

Automating Digital Oscilloscope for FPGA Validation Using Py-visa Module

Karthika S¹, Suganya S², Ramya A S³

Final Year Student, Department of IT, Velalar College of Engineering and Technology, India¹

Associate Professor, Department of IT, Velalar College of Engineering and Technology, India²

Research Scholar, Kongu Engineering College, Erode, India³

ABSTRACT: The main aim of this project is to remotely control the oscilloscope and automate the complete validation of the Field Programmable Gate Array (FPGA) signal. Programmable unit of FPGA is coded according to the requirements of fixed waveform generation. Electrical signals are analyzed through Digital Signal Oscilloscope (DSO) for the verification and debugging process. The various interfaces such as GPIB, USB, and Ethernet are used to communicate the data to oscilloscope. The pyvisa module that access the device along with its interface is done by Virtual Instrument Software Architecture (VISA) driver. Pyvisa manages the resources through ResourceManager() that find the available resources that connected to the PC. The detected device then opened to apply the oscilloscope commands. The MDO 4000 series programmers manual of the provides oscilloscope commands and functions. To ease this situation VISA libraries offer the arbitrary adapters to configure with its interface.

KEYWORDS: Digital Oscilloscope, Field Programmable Gate Arrays (FPGA), NI, NI-MAX, VISA, Pyvisa.

I. INTRODUCTION

In recent years all products consumed by the consumer have electronic circuits. So the manufacturing of a product includes electronic components that need an oscilloscope for the verification, and debugging process to analyze the numerous electrical signals. The commands are programmed in python to perform functions used to remotely control the oscilloscope settings. The FPGA is Field Programmable Gate Array which provides high precision, high flexibility, high fast, low power requirement, and low cost. When a circuit board contains an FPGA part is programmed or reprogrammed to satisfy the requirement. The Logic block is the programmable unit which contains the logic to execute the applied functions.

Tektronix MDO 4000 series oscilloscope is selected and used as experimental device. A digital oscilloscope is used for fast sampling rates, accuracy, and frequency modulation. The automation of an oscilloscope is required when continuous monitoring of signals from the board. Modern FPGA contains thousands of embedded processors and millions of logic gates allowing designers of FPGA hardware to create extremely complex and sophisticated hardware functions that are application-specific. Measuring all waveform signals from the complicated electric board by manual, is difficult and consumes a lot of time.

Paper is organized as follows. Section III describes the common communication ports like USB, LAN, or GPIB as interface to communicate with PC through different protocols. The Oscilloscope diagnose an electrical signal that draws a graph described in section IV. National Instrument MAX (Measurement & Automation Explorer) are used to configure the communication. The screenshot represents the fixing cursors and capturing of waveform automatically.

II. RELATED WORK

Oscilloscopes are the diagnostic instrument needed to measure and analysis of numerous signals and shown in graphical form. Oscilloscope used for signal processing, analysis function, fault, and error detection in the signal [1]. To detect the analog signal in the oscilloscope screen, probe act as the intermediate between the oscilloscope channel and device. Measuring probe configure the input port to receive signals generated from device under test [2]. The oscilloscope probe is connected to FPGA (Field Programmable Gate Array) is the device under test implemented to generate waveform [3]. FPGA is the prominent example for PLD (Programmable Logic Device) which is re-programmable hardware and measuring the characteristics of signal under test using multiple input channels to get digitized waveform [4]. Waveform measurement is important for semiconductors [5]. Use two different oscilloscopes

are controlled by classic methods that sent SCPI (Standard Commands for Programmable Instruments) commands through the interface RS232 and use LabVIEW to create the user interface [6]. Apply LabVIEW user interface to remotely control multiple oscilloscopes simultaneously and acquire data through internet has several common and extensional functions [7].

The proposed work in this paper is to validate the signal by remotely operate the oscilloscopes using oscilloscope commands via the interface. The oscilloscope probe is connected to FPGA (Field Programmable Gate Array) is the device under test. FPGA generates waveform to measure the signals.

