"HIGH SPEED CAMERA INTERFACE VALIDATION FOR LINK LAYER"

A Thesis Submitted in partial fulfillment of the requirement for the award of the degree

Master of Engineering *in*
Electronics &Communication Engineering

Submitted by

Simran Kashyap

RollNo.802261003

Faculty Mentor
Dr. Jaswinder Kaur
Asst.Professor
ECED,TIET,
Patiala

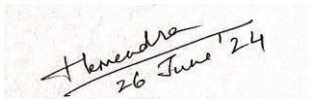
ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT
TIET, PATIALA-147004, PUNJAB
INDIA
July 2024

DECLARATION

I hereby declare that the project work entitled is " **High Speed Camera Interface validation for link layer**" in partial fulfillment of the requirement of the award of the degree **of Master of Engineering(ECE)** submitted at **Electronics and Communication Engineering Department**, Thapar institute of Engineering & Technology(Deemed to be university),Patiala is a record of work carried out under supervision of **Dr. Jaswinder Kaur**(Assistant Professor, Electronics and Communication Engineering Department, Thapar Institute of Engineering & Technology(Deemed to be university) from **JUNE 2023 TO JUNE 2024**. The Matter in this has not been submitted in part to any other university or institute for the award of any other degree.

Date:26.06.2024

Certified that the above statement made by the student is correct to the best of our knowledge and belief.



Dr.Jaswinder Kaur	Mr.Hemendra Singh
Asst.Professor-	Senior Staff Engineer
ECED,TIET,	STMicroelectronics
Patiala	Greater Noida

ACKNOWLEDGEMENT

I would like to express my gratitude to my college mentor, **Dr. Jaswinder Kaur** for her continuous support and guidance throughout my internship. His advice, wisdom, and encouragement have been invaluable in helping me navigate through the challenges and make the most out of this experience. I am thankful for his unwavering support and for being a constant source of motivation. Her guidance, support, and encouragement have been invaluable, and I am truly grateful for her contributions.

I am deeply grateful to **Dr. Kulbir Singh**, Head of the Department(ECE) and **Dr. Amit Mishra**, M.E(ECE) Coordinator, Thapar Institute of Engineering.& Technology, Patiala for providing me with a learning Environment and Infrastructure in ECED.

I Will be ignorant if I do not express my gratitude to the author of the references and other literature cited in this seminar.

Finally, I want to thank all of my colleagues for their encouragement through potential discussions and suggestions.

Simran Kashyap
802261003
M.E(ECE)

ABSTRACT

The major project undertaken was regarding the **HIGH SPEED CAMERA INTERFACE VALIDATION FOR LINK LAYER.** Crucial in modern digital imaging, converting optical images into electronic signals. This Projects highlights recent advancements in image sensor technology, focusing on Charge- Coupled Device (CCD) and Complementary Metal-Oxide-Semiconductor (CMOS) sensors. Continuous evolution in sensor technology promises further improvements in image quality, power efficiency, and functionality, driving innovation across various industries. Video timing involves the precise control of the sequence and duration of signals that govern the working of image sensors. It ensures that each pixel is read out at the correct time synchronization across the entire sensor array. Proper video timing is essential for achieving high frame rates, reducing motion artifacts, and ensuring accurate color reproduction. Recent advancements in video timingtechniques have enabled significant improvements in image sensor performance. Innovations such as global shutter technology, which captures the entire image simultaneously, have reduced motion artifacts and enhanced image quality in high-speed applications. Additionally, advanced timing algorithms have improved synchronization in multi-sensor systems, enabling more accurate 3D imaging and depth perception. Camera interfaces are responsible for transmitting image data from the sensor to the processor, ensuring that the data is transferred efficiently and accurately. The choice of interface affects the speed, resolution, and overall performance of the imaging system. Common camera interfaces include MIPI (Mobile Industry Processor Interface), USB, HDMI, and Ethernet, each with its own set of advantages and applications. USB 3.0, MIPI CSI-2, and CoaXPress 2.0, have significantly improved the performance and application range of imaging systems. These innovations are driving improvements in data transfer rates, power efficiency, and compatibility across various industries, from consumer electronics and industrial automation to medical imaging and scientific research. Ensuring compatibility and standardization across different devices and systems is important for ease of integration and flexibility.. The MIPI CSI-2 standard is widely adopted in smartphones, tablets, and other portable electronics due to its ability to efficiently handle high-resolution image data and support advanced imaging features such as high dynamic range (HDR) and multi-exposure. MIPI interfaces will remain at the forefront of enabling high- performance, energy-efficient, and versatile imaging applications across various industries.

Table of Contents

Table of Figures

CHAPTER 1

INTRODUCTION

Timing analysis is a crucial aspect of digital circuit design and verification, ensuring that the designed system meets its timing requirements for proper functionality. A logic analyzer is an essential tool used in timing analysis, providing insights into the behavior of digital signals in a system. When used in conjunction with the ZCU104 development board, a logic analyzer can help designers and engineers verify and debug the timing of their embedded systems and FPGA designs.

The control and capture platforms are connected on USB to a host PC. Each communication is sent through the USB interface, processed on the platform, and the result is sent back to the PC. This induces long response time and prevent any real-time action in response to an event. Scheduler is currently onlyavailable on ZCU104. Porting on ZCU100 is planned… To address this limitation, a Scheduler is available on ASPIC and GPZ platforms. The scheduler executes a list of commands in response to a GPIO event. It is running on the GPZ R5 real-time processor, to minimize latency between an event and the commands The event can be any of the 20 GPIOs on the ZCU104. The commands can either - Toggle another GPIO - Read or write data on the I2C / I3C bus - Wait for a given delay - Make some computation on internal variable Data read from an I2C/I3C device can be - Stored in the 32 32-bits internal variables

- Stored in a read buffer (up to 2MB) - Stored in logs (~1.8MB) 8.9.4 Software gpz_ip dll is providing an API to use the scheduler GPSDK is wrapping this dll directly, all methods of the dll are available inGPSDK. The usage is very similar. The usual scheduler usage is as follow:

Define sensitivity list: which GPIO will trigger an event, on rising and/or falling edge For each event, define the sequence of command to execute

- Upload the sequence to the R5 on GPZ
- Start Scheduler
- Events are generated, and scheduler sequences executed
- Stop Scheduler

1.2 Shutter & its types:

1.2.1 Rolling Shutter :

A rolling shutter is a method of image capture in which a camera sensor does not capture the entire image at once. Instead, it captures the image line by line, either from top to bottom or bottom to top. This method is commonly used in CMOS sensors due to its cost-effectiveness and lower power consumption.

Working

1. **Line-by-Line Capture**:

- The sensor reads out the image data one line (or row) of pixels at a time.

- Each line is exposed sequentially, with a slight delay between the capture of each line.

2. **Exposure Timing**:

- The exposure of each line starts and ends at different times.

- This can lead to temporal differences in the captured image, especially noticeable with fast-moving objects orrapid camera movements.

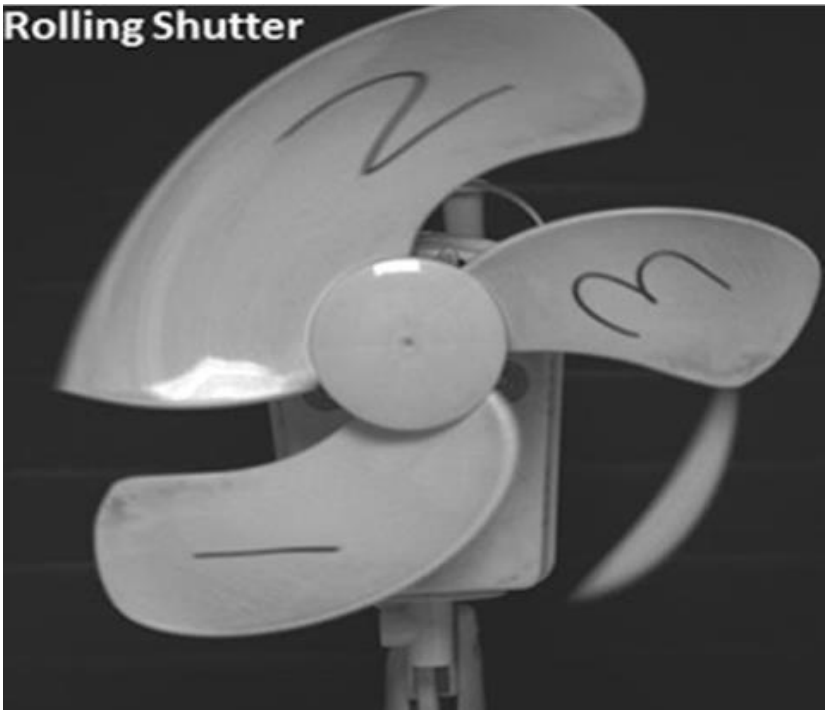


Fig1 Rolling shutter

1.2.2 Global Shutter :

A global shutter is a method of image capture where the entire sensor array is exposed to light simultaneously, capturing the entire image at once. This contrasts with the rolling shutter method, where the image is captured line by line. Global shutters are particularly useful in scenarios involving fast-moving objects or rapid camera movements, as they eliminate the motion artifacts commonly associated with rolling shutters.

WORKING:

1. **Simultaneous Exposure**:

- All pixels in the sensor array are exposed to light at the same time.
- The entire image is captured in a single instant, avoiding the temporal differences seen in rolling shutters.

2. **Readout Process**:

- After the exposure, the sensor reads out the pixel values simultaneously or in a very short time frame
.
- This ensures that the captured image represents a single moment in time.

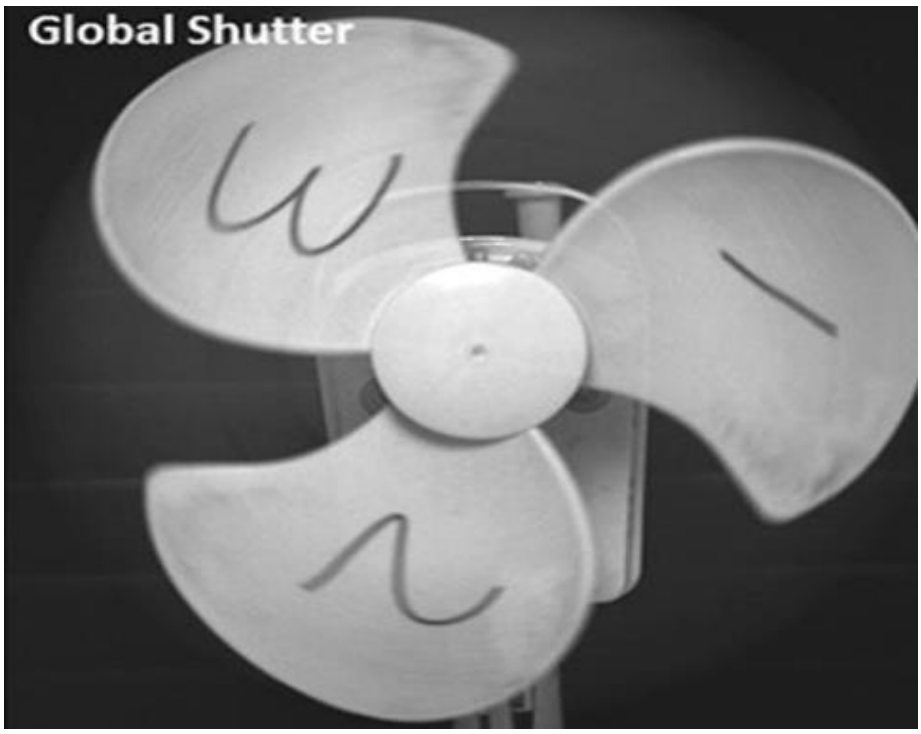


Fig2 Global shutter

1.3 CUSTOM ANALOG BLOCK

Custom Analog Block Information

Custom analog blocks are specialized circuits designed to perform specific analog functions within an integrated circuit (IC). These blocks are tailored to meet the unique requirements of a particular application, offering optimized performance, power efficiency, and functionality. Custom analog blocks are commonly used in various fields, including telecommunications, medical devices, automotive systems, and consumer electronics.

