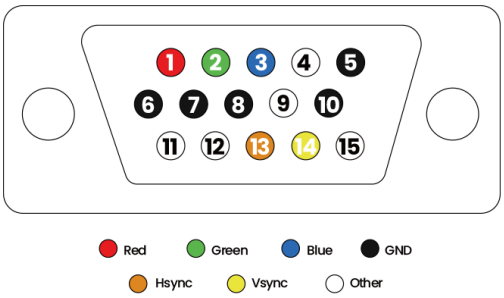


Figure 2. 5: DE-15 Connector

A DE-15 connector, often known as a VGA connector, is a 15-pin D-subminiature connector with three rows (named after their D-shaped metal shield). Figure 2.5 depicts each pin. Red, Grn, Blue, HS, and VS are the five signals out of 15 pins. The RGB signals indicate the color of a point on the screen, while HS and VS offer a positional reference for where the point should be displayed on the screen.

2.3.1 VGA timing specification:

A VGA controller circuit must generate the HS and VS timings signals and coordinate the delivery of video data based on the pixel clock. The pixel clock defines the time available to display one pixel of information [7]

Table 2. 2: Horizontal timing (line)[7]

Scanline part	Pixels	Time [µs]
Visible area	640	25.422045680238
Front porch	16	0.63555114200596
Sync pulse	96	3.8133068520357
Back porch	48	1.9066534260179
Whole line	800	31.777557100298

Table 2. 3: Vertical timing (frame)[7]

Frame part	Lines	Time [ms]
Visible area	480	15.253227408143
Front porch	10	0.31777557100298
Sync pulse	2	0.063555114200596
Back porch	33	1.0486593843098
Whole frame	521	16.683217477656

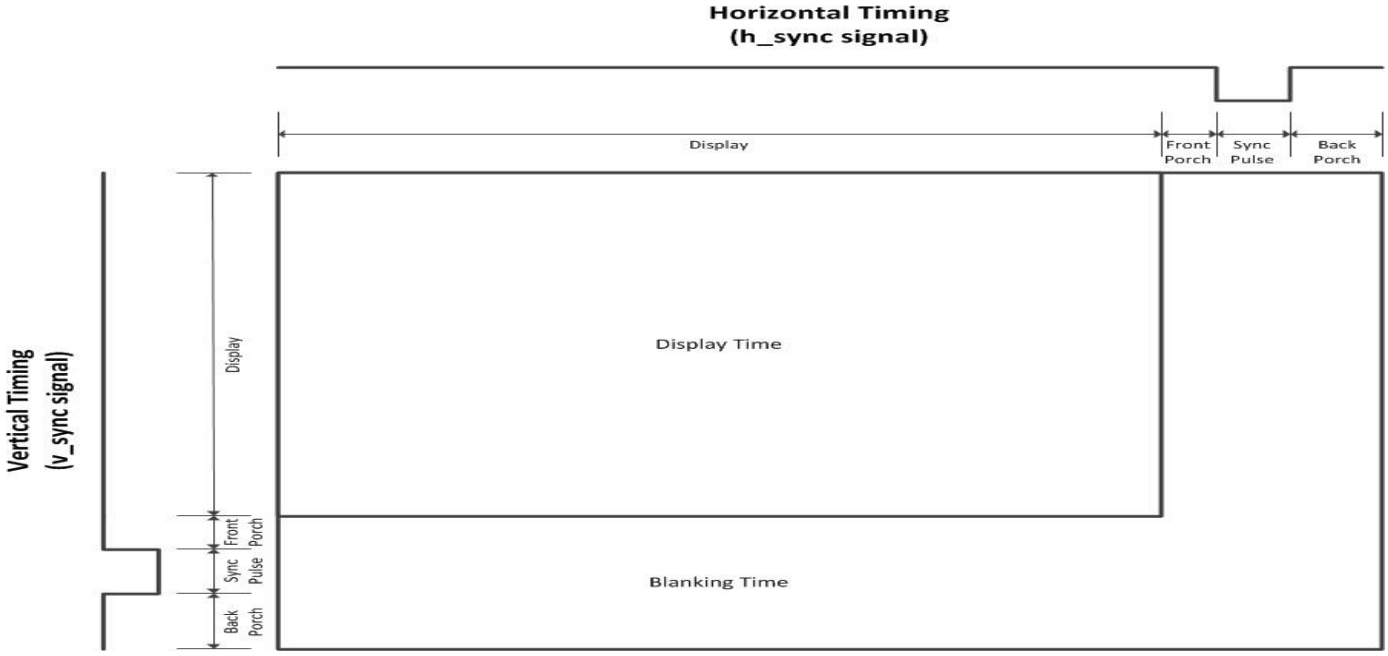


Figure 2. 6: VGA timing specification map[11]

2.3.2 VGA pin functions:

The illustration (figure 2. 5) shows the 15-pin VGA connector, its pin assignments, and size dimensions. As shown, the VGA connector has 15 holes, and each hole (pin) has its own function, as explained in the below (table 2. 4)

Table 2. 4: ZedBoard DE-15 pins description [5]

..

VGA Pin	Signal	Description
1	RED	Red vedio
2	GREEN	Green vedio
3	BLUE	Blue vedio
4	ID2/RES	Formerly Monitor ID bit 2
5	GRN	Ground (HSync)
6	RED RTN	Red return
7	GREEN RTN	Green return
8	BLUE RTN	Blue return
9	KEY/PWR	Formerly key
10	GND	Ground (VSync)
11	ID0/RES	Formerly Monitor ID bit 0
12	ID1/SDA	Formerly Monitor ID bit 1
13	Hsync	Horizontal sync
14	Vsync	Vertical sync
15	ID3/SCL	Formerly Monitor ID bit 3

2.4 Software materials

2.4.1. Vivado

2.4.1.1 Overview:

A Xilinx software suite for HDL synthesis and analysis, which replaces Xilinx ISE and adds tools for system on a chip development and high-level synthesis. Vivado is a complete rewriting and rethinking of the entire design process (compared to ISE). Vivado includes the in-built logic simulator ISIM, just like subsequent versions of ISE. Vivado also introduces high-level synthesis, with a toolchain that converts C code into programmable logic.

