TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING
PURWANCHAL CAMPUS

A FINAL YEAR PROJECT REPORT ON

PC BASED LOGIC ANALYZER

SUBMITTED BY:
DINESH KUMAR BAN (PUR074BEX011)
MANI HANG MADEN (PUR074BEX022)
MILAN RAI (PUR074BEX023)
SHRIJANA SHRESTHA (PUR074BEX042)

A PROJECT REPORT
SUBMITTED TO THE DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING IN ELECTRONICS AND COMMUNICATION ENGINEERING

MAY, 2022
TRIBHUVAN UNIVERSITY
INSTITUTE OF ENGINEERING

DEPARTMENTAL ACCEPTANCE

The report entitled "**PC based Logic Analyzer**", submitted by Dinesh Kumar Ban, Mani Hang Maden, Milan Rai and Shrijana Shrestha in partial fulfillment of the requirement for the award of the degree of "Bachelor's Degree in Electronics and Communication Engineering" has been accepted as a bona fide record of work independently carried out by them in the department.

_____

Manoj Kumar Guragai

Head of the Department
Department of Electronics and Computer Engineering
Purwanchal Campus, Tribhuvan University
Nepal.

LETTER OF APPROVAL

TRIBHUVAN UNIVERSITY,
INSTITUTE OF ENGINEERING
PURWANCHAL CAMPUS
DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

The undersigned certify that have read, and recommended by the Institute of Engineering for acceptance, a final year project report entitled "PC BASED LOGIC ANALYZER" submitted by Dinesh Kumar Ban, Mani Hang Maden, Milan Rai and Shrijana Shrestha in partial fulfillment of the requirements for the Bachelor's degree in Electronics & Communication Engineering.

_____

Bishnu Chaudhary
Supervisor/Asst. Professor
Department of Electronics and Computer Engineering
Purwanchal Campus, IOE

_____

Surendra Shrestha, PhD
External Examiner/Reader
Department of Electronics and Computer Engineering
Pulchowk Campus, IOE

_____

Manoj Kumar Guragai
Head of Department/Asst. Professor
Department of Electronics and Computer Engineering
Purwanchal Campus, IOE

DATE OF APPROVAL: -May 19 2022

COPYRIGHT©

ACKNOWLEDGEMENT

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude to our final year project supervisor, Bishnu Chaudhary, whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project. We would like to thank our campus Purwanchal Campus, Eastern wing of IOE, for giving us the opportunity to pursue our engineering studies. Furthermore, we would also like to acknowledge with much appreciation Om Prakash Dhakal, Campus Chief, IOE

Purwanchal Campus Dharan, for providing us necessary resources for the successful completion of the project.

We express our deep gratitude to our teachers, honorable Electronics and Computer Engineering department members and Head of Department, Manoj Kumar Guragai. The in-time facilities provided by the department throughout the major project are also equally acknowledgeable.

ABSTRACT

Logic analyzers are an essential instrument required for circuit debugging. Standalone logic analyzers available in the market come with many features but its cost is high. Students and hobbyists may not require all the features but an affordable alternative modular device equipped with core functionality is more practical. In this paper an integration of hardware for acquisition of data and software for processing of data is introduced to achieve low cost logic analyzer. The system is implemented on the basis of Field Programmable Gate Array using VHDL programming. The overall system consists of logic analyzer software written in C++ for running on a PC, STM32F103C6 microcontroller and Spartan 6 FPGA. The FPGA board allows capturing of high frequency signals due to its high speed data paths. The captured data is stored in the FPGA memory block before it is transferred to the PC. The GUI based analyzer software running in PC plots captured data waveform and outputs decoded data.

Keywords: Logic Analyzer, FPGA, VHDL, GUI

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATION

ACK	Acknowledgement
CS	Chip Select
FPGA	Field Programmable Gate Array
HAL	Hardware Abstraction Layer
I2C	Inter-Integrated Circuit
MISO	Master In Slave Out
MOSI	Master Out Slave In
MUX	Multiplexer
NACK	Negative Acknowledgement
PC	Personal Computer
RAM	Random Access Memory
SCK	Serial Clock
SCL	Serial Clock
SDA	Serial Data
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
VHDL	Very High Speed Integrated Circuit Hardware Description Language

1. INTRODUCTION

1.1 BACKGROUND

A digital circuit is a circuit where the signal must be one of two discrete levels. Due to the advent of semiconductor manufacture technology microprocessors, microcontrollers and lots of peripheral devices are becoming more complex. In order to see what these devices are communicating with each other to perform complex tasks we need test equipment called logic analyzer. A logic analyzer is electronic equipment that is used to debug digital circuits. Logic Analyzer reads the digital data from DUT(device under test), then the captured data are displayed on a screen to show the timing relationship between different waveforms graphically.

1.2 PROBLEM STATEMENT

Stand-alone logic analyzer are expensive as they integrate all the features, such as the digital signal reading/capturing circuit, complex trigger circuit, sample memory, power supply, embedded system to handle all the functions and electronic display in a single package, due to its expensive nature it is inaccessible to hobbyist who want to use to debug their digital circuits. Oscilloscope are mainly used for analog circuit to measure the details of the analog signal such as rise time, fall time, ringing, signal stability etc. though they can be used to record digital signal, their lack of sufficient input channel required to sample large number of digital channel simultaneously makes it unsuitable for digital circuit debug. PC based logic analyzer provides the hobbyist to analyzes digital circuit with the trade of between low cost and high sample rates , memory depth to store the captured digital data and digital signal with fast rising edge and falling edge i.e. High frequency.

1.3 OBJECTIVE

To implement PC based logic analyzer that can plot timing diagrams to show timing relationship between waveforms and perform protocol decode.

1.4 MOTIVATION

In order to complete the final year major project according to curriculum we are motivated to complete this specific project as it has never been done previously by seniors. Undertaking this project also challenges us to learn new design techniques, work with new hardware and programming languages forcing us to be out of our comfort zones giving us an opportunity to further grow as an electronics and communication engineer, this is the driving motivation for the project.

2. LITERATURE REVIEW

A logic analyzer is an electronic instrument that captures and displays multiple signals from a digital system or digital circuit. A logic analyzer may convert the captured data into timing diagrams, protocol decodes, state machine traces, assembly language, or may correlate assembly with source-level software. Logic analyzers have advanced triggering capabilities, and are useful when a user needs to see the timing relationships between many signals in a digital system.[1]

Alexey M. Romanov in his research paper "An easy to implement logic analyzer for long-term precise measurements" [2] proposed a device capable of recording continuous measurement in the long-term while in different fields of scientific research where it is necessary to make data acquisition within small periods during several hours or even days. For example real-time communication worst-case jitter analysis. Romanov utilized FPGA for data capture and octave scripts for post processing and analysis.

In a research paper by Bhavanam and Dr. Midasala [3], Spartan 3 FPGA board was used for data capture and Java based GUI to plot waveform and verify the results on PC. They verified the design using the ModelSim simulator. Hardware verification was done using a test board with a PSoC microcontroller.

Karambelkar [4] concluded that ARM controller based Logic Analyzers are low cost, provide higher speed,testability and flexibility. ARM uses RISC machines.

3. REQUIREMENT ANALYSIS

PC based logic analyzer is used to capture the digital signal values from its input channel and to show the timing relation between different digital signals. To achieve this goals, the prototype that is developed consists of the following main components:

- Probe and Input buffer.
- Acquisition of the digital signal using FPGA.
- STM32 microcontroller which reads the captured/sampled data from the FPGA to transfer it to the PC software
- Logic analyzer software which runs on the host PC to display captured data and perform other processing.

3.1 HARDWARE REQUIREMENTS

3.1.1 Microcontroller STM32F103C6

The STM32F103x4 and STM32F103x6 performance line family incorporates the high-performance ARM® Cortex™-M3 32-bit RISC core operating at a 72 MHz frequency, high-speed embedded memories (Flash memory up to 32 Kbytes and SRAM up to 6 Kbytes), and an extensive range of enhanced I/Os and peripherals connected to two APB buses. All devices offer two 12-bit ADCs, three general purpose 16-bit timers plus one PWM timer, as well as standard and advanced communication interfaces: up to two I^2Cs and SPIs, three USARTs, an USB and a CAN.

These features make the STM32F103xx low-density performance line microcontroller family suitable for a wide range of applications such as motor drives, application control, medical and handheld equipment, PC and gaming peripherals, GPS platforms, industrial applications, PLCs, inverters, printers, scanners, alarm systems, video intercoms, and HVACs.[5]

All features:

- ARM 32-bit Cortex™-M3 CPU Core

    - 72 MHz maximum frequency,1.25 DMIPS/MHz (Dhrystone 2.1) performance at 0 wait state memory access
    - Single-cycle multiplication and hardware division
- Memories
    - 16 or 32 Kbytes of Flash memory
    - 6 or 10 Kbytes of SRAM
- Clock, reset and supply management
    - 2.0 to 3.6 V application supply and I/Os
    - POR, PDR, and programmable voltage detector (PVD)
    - 4-to-16 MHz crystal oscillator
    - Internal 8 MHz factory-trimmed RC
    - Internal 40 kHz RC
    - PLL for CPU clock
    - 32 kHz oscillator for RTC with calibration
- Low power
    - Sleep, Stop and Standby modes
    - VBAT supply for RTC and backup registers
- 2 x 12-bit, 1 μs A/D converters (up to 16 channels)
    - Conversion range: 0 to 3.6 V
    - Dual-sample and hold capability
    - Temperature sensor
- DMA
    - 7-channel DMA controller
    - Peripherals supported: timers, ADC, SPIs, I^2Cs and USARTs
- Up to 51 fast I/O ports
    - 26/37/51 I/Os, all mappable on 16 external interrupt vectors and almost all 5 V-tolerant
- Debug mode
    - Serial wire debug (SWD) & JTAG interfaces
- 6 timers
    - Two 16-bit timers, each with up to 4 IC/OC/PWM or pulse counter and quadrature (incremental) encoder input
    - 16-bit, motor control PWM timer with dead-time generation and emergency stop

    - 2 watchdog timers (Independent and Window)
    - SysTick timer 24-bit down counter
- 6 communication interfaces
    - 1 x I^2C interface (SMBus/PMBus)
    - 2 × USARTs (ISO 7816 interface, LIN, IrDA capability, modem control)
    - 1 × SPI (18 Mbit/s)
    - CAN interface (2.0B Active)
    - USB 2.0 full-speed interface
- CRC calculation unit, 96-bit unique ID
- Packages are ECOPACK®

STMicroelectronics' STM32F1 series of mainstream MCUs covers the needs of a large variety of applications in the industrial, medical and consumer markets. High performance with first-class peripherals and low-power, low-voltage operation is paired with a high level of integration at accessible prices with a simple architecture and easy-to-use tools. Typical applications include motor drives and application control, medical and handheld equipment, industrial applications, PLCs, inverters, printers, and scanners, alarm systems, video intercom,etc. [5]

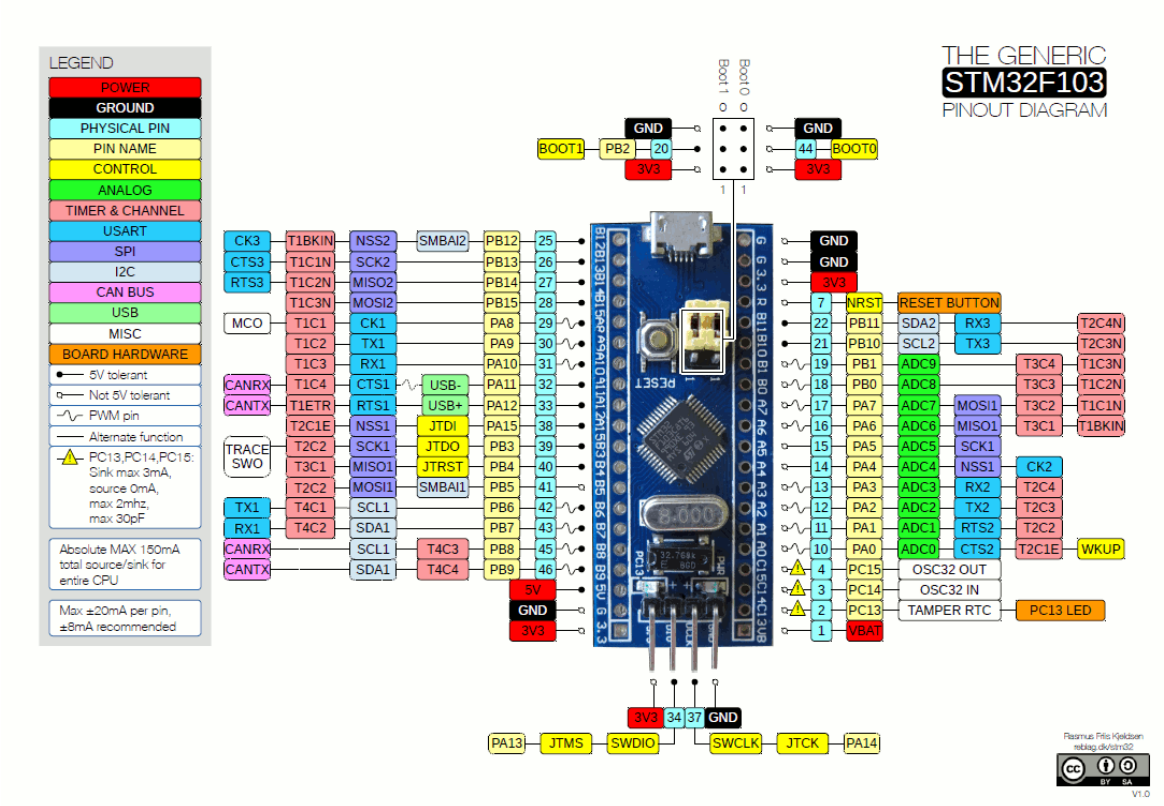


Figure 3.1 STM32F103 microcontroller
(Source:https://www.electronicshub.org/wp-content/uploads/2020/02/STM32F103C8T6-Blue-Pill-Pin-Layout.gif)

3.1.2 ST-LINK/V2

The ST-LINK/V2 is an in-circuit debugger and programmer for the STM8 and STM32 microcontrollers. The single-wire interface module (SWIM) and JTAG/serial wire debugging (SWD) interfaces are used to communicate with any STM8 or STM32 microcontroller located on an application board. In addition to providing the same functionalities as the ST-LINK/V2, the ST-LINK/V2-ISOL features digital isolation between the PC and the target application board. It also withstands voltages of up to 1000 V_{rms}.

STM8 applications use the USB full-speed interface to communicate with the ST Visual Develop (STVD-STM8) or ST Visual Programmer (STVP-STM8) software, or with integrated development environments from third-parties.