Components of Custom Analog Blocks

1. **Analog-to-Digital Converters (ADCs)**:

- Convert analog signals into digital data for processing by digital circuits.
- Custom ADCs can be optimized for resolution, sampling rate, and power consumption.

2. **Operational Amplifiers (Op-Amps)**:

- Used for signal amplification, filtering, and other analog signal processing tasks.
- Custom op-amps can be designed to meet specific gain, bandwidth, and noise requirements.

3. **Voltage Regulators**:

- Provide stable voltage levels to other components within the IC.
- Custom voltage regulators can be designed for specific voltage levels, load conditions, and efficiency.

4. **Phase-Locked Loops (PLLs)**:

- Used for frequency synthesis and clock generation.
- Custom PLLs can be optimized for jitter performance, lock time, and frequency range.

5. **Filters**:

- Used to remove unwanted frequencies from signals.
- Custom filters can be designed for specific cutoff frequencies, filter types (low-pass, high-pass, band-pass, band-stop), and quality factors.

CHAPTER 2

LITERATURE SURVEY

A capture system is a platform that allows image capture from a device (Sensor, processor…) to a host PC for post processing. It also gives the possibility to control this device and provides some specific features for test and validation. Video timing with respect to image sensors is a complex but vital component of modern video technology. It ensures that video frames are captured and displayed accurately, providing high-quality visual experiences across various applications, from consumer electronics to professional and industrial imaging systems. As technology continues to advance, ongoing research and development in video timing will further enhance the capabilities and performance of image sensors. video timing is essential for minimizing latency and reducing motion artifacts, such as skewing and wobble, which can occur with fast-moving objects or rapid camera movements. Proper synchronization between the image sensor and the display system is crucial for achieving high-quality video output. Advances in video timing techniques have enabled significant improvements in high-speed imaging,. Additionally, optimized video timing can enhance noise reduction, resulting in clearer and more detailed video signals.

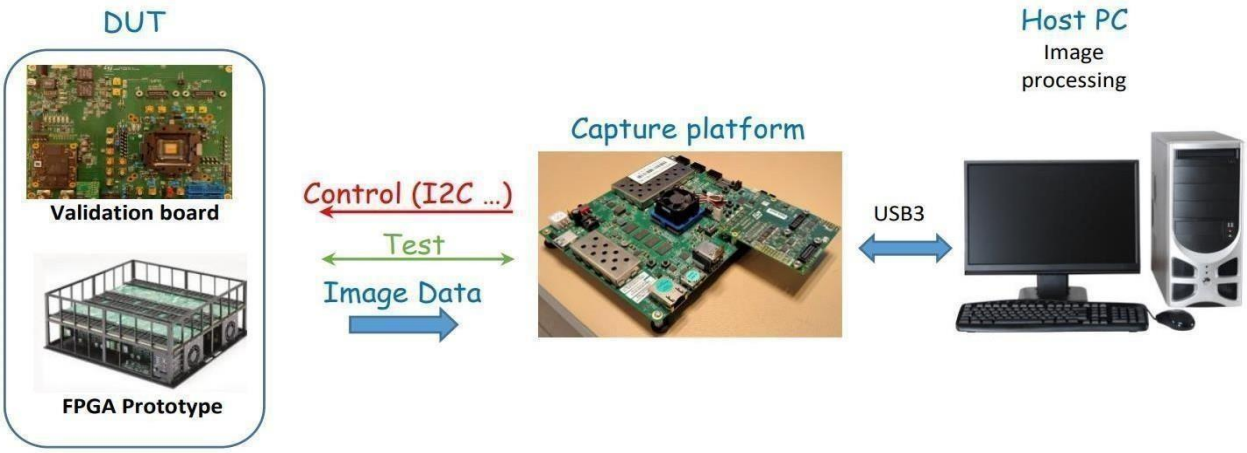


Figure3ZYNQCS12

ZCU104 U96

Figure2ZCU104& U96

A capture platform is complex system that associates various items

- Hardware➔ several boards
- FPGA➔XILINX Zynq Ultrascale Plus MPSoC
- Software➔ several software layers

2.1 ARCHITECTURE

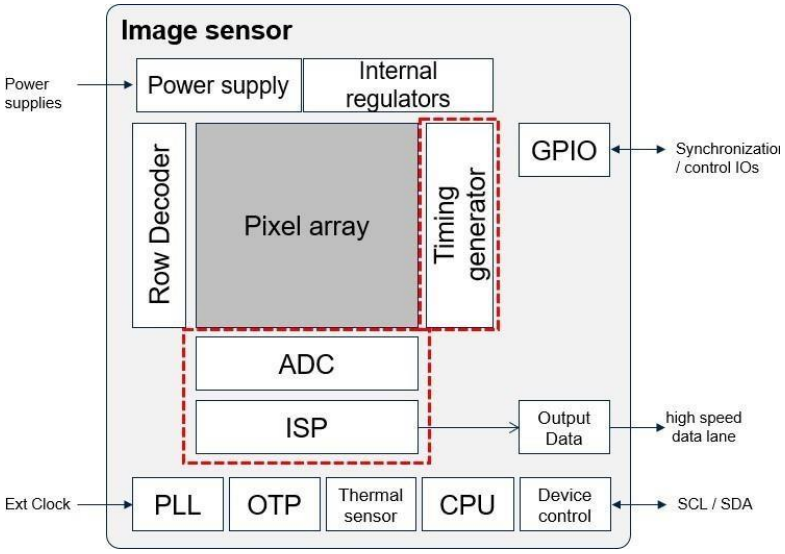


Figure4Image Sensor

2.2 Zynq MPSoC: The Xilinx Ultrascale+ MPSoC is a system on chip that combines programmable logic(PL) used to implementation custom IPs and processing system (PS) that includes hard macros suchas 64-bit processors and multiple other IPs (USB3, DDR4 controller)

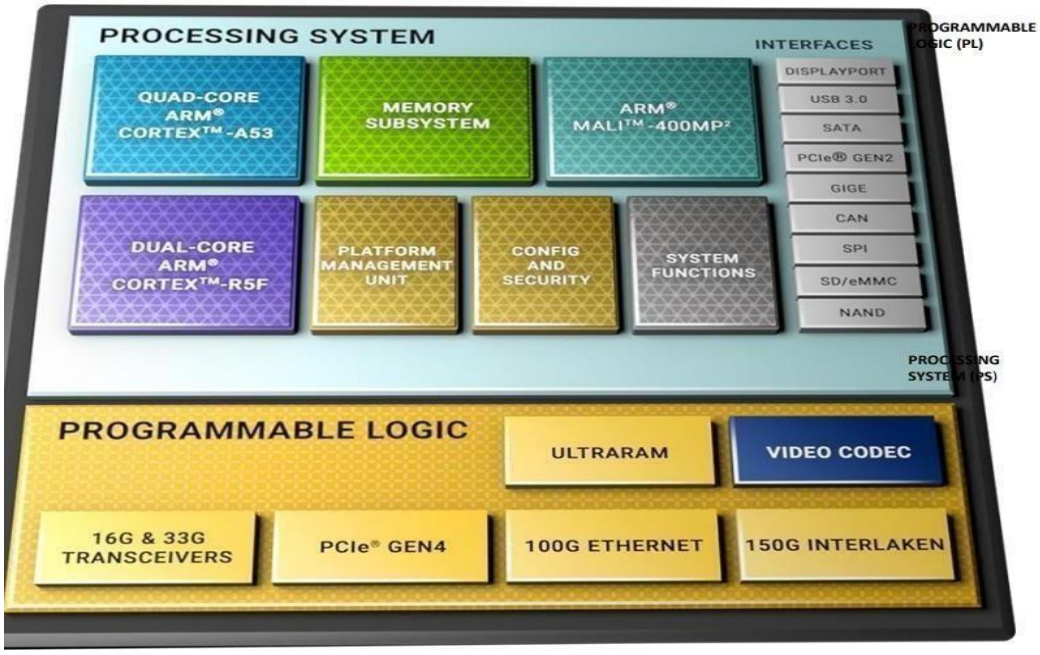
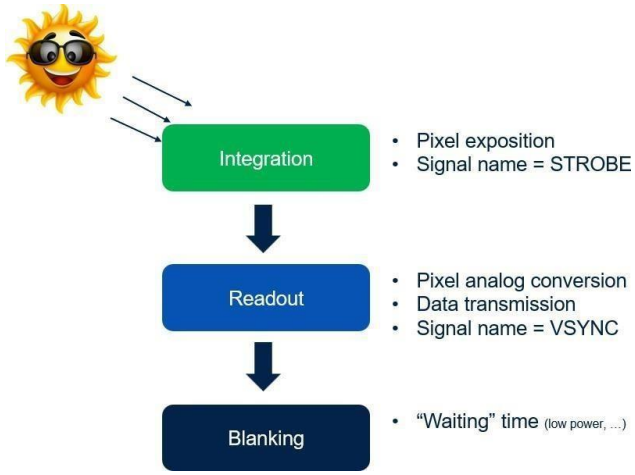


Figure4ZynqMPSoc

2.3 Video Timing

Video timing refers to the precise control of the sequence and duration of signals that define the display of video frames. Understanding video timing is crucial for working with video systems, as it ensures that images are displayed smoothly. This involves synchronization of pixel data, horizontal and vertical scanning, and the timing of various control signals. It also optimize video performance and ensure compatibility across different display technologies.



FigureVideo Timing

1. **Frame Rate**:

- The number of frames displayed per second (fps).
- Common frame rates include 24 fps, 30 fps, and 60 fps.

2. **Horizontal and Vertical Scanning**:

- **Horizontal Scanning**: The process of moving the electron beam (in CRTs) or the read/write head (in digital displays) from left to right across the screen to draw each line of pixels.
- **Vertical Scanning**: The process of moving from the bottom of one frame to the top of the next frame.

3. **Pixel Clock**:

- The clock signal that dictates the rate at which pixel data is transmitted.
- The frequency of the pixel clock is determined by the resolution and refresh rate.

2.2 ZCU104Interfaceboards–PCB4087B

The current interface board supported by ZCU104 is the PCB4087B-00B.The schematics of this board are available on BIMP board database