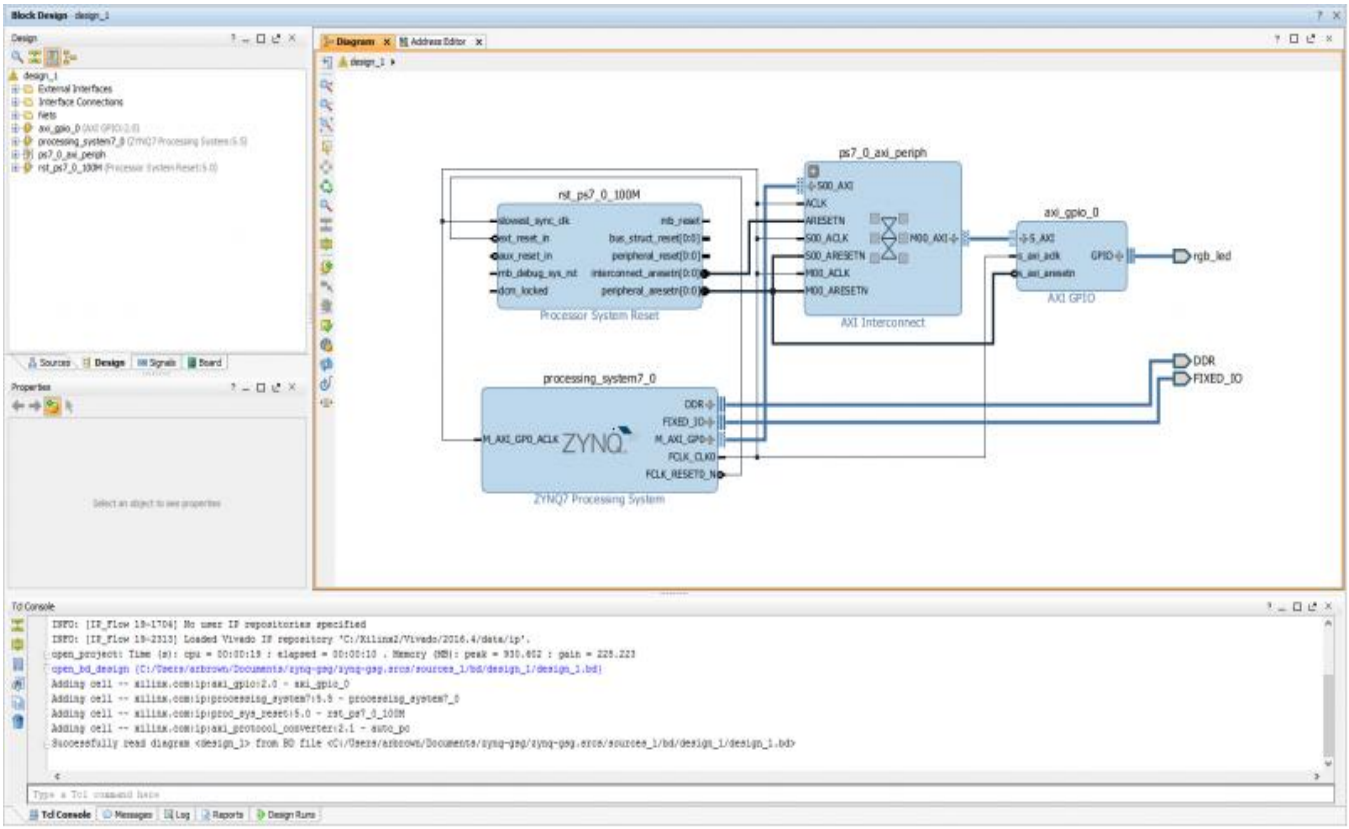


Figure 2. 7: Vivado Graphical User Interface

2.4.1.2 Features:

Vivado is an integrated design environment (IDE) featuring system-to-IC level capabilities built on a shared scalable data model and a common debug environment that was released in April 2012. Vivado offers ESL design tools for synthesizing and testing C-based algorithmic IP; standards-based packaging of both algorithmic and RTL IP for reuse; standards-based IP stitching and systems integration of all types of system building blocks; and block and system verification.

2.4.1.3 Components:

The Vivado High-Level Synthesis compiler enables C, C++ and SystemC programs to be directly targeted into Xilinx devices without the need to manually create RTL. Vivado HLS is widely reviewed to increase developer productivity, and is confirmed to support C++ classes, templates, functions and operator overloading. Vivado 2014.1 introduced support for automatically converting OpenCL kernels to IP for Xilinx devices. OpenCL kernels are programs that execute across various CPU, GPU and FPGA platforms.

2.4.1.4 Vivado Simulator:

It is a component of the Vivado Design Suite. It is a compiled-language simulator that supports mixed-language, Tcl scripts, encrypted IP and enhanced verification.

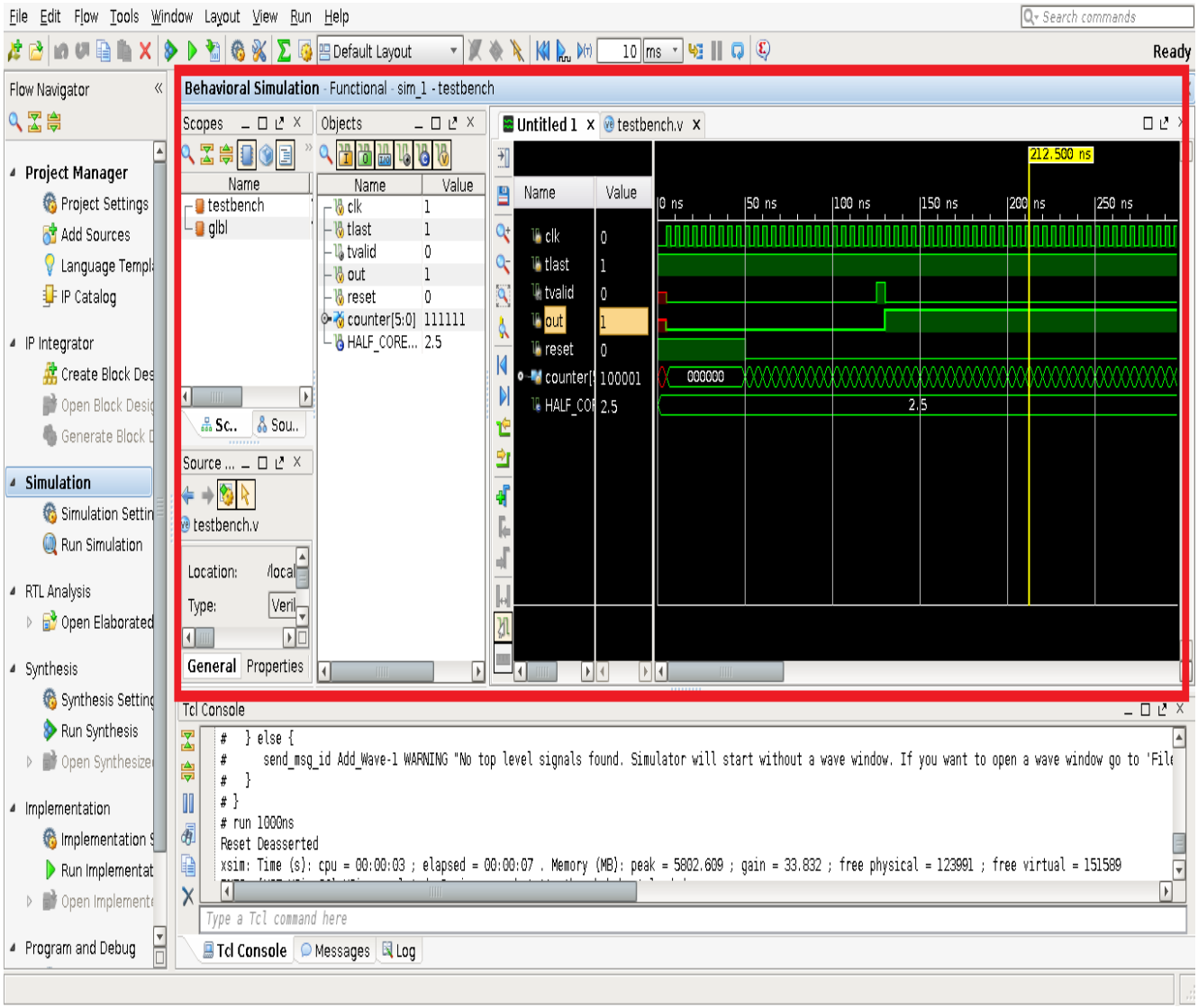


Figure 2. 8: Vivado simulator [10]

2.4.1.5 Vivado IP Integrator:

Allows engineers to quickly integrate and configure IP from the large Xilinx IP library. The Integrator is also tuned for MathWorks Simulink designs built with Xilinx's System Generator and Vivado High-Level Synthesis. [8]

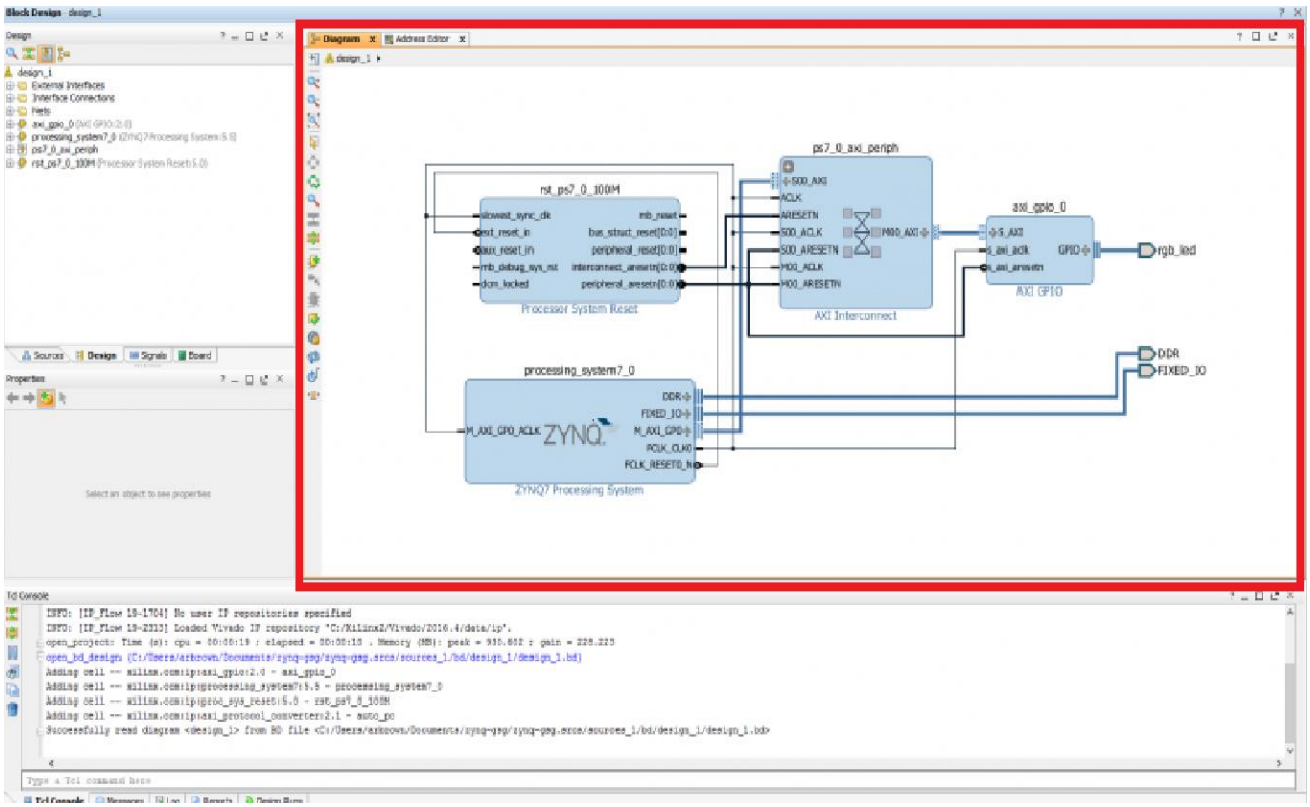


Figure 2. 9: Vivado IP integrator[8]

2.4.1.6 Vivado Tcl Store:

It is a scripting system for creating Vivado add-ons that may be used to extend and change the software's capabilities. Tcl is the scripting language that Vivado is built on. Tcl scripts can be used to call and control all of Vivado's underlying functionalities.