STM32 applications use the USB full-speed interface to communicate with the STM32CubeIDE software tool or with integrated development environments from third-parties.[6]

- Features
    - 5 V power supplied by a USB connector
    - USB 2.0 full-speed compatible interface
    - USB Type-A to Mini-B cable provided
    - SWIM specific features:
        - 1.65 V to 5.5 V application voltage support on the SWIM interface
        - SWIM low-speed and high-speed modes support
        - SWIM programming speed rates: 9.7 kbyte/s in low-speed, 12.8 kbyte/s in high-speed
        - 214017Horizontal connector reference: 214012
        - SWIM cable for connection to an application with pin headers or 2.54 mm pitch connector
    - JTAG/serial wire debug (SWD) specific features:
        - 1.65 V to 3.6 V application voltage support on the JTAG/SWD interface and 5 V tolerant inputs

- JTAG cable for connection to a standard JTAG 20-pin 2.54 mm pitch connector
- JTAG support
- SWD and serial wire viewer (SWV) communication support

- Direct firmware update support (DFU)
- Status LED blinking during the communication with the PC
- Operating temperature from 0 °C to 50 °C
- 1000 V_{rms} high isolation voltage (ST-LINK/V2-ISOL only)

Figure 3.2: ST-LINK V2

3.1.3 Xilinx Spartan 6 XC6SLX9 FPGA Development Board

The Xilinx Spartan®-6 FPGA family provides leading system integration capabilities with lowest total cost for high volume applications. It is built on 45nm technology and ideally suited for advanced bridging applications in automotive infotainment, consumer and industrial automation. Spartan®-6 FPGA family comprises two domain optimized subfamilies LX (logic optimized) and LXT (high speed serial connectivity). Spartan®-6 LX FPGAs are optimized for applications that require absolute low cost. They support up to 147K logic cell density, 4.8Mb memory, integrated memory controllers, DSP48A1 slices and high performance integrated IP with support for industry standards. Spartan®-6 LXT FPGAs are optimized to provide the industry's lowest risk and lowest cost solution for serial connectivity. LXT subfamily extends LX devices by adding up to eight 3.2Gb/s GTP transceivers and an integrated block for PCI Express® compatible endpoint block both derived from proven Virtex® FPGA family technology. [7]

- -3, -3N, -2, -1L (LX)/-3, -3N, -2 (LXT) speed grades with 0 to 85°C and -40 to 100°C temperature
- Small form factor packaging, MicroBlaze™ soft processor, diverse number of supported I/O protocols
- Advanced power management, SelectIO™ interface technology, 18Kb (2 x 9Kb) block RAMs
- Spartan®-6 provides increased system performance with up to 8 low power 3.2Gb/s serial transceivers
- 800Mb/s DDR3 with integrated memory controller, zero power with hibernate power-down mode
- Lower power 1V core voltage (LX/-1L), high performance 1.2V core voltage (LX, LXT/-2, -3, -3N)

3.1.4 Xilinx Platform Cable USB

Platform Cable USB (Figure 3.4) is a high-performance download cable attaching to user hardware for the purpose of programming or configuring any of the following Xilinx devices:

• ISP Configuration PROMs

• CPLDs

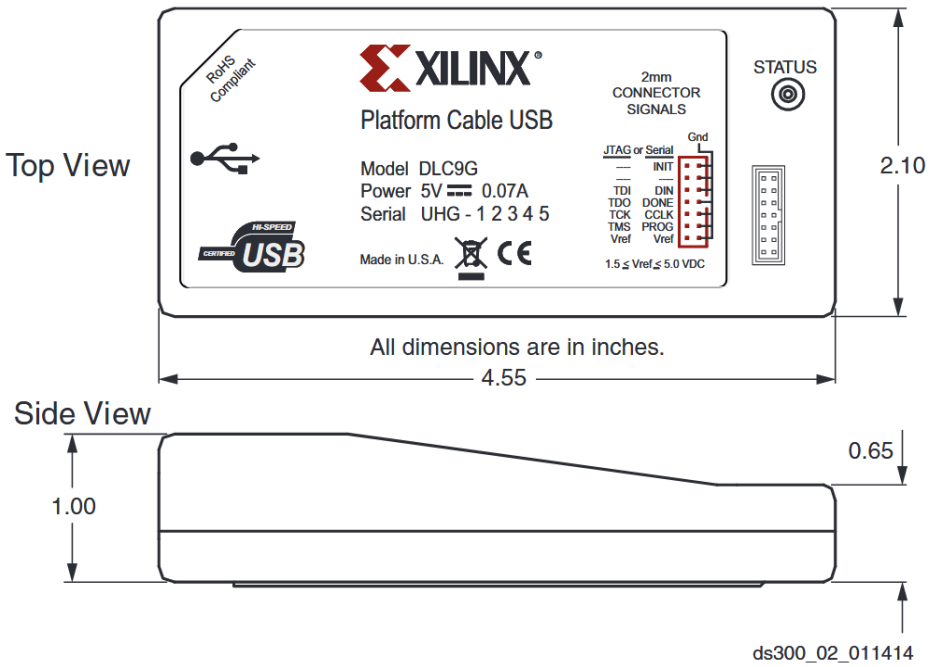
• FPGAs

Platform Cable USB attaches to the USB port on a desktop or laptop PC with an off-the-shelf Hi-Speed USB A-B cable.It derives all operating power from the hub port controller.No external power supply is required. A sustained slave-serial FPGA configuration transfer rate of 24 Mb/s ispossible in a Hi-Speed USB environment. Actual transferrates can vary if bandwidth of the hub is being shared with the USB peripheral devices.Device configuration and programming operations usingPlatform Cable USB are supported by iMPACT downloadsoftware using Boundary-Scan (IEEE 1149.1 / IEEE 1532),slave-serial mode, or serial peripheral interface (SPI). PlatformCable USB supports indirect (via an FPGA IEEE 1149.1[JTAG] port) programming of select flash memories including the Platform Flash XL configuration and storage device. Target Clock speeds are selectable from 750 kHz to 24

MHz.Platform Cable USB attaches to target systems using a14-conductor ribbon cable designed for high-bandwidth data transfers. An optional adapter that allows attachment of a flying lead set is included for backward compatibility with target systems that do not use the ribbon cable connector. [8]

Features:

- Performance
    - Utilizing CY7C68013A+FPGA new hardware solution, compatible with the original XILINX Platform Cable USB and Platform Cable USB II
    - Programs all Xilinx devices, including FPGAs / CPLDs / ISP Configuration PROMs
    - Supports JTAG, Slave-Serial and SPI programming, to configure all Xilinx devices
    - Interfaces to devices operating at 5V / 3.3V / 2.5V / 1.8V / 1.5V
    - Optional target clock frequency, supports XILINX software automatic frequency adjustment
    - Anti-static and multi-layers signal isolation, to protect the target and the programmer itself
    - Comes with adapter board and multi cables for easily connecting with official development board or other circuit boards
- Supported software
    - Xilinx ISE
    - iMPACT
    - ChipScope
    - Vivado
- Supported devices
    - Xilinx FPGA devices: UltraScale / 7 series / Zynq / Kintex / Virtex / Artix / Spartan, etc.
    - Xilinx CPLD devices: XC9500 / XC9500XL / XC9500XV / CoolRunner XPLA3 / CoolRunner-II, etc.
    - Xilinx PROM devices: Platform Flash XCF00S / XCF00P / XL, etc.



(Source:-https://www.mouser.cn/datasheet/2/903/xilinx_xili-s-a0001696175-1-1759481.pdf)

Figure 3.3:Xilinx platform Cable USB
(Source:https://www.arduitronics.com/product/1450/xilinx-platform-usb-cable-dlc9lp)

3.1.5 CP2102 Based USB to TTL Convertor Module

CP2102 chip from SiLabs is a single chip USB to UART Bridge IC. It requires minimal external components. CP2102 can be used to migrate legacy serial port based devices to USB. Hobbyists can use it as a powerful tool to make all kinds of PC interfaced projects. This module help all those who are comfortable with RS232/Serial Communication protocol, to build USB devices very easily.[9]

Working:

This is an USB2.0 to TTL UART Converter module which is based on CP2102 Bridge by SiLabs. This module can be used with Laptops which don't have a standard serial port. This module creates a virtual COM port using USB on your computer which can support various standard Baud Rates for serial communication. You just need to install the driver using a setup file which automatically installs correct driver files for Windows XP/Vista/ 7. After driver installation, plug the module into any USB port of your PC. Finally a new COM port is made available to the PC. The feature which makes it more convenient is the TTL level data i/o. So you don't need to make a RS232 to TTL converter using chips like MAX232. The Rx and Tx pin can be connected directly to the MCUs pins (assuming 5v i/o). [5]

Pinouts:

This module has 6 pin breakout which includes

- TXD = Transmit Output - Connect to Receive Pin(RXD) of Micro controller. This pin is TX pin of CP2102 on board.
- RXD = Receive Input - Connect to Transmit Pin(TXD) of Micro controller. This pin is RX pin of CP2102 on board.
- GND = Should be common to microcontroller ground.
- 3V3 = Optional output to power external circuit upto 50mA.
- 5V = Optional output to power external circuit upto 500mA
- DTR/RST = Optional output pin to reset external microcontrollers like Arduino.

Features:

- Stable and reliable chipset CP2102.

- USB specification 2.0 compliant with full-speed 12Mbps.
- Standard USB type A male and TTL 6pin connector.
- 6 pins for 3.3V, RST, TXD, RXD, GND & 5V.
- All handshaking and modem interface signals.
- Baud rates: 300 bps to 1.5 Mbps.
- Byte receive buffer; 640 byte transmit buffer.
- Hardware or X-On/X-Off handshaking supported.
- Event characters support Line break transmission.
- USB suspend states supported via SUSPEND pins.
- Temperature Range: -40 to +85.
- Size: 42mm X 15mm.
- Weight: 4g

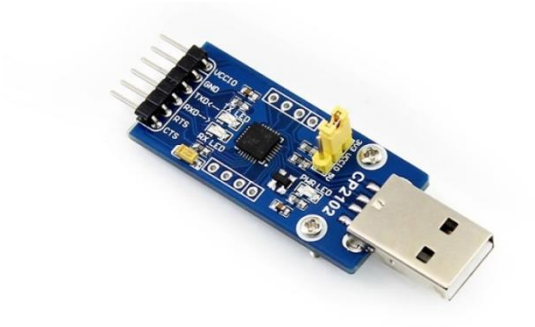


Figure 3.4: USB to Serial Converter
(source:https://www.waveshare.com/cp2102-usb-uart-board-type-a.htm)

3.1.6 SN74F244N

These octal buffers and line drivers are designed specifically to improve both the performance and density of 3-state memory address drivers, clock drivers, and bus-oriented receivers and transmitters. Taken together with the ´F240 and ´F241, these devices provide the choice of selected combinations of inverting and non-inverting outputs, symmetrical $\underline{OE}$ (active-low output-enable) inputs, and complementary OE and $\underline{OE}$ inputs.

The ´F244 is organized as two 4-bit buffers/line drivers with separate output enable ($\underline{OE}$) inputs. When $\underline{OE}$ is low, the device passes data from the A inputs to the Y outputs. When $\underline{OE}$ is high, the outputs are in the high-impedance state.

The SN74F244 is available in TI's shrink small-outline package (DB), which provides the same I/O pin count and functionality of standard small-outline packages in less than half the printed-circuit-board area.

The SN54F244 is characterized for operation over the full military temperature range of -55°C to 125°C. The SN74F244 is characterized for operation from 0°C to 70°C. [10]

Features:

- 3-State Outputs Drive Bus Lines or Buffer Memory Address Registers
- Package Options Include Plastic Small-Outline (SOIC) and Shrink Small-Outline (SSOP) Packages, Ceramic Chip Carriers, and Plastic and Ceramic DIPs

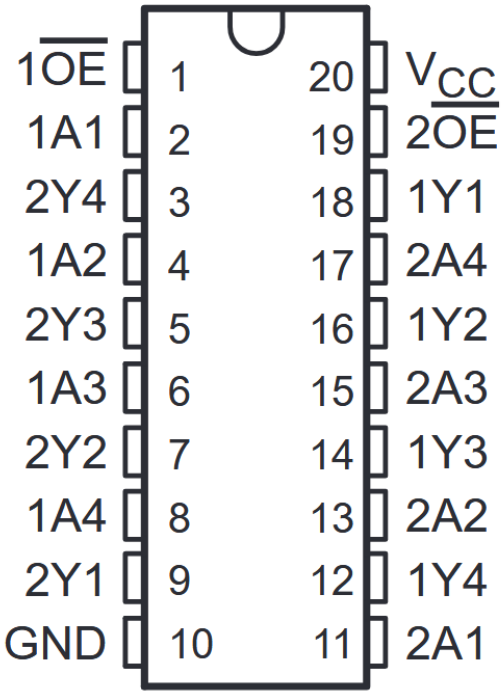


Figure 3.5: SN74F224 pinout

(source:https://datasheet.octopart.com/SN74F244N-Texas-Instruments-datasheet-8442283.pdf)

3.2 SOFTWARE REQUIREMENTS

3.2.1 Microsoft Visual Studio

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Visual Studio includes a code editor supporting IntelliSense (the code completion component) as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a code profiler, designer for building GUI applications, web designer, class designer, and database schema designer. It accepts plug-ins that expand the functionality at almost every level—including adding support for source control systems and adding new tool sets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle.

Visual Studio supports 36 different programming languages and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include C, C++, C++/CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML, and CSS. Support for other languages such as Python, Ruby, Node.js, and M among others is available via plug-ins. Java (and J#) were supported in the past. [11]

3.2.2 STM32CubeIDE

STM32CubeIDE is an all-in-one multi-OS development tool, which is part of the STM32Cube software ecosystem.

STM32CubeIDE is an advanced C/C++ development platform with peripheral configuration, code generation, code compilation, and debug features for STM32 microcontrollers and microprocessors. It is based on the Eclipse®/CDT™ framework and GCC toolchain for the development, and GDB for the debugging. It allows the integration of the hundreds of existing plugins that complete the features of the Eclipse® IDE.

STM32CubeIDE integrates STM32 configuration and project creation functionalities from STM32CubeMX to offer all-in-one tool experience and save installation and development time. After the selection of an empty STM32 MCU or MPU, or pre-configured microcontroller or microprocessor from the selection of a board or the selection of an example, the project is created and initialization code generated. At any time during the development, the user can return to the initialization and configuration of the peripherals or middleware and regenerate the initialization code with no impact on the user code.[12]

STM32CubeIDE includes build and stack analyzers that provide the user with useful information about project status and memory requirements.

STM32CubeIDE also includes standard and advanced debugging features including views of CPU core registers, memories, and peripheral registers, as well as live variable watch, Serial Wire Viewer interface, or fault analyzer.[12]

All features:

- Integration of services from STM32CubeMX:STM32 microcontroller, microprocessor, development platform and example project selectionPinout, clock, peripheral, and middleware configurationProject creation and generation of the initialization codeSoftware and middleware completed with enhanced STM32Cube Expansion Packages
- Based on Eclipse®/CDT™, with support for Eclipse® add-ons, GNU C/C++ for Arm® tool chain and GDB debugger
- STM32MP1 Series:Support for OpenSTLinux projects: LinuxSupport for Linux
- Additional advanced debug features including:CPU core, peripheral register, and memory viewsLive variable watch viewSystem analysis and real-time tracing (SWV)CPU fault analysis toolRTOS-aware debug support including Azure
- Support for ST-LINK (STMicroelectronics) and J-Link (SEGGER) debug probes
- Import project from Atollic® TrueSTUDIO® and AC6 System Workbench for STM32 (SW4STM32)
- Multi-OS support: Windows®, Linux®, and macOS®, 64-bit versions only

3.2.3 Xilinx ISE Design Suite

Xilinx ISE (Integrated Synthesis Environment)is a software tool from Xilinx for synthesis and analysis of HDL designs, which primarily targets development of embedded firmware for Xilinx FPGA and CPLD integrated circuit (IC) product families. It was succeeded by Xilinx Vivado. Use of the last released edition from October 2013 continues for in-system programming of legacy hardware designs containing older FPGAs and CPLDs otherwise orphaned by the replacement design tool, Vivado Design Suite.[13]

ISE enables the developer to synthesize ("compile") their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. Other components shipped with the Xilinx ISE include the Embedded Development Kit (EDK), a Software Development Kit (SDK) and ChipScope Pro.The Xilinx ISE is primarily used for circuit synthesis and design, while ISIM or the ModelSim logic simulator is used for system-level testing.[14]

As commonly practiced in the commercial electronic design automation sector, Xilinx ISE is tightly-coupled to the architecture of Xilinx's own chips (the internals of which are highly proprietary) and cannot be used with FPGA products from other vendors.

3.2.4 Programming Language and Framework

C++ is used to write the logic analyzer software for PCs. C++ is a general-purpose programming language developed by Danish computer scientist BjarneStroustrup as an extension to the C programming language. C++ gives programmers a high level of control over system resources and memory. C++ is an object-oriented programming language which gives a clear structure to programs and allows code to be reused, lowering development costs.[15]

We use the Win32 API for serial communication. Win32 API (officially called the Microsoft Windows API) is an application programming interface written in C by

Microsoft to allow access to Windows features. The main components of the WinAPI are:

WinBase: The kernel functions, CreateFile, CreateProcess, etc
WinUser: The GUI functions, CreateWindow, RegisterClass, etc
WinGDI: The graphics functions, Ellipse, SelectObject, etc
Common controls: Standard controls, list views, sliders, etc. [16]

WxWidgets is used for developing Graphical User Interface (GUI). wxWidgets is a C++ library that lets developers create applications for Windows, macOS, Linux and other platforms with a single code base. It has popular language bindings for Python, Perl, Ruby and many other languages, and unlike other cross-platform toolkits, wxWidgets gives applications a truly native look and feel because it uses the platform's native API rather than emulating the GUI.[17]

STM32F103C6 microcontroller is programmed using C programming language and HAL library. VHDL design is implemented in FPGA.