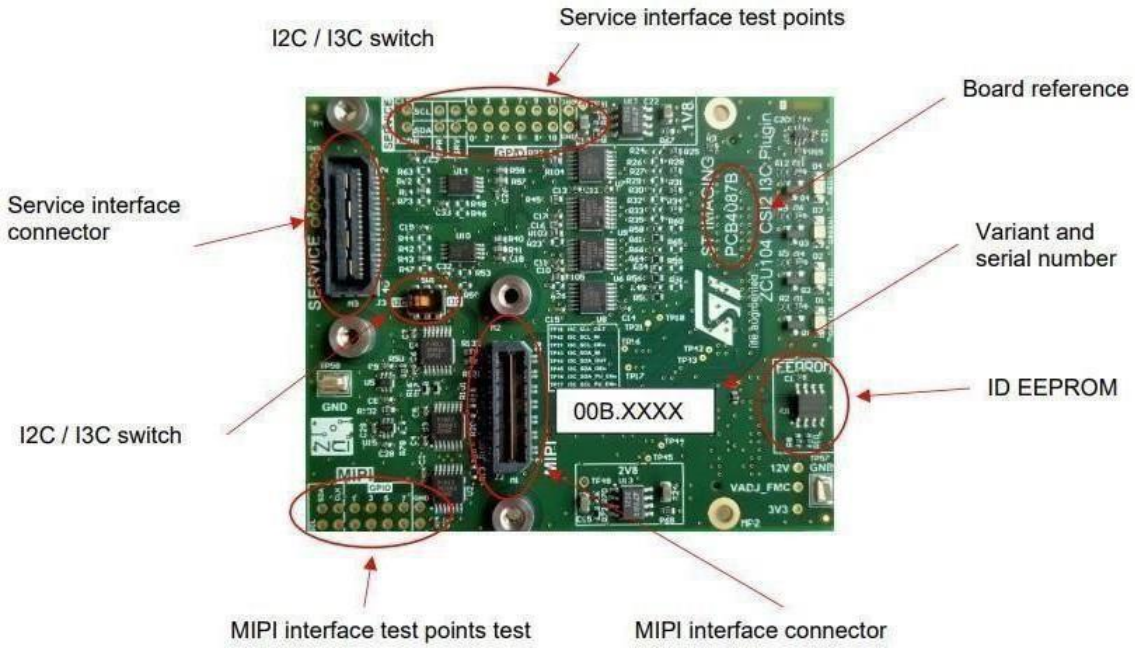
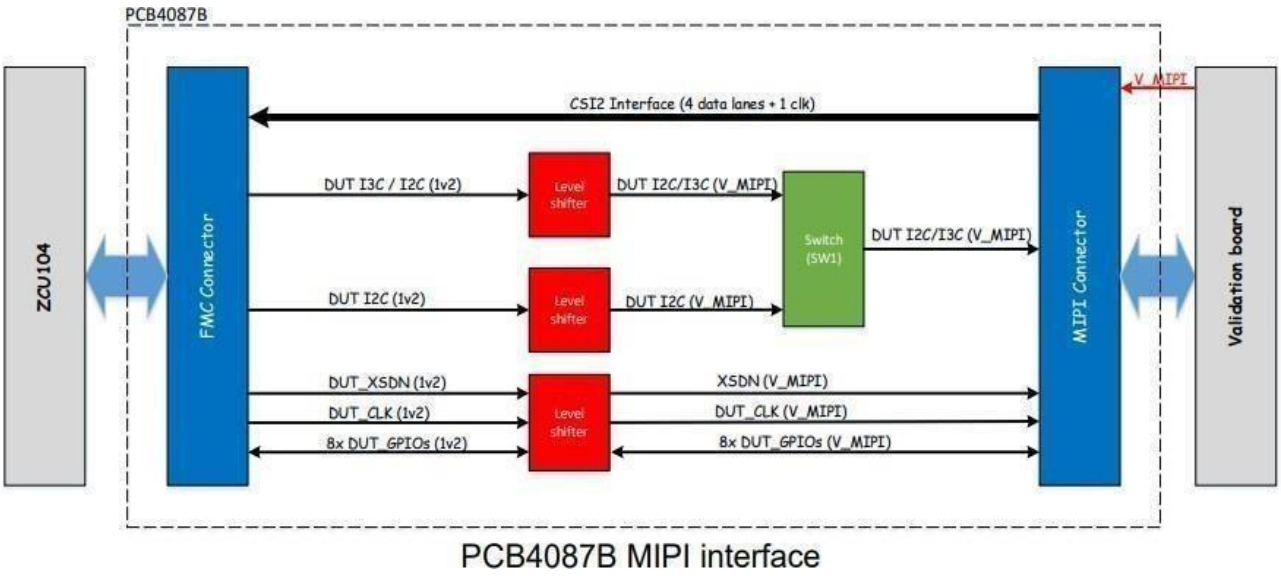


Figure6InterfacePCB4087B

Main features of PCB4087B are:

- MIPI interface connector including MIPI interface(4data+1CLK), I3C/I2C,GPIOs,ResetandClock.
- Service interface connector including I2C,SPIandGPIOs
- I3C/I2modeandLegacy I2Cmode(PCB4005B)support
- Test points on MIPI and Service interface signals



Most of interface signals are connected through voltage level shifters that may introduce some limitations Frequency

Each input/output channel of level shifters has an internal 10 kΩ pull-up resistors. These resistors must be taken into consideration when connecting pull-down resistors on the channel

2.4.1.1 ZCU104debugboard–FMC105

i) FMC connector. It can be useful to probe FMC connector signals**.**

Figure8 FMC105

ii) IOmapping on FMC105board is detailedin the table below:

Signal	FMC105	Signal	FMC105	Signal	FMC105
MIPI GPIO	J16.10	SRV GPIO 0	J1.33	SRV SCL	
MIPI GPIO	J16.12	SRV GPIO 1	J1.35	SRV SDA	
MIPI GPIO	J15.5	SRV GPIO 2	J1.10	SRV	J1.20
MIPI GPIO	J15.6	SRV GPIO 3	J1.12	SRV CLK	J1.18
MIPI GPIO	J20.2	SRV GPIO 4	J1.26	SPI CLK	
MIPI GPIO	J20.4	SRV GPIO 5	J1.28	SPI MISO	
MIPI GPIO	J16.5	SRV GPIO 6	J20.14	SPI MOSI	
MIPI GPIO	J16.7	SRV GPIO 7	J20.16	SPI CS	
MIPI SCL	J15.3	SRV GPIO 8	J1.17		
MIPI SDA	J15.4	SRV GPIO 9	J1.19		
MIPI XSDN	J16.6	SRV GPIO	J1.29		
MIPI CLK	J16.8	SRV GPIO	J1.31		

Figure9I/OMAPPINGinFMC105

2.5 Ultra 96 board

Ultra96board,also referenced ZCU100,isaZYNQUltrascale+MPSoC“lowcost”developmentboard.Itis compatible with 96Boards standard and can be ordered from AVNET.

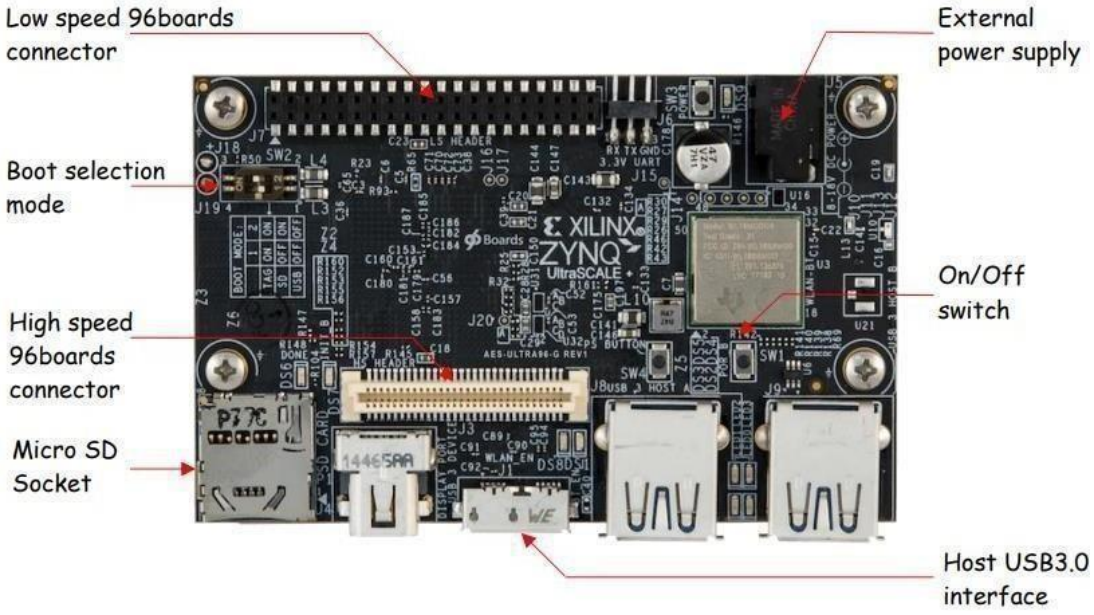


Figure10Ultra96Board

MainfeaturesofUltra96boardsusedincapturesystem are

- ZYNQUltrascale+MPSoCreferenceZU3EG-1 SBVA484

2.6 SOFTWAR AND TOOLS

2.6.1 Installing GPZ USB drivers

Prior using ZYNQ capture platforms, USB3driversmustbeinstalledonthehost PC. If the install is done correctly,and the ZCUconnected to the hostPCthoughtUSB3.0interface, hardware manager should look like the figure below.

Figure11GPZUSBDRIVERS

2.6.2 Installing UART debugger

MicroUSBtoUARTinterfaceisavailableontheZCU104fordebugpurpose,itallowscommunication with PS micro-processors thought hyper terminal.

OncethedriverinstalledandtheUSBcableconnected,3serialportswillbedetected,thefirstport detected should beselected and configured at 115200 bauds

Figure12UARTDebugger

2.6.3 Programming the MicroSD

Zynq Linux image and PL configuration file (bitstream) are embedded on the micro SD card used by ZCU104orUltra96tobooton.EachmicroSDimageversionisassociatedtoaspecificGPSDKrelease.

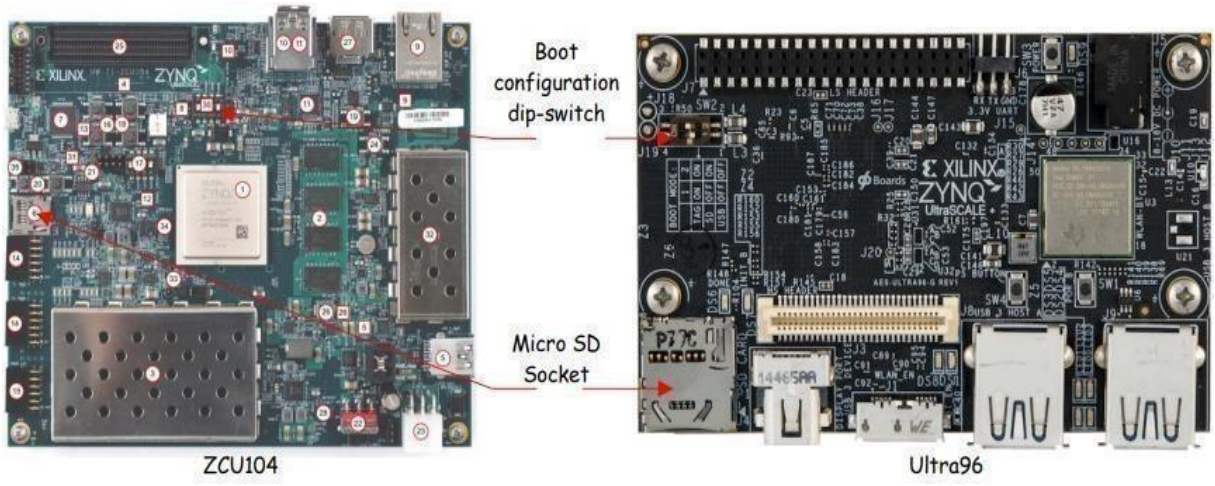


Figure13MicroSDPROGRAMMING

2.6.4 GPZ FLASH GUI

2.6.4.1 GPZflashGUI allows on board micro SD card programming.

- Useful when working on the ZCU in remote or when no USB card reader is available
- GPZ flash GUI is anupdater, it allows updating programmed SDcards only.Forempty

SDcards,useWin32diskimager.

- LaunchGPZ_Flash_GUI.exe.
- OpencompressedSDimage(.img.gzformat).
- WritetheSDcard
- RebootttheboardwiththeHyperTerminalorwithV2Wreg.

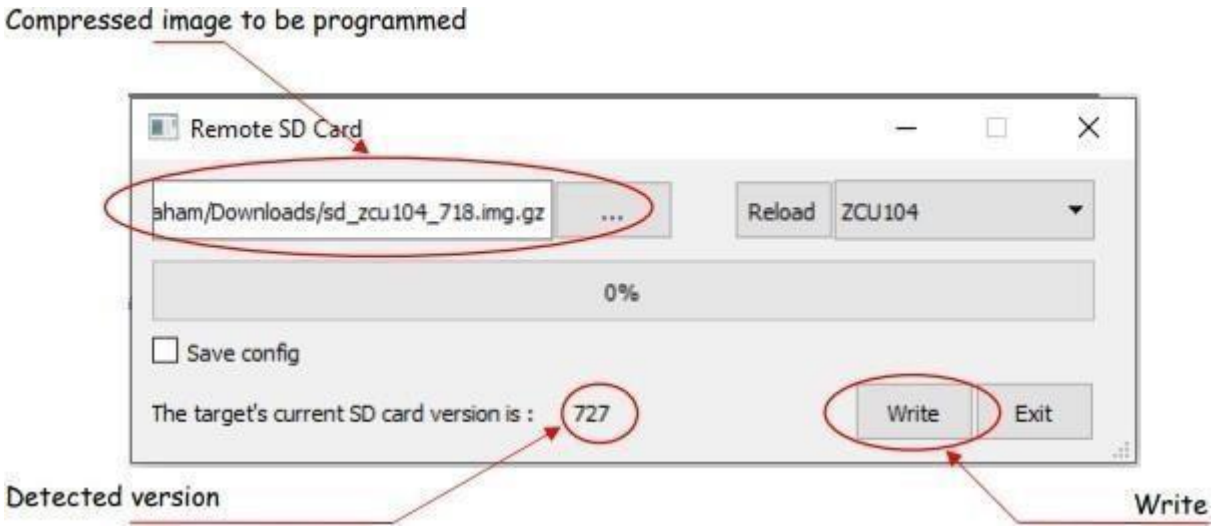


Figure14GPZFLASHGUI

2.6.5 Work Package

Using methods in GPSDK, the Work Package for each project should check the system configuration:

- Model and SD version used Update SD version GPSDK recommendation
- Daughter board model and version connected

2.7 INTERFACINGPLATFORMS

ZCU capture platforms can interface various systems for validation, characterization, purposes. Bellow some examples of interconnection.

