International Journal of Mechanics of Solids

E-ISSN: 2707-8078
P-ISSN: 2707-806X
IJMS 2023; 4(1): 15-20
Received: 04-11-2022
Accepted: 09-12-2022

Gamage Sanjeewa Samarasinghe
M.Sc. in Applied Electronics, Department of Physics, University of Colombo, Colombo, Sri Lanka

Corresponding Author:
Gamage Sanjeewa Samarasinghe
M.Sc. in Applied Electronics, Department of Physics, University of Colombo, Colombo, Sri Lanka

FPGA based logic analyzer

Gamage Sanjeewa Samarasinghe

Abstract
The FPGA-based logic analyzer is the level of a digital signal analyzing and encoding for understanding communication digital bit. In digital Instruments communications, each device internally has some sort of protocol. As an example, I2C and SPI. Some developers analyze digital signals to clarify or treble soothing problems. At that time Logic Analyzer are much more important. In this project develop a digital-level logic analyzer using FPGA based chip.
FPGA (Field-Programmable Gate Array) is a component for developing circuits using hardware definition languages. In this case, VDHL is the language, and Xilinx – Spartan 6 is the development board it used. There has a limitation to the maximum usage of FPGA. In Xilinx – Spartan 6 maximum SLICES (LUTs & Flip Flops) are 360. It can create $2^{\wedge 12}$ (4096) maximum RAM cells for devices.
In this device, the main components are SRAM and its content $2^{\wedge 12}$ memory addresses. SRAM in this device got two counting's and two controls for each side. Then SRAM content means to write and read data to SRAM using an input data analyzer. That component is Triggered by user-defined controls in python GUI using computer processing.
All of the command generates to control device using user defend python GUI. In the GUI content select sample rate and quantity. Each of parameters can change according to the user defined. So, all control defined by user according to the data use got. Sampling rate can change 1 Hz to 12.5 MHz and maximum sampling quantity was $2^{\wedge 12}$ (4096). According past experience user can decided best suitable sampling rate.
FPGA device and computer data communication done by UART (Universal Asynchronous Receiver Transmitter). Mainly hexadecimal numbers communication don for understand data to computer and ASCII character send for give command devices.

Keywords: FPGA-based logic analyzer, FPGA, Logic Analyzer, Python GUI, Python Interfacings for Instrumentation, Python Programming, Electronics Instrumentation for Digital Logic Analyzed, FPGA Instrumentation, Python Interface with FPGA, Instruments with FPGA & Python

Introduction
This is a logic analyzer for capturing logical signals for digital electronic circuit development purposes. Based on Xilinx FPGA (Field Program Gate Array) – Spartan 6 (XC6SLX9) chip with mush more limitations this research was conducted. XC6SLX9 family FPGA chip contains 90 K bit Ram space and it makes 4096 memory bytes for conducting SRAM Space. The memory counter makes $2^{\wedge 12}$ (4096) with 12 bus lines for data transfer on each side.

Background
This project uses many more components built by the FPGA programming language Cold VHDL. Each component develops according to the system development architecture. There have twelve hardware components and each component communicates with the other. Mainly in system architecture, there have memory unit-cooled SRAM. And this component got to read and write counters on each side for storing and retrieving memory controls. There have 2^{12} (4096) memory locations. Each location read and writes separately. And also, the separate two Buffer units for holding enter and out memory data to SRAM.

Methodology
Block Diagram
According to development, there have two Parts. FPGA Internal Circuity for devices and Python GUI for data representation.