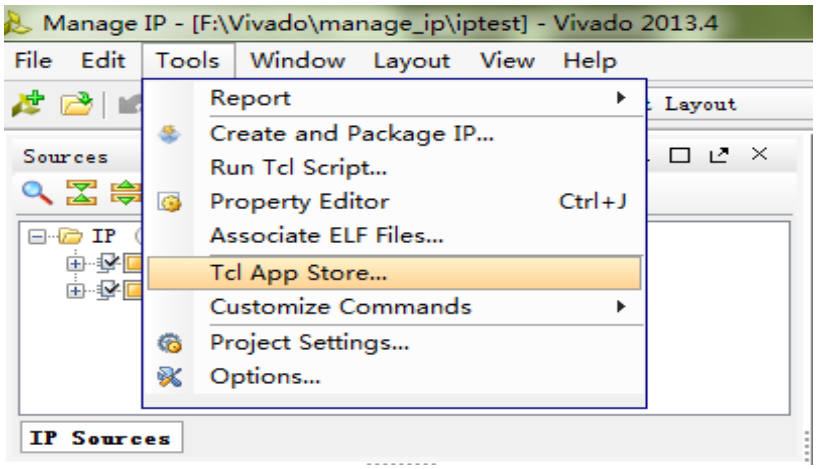


Figure 2. 10: Vivado TCL Store

2.4.2 Intellectual Property

The term "intellectual property" stands for "intellectual property." Intellectual property can refer to a variety of things, but when I refer to it, I'm referring to the libraries of HDL (hardware design language) modules that are available to us. IPs are created by users and businesses to make it easier and faster to create designs.

It's like painting vs. doing a puzzle. Rather than building every individual part of our design our self, we can just take the pieces that we need and focus on the bigger picture. As long as we respect that it's someone else's intellectual property, of course. After all, we wouldn't put together a puzzle and then say we painted it. For example, say we needed to use an Ethernet port but didn't want to write an Ethernet controller yourself. we could find an existing Ethernet controller and use it in our design. Xilinx has a huge library of IPs to use with Digilent boards, as well as an IP library built into Vivado Design Suite.

We may not only use IPs from the large libraries available, but we can also design our own IPs using Xilinx Vivado's IP packager. One advantage of building an IP rather than using files and instantiating a top module is that we may create graphically with a schematic using the Vivado Design tools. We can physically connect the pins on each of your IPs. which can be much easier than typing in inputs and outputs for each module instantiation

2.4.3 Verilog:

Verilog is a textual format for describing electronic circuits and systems that is based on the Hardware Description Language. Verilog is a programming language for electronic design that can be used for verification through simulation, timing analysis, test analysis (testability analysis and fault grading), and logic synthesis.

The Verilog HDL is an IEEE standard - number 1364. The first version of the IEEE standard for Verilog was published in 1995. A revised version was published in 2001; this is the version used by most Verilog users. The IEEE Verilog standard document is known as the Language Reference Manual, or LRM. This is the complete authoritative definition of the Verilog HDL.

2.4.4 VHDL Vs Verilog:

VHDL is more verbose than Verilog and features a syntax that is not similar to C. You have a better probability of producing more lines of code with VHDL. VHDL can also appear to be more intuitive to work with at times. It may appear that developing a program in VHDL flows more smoothly. Verilog generate statement is a powerful construct for writing

configurable, synthesizable RTL. It can be used to create multiple instantiations of modules and code, or conditionally instantiate blocks of code. Because Verilog is more of a hardware modeling language, the vocabulary is more succinct in Verilog. You'll just have to type a few lines of code, and it'll remind you of the C programming language. Verilog is stronger at hardware modeling than C, although it provides fewer programming tools. Because Verilog is less verbose than VHDL, it is more compact. Overall, Verilog differs significantly from VHDL. There are certain similarities, but their variances outweigh them.

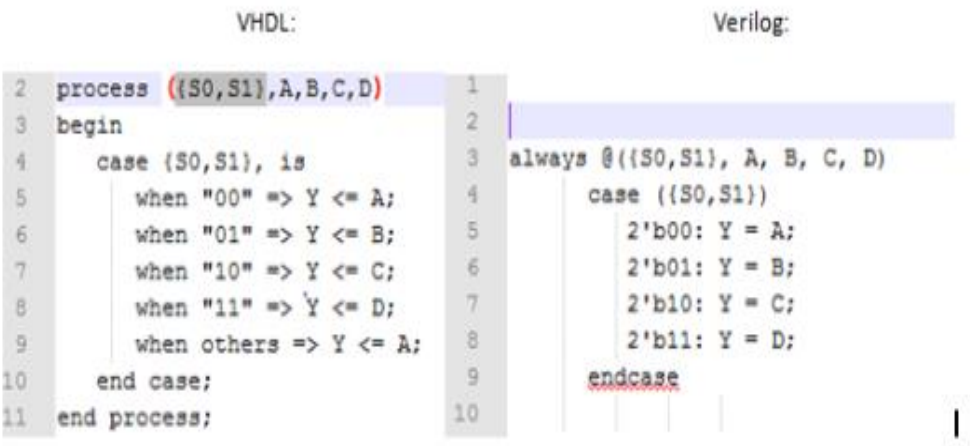


Figure 2. 11: Verilog Vs VHDL

2.4.5 SDK Vivado:

The Xilinx Software Development Kit (XSDK) is the Integrated Design Environment for creating embedded applications on any of Xilinx's award winning microprocessors like Zynq7000 SoCs.

The Software Development Kit (SDK) will then be used to create a simple software application which will run on the Zynq's ARM Processing System (PS) to control the hardware that is implemented in the Programmable Logic (PL).