2.7.1 ZCU104➔HAPS

HAPSplatformsareFPGAboardsusedforsiliconprototypingactivities.InCSI2 configurations, ZCU104 canbe connected to HAPS.

i) HAPSDX7➔FPGA prototyping platform

ii) STMC➔XP70microcontroller debugger

iii) HAPS debug daughter board(BRD579)➔STMC2toHAPSinterfaceboard

iv) HAPSCSI2daughterboard(BRD000608)➔CSI2toHAPSinterfaceboard

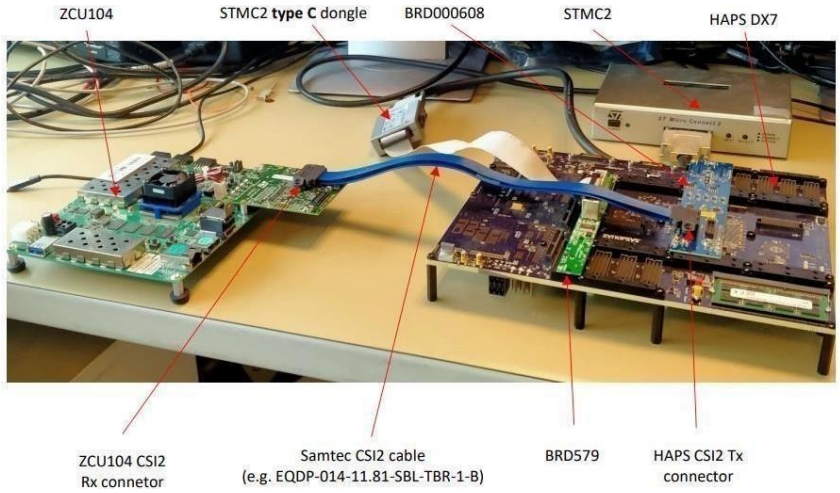


*Figure15*ZCU104➔HAPS

2.6.5 ZCu->Validation Board

ZCU104canbeconnectedtosiliconvalidationboardthroughPCB4087Binterfaceboardandwith Samtec cables (EQDP and EQCD). This setup requires PCB4087B-00B build.

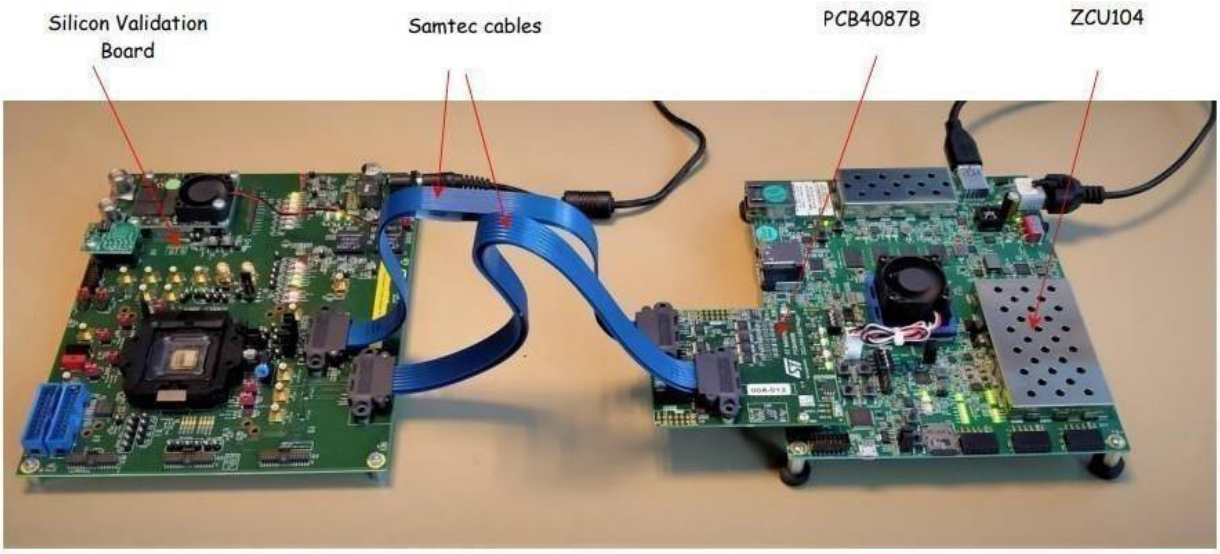


*Figure16*ZCU104➔Validation

2.6.6 Communication-I2Cmaster

ZCU104implements2I2Cinterfaces –

-**Device Under Test (DUT)**interface dedicated to DUT and located on MIPI connector.

-**Service interface:** forperipherals(e.g.voltageregulators,currentsensors…)andlocatedonserviceconnector.

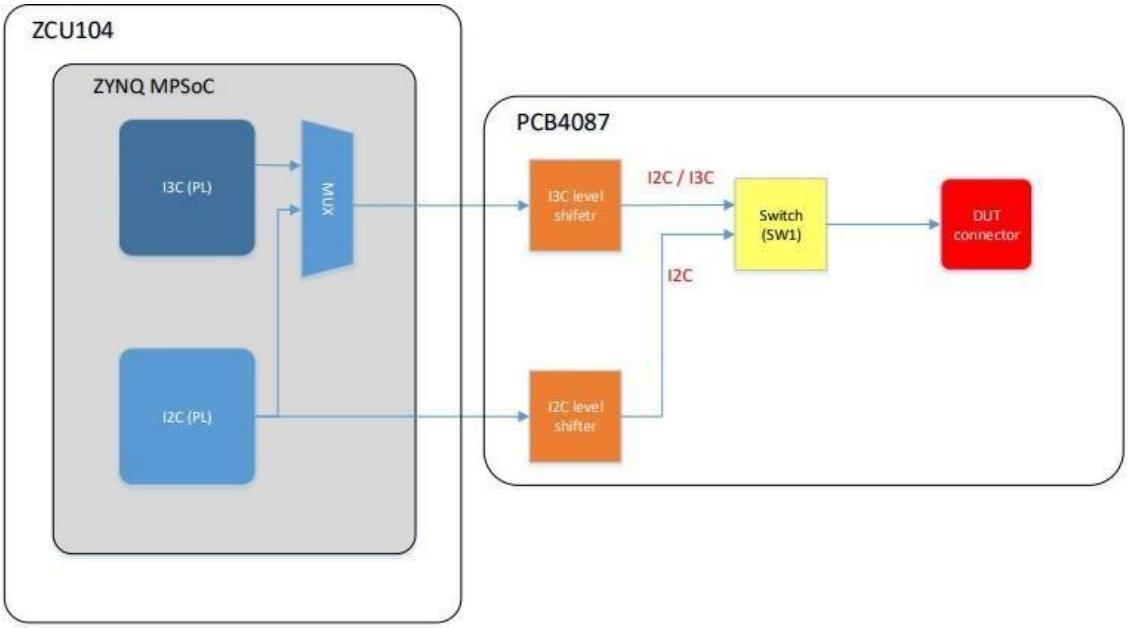
2.6.7ZCU104I2C **hardware**:

TwoconfigurationsareavailableforI2C/I3CmanagementontheZCU104system

Standard mode:

InthismodeonlyI2Cinterfaceissupported(noI3C).ThisisthePCB4087Bstandardmodewhichisinthis case compatible with PCB4005B interface board. SW1 switch must be on “I2C” position.

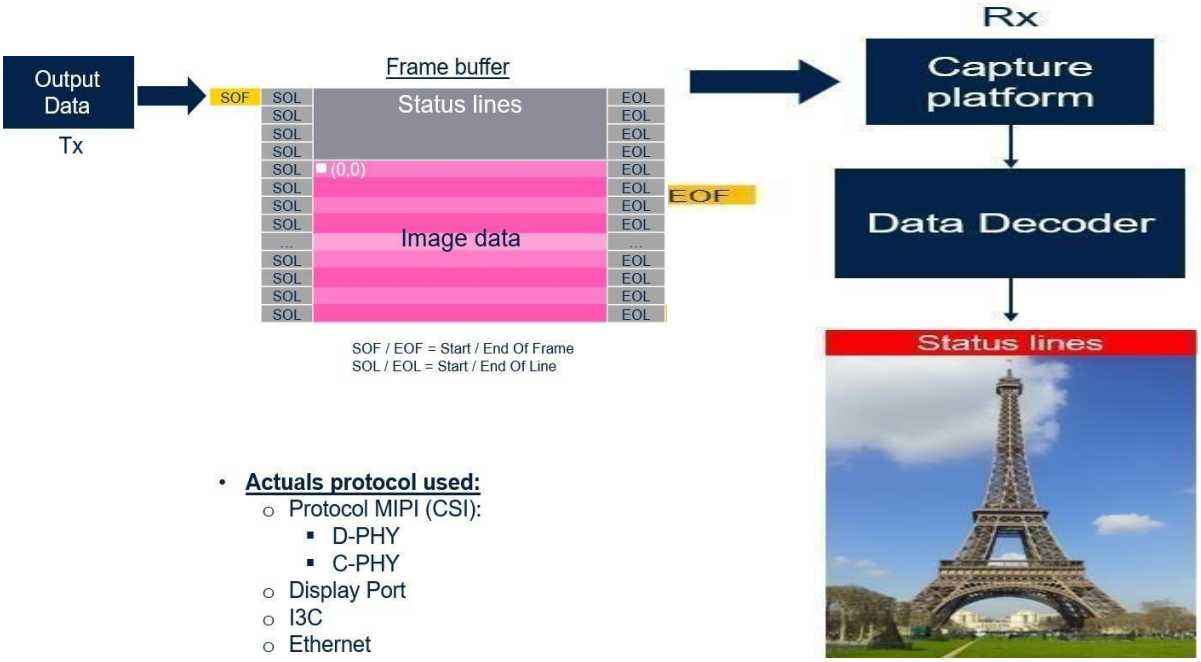
- **I3Cmode**:This mod is required I3C interface. See I3C feature for more details



-

*Figure17 Z*CU104I2Chardware

2.7 Output Data Interface



- **Actuals protocol used:**
  - Protocol MIPI (CSI):
    - D-PHY
    - C-PHY
  - Display Port
  - I3C
  - Ethernet

Figure18Output Data Interface

CHAPTER2

Project3:Link Quality Symbols

Introduction:

Link quality The High-Speed Serial Interface (HSSI) is a cornerstone of the modern communications. With the concurrent increasing of design complexity and decreasing of the time budget, the pre-silicon validation, debugging and testing of HSSIs are becoming crucial.

Thepatternsusedtovalidateahigh-speedinterfaceareLinkQualityPatterns.Theyaregeneratedandtransmitted across the interface to validate its performance and ensure its compliance with industry standards.

3.1 Link Quality Symbols

Link Quality symbols or Patterns are used to evaluate the performance of digital communication systems including wired and wireless networks, fiber optic links, and high-speed digital interfaces. These patterns consistofbinarydatasequencesthataretransmittedacrossthelinkandusedtotesttheintegrityandreliability of the link.