4. SYSTEM OVERVIEW

The proposed PC based logic analyzer consists of the logic analyzer software written in C++ that runs on the host PC, stm32f103c8t6 microcontroller, USB to Serial converter module and XilinxSpartan 6 FPGA.

4.1 BLOCK DIAGRAM

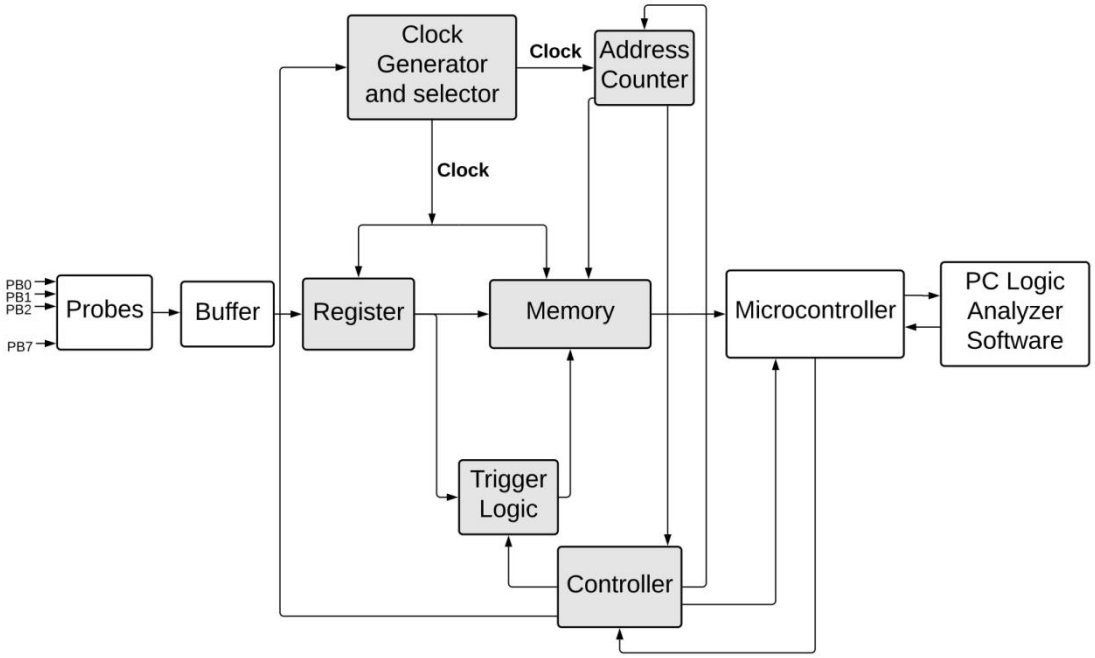


Figure 4.1: Block diagram of PC based logic analyzer

Figure 4.1 is the block diagram of the PC based logic analyzer. It consists of the logic analyzer software, microcontroller and FPGA. These different components are required to perform certain definite tasks such as digital signal data acquisition, communication between the pc software to transfer the acquired data and to display the recorded data and perform decoding function. The logic analyzer software on the host PC receives the sampled digital data from digital channel in single acquisition and displays the graphical waveform diagram. From the drawn waveform diagram, the timing relationship between different digital signals can be obtained and understood. The PC logic analyzer software also decodes widely used communication protocols like I2C and SPI as well as parallel bus data. The stm32 microcontroller is used to perform the task of transferring large data from the FPGA to the PC and

receive the memory depth and sample rate values to be used for the acquisition from the PC and set it in the FPGA. The stm32 microcontroller transfers this using the UART peripheral interface. Since the connection between the PC and microcontroller is USB, the USB to Serial converter module is used for the translation between USB to Serial. The acquisition circuit which consists of Block RAM Memory *[18]*, address counter, Clock generator and selector, register, trigger logic and control logic is implemented inside the FPGA. The Block RAM is one of the components inside the FPGA which is used to store large amounts of data hence large memory depth can be achieved which could not be achieved if microcontroller was used to store the sampled digital signal. Address counter points to the memory location where the sampled digital signal is to be stored. Trigger logic generates the trigger signal at the correct condition which enables the address counter operation and the sampling process starts until the memory is filled with the sampled digital data. Register holds the sampled data for one clock period whose output is fed to the trigger logic which generates the trigger signal on the correct input and the input to the Block Ram. The clock generator and selector consists of a simple up counter which performs division by 2 at each output to generate different clock frequency and hence different sample rate. The clock selector is a simple multiplexer circuit to select the correct clock rate/ sample rate for other components of the acquisition circuit. The control logic is the glue of the entire acquisition circuit which selects the correct sample rate, memory depth, starts and stops acquisition, generates control signals for the microcontroller and receives different control signals from the microcontroller.

5. METHODOLOGY

A logic analyzer reads digital data even though the actual signal is analog with certain amplitude and characteristics, in digital electronics the signal can only exits between 2 states i.e. either 1 or HIGH if the signal is above a certain threshold and 0 or LOW if the signal is below a certain threshold.

So a register/latch can be simply used to read the value of the digital signal which can only take 2 states. This is clarified from the figures and explanation below.

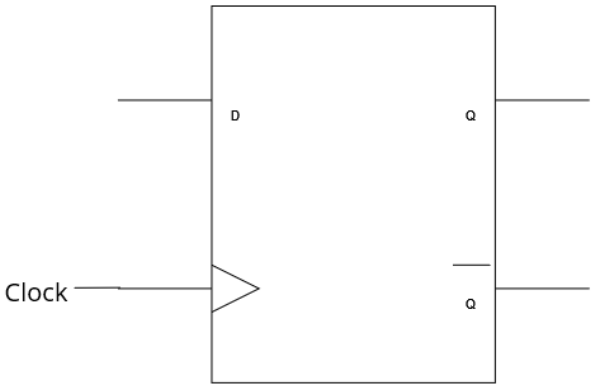


Figure 5.a: D Flip Flop

The Figure 5.a is a D flip flop which stores 1 bit of data which is present at the D input during the rising edge of the clock.

Table: 5.a: D flip flop Truth table

Clock	D	Q
1	1	1
0	0	1
1	0	0
0	1	0

Even though the actual signal at the input of the D flip flop is an analog signal, but in digital electronics the signal can be only in two state i.e. HIGH or LOW. If the input

signal exceeds a certain threshold that is registered as having HIGH input, the D flip flop stores 1 else it stores 0 during the rising edge of the clock. This is the processing of sampling or reading the digital signal, which is different from the sampling used in ADC. Now this sampled data can be stored in a memory device such as RAM

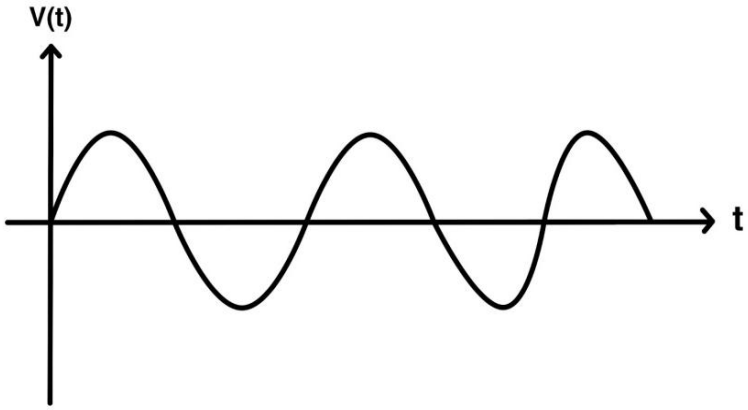


Figure 5.b: Analog Signal

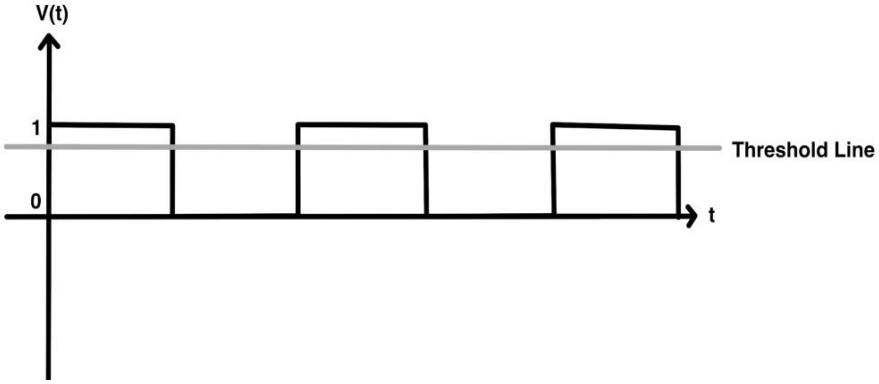


Figure 5.c: Digital Signal

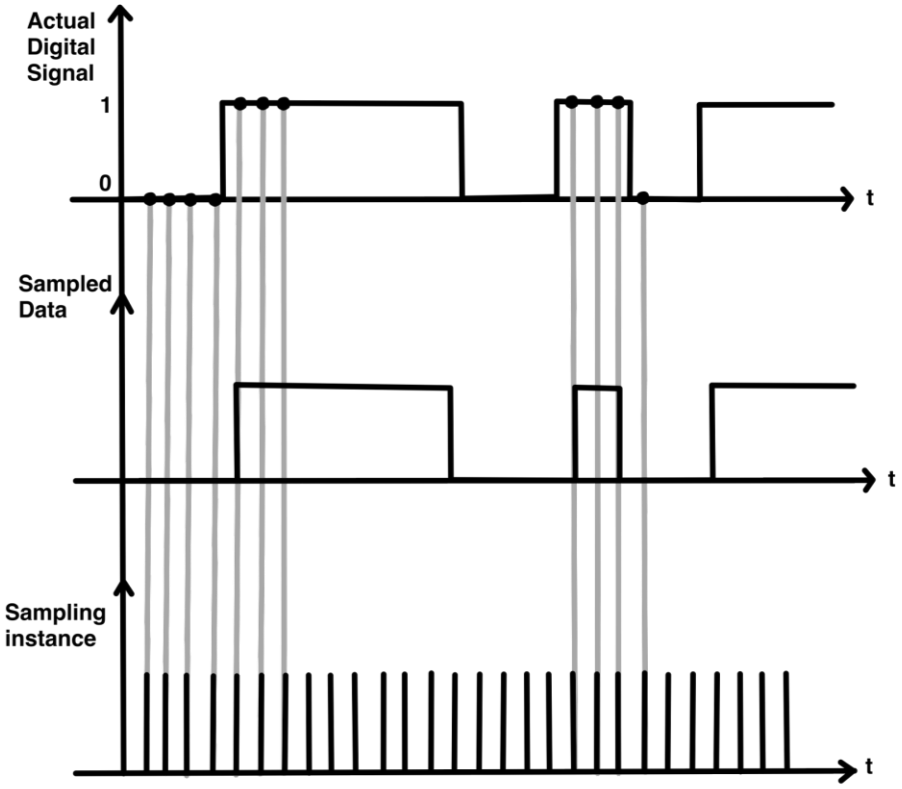


Figure 5.d: Illustration of sampling of digital signal

The number of inputs of the register is equal to the total number of channels i.e. the total number of signals that can be captured simultaneously. The captured data is stored in sample memory (RAM where read data are stored) location pointed by the address counter. This process of reading the digital data and storing in the next memory location is repeated continuously until it reaches the end of the memory location or specified sample depth. Once the sample memory is filled with the captured data, all the stored/captured data is sent to the pc where the software reads the data and displays it.

The proposed PC based logic analyzer has two main parts, one is the PC software where the users can configure the logic analyzer such as the capture/sample rate, memory depth and trigger setting i.e. triggering on rising or falling edge of certain signal or triggering on the certain word such as 0x0F. The capture/sample rate is important as it defines the resolution of the signal.

The memory depth is also important as it specifies how much data is to be captured during a single acquisition. Large memory depth can be used to capture large data such as long events or events whose trigger condition is not certain.The sample rate and the memory depth defines how long the acquisition time will be.
The software running on PC reads back all the captured data and graphically displays the signal to show the timing relationship between them.

The other part of the logic analyzer is concerned with the actual capturing of the data based on the configuration set by the user on the PC software. In order to interface with the PC, a microcontroller is used to communicate/transfer data between PC i.e. transferring configuration information such as sample rate, memory depth and trigger settings from PC to logic analyzer and transferring the captured data from FPGA block RAM to PC. Data acquisition part can be implanted on the microcontroller itself but implementing it on a FPGA would give a faster sampling rate as compared to a microcontroller and can meet the timing constraint of reading the data at specific instantaneous time. Plus the FPGA has a larger Memory resource in comparison to Microcontroller, hence large memory depth can be achieved which can be used to capture large or uncertain events successfully. The FPGA receives different configuration information from the microcontroller about the sampling rate, memory depth, trigger setting and different control signal to start the capturing process, reading the captured data and ending the capturing process. Modern FPGA has built in block RAM in its fabric, this project utilizes this block RAM to store the read data [18].

5.1 LOGIC DIAGRAM

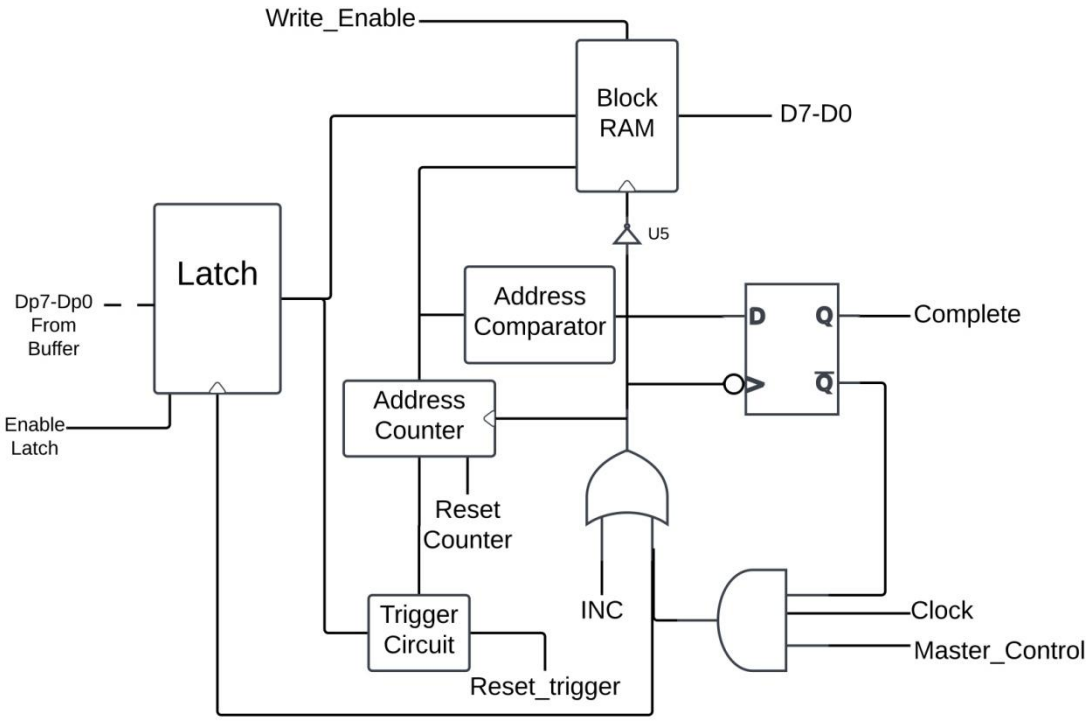


Figure 5.1: Data acquisition logic implementation inside FPGA

The data acquisition circuit which is responsible for reading the digital signals at a certain rate when a certain trigger condition is satisfied is implemented on an FPGA as shown in the Figure 5.1using VHDL.