III. METHODOLOGY

Oscilloscope connection:

The oscilloscope is connected to the computer through either of USB cable, Ethernet and GPIB. The experiment is done with tektronix 4000 series oscilloscope. On the rear panel of the oscilloscope the appropriate cable is connected to enable the communication between PC and device. The cable is the data transmission link that send the commands to remotely control the oscilloscope. On the front panel push the utility page to set IP address for Ethernet or GPIB address for GPIB connection and enable USB. Here the software connections of computer to oscilloscope is done by Virtual Instrument Software Architecture (VISA). VISA let the windows PC to acquire data from oscilloscope that runs in Tektronix\NI softwares or own custom software. VISA communication over the common connections is set by load the VISA drivers which found in appropriate CD comes with device or www.tektronix.com to download those along with OpenChoiceDesktop.

FPGA evaluation board and Programming :

The applications of Digital Signal Processing developed through a small single chip called ASIC (Application Specific Integrated Circuits). FPGA is the alternative containing millions of gates that gives fast execution and re-programmable logic unit. The waveform is generated using Virtex II FPGA (XC2V1000) run at the frequency of 150 MHz. Through program not only the microprocessor but also logic gates are under control. The waveform is generated in the form of verilog program running on the processor embedded on the FPGA. The platforms of hardware and software of the system is generated by Xilinx Platform Studio. The bit stream is uploaded to make the software and hardware together.

The evolution board with one or more communication ports JTAG, UART ports, extension slot, power supply and part of FPGA. The logic blocks are array connected through programmable interconnect. The data loads into the logic blocks in a specific address of system space and user space. The memory address starts from 0 to 0xFFFFFFFF by dividing Local Memory Bus, On-chip Peripheral Bus and Peripherals. The peripherals are connected to the On-chip Peripheral Bus.

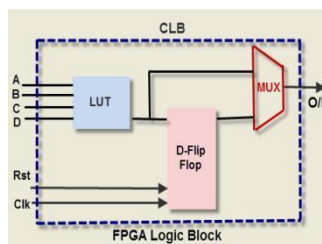


Figure 2. FPGA Logic Block

The verilog programming compiled to RTL (Register Transfer Logic). The data transferred to some conditional clock and synthesized to Electronic Design Interchange Format (EDIF) file. Process required additionally that fill register and memory to control the logic gates of an FPGA. Compiler produce bits to control fixed gate pattern and synthesized or mapped to logical gates then route EDIF to produce HEX files loaded into FPGA. The bits are downloaded from the .HEX files into hardware. The debugging of FPGA is done by software simulation.



Figure 1. Oscilloscope Probe and Circuit Interface

Probing means connect the probe wire of device to the instrument channel and the output pin.

Command and query syntax:

The manual uses BNF (Backus-Naur Form) notation for commands and queries. Commands have set form or query forms some commands have both. Device clear function is used to reset the oscilloscope and clear the output queue. Commands may enter in upper or lower case and also have abbreviations. Command message consists of five elements,

Header : Common name begin with colon (:) when two or more commands are concatenated semicolon (;) is used. If ends with question mark then it's query.

Mnemonic : Sub function of header

Argument : Comma or space separated by arguments. Arguments types are enumeration, numeric, block and quoted strings.

Comma

Space

Command syntax:<Header>[:]<Mnemonic>[<Space><Argument>[<Comma><Argument>]...]

IV. EXPERIMENTAL RESULTS

Setting system Environment:

Downloading python for windows 10 from python.org and select the current release or old releases of python. Here the version of python is 3.6 and for installation windows x86-64 installer is chosen. The executable is run to install python, after the download completed. Select the checkbox to set the command prompt path for python implementation. Next step to select the pip which is used to download other packages from python repository and select the IDLE which is the development environment. Installation process is complete by choosing the installation path then the set was successful.

Open command prompt and “pip install pyvisa” that installs the pyvisa packages. Further package installations are “pip install configparser”, “pip install python_struct”, “pip install numpy”, and ‘pip install time”. Some of the packages are by default installed if not present use the pip command to install the new releases.