There are several types of Link Quality Patterns, Including:

1. Bit Error Rate(BER)patterns
2. Pseudo-Random Binary Sequence(PRBS)patterns
3. Cyclic Redundancy Check(CRC)patterns

Overall, link quality patternsareanessentialtooltovalidateandtroubleshootdigitalcommunicationsystems. These patterns help find errors in a variety of applications, including networking, telecommunications, and storage systems.

3.1.1 Pseudo-Random Binary Sequence(PRBS)

The PRBS pattern is often used to simulate real-world data traffic and to test the integrity of a high-speed interface.BytransmittingaPRBSpatternacrossaninterfaceandcomparingthereceiveddatawiththeoriginal data, engineers can determine whether the interface is working properly and identify any errors or anomalies.

PRBS Patterns are generally used in compliance testing technologies and to identify areas for improvement. There are several types of PRBS patterns, and the most used ones are:

- **PRBS-7:**A7-bit PRBS pattern generated using a 3-stage shift register.The pattern has aperiodof2^7-1= 127bits.
- **PRBS-9:** A9-bit PRBS pattern generated using a 3-stage shift register. The pattern has aperiodof2^7-1= 127bits.
- **PRBS-15:** A15-bit PRBS pattern generated using a 3-stage shift register. The pattern has a period of of2^15-1= 32767bits.
- **PRBS-23:** A23-bit PRBS pattern generated using a 3-stage shift register. The pattern has aperiodof2^7-1= 127bits. f2^23-1= 8388607bits.
- **PRBS-31:** A31-bit PRBS pattern generated using a 3-stage shift register. The pattern has a period of of2^31-1= 2147483647bits.

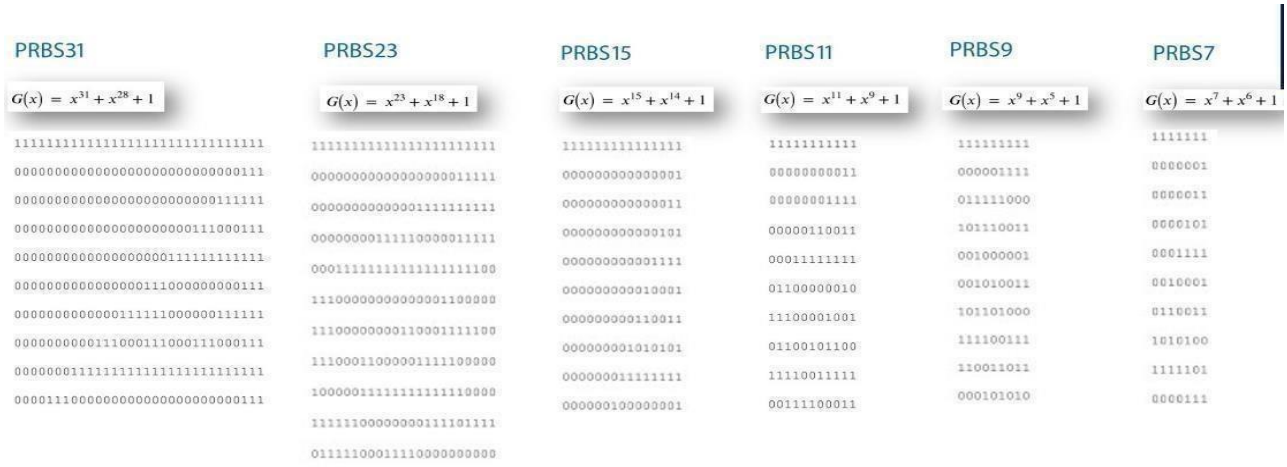


Figure23PolynomialsofvariousPRBSPatterns

3.1.2Why do we use PRBS?

PRBS patterns have several important properties that make them useful for testing digital communication systems:

1. Long sequence length: PRBS patterns have a long period, which means that the sequence will repeat itself only after a very large number of bits have been transmitted. This makes them useful for testing systems over a long period of time.
2. Good statistical properties: PRBS patterns have good random properties, meaning that the ones and zeros are distributed evenly and randomly through out the sequence. This is important for testing the performance communication systems under realistic conditions.
3. Easy to generate: PRBS patterns can be easily generated using shift registers, which are simple digital circuits that can be implemented using common logic gates.
4. Easy to synchronize:PRBS Patterns can be easily synchronized for clock recovery, which makes them useful for testing digital communication systems.

3.2Software and Tools Used:

3.1.2PyCharm-

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains. It provides code analysis, a graphical debugger, an integrated unit tester. PyCharm is cross platform with Windows, macOS and Linux versions. The Community Edition is released under the Apache License, and there is also Professional Edition with extra features – released under a proprietary license. In this project, I have used PyCharm Version 2022.3.1.

PyCharm features includes:

- Code editing and analysis: PyCharm provides helping completion ,syntax highlighting, and error highlighting, and hence making it easier to write clean and correct the Python code.
- Debugging: PyCharm includes a powerful debugger which allows the developers to step through the code and identify and fix various issues quickly.

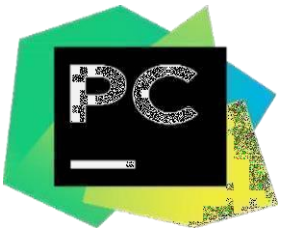


Figure19LogoforPyCharm

3.1.2 Python-

Python is the high-level, and a general-purpose programming language. Its design emphasis on the code object-oriented and functional programming. Python codes are very simple and are easy to understand and are not too complex and have the good readability.

Python is the interpreter language which means that the source code of python program is converted into the bytecodes that is executed by the Python Virtual Machine.

Andinpythoncoderunslinebylineratherthanrunningalltogether.PythonistheProgramminglanguage with many different IDEs available.Some of which are PyCharm, and VisualStudioCode .And each IDE has the unique features and benefits.

Figure20 Python Logo

E.g.:**open("file_name",mode)**

Mode–Thisparameterisastringthatisusedtospecifythemodeinwhichthefileistobeopened.The following can be used to activate a specific mode:

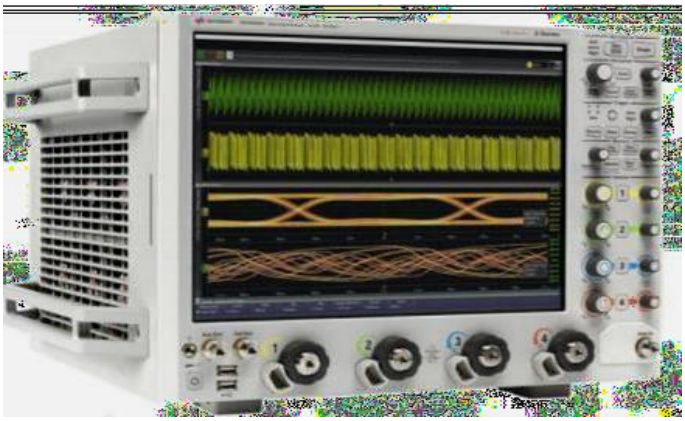
1. **"r"**–This string isusedto readonlythefile.
2. "**w"**–This isused for writing on/overthefile.
3. **"a"**–It is used to append content to the existing files.
4. **"b"**–When user wants to handle the file in the binary mode.

3.1.3 DSAZ334AInfinium Oscilloscope:

Infinium is a family of oscilloscope and logic analyzer products developed by Keysight Technologies for troubleshooting applications. These products offer a wide range of features including high bandwidth, deep memory, advanced triggering, and analysis tools for debugging and characterizing high-speed digital and mixed- signal designs.The Infiniium oscilloscopes and logic analyzer support a variety of communication

Standards including PCIExpress, USB ,DDR, HDMI, and Ethernet.

Figure21 DSAZ334AInfiniium Oscilloscope



As an intern at ST, I didn’t have the access to use the infinium oscilloscope from my ST laptop directly. So, I had to access it through another pc using Remote Access Connection on which I accessed the infinium oscilloscope through VNC Viewer.

3.1.4VNC VIEWER

VNC Viewer is used for local computers and mobile devices you want to control rom. A device such as computer, tablet, or smart phone with VNC Viewer software installed can access and take control of the desktop of a remote computer (running VNC Server) from your device, and it transmits the keyboard and mouse or touch events to

Figure22 LogoforVNCViewer

3.2 Work and Output:

PROBLEMSTATEMENT1:

UnderstoodthegenerationofPRBSthatweretransmittedoverthetransmissionlanesandwere matched With combination of 0 and 1. So, ST needed a python script that could not only generate and detect which PRBS was currently received from interface but also detect errors in the sequence.

- Setting up workspace in VNC(Virtual Network Computing)
- Setting up PyCharm in STlaptop.

- Brushing up of Python Syntax and Programming concepts.
- Understanding the generation of PRBS. Developing an algorithm for generation of various PRBS. DevelopingapythonscriptonPyCharmforeachPRBSsequence.Combiningthepythonscriptsof various PRBS into a single code.
- Developing an algorithm and code for detection and verification of various PRBS with the data provided in csv files.

- Creating a Graphic User Interface to Display the result.

ALGORITHM:

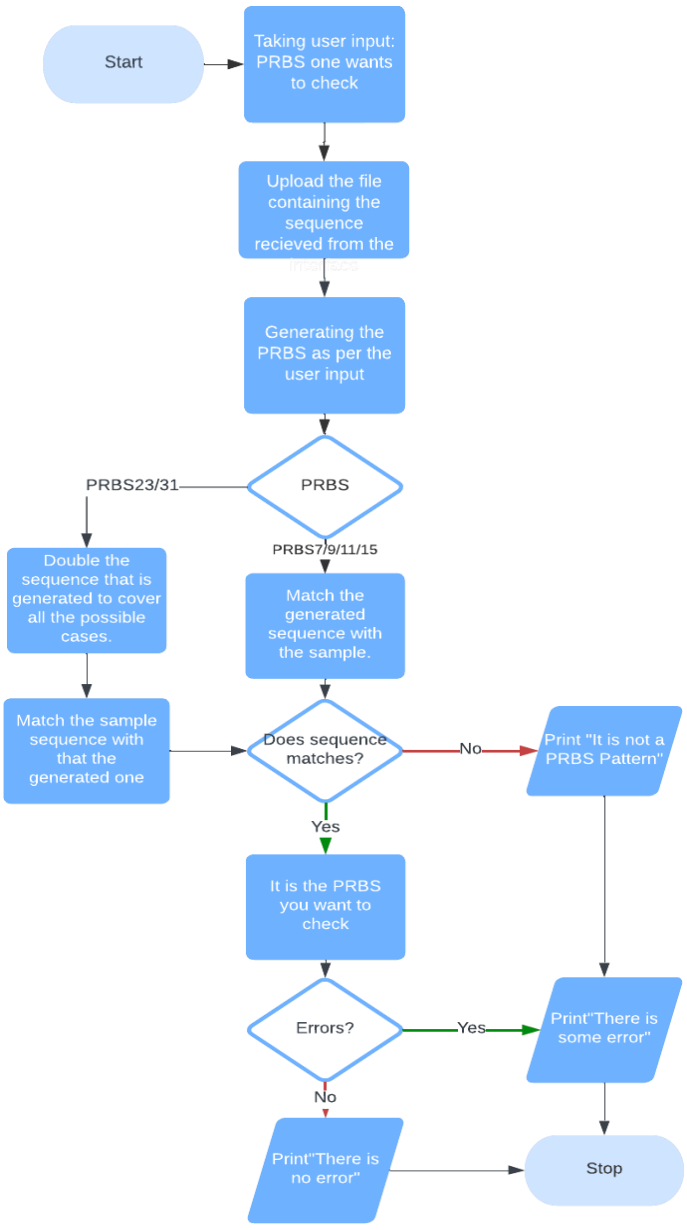


Figure23 flowchart of the code

3.2.1 A)RESULT1:(Outputofthepythonscript)

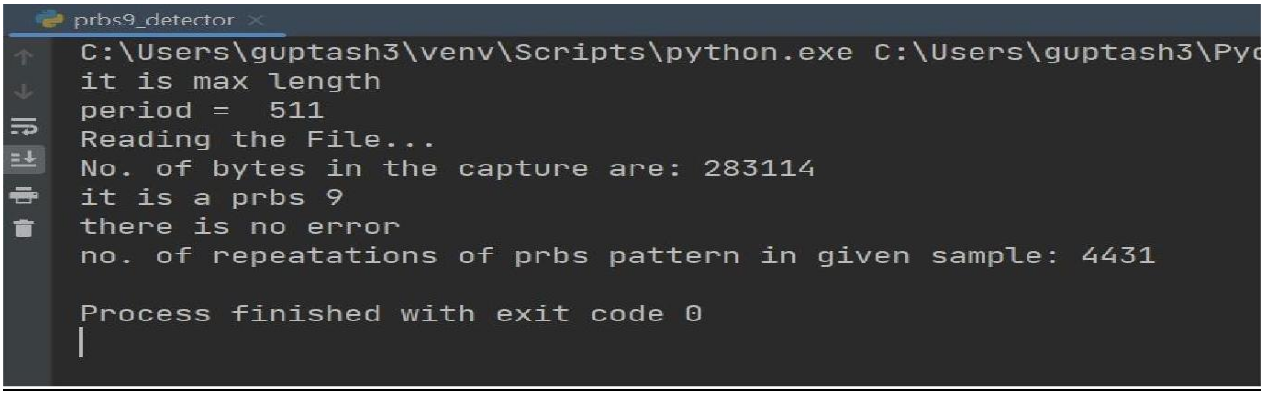


Figure24 Output of PRBS9

3.3.1B)RESULT2:(Output of The GUI)