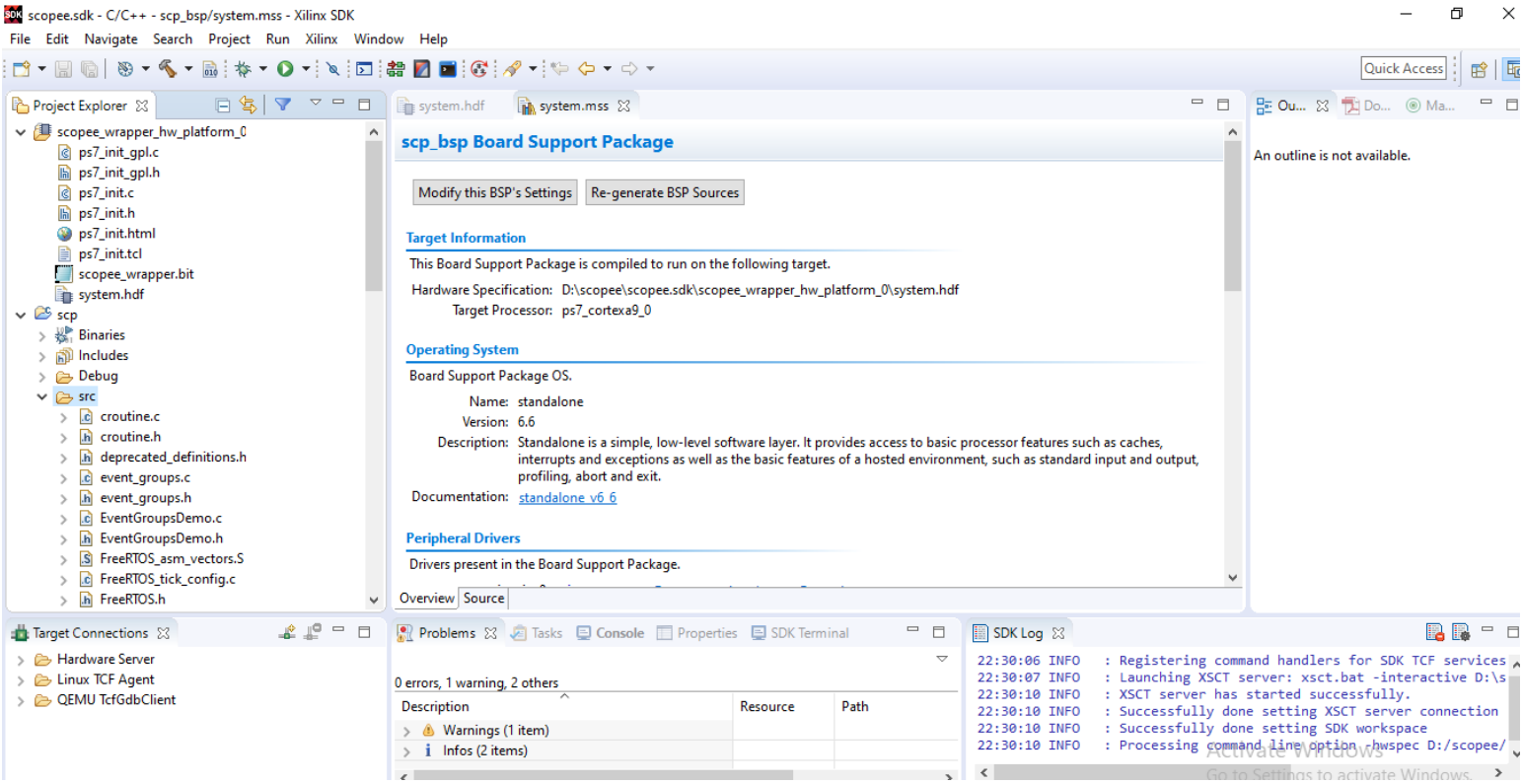
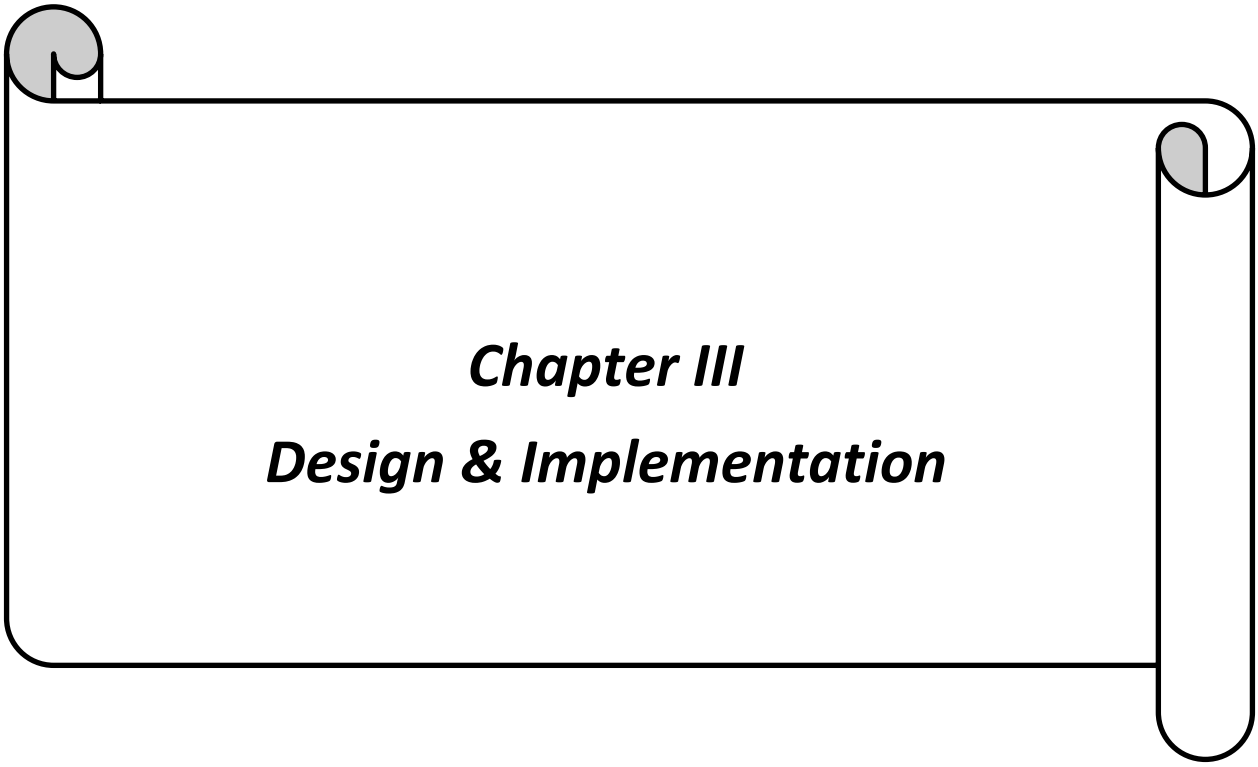


Figure 2. 12: SDK Vivado GUI

2.5 Conclusion:

In this section of our project, we concentrated on the materials and software that our supervisor selected from the open source, and we provided specifics and information about those materials.



Chapter III

Design & Implementation

3.1 Introduction:

The oscilloscope built in this project is a simple oscilloscope with the following specification:

- Input voltage range of -10V to +10V
- Bandwidth = 50MHz limited by ADC sample rate = 100MHz to avoid aliasing.
- VGA monitor display, resolution ~512 x 480

This oscilloscope is made up of several key blocks. Shown in the Figure3.1

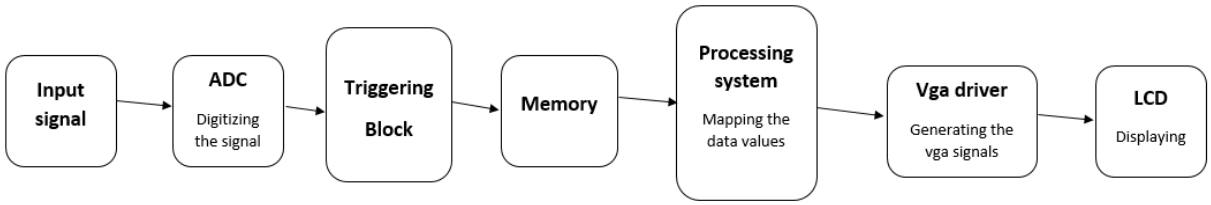


Figure 3. 1: digital storage oscilloscope system key blocks diagram

3.2 Input signal

The voltage level of the signal should be in the range of 0.478 Volt to 1.49 centered at 0.99 Volt (VD =3.3), so the input voltage should be amplified by gain of 0.051 and biased by +0.989 Volt, low pass filter is used to ovoid aliasing with cutoff frequency W0 = 50 Mhz. the proposed circuit is shown in the figur3.2.

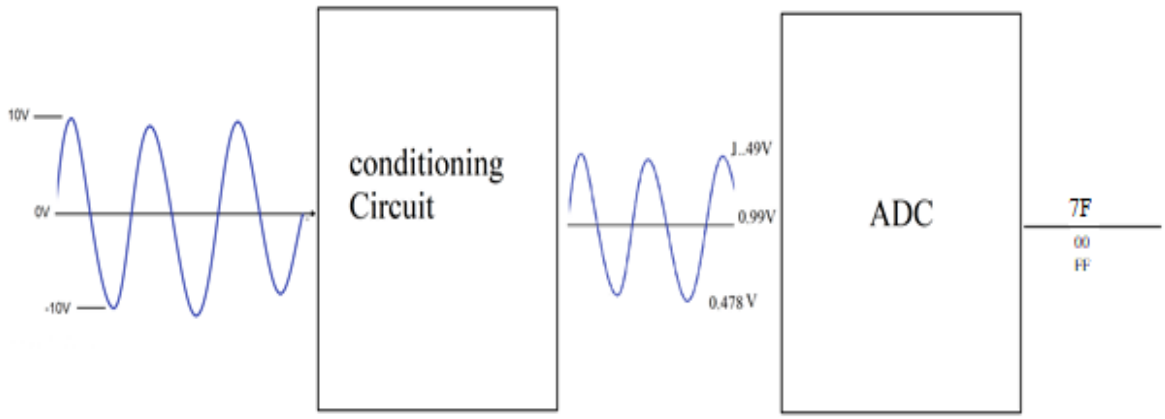


Figure 3. 2: proposed front end circuit

3.3 ADC

the AD 9288 is operating in the mode of Stand by Channel B so that only Channel A input signal will be sampled the complement output format is selected, the ENC A is clocked by 100 Mhz. the channel output pins are connected to 8 I/O pins of JB1 Pmod of the Zedboard, the ENC A is connected to the pin1 of the JA1 Pmod, the low and high signal from the VCC and GROUND pins of the JC1, JB1, JA1 Pmods. The table3.1 shows connection between ADC chip and Zedboard. and the figure 3.3 shows our Zedboard connected to the ADC chip.