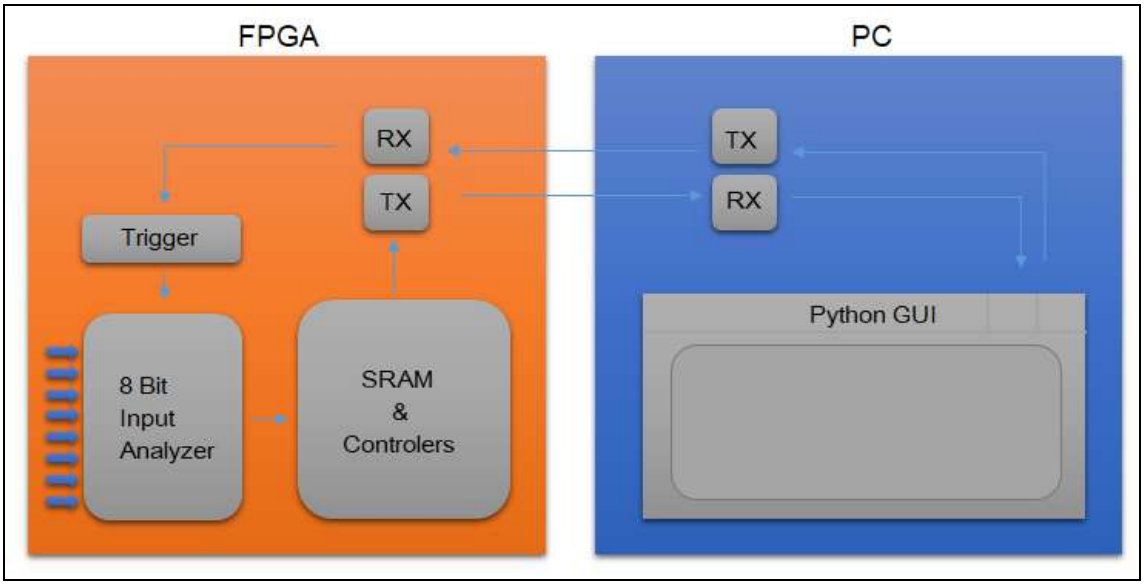

Fig 1: Frist level Block Diagram

2.2. Internal Device Diagram

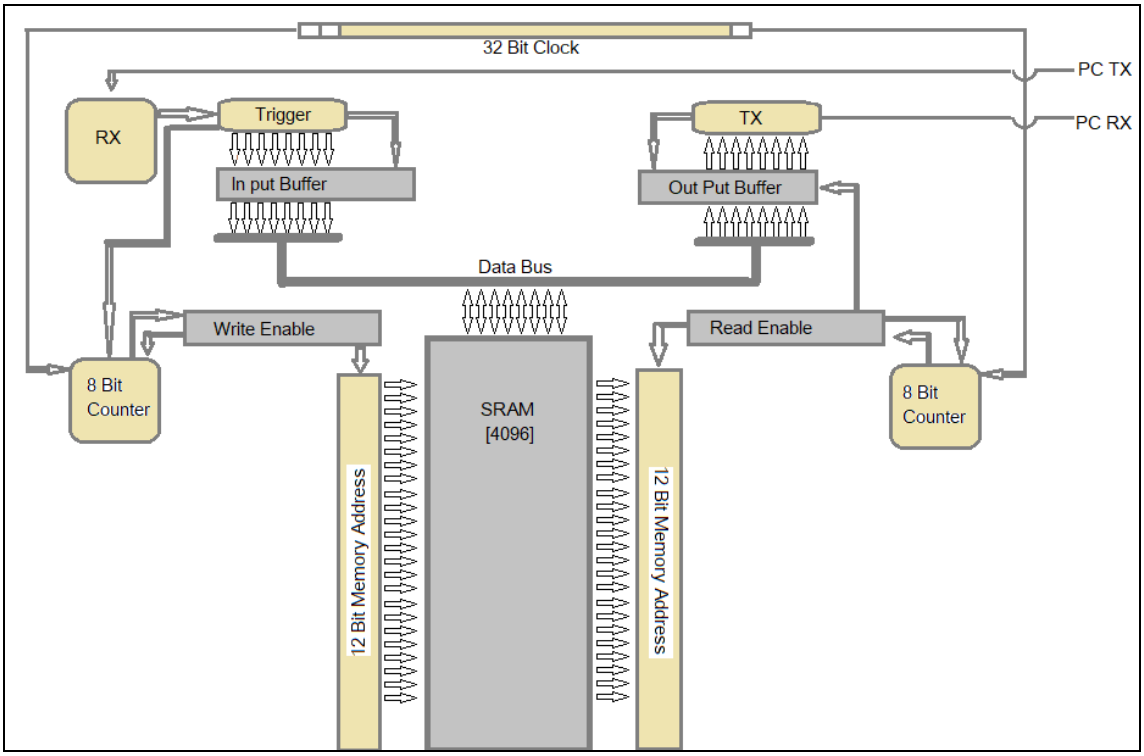

Fig 2: Internal Device Diagram

2.3. VHDL Program

Using the Xilinx ISE development tool, it creates towel internal devices in Spartan 6 FPGA for the operation of Logic Analyzer. Each device interconnects with the other for the combination of all operations in parallel.