Module description:

Import the packages and required modules where the build-in functions support the operations. Pyvisa info is the command to verify that pyvisa is installed properly.



```

Command Prompt
Microsoft Windows [Version 10.0.18362.1]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\wdt_user1>pyvisa-info
Machine Details:
Platform ID: Windows-10-10.0.18362-SP0
Processor: Intel64 Family 6 Model 158 Stepping 11, GenuineIntel

Python:
Implementation: CPython
Executable: c:\users\wdt_user1\appdata\local\programs\python\python36\python.exe
Version: 3.6.4
Compiler: MSC v.1900 64 bit (AMD64)
Bits: 64bit
Build: Dec 19 2017 06:54:40 (#v3.6.4:d48cecb)
Unicode: UCS4

PyVISA Version: 1.10.1

Backends:
ni:
Version: 1.10.1 (bundled with PyVISA)
#1: C:\WINDOWS\system32\visa32.dll:
found by: auto
bitness: 64
Vendor: Tektronix
Impl. Version: 4194816
Spec. Version: 4194304
#2: C:\WINDOWS\system32\visa64.dll:
found by: auto
bitness: 64
Vendor: Tektronix
Impl. Version: 4194816
Spec. Version: 4194304

py:
Version: 0.3.1
ASRL INSTR: Available via PySerial (3.4)
USB INSTR: Available via PyUSB (1.0.2). Backend: libusb1
USB RAW: Available via PyUSB (1.0.2). Backend: libusb1
TCP/IP INSTR: Available
TCP/IP SOCKET: Available
GPIO INSTR:
Please install linux-gpio to use this resource type.
No module named 'gpio'

C:\Users\wdt_user1>
    
```

Pyvisa : The package that supports VISA for controlling the instruments through the interface GPIB, USB, LAN. Pyvisa package supports the device taken for this experiment also support interface chosen. Python can call the VISA shared library function and directly access the device interface.

ConfigParser : Configparser is module that gives ConfigParser class. The extension of the file is .INI the below is configuration file looks like,

```

[dso_ch]
ch = ch1

[resource]
usb = USB0::0x0699::0x040C::C010311::INSTR

[scale]
v = 1
h = 1
    
```

The above configuration file can able to change the input instantly before the execution starts instead of changing the values in the python file. Here the name of INI file is dso_config.ini and the values are accessed by the following,

```

import configparser
config = configparser.ConfigParser()
config.read('dso_config.ini')
print (f "{ config ['dso_ch'] ['ch'] } : SCALE{ config ['scale'] ['v'] } ")
    
```

“config” is the variable name and this act like dictionary in python.

“config.read()” the function that read the INI file.

“config['dso_ch'] ['ch]” gives the value as ‘1’ and the type of the data from INI file is always strings.

Struct: Struct is a module that have the capability of converting Python values from C structures. The raw data from the captured image is packed and stored in a variable. Unpack is function from struct to get the original values from packed data. The unpacked values are stored in a numpy array.

Numpy: There is no dependency in numpy package. Creates a powerful n-dimensional array. Here the data points from image that captured are stored in the numpy variable.

Matplotlib: Package for data visualization that plots 2D graphs from data in an array. The stored data points are plotted to verify the points whether the waveform is generated correctly or not.

Once the connections are made then the setup is ready to work under the program control. Run the python file with extension .py consists of oscilloscope commands from the programmers manual by the tektronix.

The following sample python code that runs, stores and measures the signal. This will help at the situation of continuous monitoring of waveform which is know previously. Once the code is written this will help for other signals.

```
import pyvisa
import pylab
res = pyvisa.ResourceManager()
scope = res.open_resource(f"{config['resource']}['usb']}")
scope.write("FPANEL:PRESS defaultsetup, menuoff")
scope.write("ch1: SCALE 1")
scope.write("HORIZONTAL: SCALE 10")
scope.write("CURSOR: function screen\n")
scope.write("acquire: mode hires ")
if(scope.query("acquire: state? \r") == 0):
scope.write("acquire: state run")
time.sleep(5)
scope.write("acquire: state run")
scope.write("cursor:vbars:position1 0")
scope.write("cursor:vbars:position2 1")
scope.write("cursor:hbars: position1 0; position2 10")
```