Table 3. 1: ADC and Zedboard interfacing

ADC chip pin	Zedboard pin
ENC A	Pin 1 of JA1
$D0_A$-$D7_A$	Pin 1 to Pin 8 of JB1
DFS	VCC of JA1
S1	GND of JB1
S2	VCC of JC1
VD	VCC of JA1
VDD	VCC of JC1

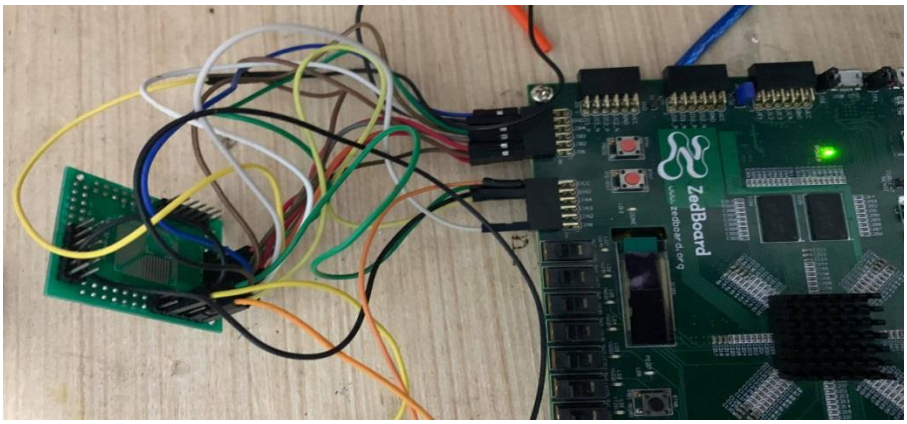


Figure 3. 3 ADC and Zedboard interfacing

3.4 Trigger Block:

The trigger block is responsible of reading the data from the ADC and save it in memory, in this oscilloscope the positive slope triggering is used to display the signal, so the trigger is providing with algorithm to make sure the trigger point is displayed in the center of the screen which the sample data where the trigger occur is saved at the center of the memory.

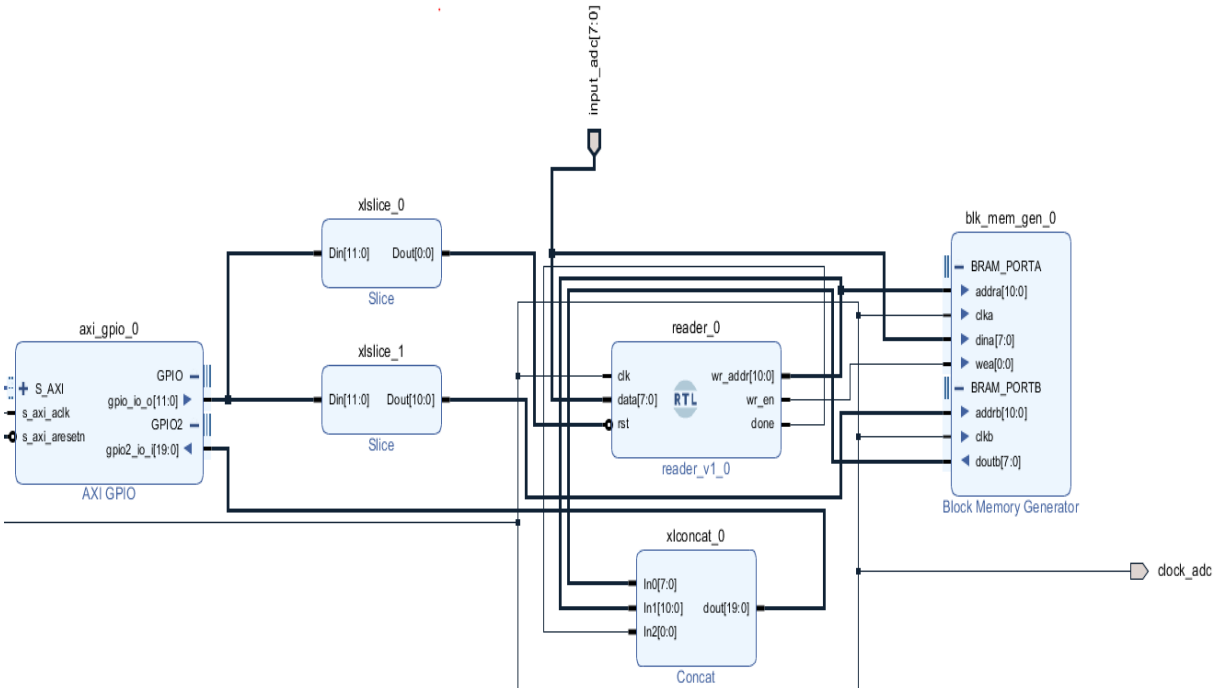


Figure 3. 4: Trigger Block

The figure 3.4 shows part of our block design created using Vivado, this part is the Trigger which is formed of:

- Blk_mem_gen_0: dual port the memory that will save the sampled data it is generated using tool called bram memory generator, our memory is 2 K =2048 of size with 8-bit data width the address width will be 11-bit, port A for writing and B port for reading, memory is in the write or read state according to input WE.
- Input_ADC [7:0]: a port to receive the ADC outputs.
- Axi_gpio_0: an IP created by Vivado used as an interface between processing system and the trigger block ,it has 2 channel , first channel to send a the reading addresses and

a RST signal to clear the done signal when the processing system is reading. the other channel the data from memory and last address written to and the done signal that indicate that the memory is full.

- For the Slice and Xconcate are just tools to get sclicing an data bus or concatenating many data buses in one data bus.
- reader_0: RTL file created by us shown in the figure3.6, this module will control memory writing and reading as described using the diagram in figure 3.5.

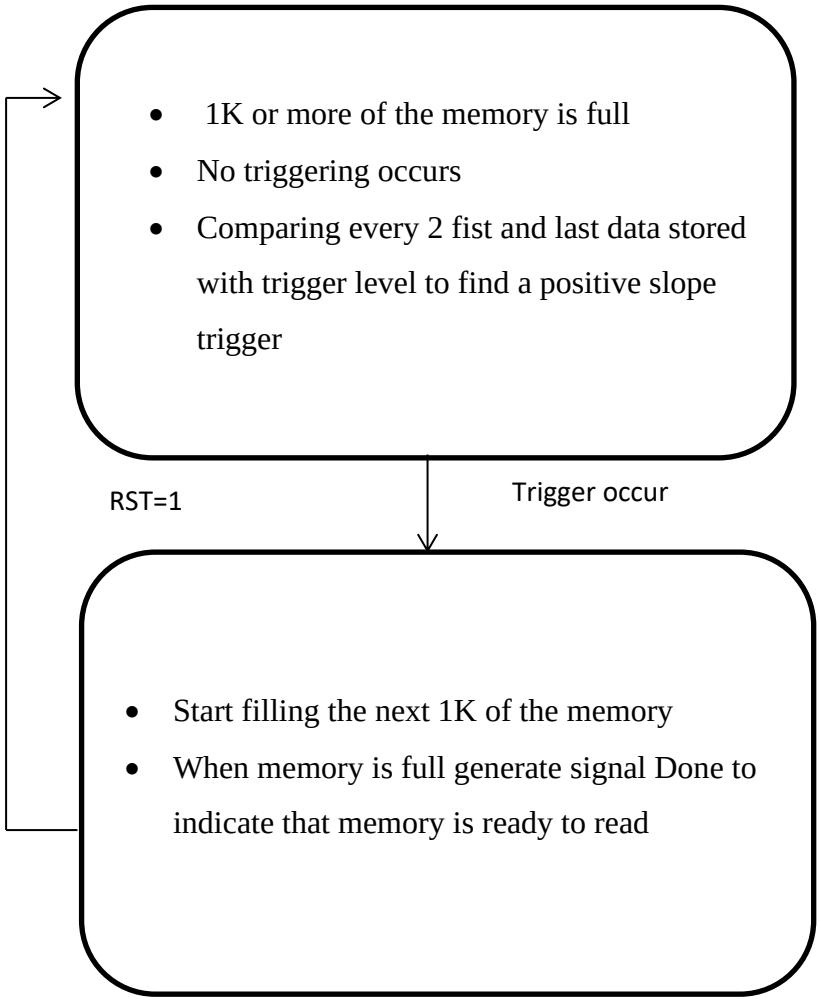


Figure3.5 : RTL reading & writting process

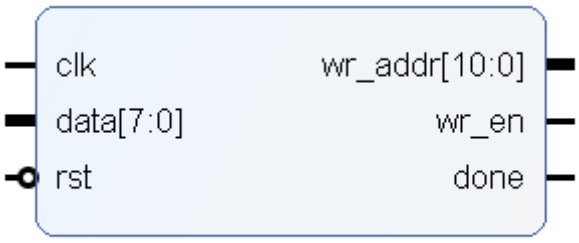


Figure3.6 : reader_0 RTL

3.4.1 Reader Module simulation

The block is tested using Vivado simulator, the Reader module was modified and a 8 bit variable register called value is added as an ADC input. the voltage level of triggering is 5, the clock period is 100 ns and the result of the simulation is shown in the figure 3.7.