This device's main component was SRAM. Each read-and-write operation is done separately. Using Time dividing rule each timer counter works separate clock signal. It handles by the 32-bit clock and read and write counter using a 2^{12} data line.

Each counter control by write and read enable units and Python GUI trigger each command separately. According to the timer selection, its cam generates a 12.5MHz signal for capturing the 8 Bits input given. There are

Using FPGA UART protocol device communications are done with the computer. RS323 serial communication chip interfacing each device.

2.4. Python Interfaces

There have several GUIs for displaying the output of data. Graphs for graph each input represents logic level 0 or 1. There have 8 graphs in the same GUI for identifying each input. And also, a Toolbox for the exam each access with numbers.

Python GUI added several features. Serial Monitor and SPI and I2C modulation are the each of features. There are several text editors' windows for modulation. Serial monitor window monitor logic level fix frequency.

Implementation

This device is mainly concerned with the hadal logic level in various digital devices. Logic Analyzer anomaly works with high speed like MHz range and gets more samples like thousand for the sample's logic signals.

Logic signal analysis is much more complex without modulation. Especially communication signals like I2C and SPI. So, this device store row data for modulation.

Manly, there must be a user-defined frequency range for the sampling logic level of each signal. Because sampling signal frequency can be changed according to each situation. In this case, the 1Hz to 12.5 MHz frequency range can change and might be entered into sampling.

Xilinx FPGA (Field Program Gate Array) – Spartan 6 (XC6SLX9) chip can have a maximum ram space of 90 Kbit. So, there have more limitations to using FPGA. This SRAM can contain only $2^{\wedge 12}$ (4096) memory cells with byte size.

Using a computer interface, it can communicate with an FPGA device using the UART protocol. It communicated 9600 bits per second speed. And also trigger signal also use define. The user can control much more variables in this device.