Master_control, Enable Latch, Reset_trigger, Reset_counter, INC and Enable Latch are external control signals which are set and reset by the Microcontroller.

Clock is the internal clock Signal obtained by the clock generator and selector which generates multiple clock frequencies and selects one, which is then fed to the other acquisition circuit component for operation. While the clock generator and selector is itself driven by an external oscillator which in this case is a 50 MHz oscillator which is already available on the FPGA development board. Dp7-Dp0 is the probed signal from the buffer. D7-D0 is the 8 bit wide parallel bus interface with the microcontroller through which the sampled data for the single acquisition is transferred to the microcontroller. Complete signal is output signal, which is set when the acquisition process completes to inform the microcontroller to start reading the data from the Block RAM and transfer it to PC. The detailed operation of the entire acquisition process is explained below.

Initially the master_enable and Q complement of D flip flop are set(1) enabling the AND gate, which allows the clock pulses to pass through the AND gate as both the master_enable and Q complement are set to 1, the output of the AND gate will follow the clock signal. Enable Latch is set to High initially which enables the Latch to read the digital signal of 8 bit width on the positive edge of the clock which is driven by the output of the AND gate. The output of the latch is fed to the input of the block RAM and trigger circuit. The address counter is positive edge triggered and only counts when enable_counter is set to '1' and reset_counter is set to '0'. The data is written to the Block RAM at the address pointed by the address counter on the negative edge of the clock and when the write_enable is set to '1'. The D flip flop D input is driven by the output of the address comparator circuit, whose output will be set to high as the address counter reaches the end of the memory depth .

In order to start the acquisition process the master_control signal is set to high, such that the clock is passed through the AND gate to latch, address counter, Block RAM and the D flip flop. On the positive edge of the clock the latch reads and stores the digital data available on the probe, while on the negative edge the data read by the latch is written to the Block RAM on start address 0X0000. On the next clock pulse the new data from the latch is overwritten to the same address 0X0000 until the trigger signal is not generated which enables the counter. As the correct trigger condition is met, the trigger circuit sets the trigger signal to High which enables the counting operation of the address counter and subsequent samples are stored on the 0x0001, 0x0002, 0x0003 and so on address location in the Block RAM. This process only stops once the address comparator, which is simply an array of XNOR gate and AND gate, sets its output to HIGH such that the D flip flop input will be high which causes the Q compliment output to be reset 0 on the negative edge of the clock. This effectively disables the AND gate, as if any input is set to 0 the output is 0 we can see this from the truth table, such that no further clock is passed through hence stopping the sampling process. As the Q complement is 1 the Q output of the D flip flop is 1, hence the complete signal is set to 1.

This complete signal is continuously polled by the microcontroller which is used to indicate completion of the sampling process.

Now the microcontroller resets the master control, write enable and latch enable signal. The address counter is reset by the microcontroller such that the address pointed will be reset to 0X0000. Now to read the sampled data from block RAM though the D7-D0 parallel bus the microcontroller will provide INC signal pulse, for entire memory depth such that if the selected memory depth is 4 KB microcontroller will provide INC pulse for 4096 time , until the entire sampled data is read. The thing to remember is that since the write_enable is reset to LOW no new data can be written on the block RAM, hence the read data is actually the sampled data itself. The INC signal is ORed with the clock signal so that only the address counter, Block RAM and D flop flip are driven by the INC pulse.

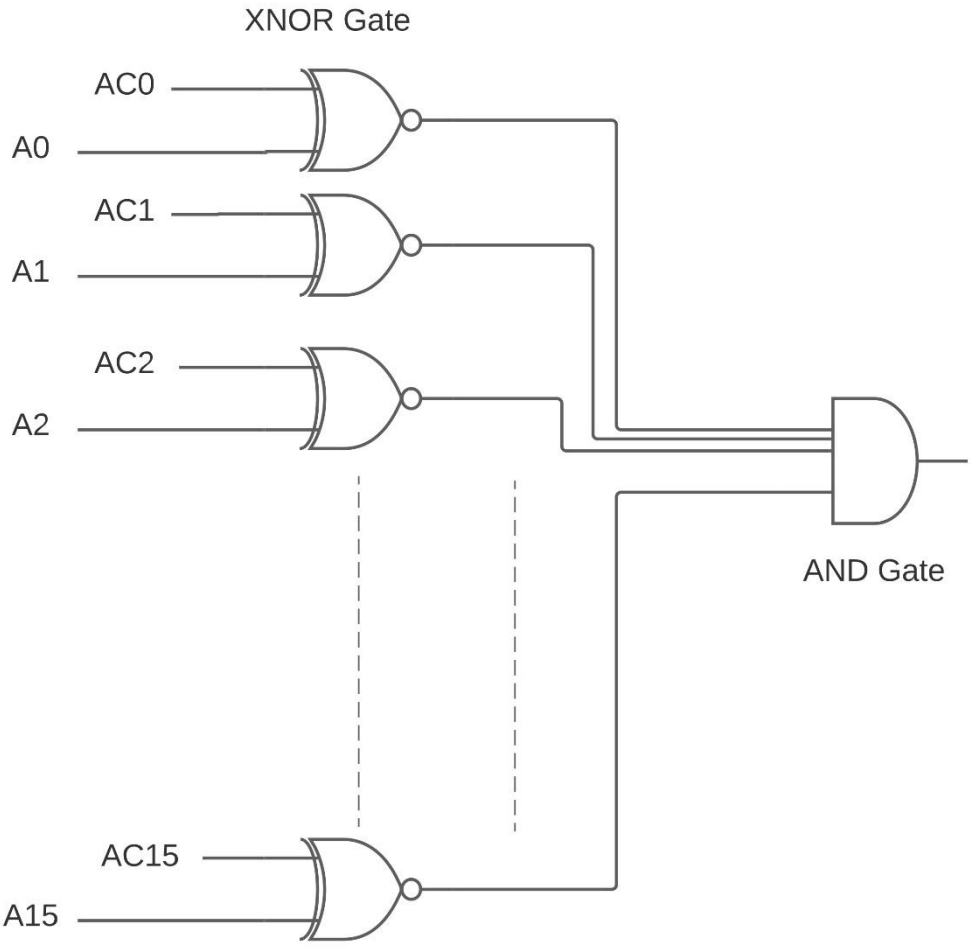


Figure 5.2: Circuit Diagram of Address comparator

The address comparator shown in the Figure 5.2 is used to identify the end of the memory depth and stop the acquisition process. Our proposed logic analyzer provides the following selectable sample depth which corresponds to the following address.

Table 5.1: Sample depth and its corresponding address

S.N.	Sample depth(KB)	Address
1	1	0b0000001111111111
2	2	0b0000011111111111
3	4	0b0000111111111111
4	8	0b0001111111111111
5	16	0b0011111111111111
6	32	0b0111111111111111
7	64	0b1111111111111111

Once a proper sample depth is selected by the user in the logic analyzer software, the microcontroller sets the address comparator register to appropriate address values so that once the address counter reaches that specific address the output of the address comparator becomes HIGH. Which in turn resets the Q compliment output of D flip flop on the subsequent negative edge of the clock and stops the acquisition process.
The XNOR gate only produces output HIGH when both the inputs have the same value. Hence by driving the array of XNOR gate with the output of the address comparator register we can compare the address counter output with it, once it matches i.e. the memory is filled for the required memory depth address comparator output is set to HIGH.

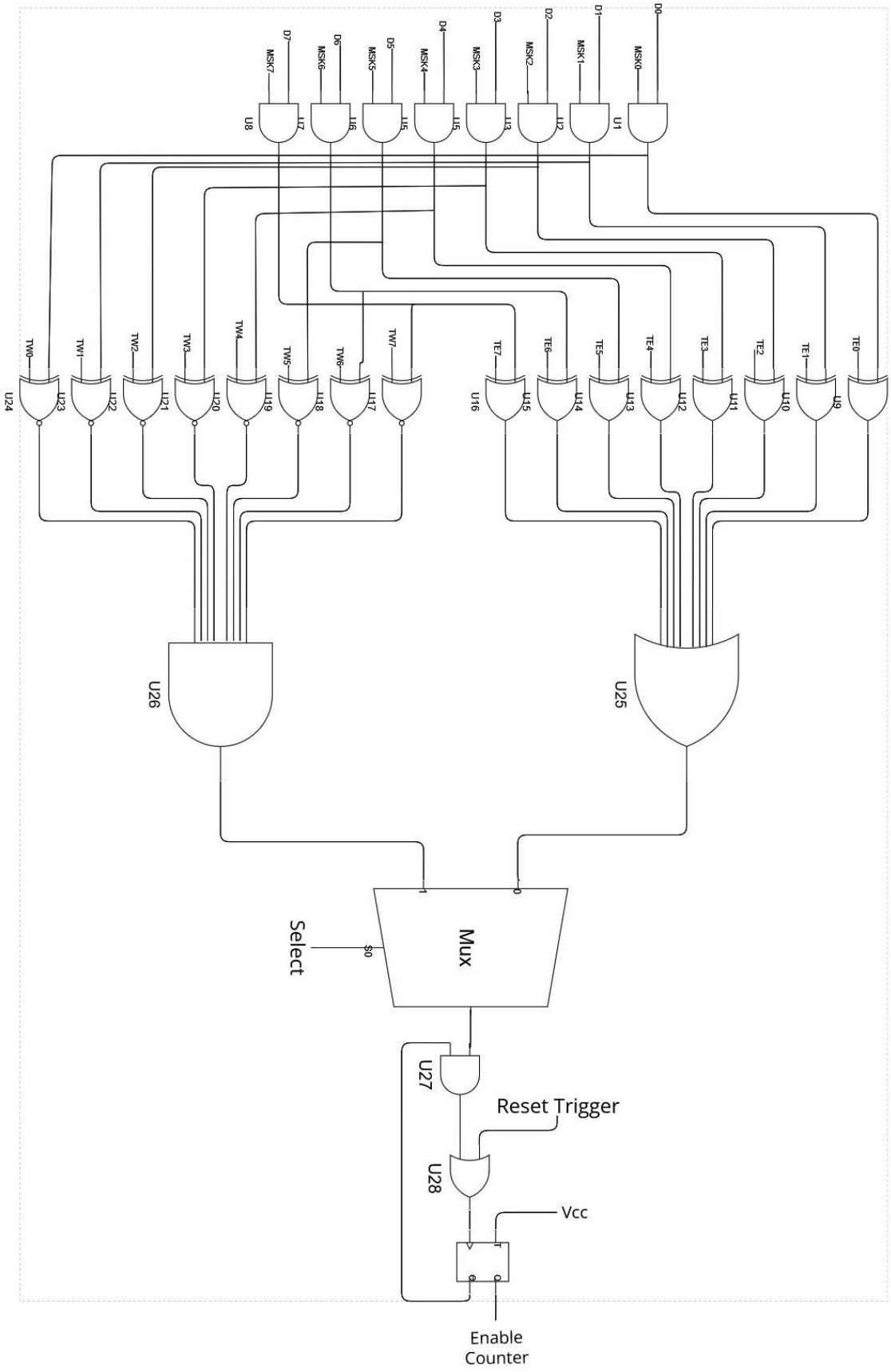


Figure 5.3: Trigger Circuit

The trigger circuit is shown in Figure 5.3. The U1 to U8 are AND gates which are used to MASK the signal. XOR gates from U9 TO U16 are used to select either positive trigger or negative edge trigger based on the input TE0-TE7.
For a positive edge trigger one of the inputs of the XOR gate is reset to LOW, hence according to the truth table the rising edge of a signal passes as it is causing the trigger circuit to generate the trigger signal to enable the counter. For a negative edge trigger one of the inputs of the XOR gate is set to HIGH which causes the output to the inversion of the input, such that the negative edge on the sampled signal will be positive. This generates the trigger signal on the negative edge. The outputs from these gates are logically ORed by U25 and fed to 2X1 MUX.

XNOR gate from U17 TO U24 is used to trigger on a word. The input TW0-TW7 sets which word to trigger on. Since the output of the XNOR gate is only set to HIGH when the input value matches. Outputs from these XNOR are logically AND by U26 so that trigger is generated when only all the bit values are matched. The AND output fed to the other input of 2X1 trigger select MUX.

The truth table for the 2X1 trigger select MUX is shown in Table 5.2.

Table 5.2: Trigger type selector MUX truth table

Select	output
0	Edge trigger
1	Word trigger

If the user wants to start the acquisition process without any trigger condition, it can also be done. The microcontroller takes care of it by generating the trigger signal through the reset trigger control signal. The U27, U28 and T flip flop are just simple logic circuits to generate the trigger signal and prevent any undesirable behavior once the trigger signal is generated. This arrangement also allows the microcontroller to set the trigger signal to immediately start the acquisition process.

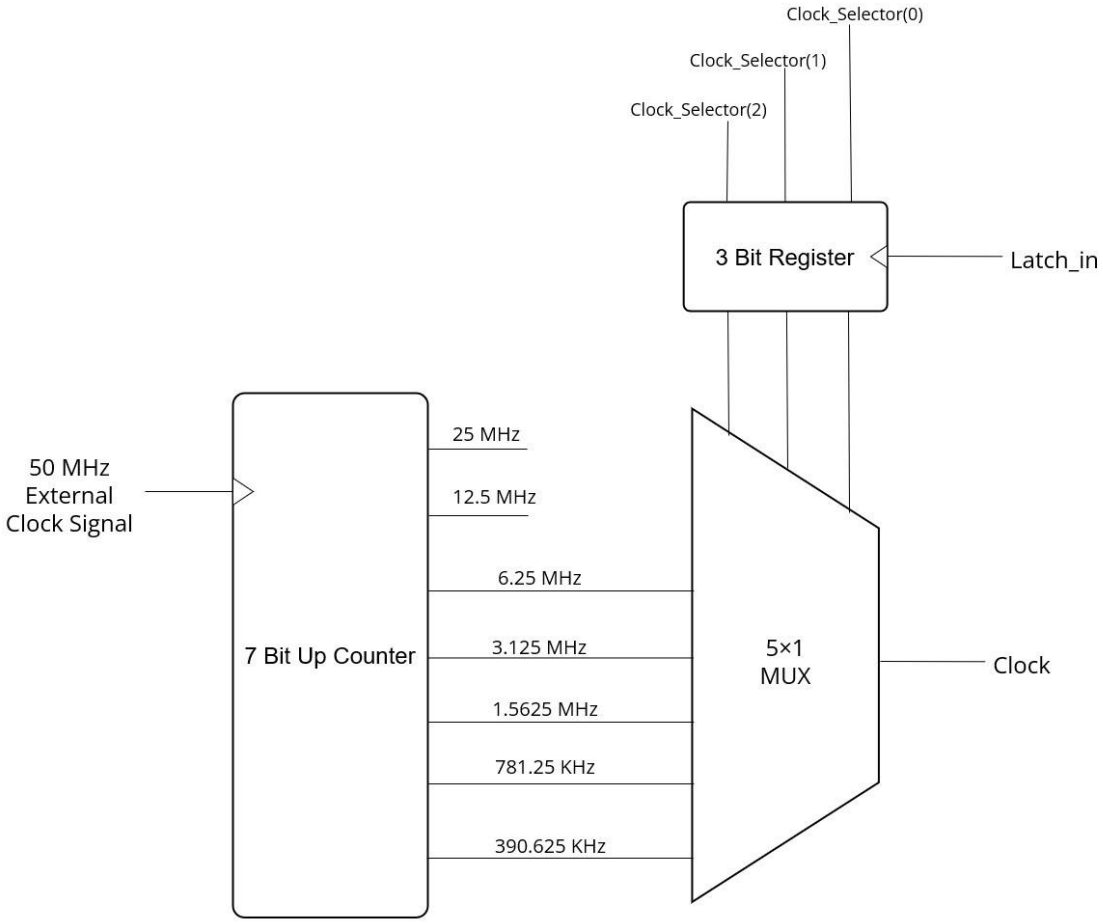


Figure 5.4: Circuit Diagram of Clock generator and selector

The Figure 5.4 is of the clock generator and selector circuit. It consists of a 7 bit up counter, 5X1 MUX and 3 bit register that is used to hold the selection bit value. The up counter works as the divide by 2 at each output, hence various sample rates are available as shown in the Figure 5.4. Depending upon the requirement the user can select the appropriate sample rate.