Figure 3.7: Reader Module Simulation

3.4.2 Comments:

- Every negative edge a new sample is generated To make sure we get the value from the ADC we made the write signal hold for two clock signal in order to make the clock falling in the middle of writing, besides the memory is active high clock so we write to memory every rising edge and get the value at the falling edge, we lose a sample at every writing but since the sampling rate is 100 mega sample per second the difference is negligible.
- address of the memory is changed after the write finished, so the address changes every falling edge of wr_en signal to make sure that no address change before a write is done.
- Two counters are used to count the first and second halves of the memory.
- In order to get the trigger point in the center of the 2-k memory the address is vector data type making the memory as circular memory, when the done is high the writing address is fixed in order to send it later to the processing
- system as the last address written to. Shown in figure 21
- as it is shown in the figure 21 the done signal is generated after the memory is full. And the trigger point has been detected successfully.
- the RST signal will reset the counters and done, no need to reset the address since the last address written to is saved every done signal.

3.5 Display Block:

This part of the design is responsible for drawing the background image and signal waveform. The figure 3.8 shows a part of the block design which is the display block which is formed of:

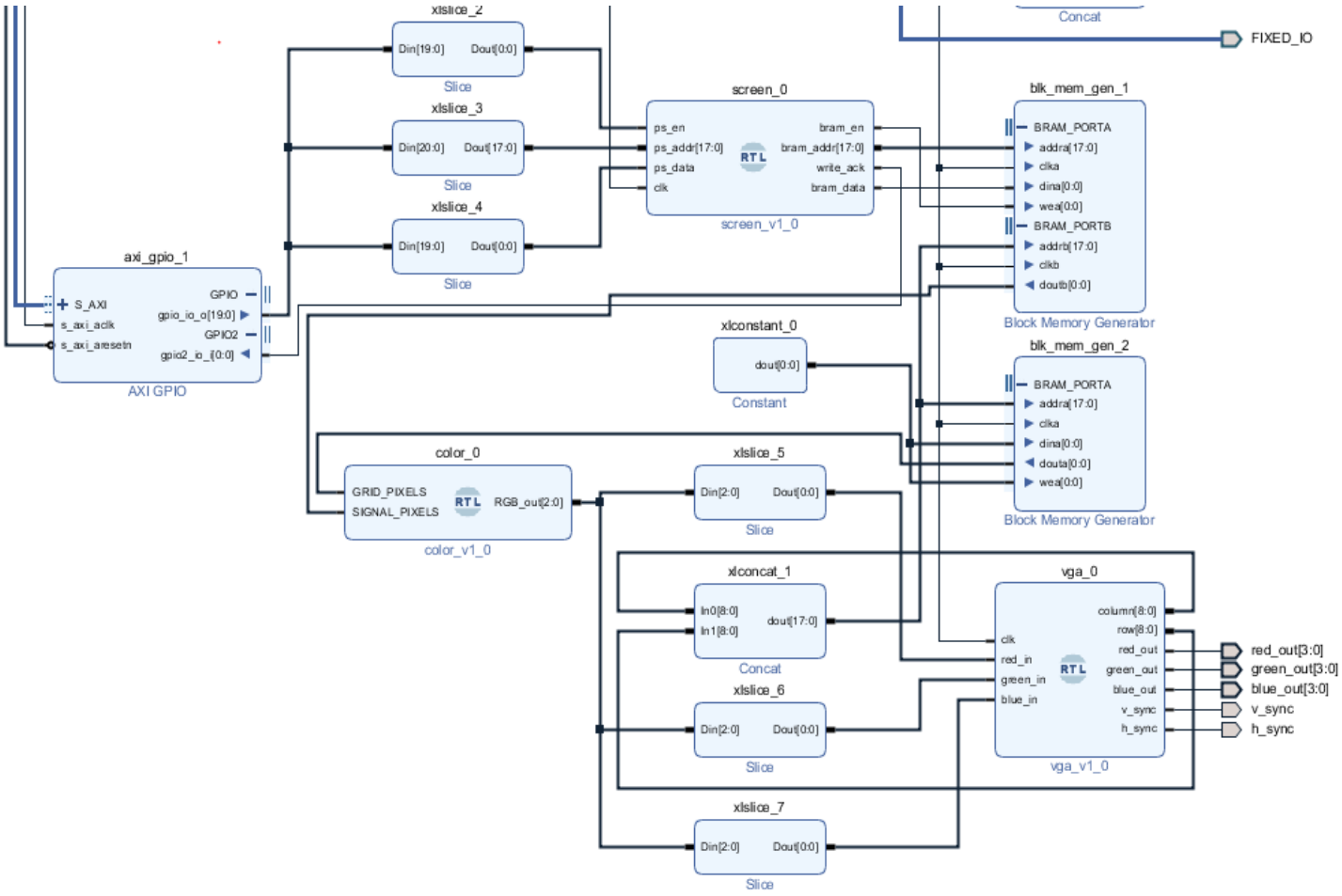


Figure 3.8: Display Block Diagram

- Blk_mem_gen_1: dual port memory generated using tool called bram memory generator, this memory will hold the signal wave form data, the resolution used here is 512x480 pixels so the screen holds 245760 pixels then the memory should of 245760 as size, port A for writing and B port for reading, the color of the signal is blue so just 1 bit to indicate that the pixel is blue or not that's why the data width of the memory is. Since there are 245760 cells so the address width is 18 bits. The address of this memory is the pixels' location which is the row and the column index, this memory will be read later by the a vga driver to generate the appropriate color in the right location.
- Blk_mem_gen_2: memory generated using tool called bram memory generator, this memory will hold the Background image named (Grid) which is shown in the figure3.12, this memory has the same size as the blk_mem_gen_1 and also

the addressing. The color used for background is black and white so only 1 bit to describe the color.

- Axi_gpio_1: an IP created by Vivado used as an interface between processing system and the displaying block, it has 2 channel, first channel sends the pixel signal and location and enable signal to write to the memory other channel will hold an acknowledgement signal to know that a data is written, this is used to ovoid overwriting.
- Screen_0: RTL file created by us shown in the figure 3.9, this module will control blk_mem_gen_1 writing and reading.

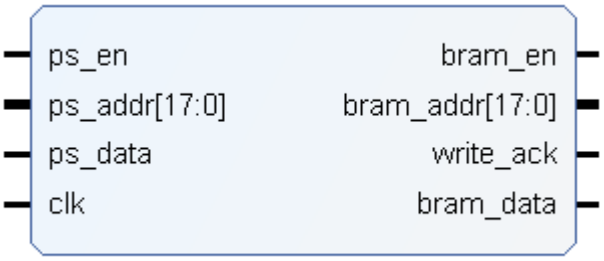


Figure3.9: screen_0 RTL

the role of this module is as follow:

  - write to the memory blk_mem_gen_1 according to the pixel location ps_addr[17:0] and pixel signal ps_data that hold the signal of the pixel that construct our signal wave form in the screen.
  - bram_en: to enable and disable writing .
  - Write_ack: 1 means write in progress, 0 write is finished.to avoid overwriting.

- Color_0: RTL file created by us shown in the figure ??, this module will decide which color will be sent to the vga driver ,or in other meaning which image will be displayed . this module makes sure that the signal is displayed above the grid and not the opposite. this module will send to the vga driver the background signals only if the wave form signal is absent (SIGNAL_PIXELS=0).

Figure3.10 : Color_0 RTL

- Vga_0: RTL file created by us shown in the figure3.11, this is an VGA driver that will generate the vga signal according to the vga protocol of 512x480, it was created from the vga protocol of 640x 480 and modified in the timing specification. the modifications to get the following timing specification are:
  - The Vertical display duration is 480 lines
  - The Horizontal display duration is 512 frame
  - Vertical back porch time is 29 lines
  - Horizontal back porch is 48+64=112, 64 frames shouldn't be displayed on each side.