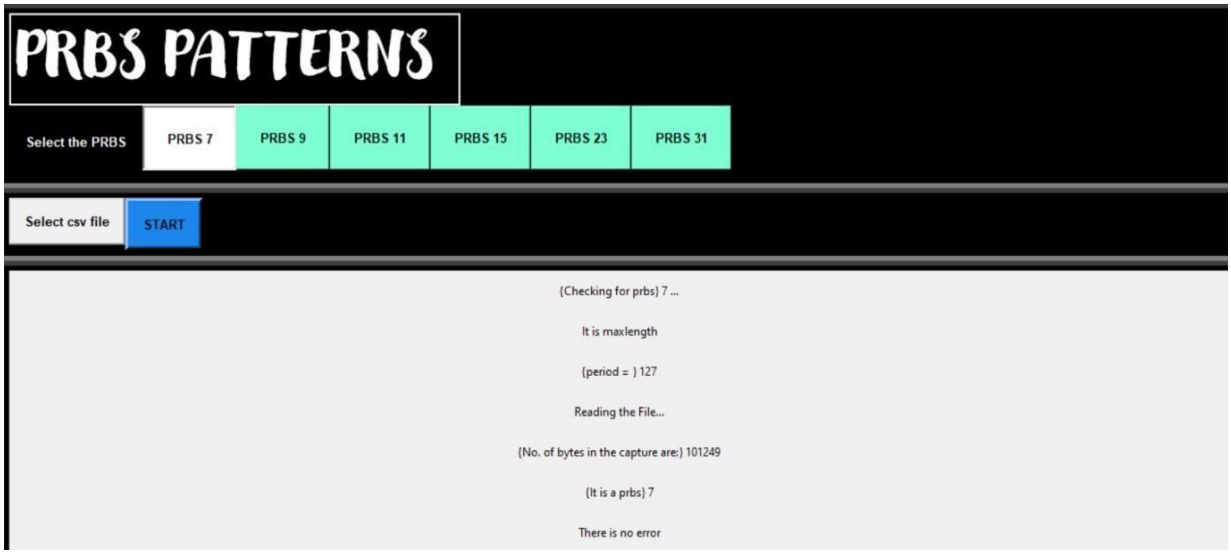


Figure25 Graphic User Interface

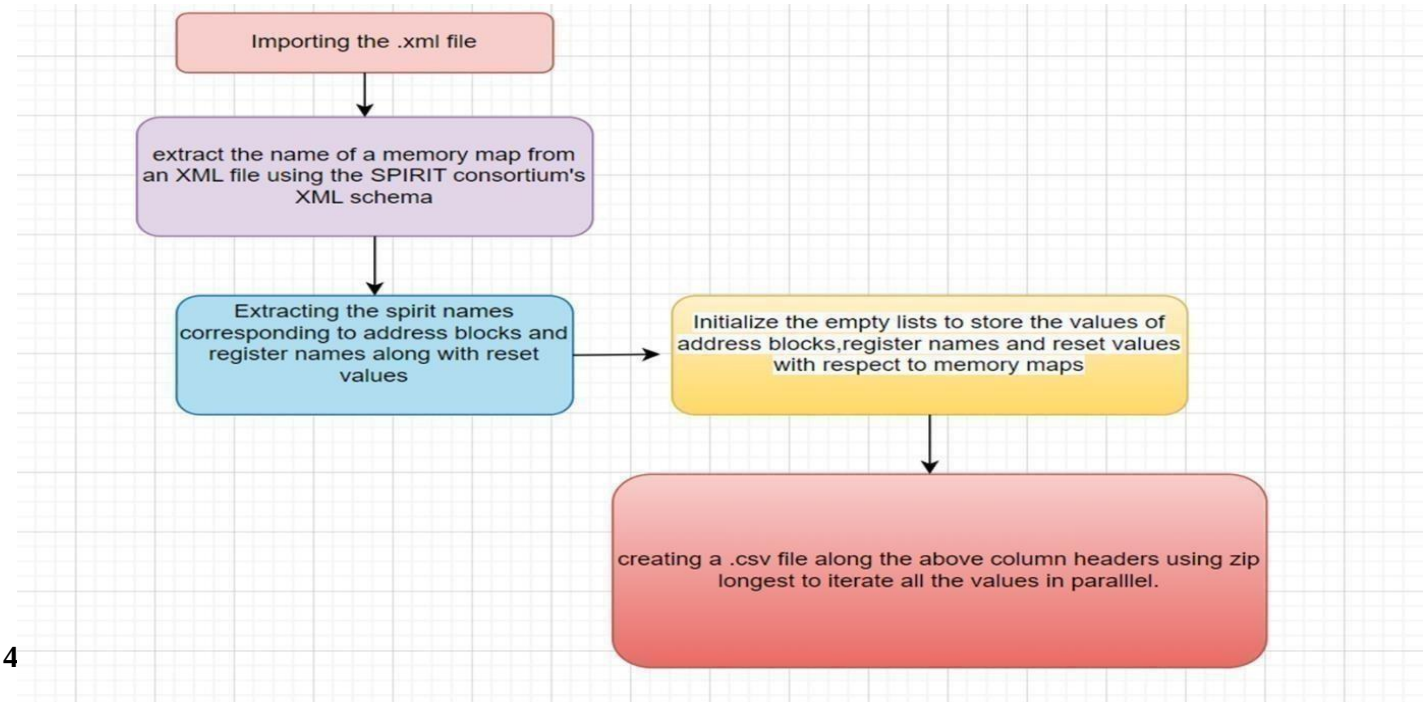
CHAPTER4
XML PARSER

4.1XML PARSER:

To Parse All The Specific REGISTERS And ADDRESS BLOCKS From XML Files In A Directory And Write The Output To a .xls File. Further Similarly From A Given .py Python File Extract All The
.Readreg And .Writereg REGISTERS AND ADDRESS BLOCKS And Compare Both The Registers And Address Block Columns And Find All The Accessed Regs And Make A .Csv File Of It It.

WORKINGFLOW-

- The Code Parses A Set Of XML Files And Extracts Specific Data From Them.
- It Uses The Os Module For File System Interaction, Allowing The User To Select Specific Folders And Files To Be Parsed.
- The Xml. Etree.Elementtree Library Is Used To Parse The XML Files And Extract The Desired Data.
- The Extracted Data Is Written To A Csv File And An Excel Workbook For Further Analysis.
- The Code Uses Extraction Statements To Check For Corresponding Address Blocks And Register Names From The Given Curtiss.Py File
- The Code Has Some Innovations, Such As File System Interaction, XML Parsing, File Handling, String Manipulation,ConditionalStatements,ExcelWorkbookCreation,WorksheetManipulation,AndWorkbook Saving.
- The Code also faces some major challenges such as File Format Changes, Large Input Data, Errors, And Excel Workbook Size



4

Figure26 Workingflowofxmlparser

4.1.1 InputCurtiss.pyfile:

```
1 usage
@property
def aci_lane_count_set(self):
    self._aci_lane_count_set = self.read_reg(self.regs.STREAM_STATIC_ACICD_LANE_COUNT_SET)
    return self._aci_lane_count_set

@aci_lane_count_set.setter
def aci_lane_count_set(self, value):
    aci_range = range(0, 2)
    assert value in aci_range, \
        'aci_lane_nb should be in %s, got %d' % (aci_range, value)
    self._aci_lane_count_set = value
    self.write_reg(self.regs.STREAM_STATIC_ACICD_LANE_COUNT_SET, self._aci_lane_count_set)

1 usage
@property
def aci_link_rate_set(self):
    self._aci_link_rate_set = self.read_reg(self.regs.STREAM_DYNAMIC_ACICD_LINK_RATE_SET)
    return self._aci_link_rate_set
```

Figure27 Input curtiss.py file

i)Input.xmlfile:

```
<spirit:name>DEVICE_MODEL_ID</spirit:name>
<spirit:displayName>DEVICE_MODEL_ID register</spirit:displayName>
<spirit:description>DEVICE_MODEL_ID register</spirit:description>
<spirit:addressOffset>0x0</spirit:addressOffset>
<spirit:size>32</spirit:size>
<spirit:access>read-only</spirit:access>
▼<spirit:reset>
  <spirit:value>0x53544343</spirit:value>
</spirit:reset>
▼<spirit:field>
  <spirit:name>VALUE</spirit:name>
  <spirit:description>Device identification in ASCII (S3L9)</spirit:description>
  <spirit:bitOffset>0</spirit:bitOffset>
  <spirit:bitWidth>32</spirit:bitWidth>
  <spirit:access>read-only</spirit:access>
</spirit:field>
</spirit:register>
▼<spirit:register>
<spirit:name>DEVICE_REVISION</spirit:name>
<spirit:displayName>DEVICE_REVISION register</spirit:displayName>
<spirit:description>DEVICE_REVISION register</spirit:description>
<spirit:addressOffset>0x4</spirit:addressOffset>
<spirit:size>32</spirit:size>
<spirit:access>read-only</spirit:access>
▼<spirit:reset>
  <spirit:value>0xa000000</spirit:value>
</spirit:reset>
▼<spirit:field>
  <spirit:name>LOGIC_BE_REVISION</spirit:name>
  <spirit:description>C40 die back end revision</spirit:description>
  <spirit:bitOffset>16</spirit:bitOffset>
  <spirit:bitWidth>8</spirit:bitWidth>
  <spirit:access>read-only</spirit:access>
</spirit:field>
▼<spirit:field>
  <spirit:name>LOGIC_FE_REVISION</spirit:name>
  <spirit:description>C40 die front end revision</spirit:description>
```

Figure28 Input.xml file

ii)Accessed registers from .xml file:

	Memory Map	Address Name	Register Name	Field Name	Reset Value
2	UI	REVISION_STATUS	DEVICE_MODEL_ID	VALUE	0x53544343
3	UI	REVISION_STATUS	DEVICE_REVISION	LOGIC_BE_REVISION	0xa000000
4	UI	REVISION_STATUS	DEVICE_REVISION	LOGIC_FE_REVISION	0x20400
5	UI	REVISION_STATUS	FW_REVISION	BUGFIX	0x17
6	UI	REVISION_STATUS	FW_REVISION	MINOR	0x0
7	UI	REVISION_STATUS	FW_REVISION	MAJOR	0x0
8	UI	REVISION_STATUS	UI_REVISION	MINOR	0x0
9	UI	REVISION_STATUS	UI_REVISION	MAJOR	0x0
10	UI	REVISION_STATUS	FW_PATCH_REVISION	MINOR	0x0
11	UI	REVISION_STATUS	FW_PATCH_REVISION	MAJOR	0x0
12	UI	REVISION_STATUS	ROM_MD5	VALUE	0x0
13	UI	REVISION_STATUS	DIE_TRAC	VALUE	0x0
14	UI	SYSTEM_STATUS	FRAME_COUNTER	VALUE	0x0
15	UI	SYSTEM_STATUS	ERROR_CODE	VALUE	0x0
16	UI	SYSTEM_STATUS	ERROR_STATUS	VALUE	0x0
17	UI	SYSTEM_STATUS	CURRENT_ACTIVE_LANE_NBR	CURRENT_ACTIVE_LANE_NBR_SF0	0x0
18	UI	SYSTEM_STATUS	CURRENT_ACTIVE_LANE_NBR	CURRENT_ACTIVE_LANE_NBR_SF1	0x0
19	UI	SYSTEM_STATUS	CURRENT_ACTIVE_LANE_NBR	VALUE	0x10
20	UI	SYSTEM_STATUS	ACICD_ACI_POWER_STATE	VALUE	0x9