5.2 FLOWCHART

Flowchart of the microcontroller section of the PC based logic analyzer is shown above. As the microcontroller starts it initializes the default sample rate which is 1.6525 MHz and default sample depth which is 8 kilo Bytes.

Microcontroller then continuously waits for command from the Logic analyzer software running on the host PC through the UART peripheral interface. While waiting it continuously polls for any new data available on the UART receiver buffer. The polling of the command creates idle time during which the microcontroller could be doing other work but for this implementation and design, it doesn't need to perform other operations until it receives any kind of command from the Logic analyzer software running on the host PC. Hence due to this approach there is no need to worry about using the polling approach as it is quite reasonable.

Upon receiving a new command from the logic analyzer software, the command could be either to start the acquisition process or set some parameter within the acquisition circuit of the logic analyzer implemented within the FPGA.

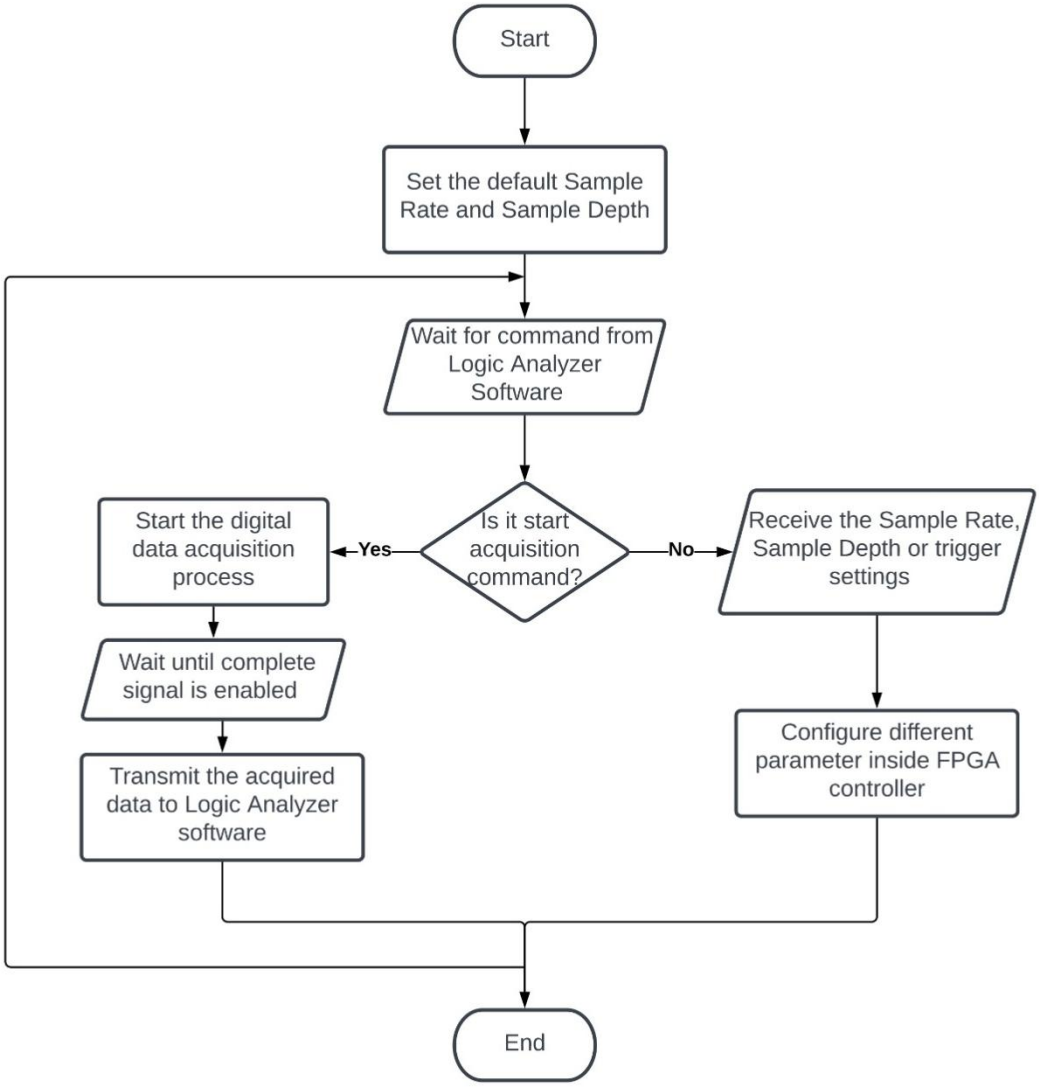


Figure 5.5: Flowchart of stm32 microcontroller Program

If the command is to start the acquisition, master controller, latch_enable, write_enable control signals are enabled which start the sampling process.

Then the microcontroller waits for the complete control signal to be set by the acquisition circuit by continually polling it.

Once the complete signal is set, the microcontroller reads all the sampled data from the Block RAM within the FPGA by using necessary control signals. Finally the sampled values are sent to the logic analyzer software and it is displayed and analyzed.

If the command is to set different parameters such as the sample rate, the microcontroller receives the parameter and sets it within the acquisition circuit using various control signals.

5.3 ALGORITHM

In order to decode the sampled data different algorithms are used depending on which protocol is to be decoded. Our developed PC based logic analyzer software can decode I2C and SPI communication protocol and 8 bit parallel bus data.

Following are the algorithms developed that are used in the logic analyzer software to decode these protocols.

5.3.1 Algorithm to Decode SPI Packet

Since SPI is a synchronous serial interface we use the clock signal to decode the packet.

Following are the steps to decode a SPI packet:

1. Find the point where the CS signal goes LOW.
2. Scan the SCK channel for the rising edge points.
3. Read the value of MISO and MOSI channel at these points for 8 consecutive rising edges to form 8 bit data words.
4. Convert the 8 bit word that represents data into HEX form and display it.
5. Repeat steps 2 and 3 until the CS signal goes HIGH.

5.3.2 Algorithm to Decode I2C Packet

Since I2C is a synchronous serial interface we use the clock signal to decode the packet.

Following are the steps to decode an I2C packet:

1. Find the start condition i.e. when the SDA line goes LOW followed by SCL line going LOW.
2. Scan the SCL channel for rising edge points.
3. Read the value of SDA line at these points for 7 consecutive rising edges for 7 bit address followed by 1 bit READ or WRITE bit if it's an address packer or read the value of SDA line at these points for 8 consecutive rising edges for data packet.
4. Obtain HEX equivalent value and display it.
5. Read the SDA line at the 9th rising edge to obtain either ACK or NACK bit.
6. Display the entire packet.
7. Repeat step 2 to 6 until stop condition is identified i.e. SCL line goes HIGH followed by SDA line going HIGH.

6. SIMULATION & OBSERVATION

Following are the screenshots of the simulation obtained from ISE simulator, obtained after writing VHDL implementation of the acquisition circuit.

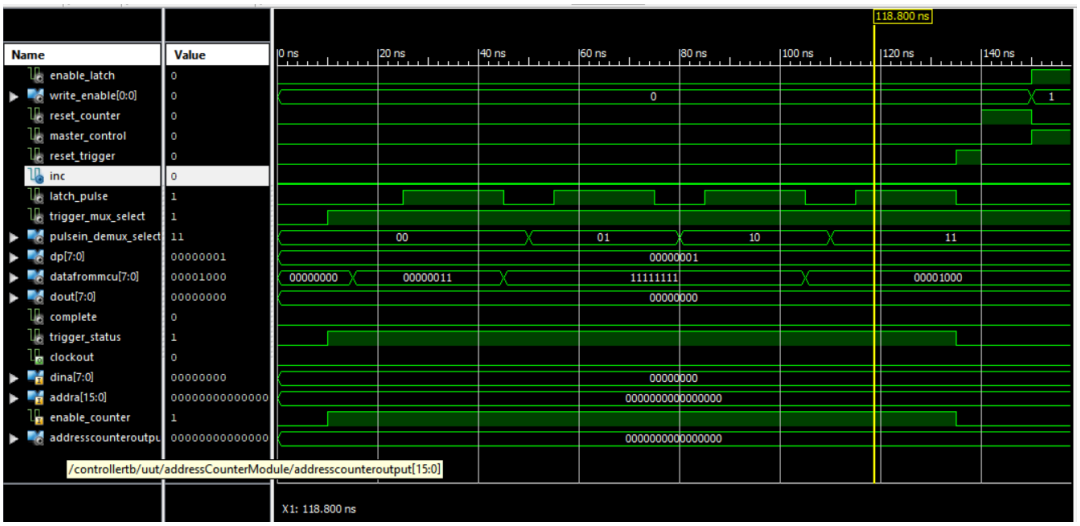
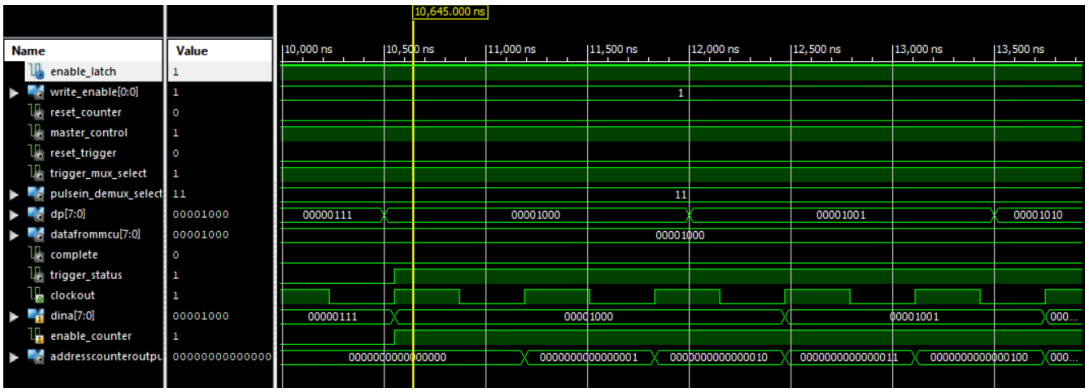


Figure 6.1: Simulation showing setting default parameter.



igure 6.2: Simulation showing start of acquisition process.

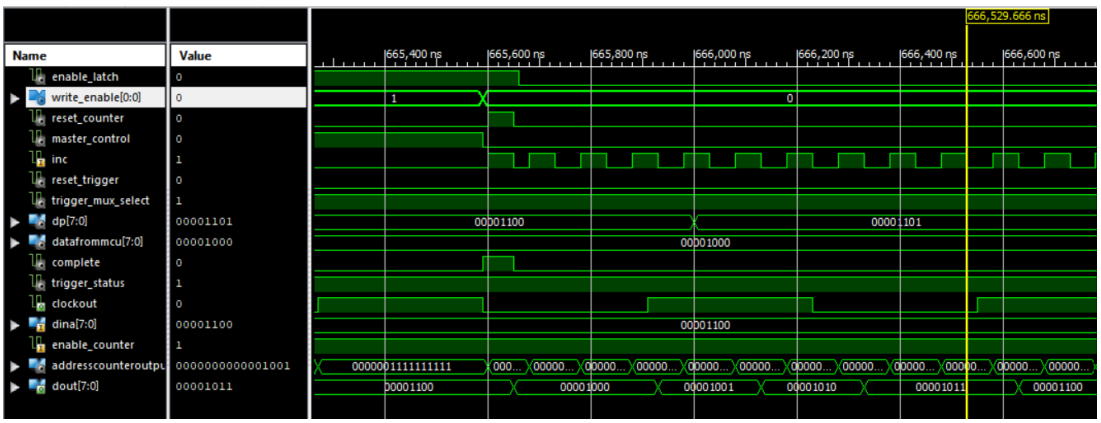


Figure 6.3: Simulation showing the end of the acquisition process.

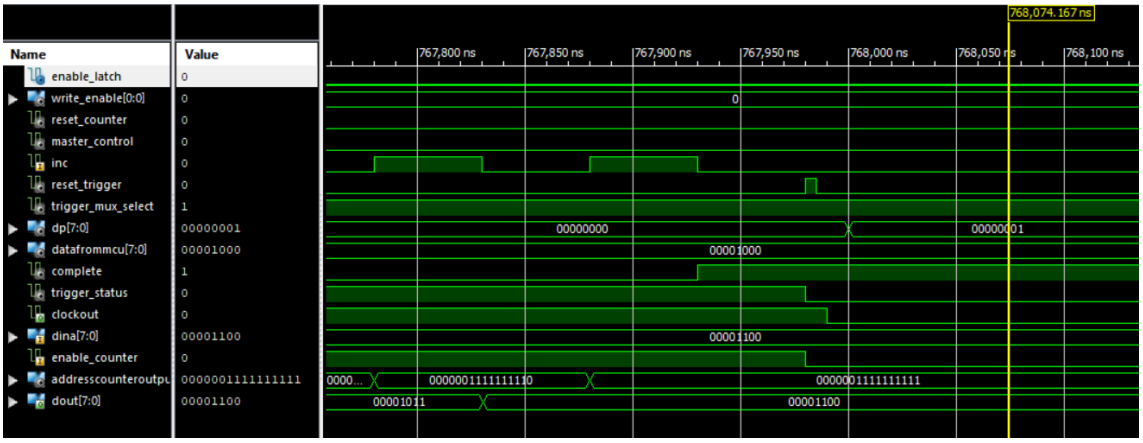


Figure 6.4: Simulation showing the end of the sampled data reading process.

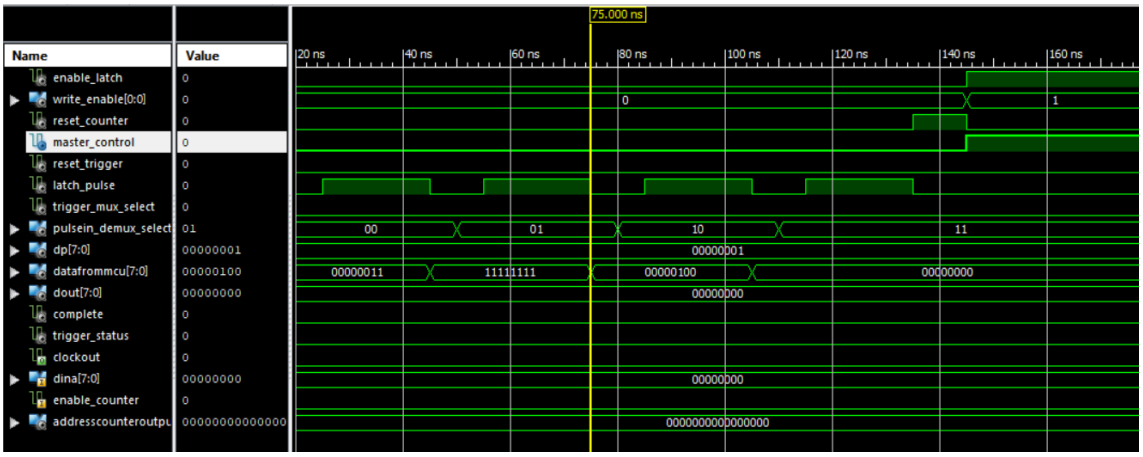


Figure 6.5: Simulation showing setting rising trigger on channel 3.