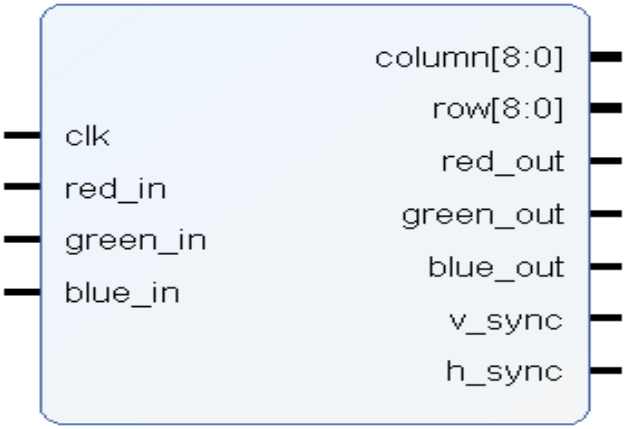


Figure 3.11 : Vga_0 RTL block

.

3.5.2 GRID Image:

GRID image is stored as COE file in a generated memory blk_mem_gen_2, the GRID is shown in figure3.12 the pixel's values representing the signal are stored in another memory blk_mem_gen_1, each memory address holding a pixel's value and the addresses are the pixel location in the screen (column and row indexes), writing to blk_mem_gen_1 is controlled by a created module named SCREEN, this module received from the processor the pixel locations

The grid image is drawn using an image editor PAINT, the voltage and the time division was calculated as follow:

- For the time division :100 mega samples per seconds and since we are ignoring a sample on each reading from ADC that's make us getting 50 mega simple per second ,which means a sample is captured every 20 x 10^{-9} second =20 Nano seconds. And every pixel location is a sample .50 pixels takes 50 x 20 ns which is 1000 ns = 1 us, and every square is a 50 pixel length
- For voltage division: the signal is a 10 V peak and we choose the signal to be drawn vertically in the range of 400 pixels leaving 40 pixels on each side ,so 10 Volt on each 200 pixels that make 50 pixels as 50x10volt /200pixel= 2.5 volt each 50 pixels which the length per square .
- The voltage and the time division are drawn in the free space of the GRID.
- The file compatible by the memory is COE file type, the GRID file was converted to. COE file.

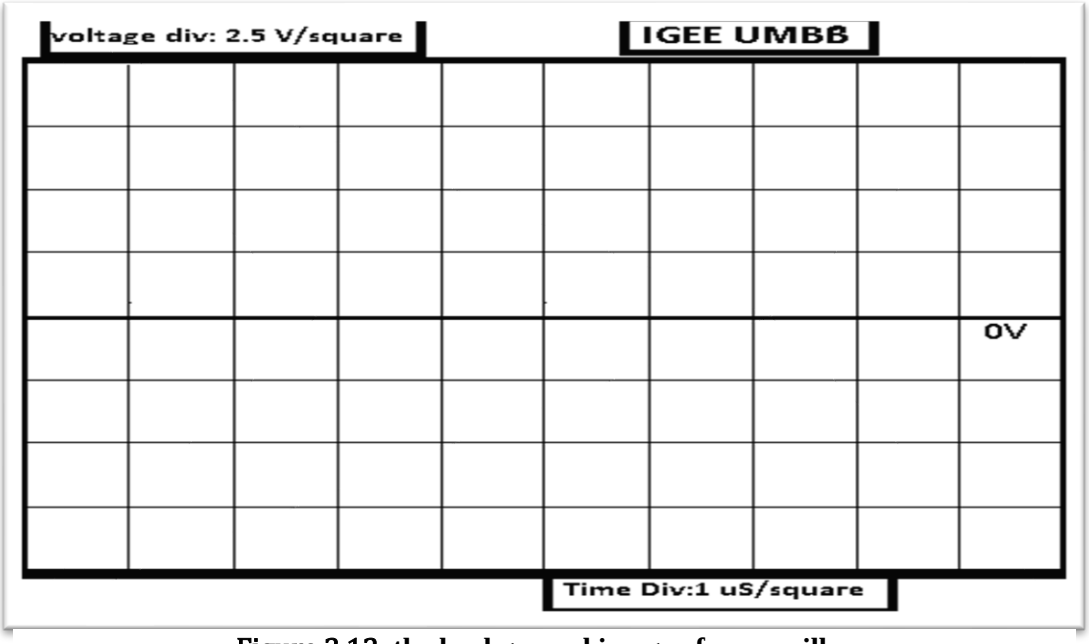


Figure 3.12: the background image of our oscilloscope

3.5.3 VGA Driver simulation:

After simulating the VGA driver, we will get the results shown on figure 3.13

Figure 3.13: VGA Driver Simulation

Comments:

- v_sync is low between 480+29+10=519 and 521

- h_sync is low between 480+64+512= 704 and 704+96=800
- column is incremented after H_counter is 112 and reset at 704
- row is incremented each time H_couter is 800, which defined that the scan is from the left to right and from the top to bottom. The time specifications desired are accomplished by this module.

3.6 Processing system Block:

The block design created in Vivado will be implemented in Pregaming Logic of the chip which is the FPGA, so now the processing need to control and calculate the pixels location and signals, we create an C application using the SDK of Vivado, and the flowchart shown on Figure3.14 describe the operation of our application

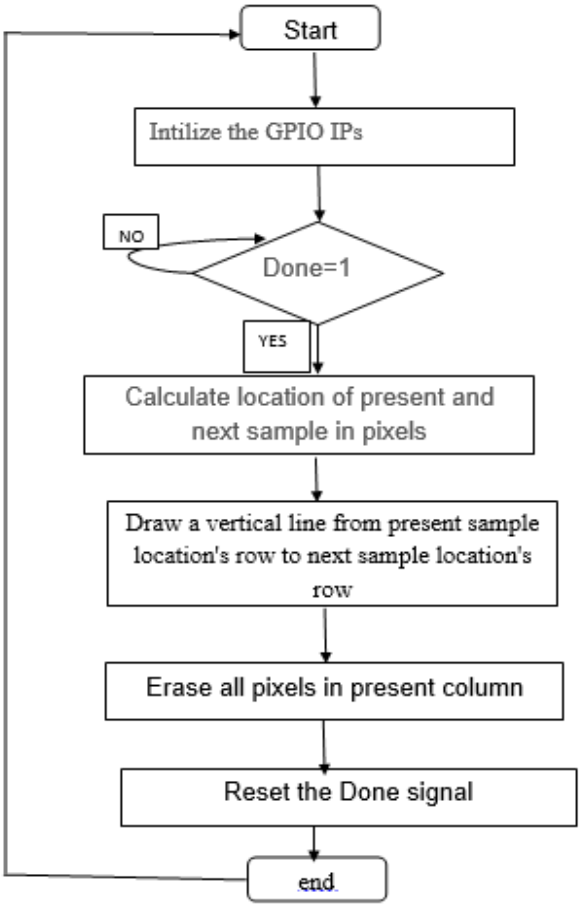


Figure 3. 14: The main program flowchart

The SDK provided us with libraries and header files to build our application our project from these libraries we used:

- Xparameter.h: define the GPIO's addresses refered by the Processing system.
- Xgpio.h, xgpio.c: contain the functions deals with gpios operation, like initializing the gpios, write and read from the Gpios

We create functions needed in our application; these functions are:

- Read_gpio_ trigger():read the data from memory of data blk_mem_gen_0.
- Read_gpio_trigger_addr(): read the beginning of the memory , by adding 1 to the last address written to the memory.
- Check_done_signal(): to check if the memory is full.
- RST_trigger(): sending a rst signal to the reader module
- Write_pixel(): uses as parameters pixel signal ,pixel_enable, pixel location or address in the signal wave memory blk_mem_gen_1.this function for storing the signal wave form pixels. The pixel calculated as follow:

pixel location =the vertical_center –sample_value*screen_ratio.

- The screen ratio is found as follow: we defined the signal to be displayed in 400 pixels vertically, so 200 pixels for each 10 volts, and the 10 volts enter the ADC as 0.5 volts, using the previous formula of calculating the ADC output of 0.5 gives 102 in decimal, so the ratio will be 102/200

3.7 Displaying testing:

We test the overall design without connecting the ADC outputs to the Zedboard, and we modified the module file reader to add a variable called that increment in triangular manner. this value will be stored in blk_mem_gen_0, this mean a signal is generated internally in the FPGA. Then we loaded the program to the Zedboard and the result in the screen is shown on the figure figure 3. 15.