1. 'res' is the variable where the resource is managed and store the available resource's access to another variable 'scope' that was opened.
2. FPANEL: PRESS is the header and mnemonic that takes care of front panel buttons of the oscilloscope. Here the defaultsetup button is pressed to return back the oscilloscopes original state.
3. Acquire query returns the message about the acquisition parameters like acquire: state that returns (ON/OFF/RUN/STOP) if the output 0 means the state of oscilloscope is STOP else if result is 1 then at RUN state and acquire: mode is set as hires that displays the better quality of signal. The sequence of waveform generated is predefined so after a specific time the state of DSO goes to STOP. Sleep is applied for some time to get the waveform comes to the position. Then it gets easy to fix the cursor.
4. CH1 refers channel 1 among the four channels is simply with scale mnemonic used to set the vertical scale is 1 volt. Horizontal scale is 10 seconds. The grid is at the scale of 1V vs 10S.
5. Cursor is used to check the amplitude and time between sample waveform points. There are two cursors horizontal an vertical cursor. If the cursor is disabled, function 'screen' makes that appears on the screen.
6. The cursor of vertical bars are set at the position of 0V and 1V. Horizontal bars of are at the position of 0S and 10S.
7. The number after the mnemonic represents the position of bar and simply the number indicates that it is Volt or Second. If want to set at the millisecond or millivolt position, the representation should be "0E-3, 1e-03, 10e-3". This indication says 0, 1, 10 millisecond or millivolt. For microsecond or micro-volt "100e-06, 2e-06, 4e-6" this is the format. Else it is possible to use decimal values like "1.2V, 0.000100S, 1.0004V " means 1 volt 2 millivolt, 100 microseconds, 1 volt 4 microvolt respectively.

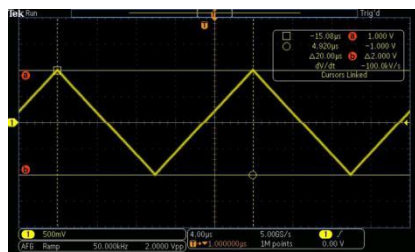


Figure 5. Vertical and Horizontal Cursor

8. It is possible to put single by activate the trigger function. Triggering is to capture very less frequency signals. This function does not run the screen with the default voltage level instead wait for signal and stop the signal at the trigger level when the amplitude of signal attains the slop fall or raise 'TRIGGER:A:EDGE:SLOPE FALL/RAISE'. For example 'TRIGGER:A:LEVEL: ch1 200e-06' this gets the waveform at 200 microvolt fall or raise, if fall once the signal drops from 200mv else if it is raise stops the run state when it raises from 200mv.
9. The command for A edge trigger "TRIGGER: A: EDGE: SOURCE ch1" that sets the trigger for source channel 1. After the signal is acquired then set the cursor according to it. Cursor also have mode like track and independent. Track only applies when the display: XY: withyt is set to off that tie the cursor together moves parallel one cursor dependent another. Independent can make it move freely without any dependency.

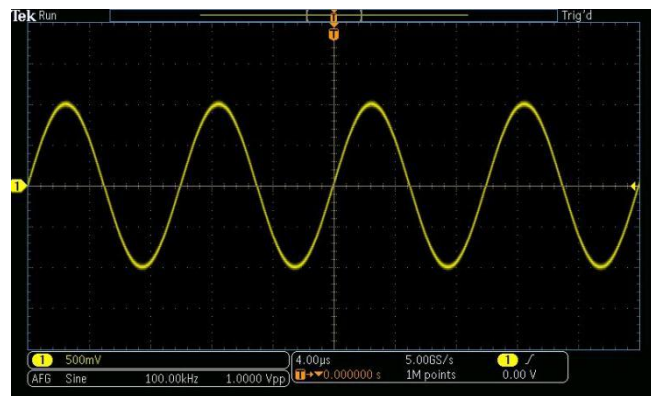


Fig 4. Triggered Wave

10. In addition to this python changes the intensity of displayed waveform DISPLAY:INTEITY:WAVEFORM 100 set 100% intensity. Increases the visibility of waveform on the screen.
11. After the cursor fixed then “save: Image :fileformat PNG/TIFF/JPEG” command is used to save the waveform. The inksaver is set off because it search for any printer or other printing device after the save command is executed. Then the “hardcopy start” is to save the image at the location. Use the ‘scope.read_raw()’ function to read the packed image values. This read data is stored in the variable and write it to another file at the location ans save it.