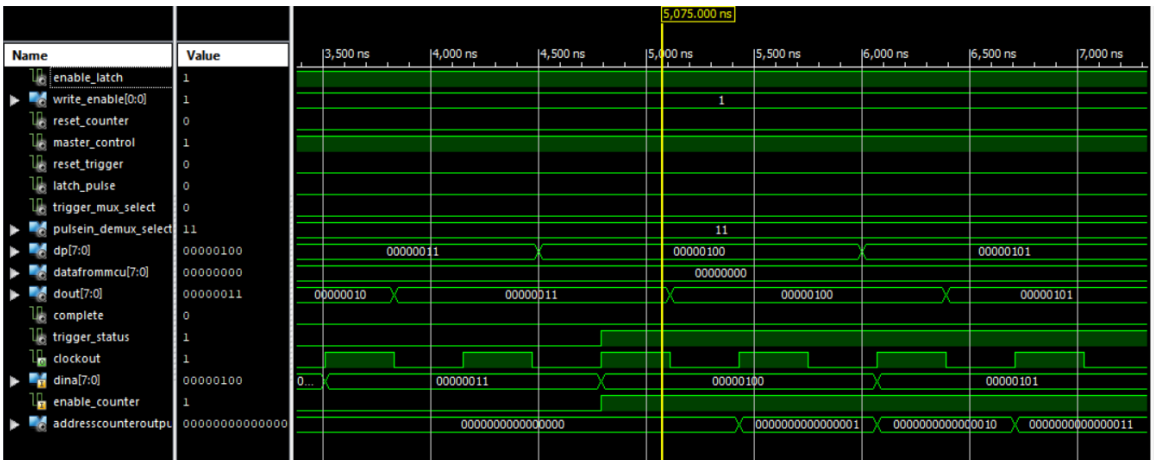


Figure 6.6: Simulation showing trigger signal generated.

From the above simulation screenshot we verified that the acquisition circuit design implemented using the FPGA using VHDL works as intended.

7. RESULT &DISCUSSION

The developed Logic analyzer prototype operates as designed. The acquisition circuit implemented in the FPGA using VHDL reads the digital signal from input channels and stores them in the Block RAM. Once the acquisition completes the microcontroller reads the entire data from the block RAM and transfers it to the PC using serial interface. The Logic analyzer software running on the host PC displays the sampled data as in a timing diagram and performs protocol decoding operation as required.

Photos and screenshot below shows the logic analyzer hardware and software prototype.

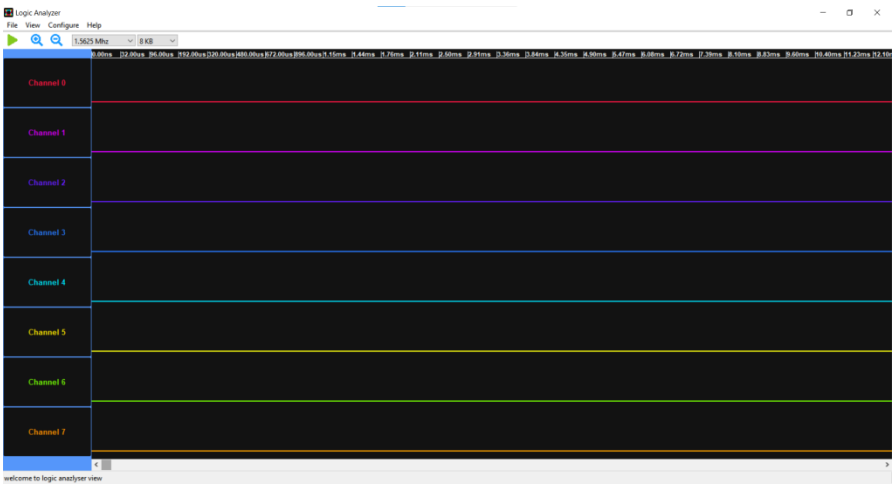


Figure 7.1: Screenshot of logic analyzer software.

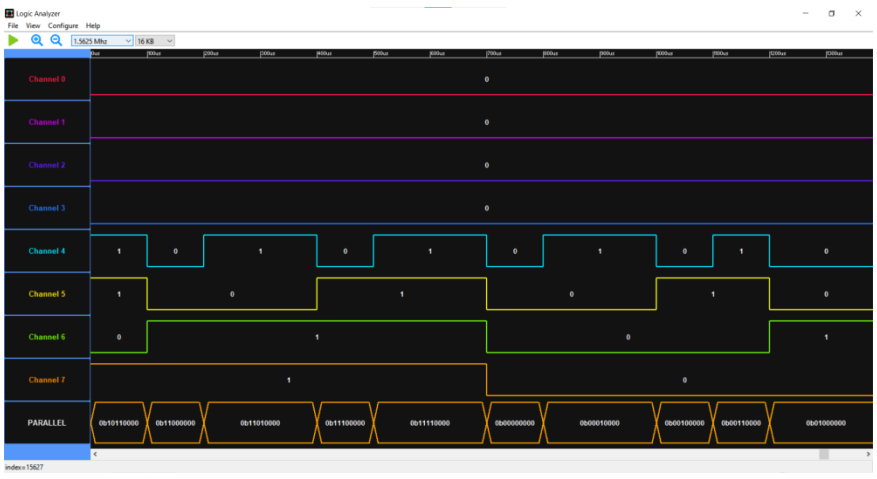


Figure 7.2: Logic analyzer software displaying parallel decoded data

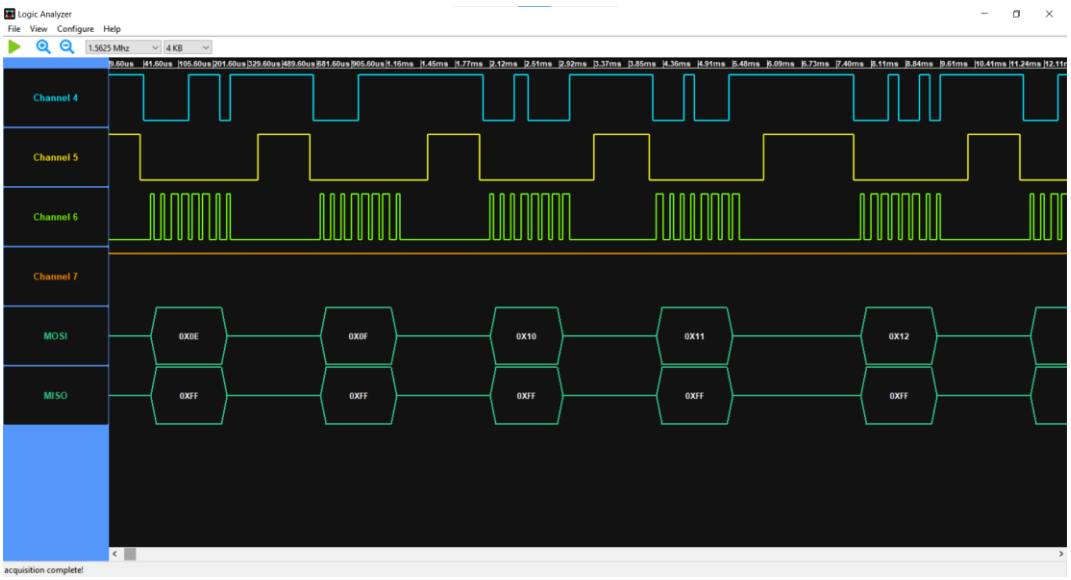


Figure 7.3: logic analyzer software displaying decoded SPI packet

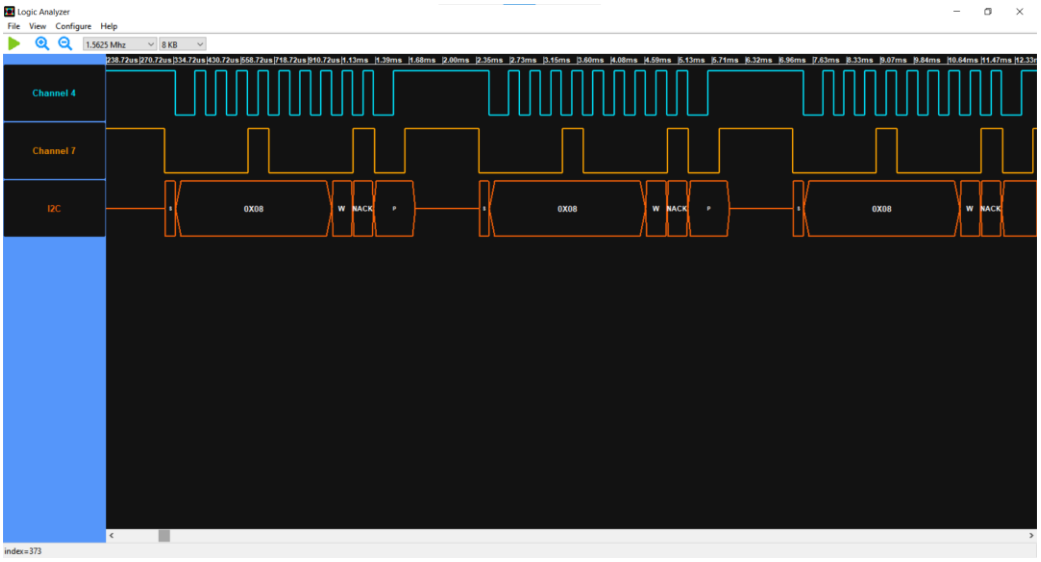
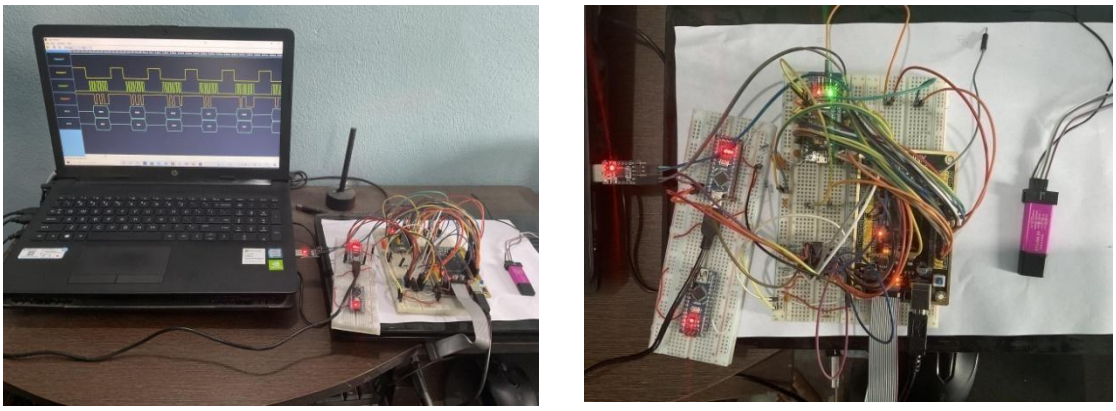


Figure 7.4: Logic analyzer software displaying decoded sampled I2C packet



(A) (B)

Figure 7.5: (A) and (B) PC based Logic Analyzer test setup

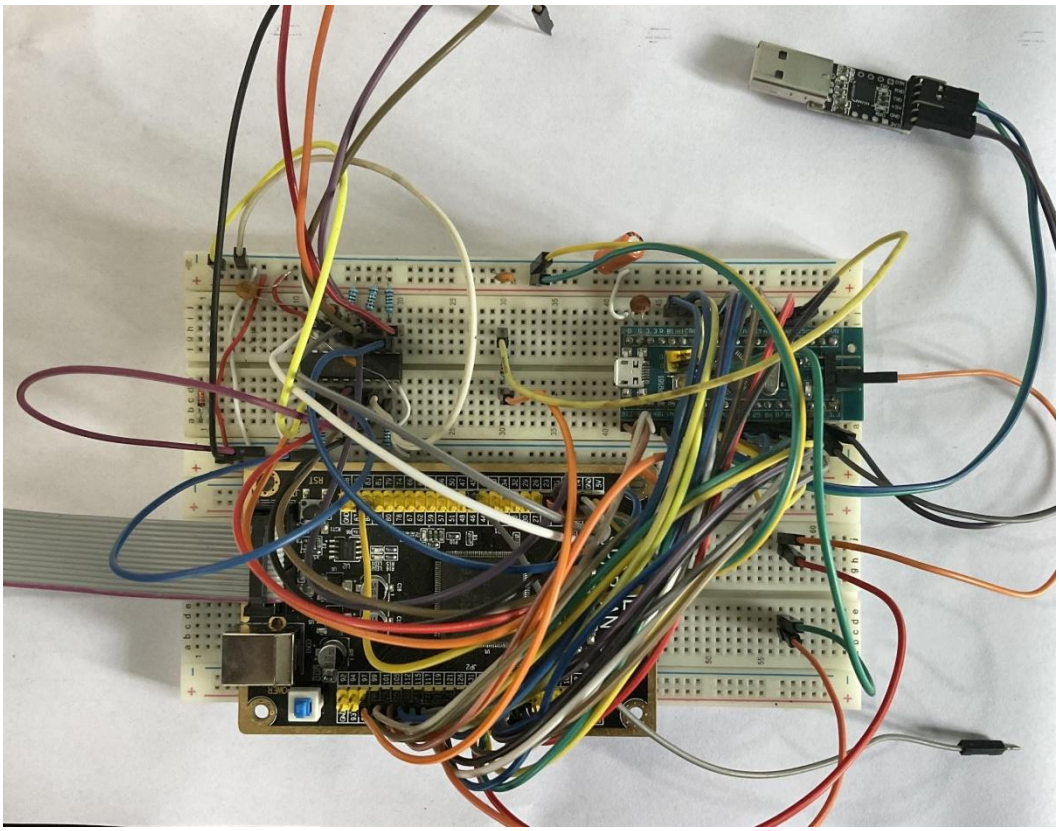


Figure 7.6: PC based Logic Analyzer hardware prototype Top View

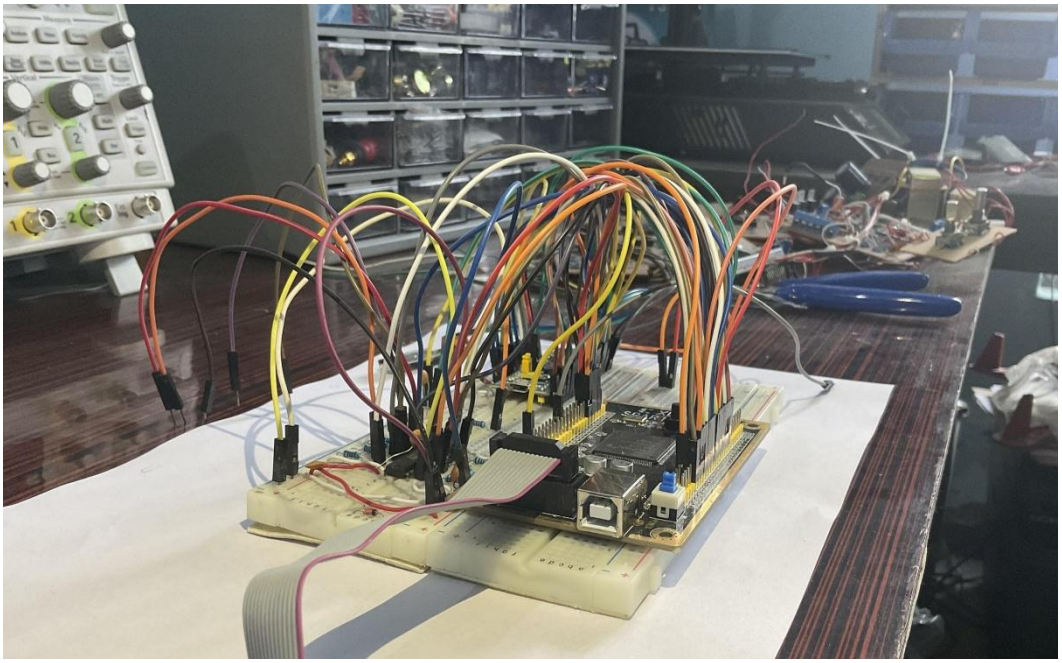


Figure 7.7: PC based Logic analyzer size view

8. CONCLUSION

Hence from the result, we can observe that we have successfully designed and developed PC based logic analyzer that can display waveform diagram from each channel to show their timing relationship and also provides feature to perform protocol decode such as I2C, SPI and parallel data.

9. APPLICATION & LIMITATIONS

9.1 APPLICATIONS

- Uncover hardware defects that are not found in simulation.
- Debug and verify digital system operation.
- Trace and correlate many digital signals simultaneously.
- Protocol Decode.
- Timing Diagram visualization.

9.2 LIMITATIONS

- Breadboard construction leads to uncertain behavior sometimes.
- Parasitic capacitance and inductance introduced due the breadboard and jumper wire, which limits signal with sharp edges.
- Limited acquisition speed in comparison to commercial logic analyzer.
- Limited memory depth in comparison to high end standalone logic analyzer.
- The logic analyzer software can only decode I2C, SPI protocol and data from a parallel bus. Other various protocol decoding, such as UART, functions are lacking and need to be added.
- The developed prototype of logic analyzer software and firmware still has bugs which can only be identified by its extensive use.
- Probing conditions should be taken into consideration such that it does not actually load the actual digital circuit under test.