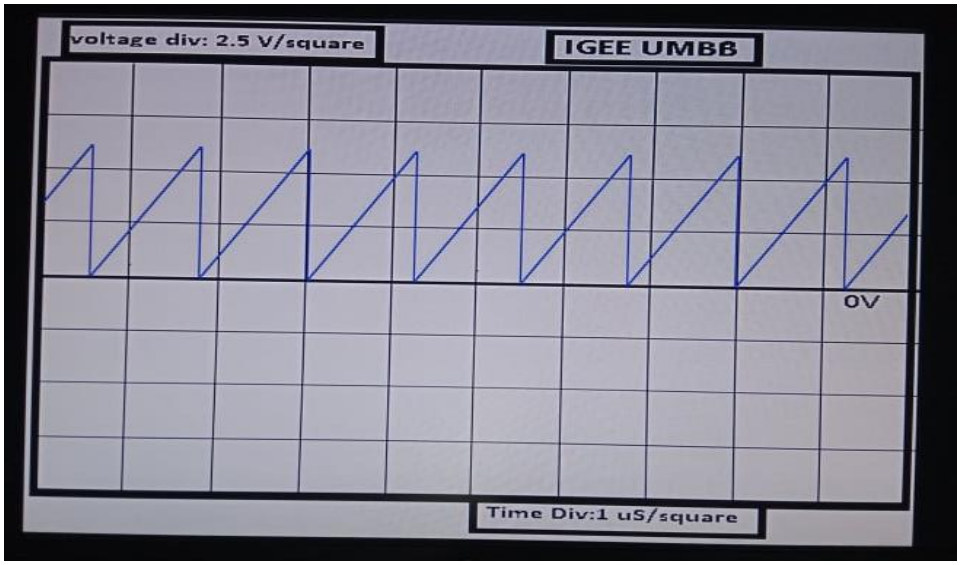


Figure 3. 5: Displaying of generated triangular signal

Comments:

- Signal is displayed perfectly which the software part of the project is correct.

3.8 Conclusion:

This chapter represent the software development of our Digital oscilloscope, where we used the Vivado software to design the different blocks such as the trigger block and the Video driver block, each block contains IPs that are customized IPs developed using Verilog. The processing system block is responsible for data mapping and displaying the electrical signal.

Chapter IV

Final results

4.1 Introduction

The project is finally implemented and the program is loaded to the ZedBoard via JTAG USB. We tested our ADC using a generated signal divided by 10 using voltage divider. We tried several levels of voltage. The signal source is the mini-function generator shown in figure 4.1. And the final results are shown in figure 4.2 and figure 4.3.

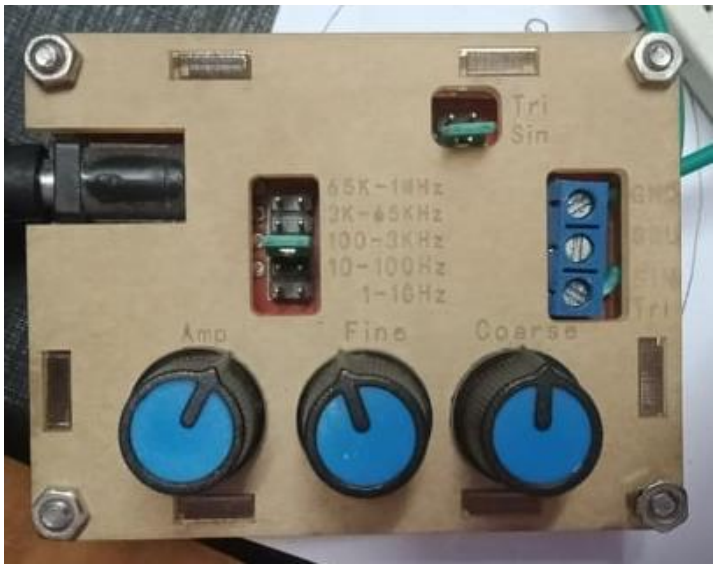


Figure 4. 1: Mini function generator

Figure4.2 : Oscilloscope final implementation

Figure 4. 2: Testing with low frequency

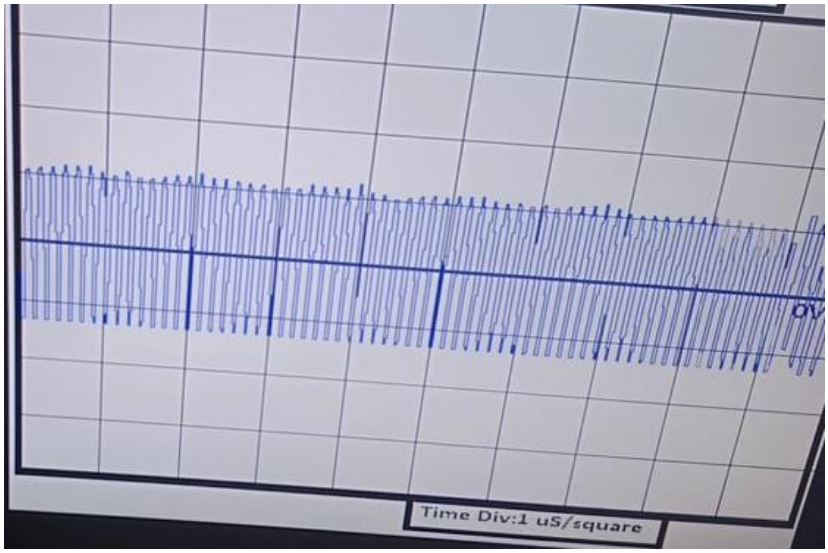


Figure 4. 3: Testing with high frequency

Comments:

- The oscilloscope shows a noticeable improvement in performance with high frequency signal comparing to the low frequency signal.
- When the frequency is too low, the display is meaningless.

4.2 Conclusion:

This oscilloscope did not meet the desired results. However, the software simulation did perform as we have anticipated. We can conclude from this that our oscilloscope has clocking and timing issues to get the desired signal from the ADC.

•*Conclusion*

The oscilloscope can measure a wide range of phenomena with the right sensor. it is a critical tool for solving today's demanding measurement difficulties and it is must-have for anybody designing and repairing electronics equipment.

In this project we have developed a prototype of a single channel digital oscilloscope using SOC which combines an FPGA with ARM microprocessor.

In our oscilloscope, the input signal is sampled using an external ADC. The sampled data is then stored in programmable memory when positive edge triggering occurs using a block that is implemented on the PL of the FPGA. This data will be sent to the processor where an application is written to map it to pixel locations that are read by a PL block to generate the VGA signal.

We have simulated our design implemented in the PL and it was very satisfying. And we also test an internally generated signal and it was displayed perfectly. However, when assembled our oscilloscope, we found a poor performance supposedly the clocking of the ADC is not well understood due to our poor experience with this kind of ADC and time constraints.

Futur work :

- the clocking of the RTL module "reader " need some modification to respect the clocking of the adc
- add some user input interface to adjust some parameter like voltage division and time division
- we will use LVDS cable to replace VGA in order to accelerate the display .
- add functions to calculate some voltage parameter as mean square value .
- we use another channel to display the frequency domain of the signal and create a RTL module for it .

References

[1] Oscilloscope Fundamentals_tektronix https://docs.rs-online.com/0640/0900766b80da861a.pdf

[2] Oscilloscope Triggering.pdf (csufresno.edu) Date accessed: 08/20/2021

[3] Projects, H. (2018, December 23). *About oscilloscope - oscilloscope trigger control*. Electronics Circuits & Tutorials. Retrieved September 18, 2021, from https://www.hobbyprojects.com/oscilloscope_tutorial/oscilloscope_trigger_controls.html?no_redirect=true. TRIGGER

[4] Sampling theorem. Sampling Theorem - an overview | ScienceDirect Topics. (2007, October 24). Retrieved September 18, 2021, from https://www.sciencedirect.com/topics/engineering/sampling-theorem. SAMPLING

[5] Xilinx Zedboard. (2012). Zynq™Evaluation and Development Hardware User's Guide. *https://digilent.com/reference/_media/zedboard:zedboard_ug.pdf* (Version 1.1). Retrieved from https://digilent.com/reference/_media/zedboard:zedboard_ug.pdf.

[6] 8-Bit, 40/80/100 MSPS Dual A/D Converter AD9288. https://www.analog.com/media/en/technical-documentation/data-sheets/AD9288.pdf. (2017, July 16).

[7] VGA signal 640 x 480 @ 60 Hz industry STANDARD timing. (2011, February 11). Retrieved September 18, 2021, from http://tinyvga.com/vga-timing/640x480@60Hz.

[8] Vivado Design Suite User Guide_Creating and Packaging Custom IP_UG1118 (v2019.2) March 2, 2020 https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_2/ug1118-vivado- creating-packaging-custom-ip.pdf

[9] INITIATION AU LANGAGE VERILOG_Jean Louis Noullet - 1995-2005 - 5ème édition rev.5 https://www.aime-toulouse.fr/DOCPDF/CAO/TPCAO/verilog.pdf

[10] Vivado Design Suite User Guide_Design Flows Overview_UG892 (v2020.2) February12, 2021

[11] MLM (https://electronics.stackexchange.com/users/11915/mlm), VGA Decoding - Dealing with tolerances, URL (version: 2013-12-11): https://electronics.stackexchange.com/q/92900