Synthesized RTL view

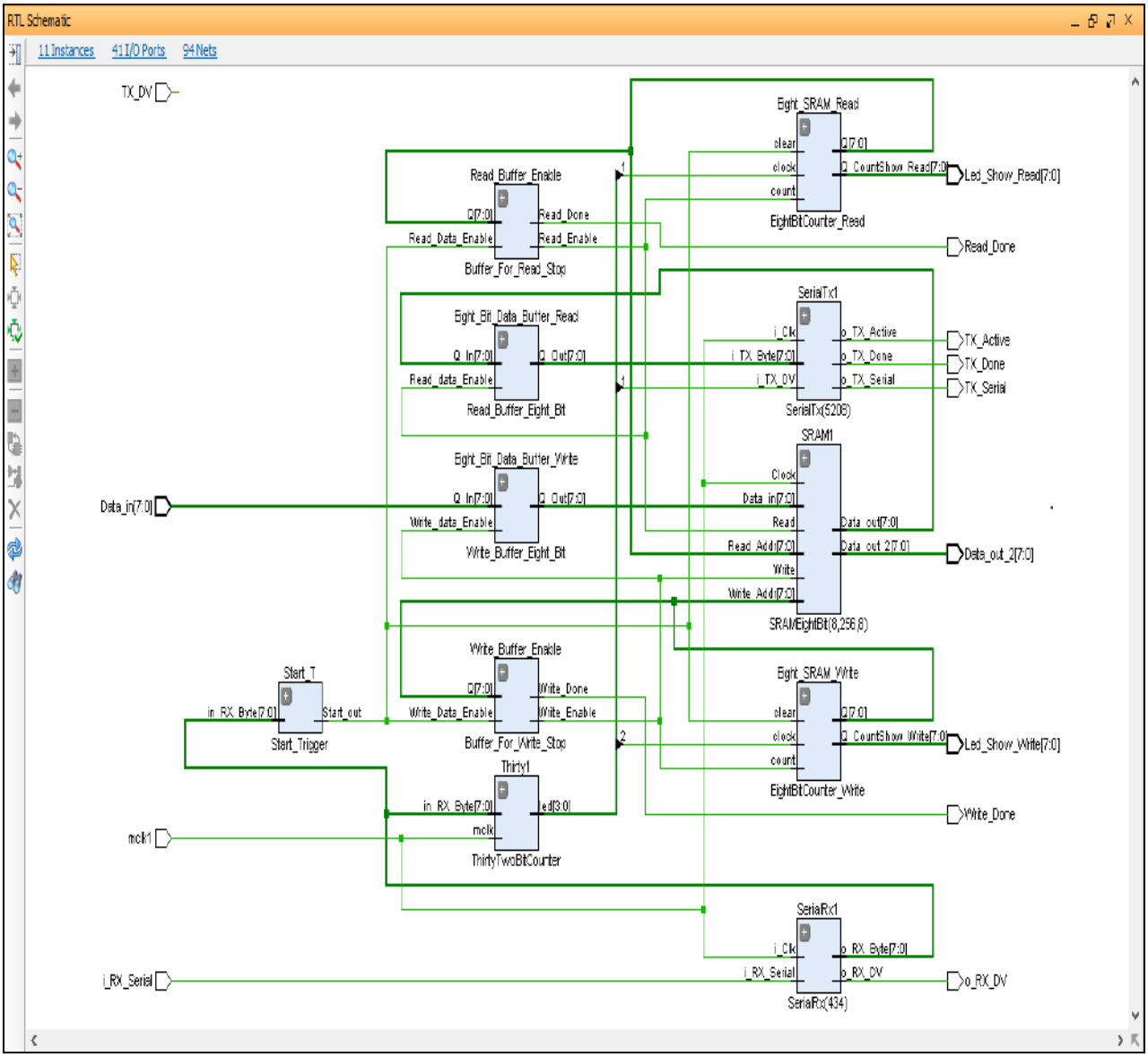

Fig 3: Synthesized RTL Diagram

FPGA Development Board Setup

This Development board allocates with Xilinx FPGA (Field Program Gate Array) – Spartan 6 (XC6SLX9) chip. It contains 9152 logic cells and 11440 Flip-Flops. And also, Maximum distributions of RAM (Kbits) 90. Then the total number of I/O Banks is 4 and the maximum number of users I/O is 200.

Using each I/O block to indicate SRAM read and write sequence each one separately. Each side has a towel data line for reading 4096 memory cells.

Data read 8 pin controls by Arduino Mega with an 8-bit counter for generating different each pin.

Fig 4: Development Setup with FPGA Board

Python GUI for Device

This program writes in Python programing language. There have several modules in this code, but the main GUI content is all of the functional capabilities.