Further more commands are specified in the manual if required that have to be applied in the .py file.

- Source decides which channel’s waveform is to display on the screen. The below sine waveform emerges when the oscilloscope is in run mode and the signal is probed to channel 1 which is displayed yellow. Trigger also mentioned as 1.
- Fpanel is the front panel, instead of pressing buttons by manual use press command and button name.
- Horizontal and vertical scale commands to fix the unit scale value for both x and y axes. There are eight divisions in vertical direction one division has 500mv so totally 4v. In horizontal direction each division is 4 microsecond for ten divisions totally 40 microseconds.

V. CONCLUSION

In this project the work done is to automate the manual work of oscilloscope by using instrument commands. The appropriate commands that support the Tektronix oscilloscope work under python. Pyvisa is the module interface between oscilloscope and device. The command sends via the communication interfaces like USB, Ethernet, and GPIB. Among these interfaces Pyvisa supports the communication with the different protocols. Here NI-MAX is the application that contains all the drivers necessary for the different communication ports with different communication protocols. The role of this driver is to find the device that connected to the appropriate interface and send commands through the medium this is the way to access the device. FPGA is the board and logic units are programmed to generate different waveforms . The output pin where the signals get is probed to the oscilloscope channel to measure. The FPGAs logic block is programmed using Verilog and itsre-programmable to change the sequence.

REFERENCES

- [1] Zhang Qinchuan, Huang Wuhuang, Zeng Hao, “Study on enhancing features of digital oscilloscope to analyze the envelope of signal”, International Conference on Electronic Measurement & Instruments, pp. 30, 2017.
- [2] Michael T. McTigue, Kenneth W. Johnson, Edward Vernon Brush, “Measurement Probe Providing Different Levels of Amplification for Signals of Different Magnitude”, United States Patent, (2016).
- [3] Anna Goman, “Waveform Generator Implemented in FPGA with an Embedded Processor”, Linköping University, Sweden,(2003), pp. 9-24.
- [4] Vikas Kumar, N. Chandrasekhar, S.K. Albert, J. Jayapandian, “Analysis of Arc Welding Process Using Digital Storage Oscilloscope”, Measurement, (2016), pp. 1-12.
- [5] David F. Bacon, Rodric Rabbah, Sunil Shukla, “FPGA Programming for the Masses”, ACM queue, (2013).
- [6] K .Shimazoe, H. Takahashi, J . Y. Yeom, T. Furumiya, J. Ohi, “Multi- channel Waveform, sampling ASIC for radiation detection and measurement”, Radiation Measurement,(2013), pp. 87-89.
- [7] Roland Szabo, Aurel Gontean, Ioan Lie, Mircea BabaiNa, “Oscilloscope Control with PC”, International Journal of computers and communication, (2009), pp. 33-38.
- [8] Wanzeng Cai, Binwen Wang, Shoulong Zhang, “Remote Control and Data Acquisition of Multiple Oscilloscopes Using LabVIEW”, International Conference on Computer Technology, Electronics and Communication, (2017), pp. 920-923.
- [9] K.P.S. Rana, R. Singh , K.S. Sayann, “Correlation based novel technique for real time oscilloscope triggering for complex waveforms”, Measurement, (2009), pp. 299-311.
- [10] S. Reinhardt, C. Butschkow, S. Geissler, A. Dirnaichner, F. Olbrich, C.E. Lane, D. Schröer, A.K. Hüttel, “Lab::Measurement—A portable and extensible framework for controlling lab equipment and conducting measurements”, Computer Physics Communications, (2019), pp. 216–222.
- [11] LabVIEW URL <https://www.viewpointusa.com/labview/what-is-labview-used-for/>
- [12] National Instruments URL <https://www.ni.com/en-in/innovations/platform.html>