10. FUTURE SCOPE

The developed logic analyzer prototype is useful for debugging digital circuits, especially various cheaply available microcontrollers which have various communication peripherals. It is very useful to obtain a simple view of a waveform diagram and obtain timing relations between them, decoding of some widely used protocol and digital circuit debugging. The limitation can be reduced by developing a dedicated single PCB for the hardware portion. By utilizing a better high speed buffer we can increase the frequency of input signal that is to be sampled. The speed of data transfer can be further improved by using USB to transfer data instead of Serial communication. The logic analyzer software portion can be further improved by adding different protocol decoding functions and debugging bugs encountered in the duration of its operation.

11. REFERENCES

[1] Keysight, "Feeling Comfortable with Logic Analyzers" (PDF). *keysight.com*. Agilent Technologies, Inc. Retrieved 28 November 2012.

[2] Alexey M. Romanov, "An easy to implement logic analyzer for long-term precise measurements".HardwareX, Volume 9, e00164, ISSN 2468-0672, 2021 https://doi.org/10.1016/j.ohx.2020.e00164.

[3] Bhavanam Vasujadevi and Dr Midasala, "Implementation of FPGA based LOW-COST Logic Signal Analyzer (LSA)". International Journal of Recent Trends in Engineering and Technology, 2010

[4] Varsha Karambelkar and A.A. Shinde, "Digital Signals by Low Cost ARM Based Logic Analyzer". International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-1, Issue-5, 2012

[5] STMicroelectronics, "STM32F103C6", Datasheet, 2015

[6] STMicroelectronics, "ST-LINK/V2", Datasheet, 2020

[7] Mouser, "Spartan-6 LXT fpgas". Mouser, 2019.Accessed: May 9, 2022, from https://www.mouser.com/new/xilinx/xilinx-spartan-6-lxt-fpgas/

[8] Xilinx, "Platform Cable USB", Datasheet, 2014

[9] Sunrom Electronics "CP2102 - USB-TTL UART module". Sunrom Electronics. (n.d.). Retrieved May 9, 2022, from https://www.sunrom.com/p/cp2102-usb-ttl-uart-module

[10] Texas Instruments, "SN54F244, SN74F244 Octal buffers/drivers with 3-state outputs". Texas Instruments, 2007

[11] J-Martens. "Visual Studio Documentation." *Microsoft Docs*, 2022 docs.microsoft.com/en-us/visualstudio/windows/?view=vs-2022.

[12] STMicroelectronics, "STM32CubeIDE". STMicroelectronics. (n.d.). Retrieved May 17, 2022, from https://www.st.com/en/development-tools/stm32cubeide.html

[13] Xilinx, "Ise Design Suite". Xilinx. (n.d.). Retrieved: May 17, 2022, from https://www.xilinx.com/products/design-tools/ise-design-suite.html

[14] Xilinx, “ISE In-Depth Tutorial”. Xilinx. (n.d.). Retrieved: May 25, 2022, from https://china.xilinx.com/content/dam/xilinx/support/documents/sw_manuals/xilinx13_1/ise_tutorial_ug695.pdf

[15] C++ introduction. (n.d.). Accessed: May 17, 2022, fromhttps://www.w3schools.com/cpp/cpp_intro.asp

[16] Rip Tutorial, “Win32 API tutorial => getting started with win32 API” (n.d.). Accessed: May 17, 2022, from https://riptutorial.com/winapi

[17] Team, wxW. (n.d.). “wxWidgets Overview”. wxWidgets. Retrieved May 17, 2022, from https://www.wxwidgets.org/about/

[18] Xilinx, “Spartan-6 FPGA Block RAM Resources”. Xilinx, 2011Accessed: December 18, 2021, from https://www.xilinx.com/support/documentation/user_guides/ug383.pdf

12. APPENDIX

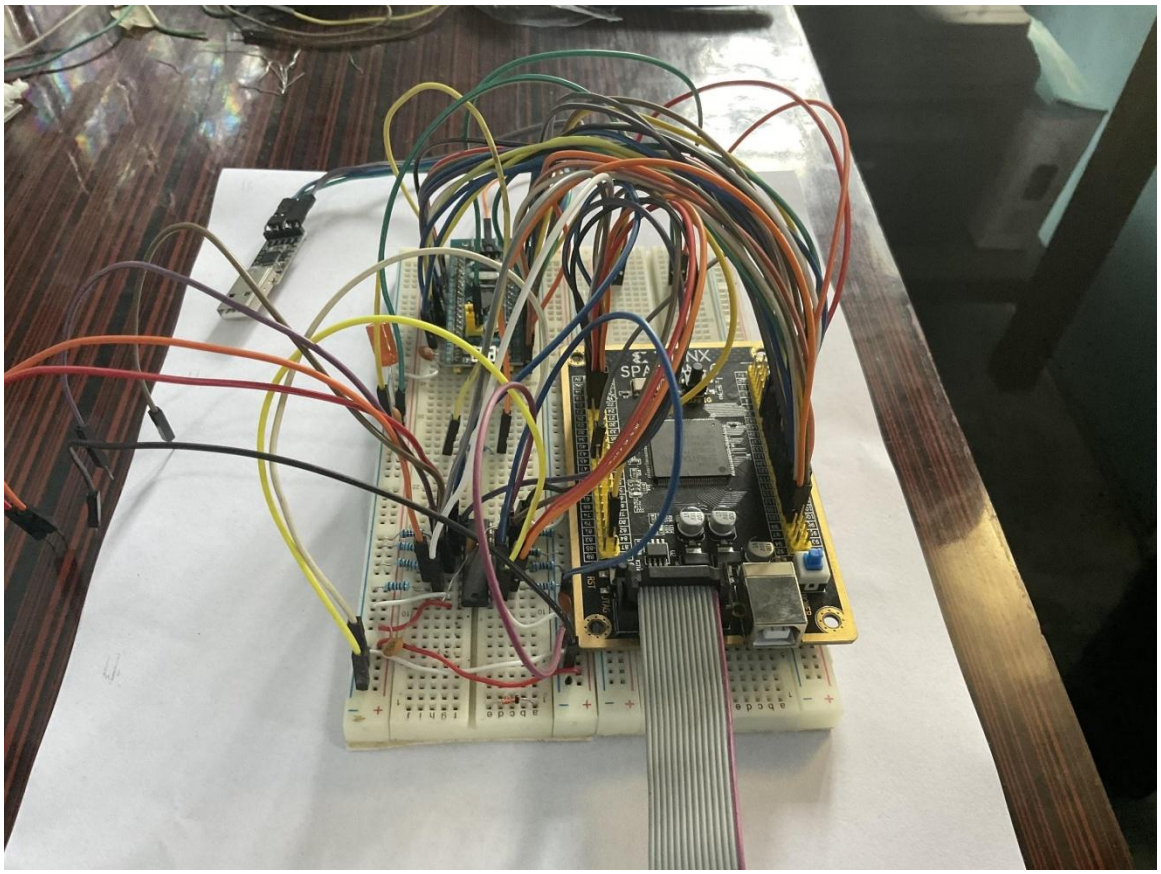


Figure 12.1: PC based logic analyzer Hardware prototype

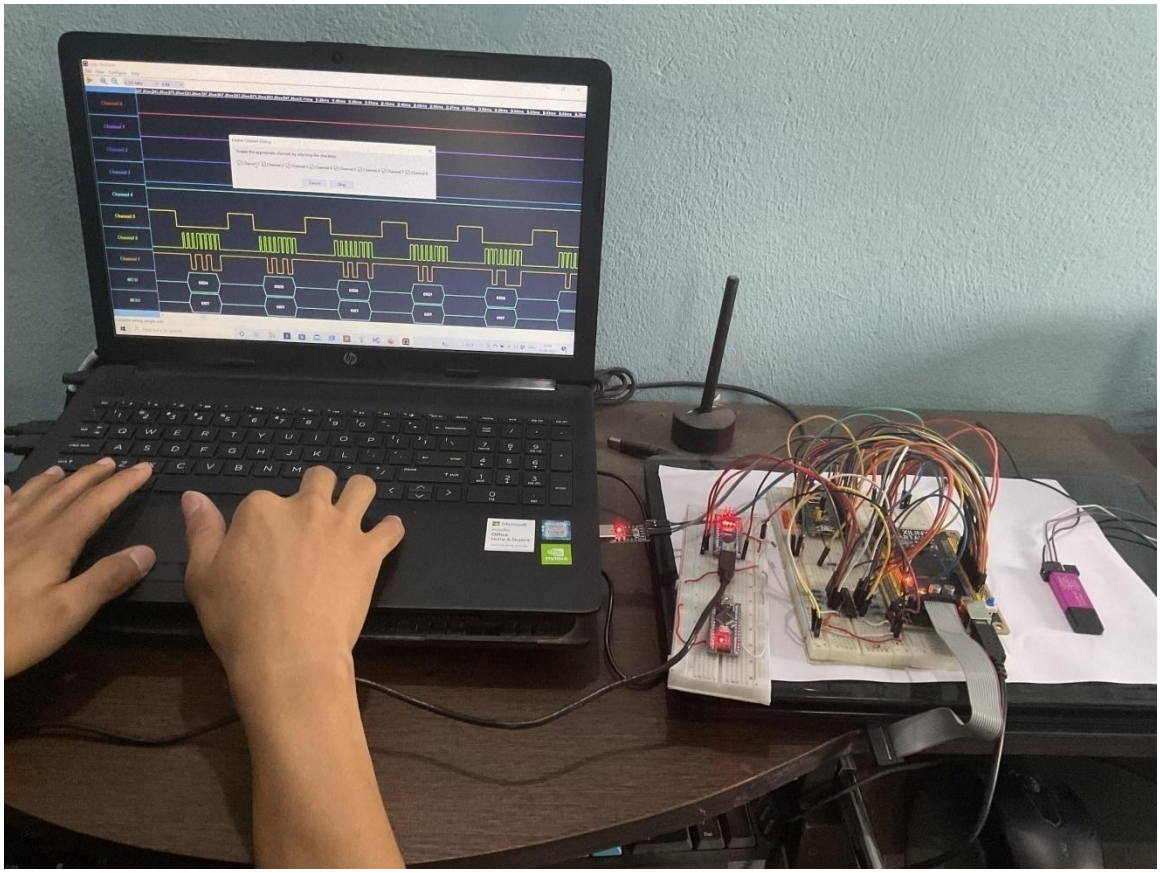


Figure 12.2: PC based logic analyzer prototype test setup

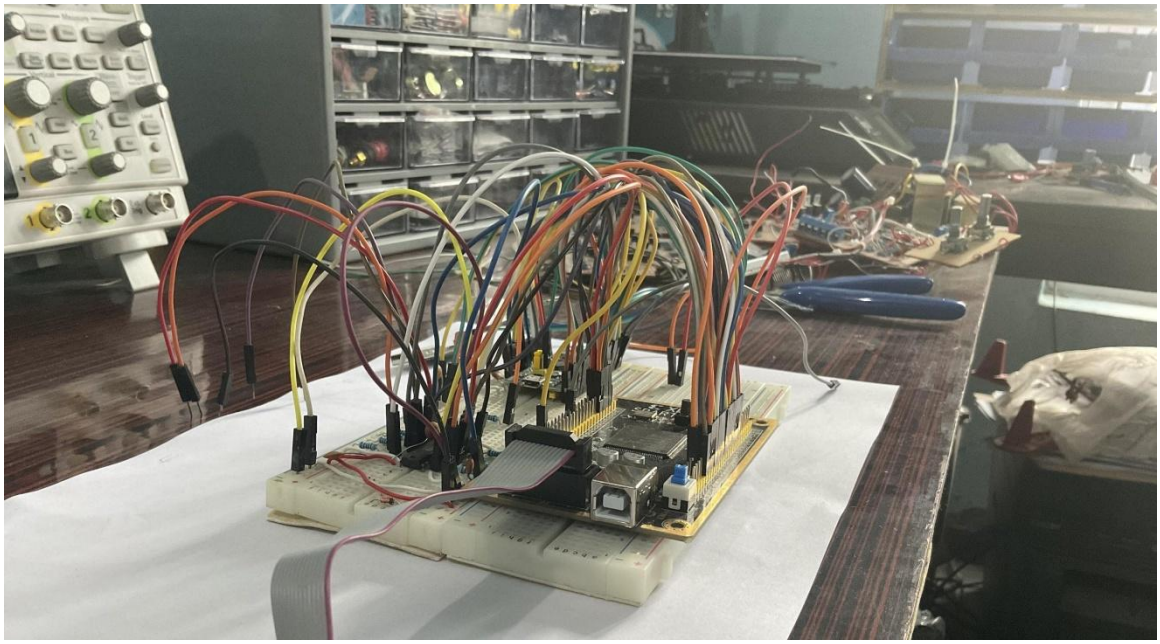
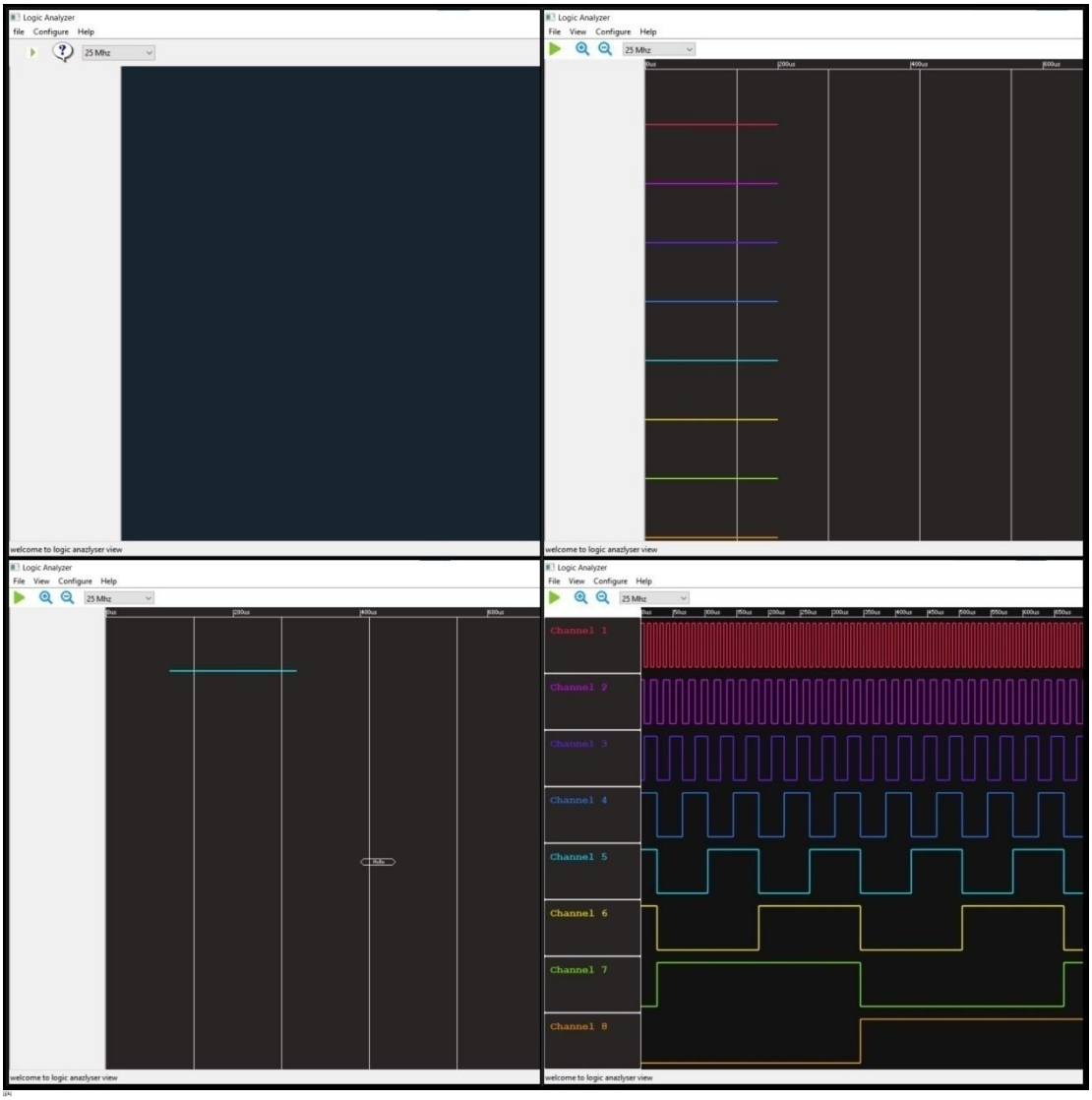
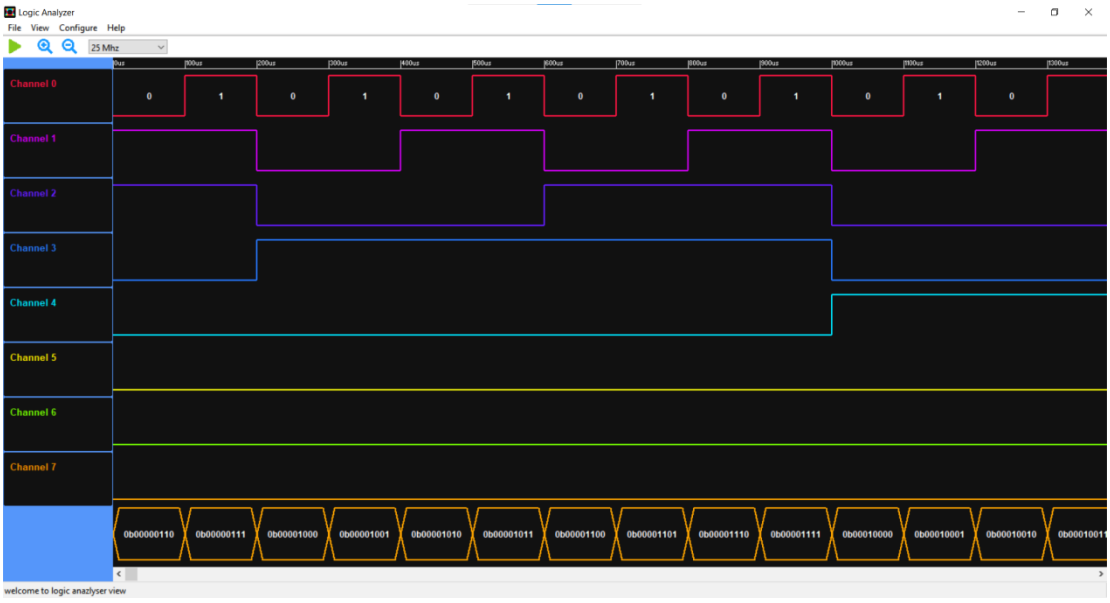