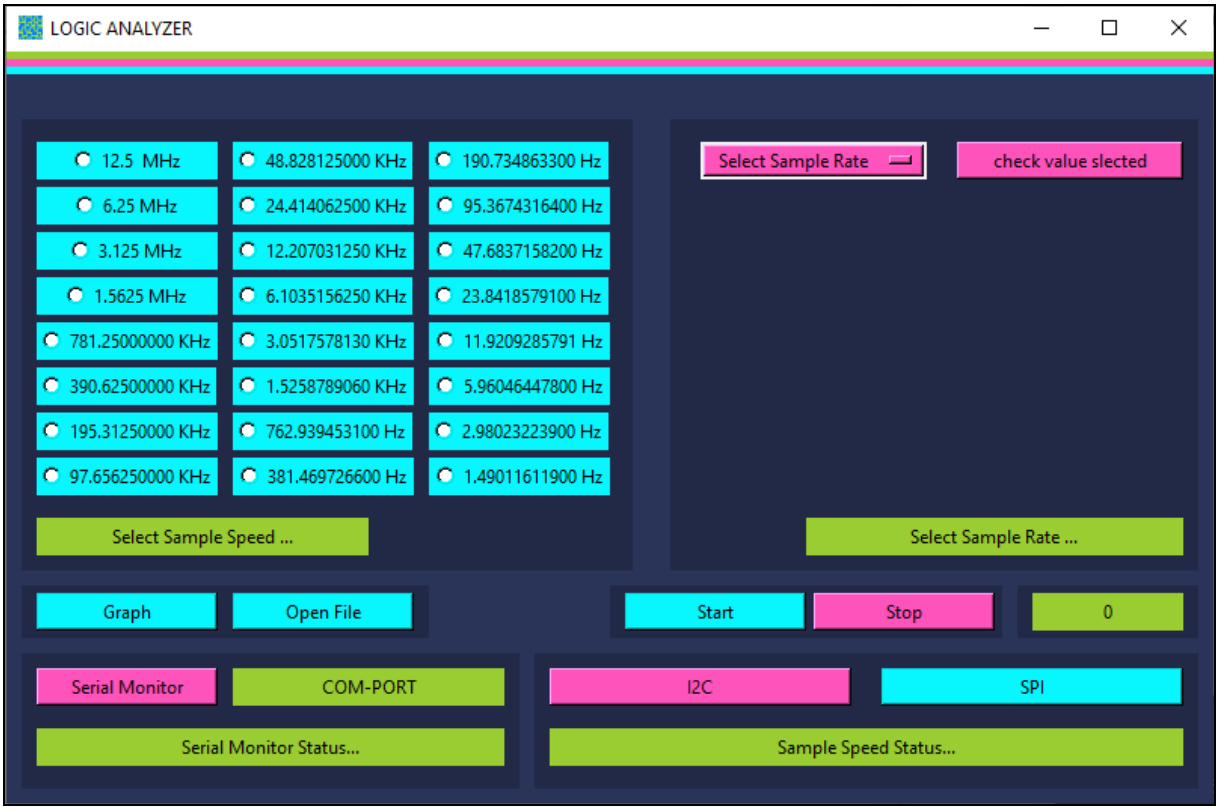

Fig 5: Python GUI for Logic Analyzer functional interface

4. Results and Discussion

4.1. Results: This device mainly contains two parts. One part run on the FPGA development board and the other run on a PC. Their communication builds in these two parts. That reason communicates more critically for the result. There has a data filter algorithm for collecting correct row data to make a good result. Then it creates raw data content text file and it encodes and represents in graphics. Several windows and text editors run to get the end product.

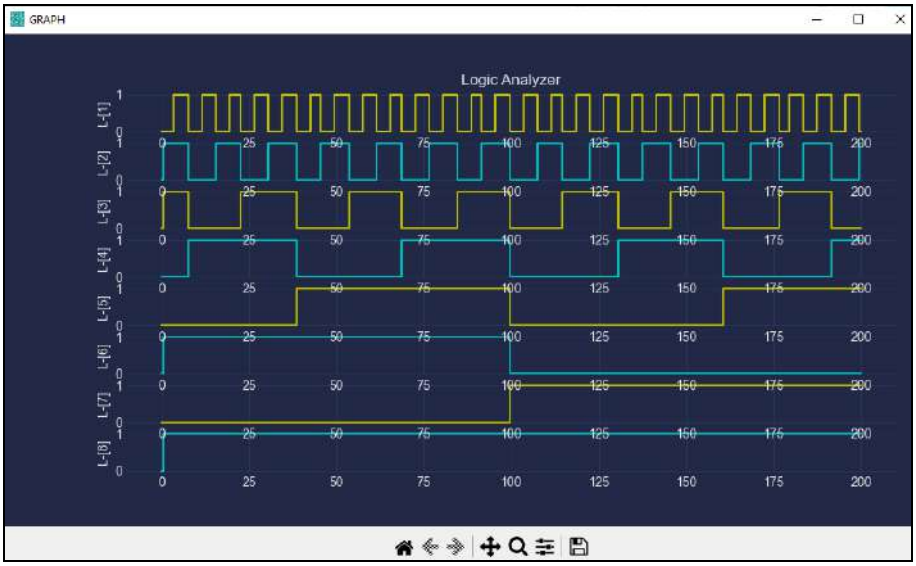

Fig 6: 100Hz 8-Bit Clock Signal Capture by Device [sample rate 23.8418579100 kHz]