Figure29 AccessedRegisters.xml

(i) Accessed Registers :

```
CONTROL.COMMAND
SYSTEM_STATUS.ERROR_CODE
CONTROL.COMMAND
SYSTEM_STATUS.ERROR_CODE
DEBUG.MINIMUM_LATCH_DURATION_INTER_USEC
DEBUG.MINIMUM_LATCH_DURATION_INTRA_USEC
STREAM_DYNAMIC.INTRA_FRAME_BLANKING
STREAM_DYNAMIC.FRAME_READ_OFFSET
STREAM_DYNAMIC.FRAME_ACTIVE_OFFSET
DEBUG.LP_EXIT_DURATION
DEBUG.ULP_EXIT_DURATION
STREAM_STATIC.LATCH_POINT
DEBUG.ISL_LINES_BOT
DEBUG.ALPM_WAKEUP_ACTIVE_OFFSET_LINES
DEBUG.ALPM_WAKEUP_OFFSET_USEC
DEBUG.FIRST_FRAME_BOOST
DEBUG.ISL_SF0_OFFSET_USEC
STREAM_CTX_0.ACI_VALID_BIT_FLAG_ENABLE
STREAM_CTX_1.ACI_VALID_BIT_FLAG_ENABLE
SYSTEM_STATUS.CLK_PIXEL
SYSTEM_STATUS.CLK_SYS
SYSTEM_STATUS.CLK_MCU
REVISION_STATUS.FW_REVISION
REVISION_STATUS.UI_REVISION
REVISION_STATUS.FW_PATCH_REVISION
REVISION_STATUS.DEVICE_REVISION
REVISION_STATUS.DEVICE_MODEL_ID
SYSTEM_STATUS.ACICD_ACI_VERSION
SYSTEM_STATUS.ACICD_MAX_LANE_COUNT
SYSTEM_STATUS.ACICD_ALPM_CAPABILITY
SYSTEM_STATUS.ACICD_MAX_LINK_RATE
SYSTEM_STATUS.ACICD_RAW_FORMATS_SUPPORTED
DEBUG.STREAM_STALL
SYSTEM_STATUS.CLK_PLL_SYS
```

Figure30 Accessed Registersfrom.py

(ii) Final Accessed .readreg and. writeregRegisters:

	Memory Map	Address Name	Register Name	tested
2	UI	REVISION_STATUS	DEVICE_MODEL_ID	Accessed ok
3	UI	REVISION_STATUS	DEVICE_REVISION	Accessed ok
4	UI	REVISION_STATUS	DEVICE_REVISION	Accessed ok
5	UI	REVISION_STATUS	FW_REVISION	Accessed ok
6	UI	REVISION_STATUS	FW_REVISION	Accessed ok
7	UI	REVISION_STATUS	FW_REVISION	Accessed ok
8	UI	REVISION_STATUS	UI_REVISION	Accessed ok
9	UI	REVISION_STATUS	UI_REVISION	Accessed ok
10	UI	REVISION_STATUS	FW_PATCH_REVISION	Accessed ok
11	UI	REVISION_STATUS	FW_PATCH_REVISION	Accessed ok
12	UI	REVISION_STATUS	ROM_MD5	
13	UI	REVISION_STATUS	DIE_TRAC	
14	UI	SYSTEM_STATUS	FRAME_COUNTER	
15	UI	SYSTEM_STATUS	ERROR_CODE	Accessed ok
16	UI	SYSTEM_STATUS	ERROR_STATUS	
17	UI	SYSTEM_STATUS	CURRENT_ACTIVE_LANE_NBR	
18	UI	SYSTEM_STATUS	CURRENT_ACTIVE_LANE_NBR	
19	UI	SYSTEM_STATUS	CURRENT_ACTIVE_LANE_NBR	
20	UI	SYSTEM_STATUS	ACICD_ACI_POWER_STATE	

Figure31 Final Accessed Registers

4.2 XML Parser for aci non-image packets.py

(i) Inputacinon-image.py

```
def get_custom_pkt_info(self):
    en = False
    if self.bench.device.read_reg(self.bench.device.regs.STREAM_STATIC_CUSTOM_SIZE):
        en = True
    return en

def embedded_packets(self, enable, datatype):
    self.bench.device.aci_islgen_embedded = {"en": enable,  # enable/disable
                                             "embedded_datatype": datatype,  # 0x10 for Image else 0x70 [70 -
                                             }

def get_embedded_pkt_info(self):
    en = False
    if not self.bench.device.read_reg(self.bench.device.regs.DEBUG_GENERIC_EMBEDDED_DATA_DISABLE):
        en = True
    return en

def frame_interrupt_packet(self, enable):
    if enable:
```

Figure32 Input aci.py file

(ii) Final Accessed. Readreg and. writeregRegisters:

UI	STREAM_CTX_0	EN_FRAME_INTERRUPT_PKT	GAIN_CHANGE_ENABLE	read-write 0x0	
UI	STREAM_CTX_0	EN_FRAME_INTERRUPT_PKT	INTEGRATION_TIME_CHANGE_ENABLE	read-write 0x0	
UI	STREAM_CTX_0	EN_FRAME_INTERRUPT_PKT	IMAGE_SIZE_CHANGE_ENABLE	read-write 0x0	
UI	STREAM_CTX_0	EN_FRAME_INTERRUPT_PKT	ACI_EN_FRAME_INTERRUPT_PKT	read-write 0x0	
UI	STREAM_CTX_0	ACI_EN_FRAME_ATTR_PKT	VALUE	read-write 0x0	ACCESSED ok
UI	STREAM_CTX_1	SUB_FRAME_SIZE_X	VALUE	read-write 0x0	
UI	STREAM_CTX_1	SUB_FRAME_LINE_LENGTH	VALUE	read-write 0x0	
UI	STREAM_CTX_1	GAIN	VALUE	read-write 0x0	
UI	STREAM_CTX_1	INTEGRATION_TIME	VALUE	read-write 0x0	
UI	STREAM_CTX_1	EN_FRAME_INTERRUPT_PKT	IMAGE_RESOLUTION_CHANGE_ENABLE	read-write 0x0	
UI	STREAM_CTX_1	EN_FRAME_INTERRUPT_PKT	GAIN_CHANGE_ENABLE	read-write 0x0	
UI	STREAM_CTX_1	EN_FRAME_INTERRUPT_PKT	INTEGRATION_TIME_CHANGE_ENABLE	read-write 0x0	
UI	STREAM_CTX_1	EN_FRAME_INTERRUPT_PKT	IMAGE_SIZE_CHANGE_ENABLE	read-write 0x0	
UI	STREAM_CTX_1	EN_FRAME_INTERRUPT_PKT	ACI_EN_FRAME_INTERRUPT_PKT	read-write 0x0	
UI	STREAM_CTX_1	ACI_EN_FRAME_ATTR_PKT	VALUE	read-write 0x0	ACCESSED ok
UI	STREAM_ISL	HEADER_DATA	VALUE	read-write 0x0	
UI	STREAM_ISL	FOOTER_DATA	VALUE	read-write 0x0	
UI	STREAM_ISL	CUSTOM_DATA	VALUE	read-write 0x0	
UI	DEBUG	PATGEN_FW_CTRL	VALUE	read-write 0x0	
UI	DEBUG	ISLGEN_FW_CTRL	VALUE	read-write 0x0	

Figure33 final accessed register from. aci

Chapter 5
HTMLPARSER

(i) Libraries Involved:

BeautifulSoupisaPythonlibraryforpullingdataoutofHTMLandXMLfiles.Itworkswithyour parser to provide idiomatic was navigating, searching ,and modifying the parse tree

SYNTAX:

```
from bs4 import BeautifulSoup

with open("index.html") as fp:
    soup = BeautifulSoup(fp)

soup = BeautifulSoup("<html>data</html>")
```

Figure34 ynatx.html

(ii) Working Flow:

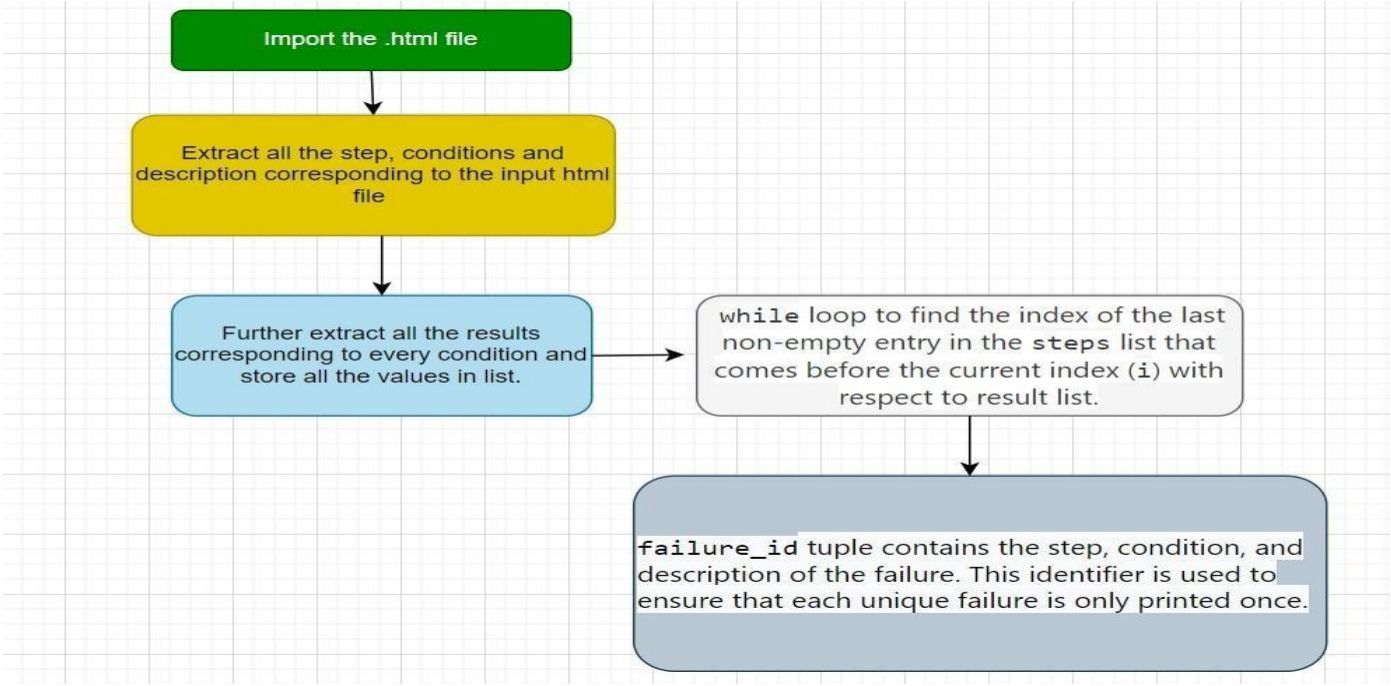


Figure35 Wokringflow.ht5.1.3

(iii) Input html file:

5		ACI source signals LFPS, to establish link common-mode voltage on all confgured lanes												
	Pass1													
		IF (LT_TPS1_DURATION != 0 AND LT_TPS2_DURATION != 0): ACI source signals LFPS for time TLFPS-LONG as defned in Table 5.5 of the ACI Specifcation	FAIL	TLFPS = 888ns (Out of limits) at offset 15	FAIL	LFPS is not detected	FAIL	LFPS is not detected	PASS	TLFPS-LONG = 74057ns	PASS	TLFPS-LONG = 74060ns	PASS	TLFPS-LONG = 74061ns
	Pass2													
		ACI source signals LFPS within limits of TLFPS-CYCLE and LFPS_DUTY_CYCLE, as defned in Table 5.5 of the ACI Specifcation	PASS		FAIL	LFPS is not detected after SLEEP	FAIL	LFPS is not detected after SLEEP	PASS		PASS		PASS	
6		ACI source stops signaling for TSILENCE, after completing TLFPS-LONG												
	Pass1													
		ACI source remains in TSILENCE for time defined in Table 5.5 of the ACI Specification	FAIL	TSILENCE = 125ns (Out of limits) at offset 2414	FAIL	LFPS is not detected after SLEEP	FAIL	LFPS is not detected after SLEEP	FAIL	TSILENCE = 129ns (Out of limits) at offset 254145741	FAIL	TSILENCE = 124ns (Out of limits) at offset 279217946	FAIL	TSILENCE = 124ns (Out of limits) at offset 277237586
END														
		Test result	FAIL		FAIL		FAIL		FAIL		FAIL		FAIL	

Figure36 input.html

(iv) CODE SNAPSHOTS:

```
from bs4 import BeautifulSoup

# Load the HTML file
with open('3_4_6_Dynamic_ALPM_Lane_Count_Change__FAIL.html', 'r') as f:
    html = f.read()

# Create a BeautifulSoup object
soup = BeautifulSoup(html, features: 'html.parser')

# Find the table element
table = soup.find("table")

# Extract the 0th index column (Step)
steps = [row.find_all("td")[0].text.strip() for row in table.find_all("tr")[2:]]

# Extract the 1st index column (Condition)
conditions = [row.find_all("td")[1].text.strip() for row in table.find_all("tr")[2:]]

# Extract the 2nd index column (Description)
descriptions = [row.find_all("td")[2].text.strip() for row in table.find_all("tr")[2:]]
```

```
if "fail" in result.lower():
    # Look back to find the corresponding step and condition
    step_index = i
    while step_index >= 0 and not steps[step_index]:
        step_index -= 1
    condition_index = i
    while condition_index >= 0 and not conditions[condition_index]:
        condition_index -= 1

    # Check if the step is 'END', if so, skip printing this condition result
    if step_index >= 0 and steps[step_index] == 'END':
        continue

    # Create a unique identifier for the current failure
    failure_id = (steps[step_index], conditions[condition_index], descriptions[i])

    # Check if the current failure has not been printed yet
    if failure_id not in printed_failures:
        # Print the step, condition, description, and test result
        print(f"Step: {steps[step_index]}, Condition: {conditions[condition_index]}, Description:
        # Add the current failure to the set of printed failures
        printed_failures.add(failure_id)
```

Figure37CodeSnapshots

(v) OUTPUT:

```
C:\Python37\python.exe "C:\Users\kashyaps\OneDrive - STMicroelectronics\Desktop\Summarized_docs_st\Final_html(merger) (2).py"
Step: 6, Condition: Pass6, Description: ACI source signals scrambled 0 Data Link Symbols during TPS2, Test Result: FAIL
Test Result: FAIL


Process finished with exit code 0
```

Figure38 Output

Chapter–6 Conclusion And Future Scope

6.1 Conclusion:

The control and capture platforms are connected on USB to a host PC. Each communication is sent through the USB interface, processed on the platform, and the result is sent back to the PC. This induces long response time and prevent any real-time action in response to an event. Scheduler is currently only available on ZCU104.
Porting on ZCU100 is planned… To address this limitation, a Scheduler is available on ASPIC and GPZ platforms. The scheduler executes a list of commands in response to a GPIO event. It is running on the GPZ R5real-time processor, to minimize latency between an event and the commands.

The ZCU104 board is well-positioned to meet the demands of future technological advancements. As the industry moves towards higher resolutions and more sophisticated AI and machine learning applications, the ZCU104 and its successors will continue to play a critical role in driving innovation.

In conclusion, the ZCU104 development board offers a comprehensive and powerful platform for a wide range of applications. Its combination of high-performance processing. As advancements in AI, communication, and embedded systems continue to evolve, the ZCU104 is poised to remain a key player in the development and prototyping of next-generation solutions. The ZCU104 is ideal for prototyping cutting-edge technologies inAI, machine learning, IoT, and high-speed communication, making it a valuable tool for engineers and researchers. The support for SDSoC Development Environment streamlines the development process, enabling efficient design, simulation, and deployment of applications.

The future scope of capture systems like the ZCU104 is vast and promising, driven by advancements in AI, communication technologies, and industry-specific applications. These systems will continue to evolve, offering more powerful, versatile, and solutions to meet the demands of a rapidly changing technological landscape, ongoing advancements aimed at supporting higher resolutions, faster refresh rates, enhanced audio capabilities, and better integration with emerging technologies. These improvements will continue to make HDMI a vital interface in both consumer electronics and professional applications.

The integration of ARM Cortex-R5 processors, and a MP2 GPU with FPGA capabilities provides a robust platform for complex computational tasks and real-time processing and Comprehensive Connectivity of the board's extensive connectivity options, including Gigabit Ethernet, USB 3.0, DisplayPort, PCIe, and SATA, ensure seamless integration with a variety of peripherals and external devices. Flexible Memory Options With 4GB of DDR4 memory, QSPI Flash, and SD card support, the ZCU104 offers ample storage and memory flexibility for diverse applications. The ZCU104 is ideal for developing and prototyping cutting-edge technologies in AI, machine learning, IoT, and high-speed communication, making it a valuable tool for engineers and researchers.

6.2 Future Scope:

High-Definition Multimedia Interface (HDMI) is a widely used interface for transmitting high-definition video and audio between devices. Since its introduction, HDMI has undergone several iterations, each improving upon the capabilities and performance of the interface. The future scope of HDMI, particularly in high-speed applications, is promising due to several factors:

1. Increasing Demand for Higher Resolutions and Refresh Rates

- **8K and Beyond**: As display technology advances, there is a growing demand & higher resolutions such as 8K and even 10K. HDMI 2.1 already supports 8K resolution at 60Hz, and future versions are expected to handle even higher resolutions and refresh rates.
- **High Dynamic Range (HDR)**: HDR technology enhances color accuracy of images. The future of HDMI will likely involve better support for HDR formats, providing a more immersive experience.

2. Enhanced Gaming Experiences

- **Variable Refresh Rate (VRR)**: HDMI 2.1 introduced VRR, which reduces screen tearing and stuttering ingames. As gaming becomes more advanced, HDMI will continue to evolve technologies, making gaming smoother and more responsive.
- **Quick Frame Transport (QFT)**: This technology reduces latency, which is crucial for gaming. Future HDMI versions will likely improve on QFT to provide even lower latency.

3. Enhanced Processing Capabilities

- **AI and Machine Learning**: With the rise of AI and machine learning, future capture systems will increasingly integrate AI capabilities directly into the hardware. This will enable faster datasets, real- time inference, and improved decision-making capabilities.
- **Edge Computing**: Capture systems will play a critical role in edge computing, where data is processed closer to centralized data center. The ZCU104 and its successors will support more powerful processors and advanced algorithms to handle complex tasks locally.

4. Advanced Communication Technologies

- **5G and Beyond**: The ongoing rollout of 5G networks and the development of 6G will require advanced capture systems to manage the increased data throughput, lower latency, connectivity. Future capture systems will need to support higher bandwidth and faster data rates.
- **Internet of Things (IoT)**: As the number of connected devices grows, capture systems like the ZCU104 will be crucial for managing and processing data from IoT devices. They will need to support a variety of communication protocols and handle diverse data streams efficiently.

5. Improved Data Security

- **Cybersecurity**: With increasing concerns over data security, future capture systems will incorporate advancedsecurity features such as hardware encryption, secure boot, and tamper detection to sensitive data.
- **Blockchain and Distributed Ledgers**: Integrating blockchain technology into capture systems could enhancedata integrity and security, ensuring that data captured and processed is immutable and traceable.

6. Improved Audio Capabilities

- **Higher Bit Rates**: As audio technology progresses, HDMI will need to support higher bit rate to deliver uncompressed, high-fidelity audio.

7. Connectivity and Integration

- **Unified Connectivity**: HDMI aims to provide a unified solution for video, audio, and data transfer. Futureversions will likely enhance integration with other interfaces and technologies, such as USB-C and Thunderbolt.
- **Internet of Things (IoT)**: As IoT devices become more prevalent, HDMI may evolve to support better connectivity and control over these devices.

REFERENCES

[1] Zynq UltraScale+ MPSoC Data Sheet: Overview (DS891)

[2] Zynq UltraScale+ MPSoC Technical Reference Manual (UG1085)

[3] Zynq UltraScale+ MPSoC Data Sheet: DC and AC Switching Characteristics (DS925)

[4]UltraScale Architecture PCB Design User Guide (UG583)

[5]UltraScale Architecture-Based FPGAs Memory IP LogiCORE IP Product Guide (PG150)

[6]UltraScale Architecture GTH Transceivers User Guide (UG576)

[7] UltraScale Architecture Gen3 Integrated Block for PCI Express LogiCORE IP Product Guide (PG156)

[8] Silicon Labs CP210x USB-to-UART Installation Guide (UG1033)

[7] Ter a Term Terminal Emulator Installation Guide (UG1036)

[8] Vivado Design Suite User Guide: Using Constraints (UG903) The following websites provide supplemental material useful with this guide

[9] Micron Technology: www.micron.com (MT40A256M16GE-075E, MT25QU512ABB8ESF-0SIT data sheets)

[10] Standard Microsystems Corporation (SMSC): www.microchip.com (USB3320 data sheet)

[11] SanDisk Corporation: www.sandisk.com
[12] SD Association: www.sdcard.org

[13] Silicon Labs: www.silabs.com/Pages/default.aspx (SI5341B, Si570, Si5319C, Si53340, CP2108 data sheets)

[14] Texas Instruments: www.ti.com/product/DP83867IR (TI DP83867 data sheet)

[15] PCI: https://pcisig.com/specifications/pciexpress/M.2_Specification/

[16] Maxim Integrated Circuits: https://www.maximintegrated.com

M. Tech Thesis

ORIGINALITY REPORT

7%	6%	4%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	discover.realvnc.com Internet Source	"Chapter 1 On the Use of System-
2	fliphtml5.com Internet Source	
3	www.scilit.net Internet Source	
4	blog.51sec.org Internet Source	
5	Submitted to Morgan Park High School Student Paper	
6	www.coursehero.com Internet Source	
7	Submitted to University of Rwanda Student Paper	
8	technodocbox.com Internet Source	
9	Xabier Iturbe, Didier Keymeulen, Patrick Yiu, Daniel Berisford, Robert Carlson, Kevin Hand, Emre Ozer.	

	1%
	1%
	1%
	1%
	1%
	<1%
	<1%
	<1%
	<1%

on-Chip Technology in Next-Generation Instruments Avionics for Space Exploration", Springer Science and Business Media LLC, 2016
Publication

10	doczz.net Internet Source	<1%
11	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
12	vn.element14.com Internet Source	<1%
13	www.ncbi.nlm.nih.gov Internet Source	<1%
14	Ton Duc Thang University Publication	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

STMicroelectronics Private Ltd.
Plot No.1, Knowledge Park-III, Greater
Noida – 201 308.
Uttar Pradesh, India Tel :
+91-120-4003000
Fax :+91-120-4003007
Email : infostm.india@st.com CIN :
U32109DL1990FTC039906
www.st.com

June 28, 2024

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Simran KASHYAP** has undergone internship in our **AMG** Group from **June 02, 2023 to June 28, 2024** and has successfully completed the project on: **High Speed Camera Interface Validation For Link Layer**

During the internship period, Simran KASHYAP was found to be sincere and professional in conduct.

We wish her all the best for the future endeavors.

for **STMicroelectronics Pvt. Ltd.**

Sanjay Kumar PIPLANI
India Talent Acquisition Head