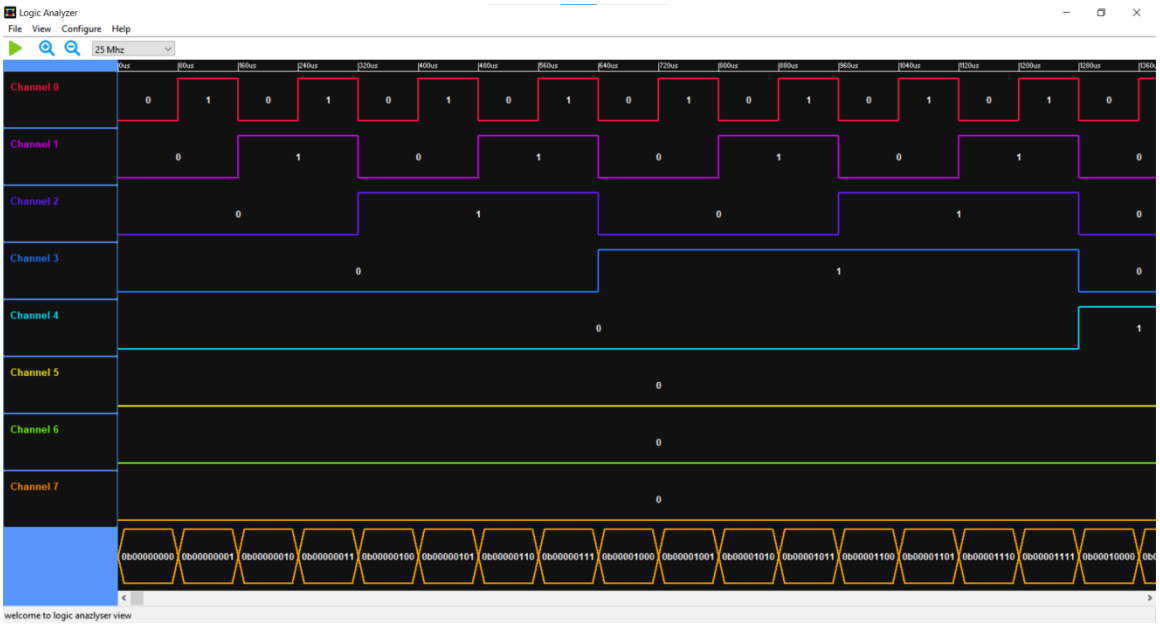
Figure 12.3: PC based Logic analyzer hardware prototype alternate view


Logic Analyzer
File Configure Help
25 Mhz
welcome to logic anazlyser view
Channel 1
Channel 2
Channel 3
Channel 4
Channel 5
Channel 6
Channel 7
Channel 8

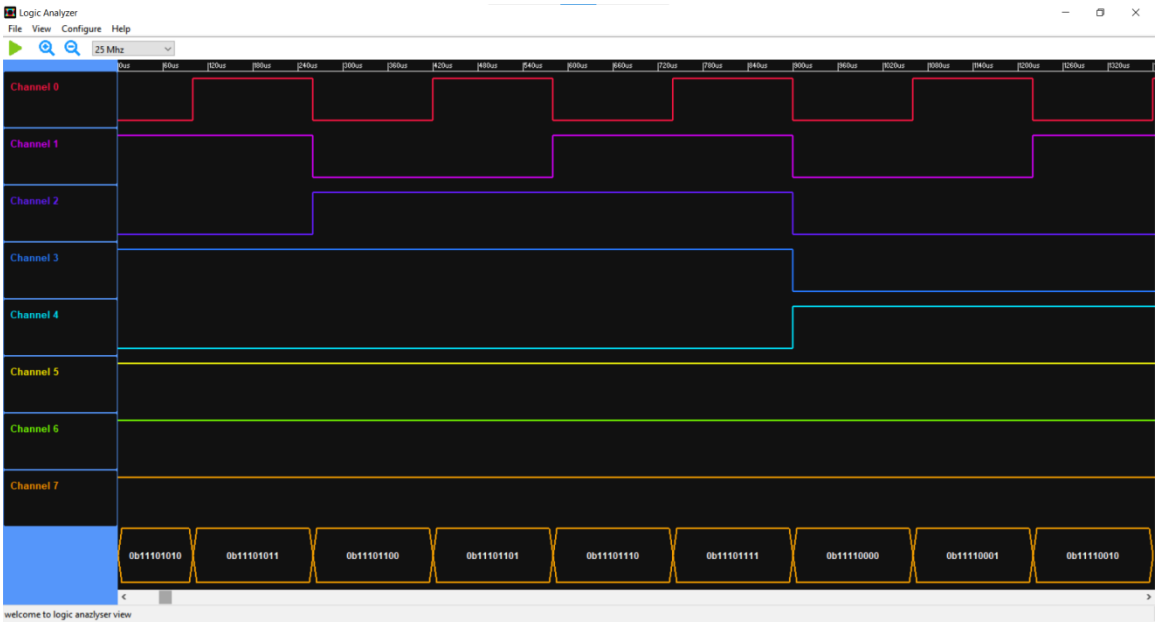
(a)


Logic Analyzer
File View Configure Help
25 Mhz
Channel 0
Channel 1
Channel 2
Channel 3
Channel 4
Channel 5
Channel 6
Channel 7
0b00000110
0b00000111
0b00001000
0b00001001
0b00001010
0b00001011
0b00001100
0b00001101
0b00001110
0b00001111
0b00010000
0b00010001
0b00010010
welcome to logic anazlyser view

(b)



(c)



(d)

Figure 12.4:(a), (b),(c) and (d) Stages of software development versions

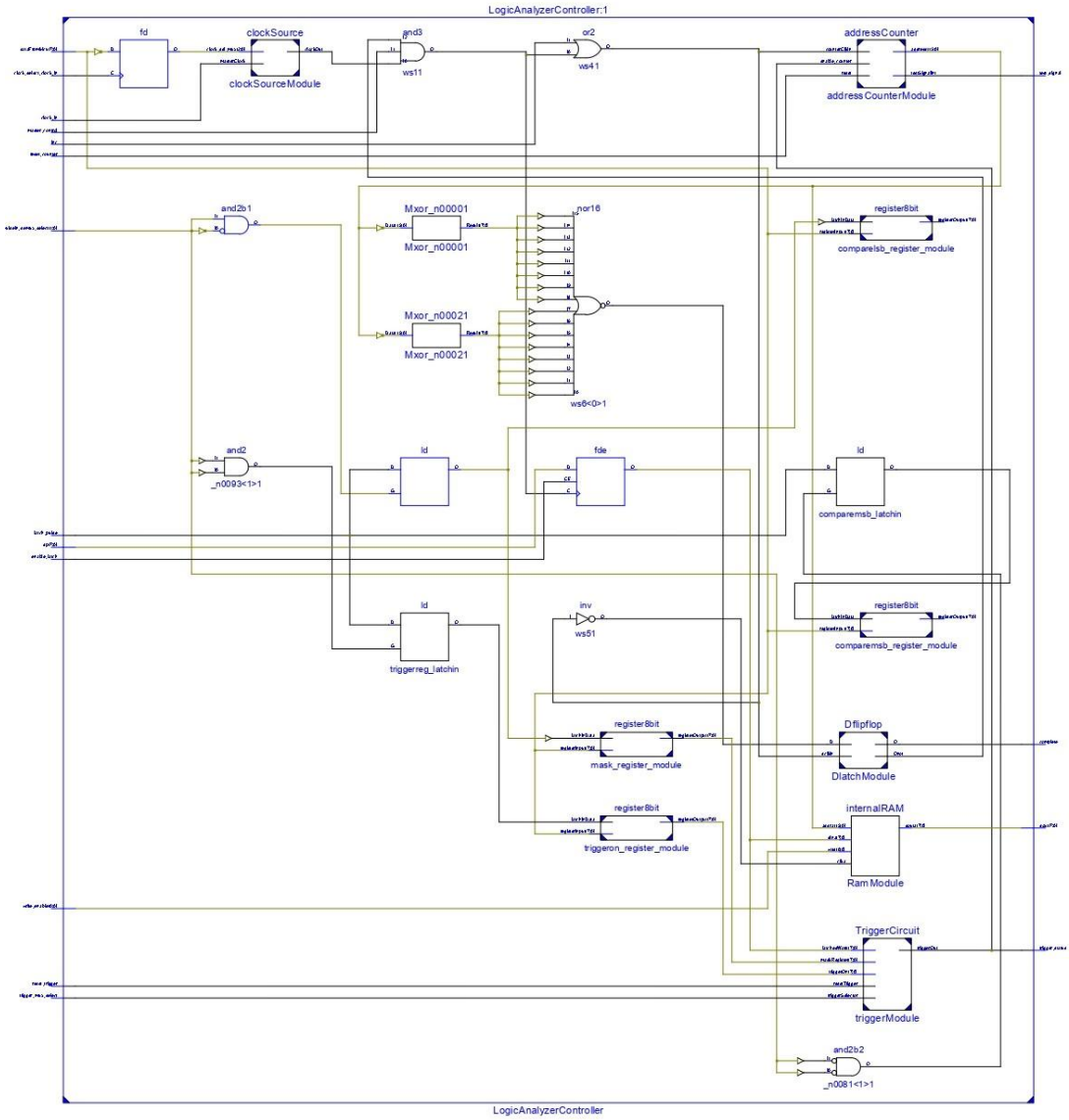


Figure 12.5: RTL view of acquisition circuit implemented inside FPGA

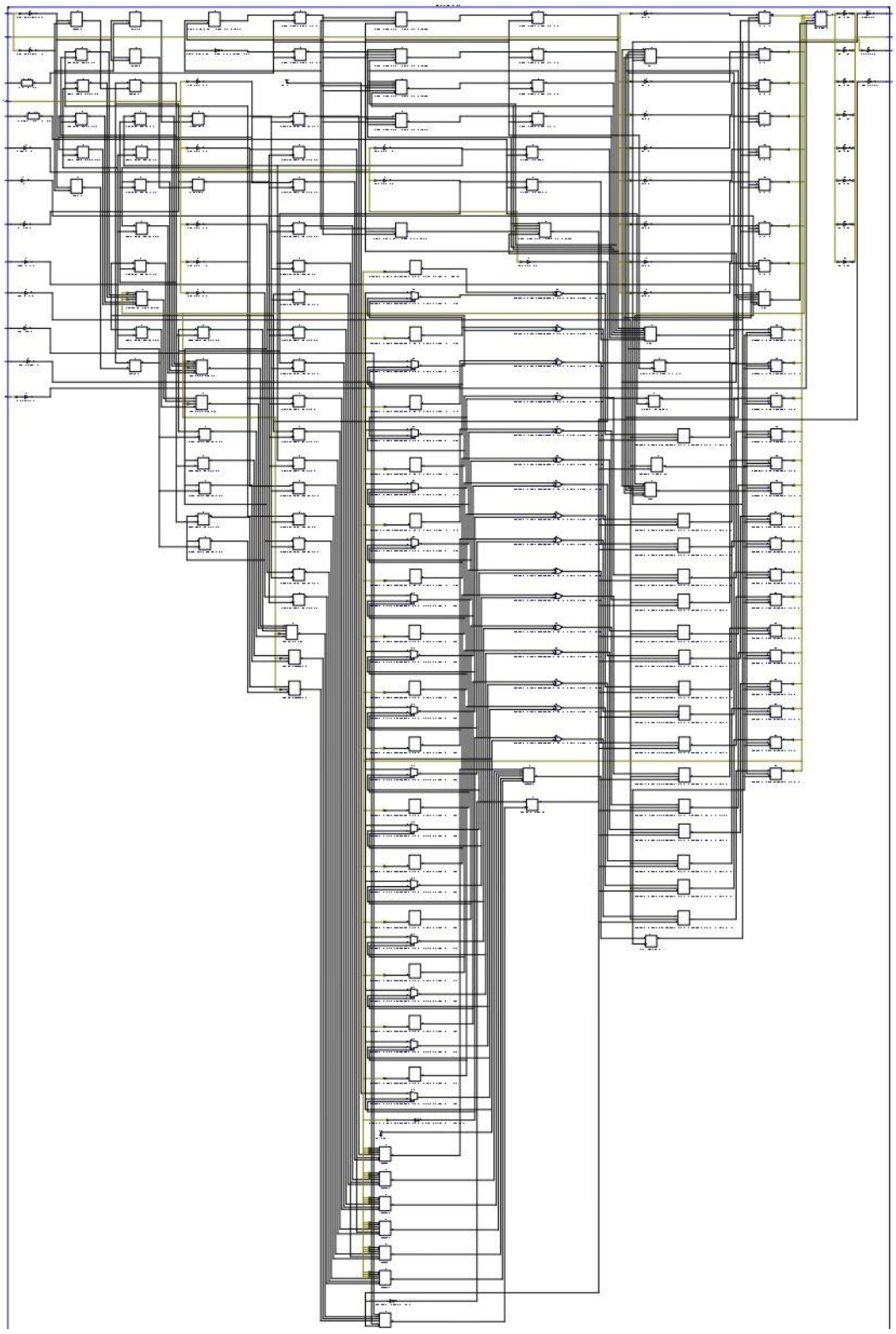


Figure 12.6: Technology View of Acquisition circuit implemented inside FPGA