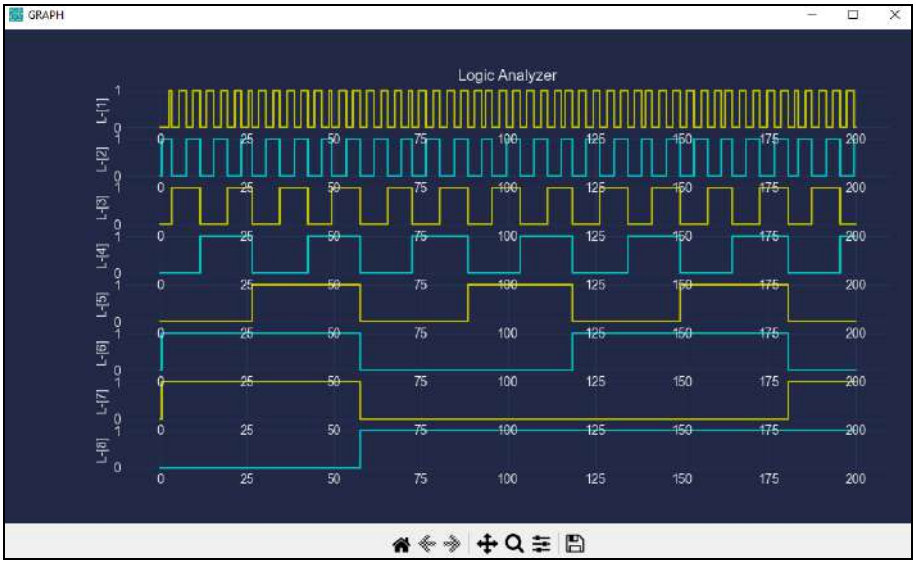

Fig 1: 10 KHz 8-Bit Clock Signal Capture by Device [sample rate 390.625000 kHz]

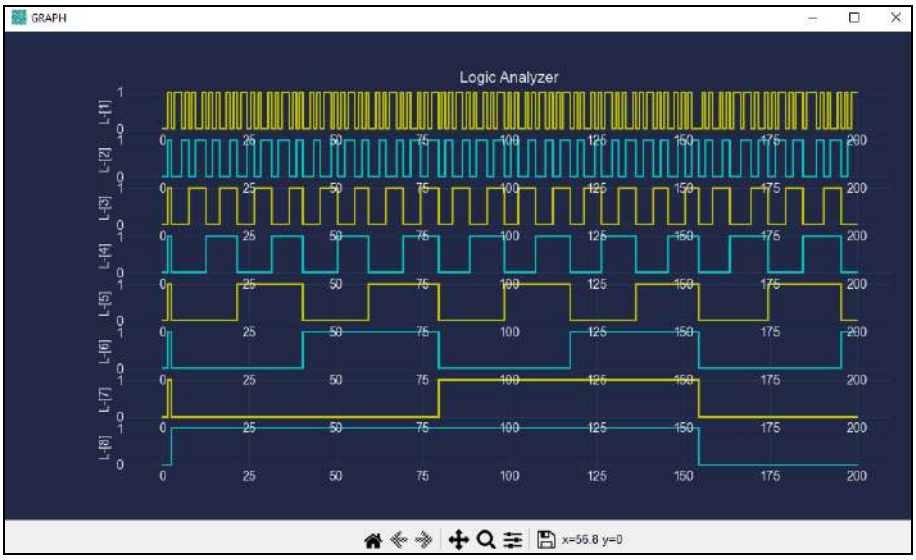

Fig 8: 1MHz 8-Bit Clock Signal Capture by Device [sample rate 6.250000MHz]

4.2. Discussion

Using Logic analyzer must want parties. Because it depends on your choice to sample signal. There has a low frequency for a sample and also it can be a high frequency. Then the user must have the experience choice of what frequency is selected for the sample. It can be five times higher than the

sampling signal frequency.

But if used under sample frequencies it not works well in this software. Using more sampling rates with more samples can get a good result. There have more techniques for generating more samples in a written way. Always user must try to get continuous data rang.

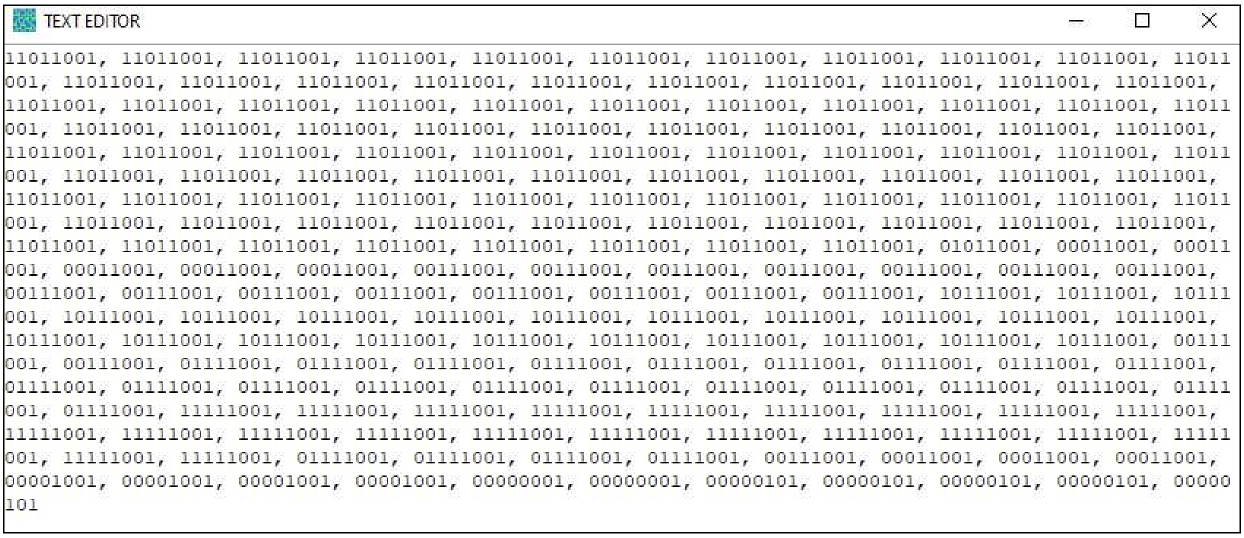

Fig 9: Row Capture Data for I2C SPI Modulation

5. Conclusions

The FPGA type of use depends on the performance of the device. More memory spaces create more advantages for development. But the problem was that small-size FPGA chip resources can't generate the best performance. According to this project FPGA also get maximum SRAM, not enough to get samples with fast frequency signal.

6. Acknowledgements

According to five months very much hard work was done to succeed in this project. Without any advice for path correction and writing guidelines create suitable devices never append. This search project supervised by Dr. M.K Jayananda was also very much helpful for the result. And also credit must apply to Dr. Hiran Jayaweera and Dr. Lakmal Weravarna added for conducting this project.

The report, writing guidelines line makes very much acceding to Prof. Upul J Sonnadara. So it will parent regarding him also.

Then also all unnamed Physics department personnel helps with conducting This project.

7. References

1. Dr. BK Mishra, *et al.* Design and simulation of UART for communication between FPGA and TDC ..., ResearchGate. International Journal of Scientific & Engineering Research, 2014. Volume 5. Available at: https://www.researchgate.net/publication/354311989_Design_and_Simulation_of_UART_for_Communication_between_FPGA_and_TDC_using_VHDL (Accessed: December 20, 2022).
2. Bhavanam DSN, Midasala V. Implementation of FPGA-based low-cost Logic Signal Analyzer (LSA), research gate. Int. J. of Recent Trends in Engineering and Technology; c2010. Available at: https://www.researchgate.net/publication/266393772_Implementation_of_FPGA_based_LOW-COST_Logic_Signal_Analyzer_LSA (Accessed: December 20, 2022).
3. Minseok KIM, Koichi ICHIGE, Hiroyuki ARAI. Implementation of FPGA based fast DOA estimator using unitary music, IEEE Xplore. IEICE TRANS. ELECTRON; c1999. Available at: https://ieeexplore.ieee.org/document/1285009 (Accessed: December 20, 2022).
4. Low-cost and programmable CRC implementation based on FPGA. YouTube. YouTube; c2021. Available at: https://www.youtube.com/watch?v=ngLnHqjaFRI (Accessed: December 20, 2022).
5. hardwarebee.com. 2023. FPGA Logic Analyzer. [ONLINE] Available at: https://hardwarebee.com/WikiBee/fpga-logic-analyzer/. [Accessed 14 December 2022].
6. Github.com. 2023. enxor-logic-analyzer. [ONLINE] Available at: https://github.com/lekgolo167/enxor-logic-analyzer. [Accessed 14 December 2022].
7. Hackaday.com. 2023. A MODULAR LOGIC ANALYZER FOR FPGAS. [ONLINE] Available at: https://hackaday.com/2019/05/26/a-modular-logic-analyzer-for-fpgas/. [Accessed 14 December 2022].
8. Bastli.ethz.ch. 2023. BitHound - An FPGA Based Logic Analyzer. [ONLINE] Available at: https://bastli.ethz.ch/index.php?page=bithound. [Accessed 14 February 2023].4 January 2023].