**eCPRI Implementation for 4G/5G Mobile Network**

**ME ZG623T: Project Work**

**By**

**Selvaraj V**

**2018HT01537**

**Project work carried out at**

**Altiostar Network India Pvt Limited.**

****

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE PILANI (RAJASTHAN)**

**Dec 2020**

**eCPRI implementation of the 4G/5G Mobile Network**

**ME ZG623T: Project Work**

**By**

**Selvaraj V**

**2018HT01537**

**Project work carried out at**

**Altiostar Network India Pvt Ltd.**

Submitted in partial fulfillment of

M-Tech Embedded system degree programme

Under the Supervision of

Apparoa Paidi, Director Radio Design Head AltioStar Network India Pvt Ltd.

****

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE**

**PILANI (RAJASTHAN)**

**Dec 2020**

**CERTIFICATE**

This is to certify that the Dissertation entitled “**eCPRI implementation of the 4G/5G Mobile Network**” and submitted by “**Selvaraj V**” having ID-No. **2018HT01537** for the partial fulfillment of the requirements of M.Tech. **Embedded system** degree of BITS, embodies the bonafide work done by him/her under my supervision.



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of the Supervisor

Place : \_Bengaluru\_\_\_\_\_\_\_\_ Apparoa Paidi, Director of Radio Head design

Altiotar Networks India Pvt Ltd

Date : \_\_27/09/2020\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name, Designation & Organization &Location

Contents

[1) Introduction 7](#_Toc57749312)

[1.1. Mobile network 7](#_Toc57749313)

[1.2. Core network 7](#_Toc57749314)

[1.3. Radio head 7](#_Toc57749315)

[2) CPRI 8](#_Toc57749316)

[3) eCPRI 9](#_Toc57749317)

[1.1. Packet format: 10](#_Toc57749318)

[1.2. Message Type #4: Remote Memory Access 11](#_Toc57749319)

[1.3. FPGA architecture 12](#_Toc57749320)

[1.4. Hardware Component 12](#_Toc57749321)

[1.5. Software component 13](#_Toc57749322)

[4) Test bed 13](#_Toc57749323)

[1.1. Simulation Test bench 13](#_Toc57749324)

[1.2. Block diagram of the test jig 13](#_Toc57749325)

[5) Risk 14](#_Toc57749326)

[6) FSM 14](#_Toc57749327)

[7) Block diagram 16](#_Toc57749328)

[8) Simulator output 17](#_Toc57749329)

[1. 18](#_Toc57749330)

[8.1. epcri\_rx 18](#_Toc57749338)

[8.2. ecpri\_tx 18](#_Toc57749339)

[8.3. ram\_cpri\_packet 19](#_Toc57749340)

[8.4. ram\_cpri\_payload 19](#_Toc57749341)

[19](#_Toc57749342)

[8.5. ram\_recv\_eth\_packet 20](#_Toc57749343)

[9) Platform specific changes 20](#_Toc57749344)

[10) Critical decisions 20](#_Toc57749345)

[11) Conclusion 21](#_Toc57749346)

[12) Future activity 22](#_Toc57749347)

[13) Reference 22](#_Toc57749348)

[14) Software repository link 22](#_Toc57749349)

[Figure 1 4G -5G Network 7](#_Toc57549301)

[Figure 2 CPRI Specification 8](#_Toc57549302)

[Figure 3 CPRI IP payload 8](#_Toc57549303)

[Figure 4 eCPRI Packet Payload 10](#_Toc57549304)

[Figure 5 eCPRI Packet 10](file:///D:\mine\google_drive\notes\bits_wilp\sem_4\dissertation\final_report\2018HT01537-Midsem_Report_Dissertation.docx#_Toc57549305)

[Figure 6 eCPRI header 11](file:///D:\mine\google_drive\notes\bits_wilp\sem_4\dissertation\final_report\2018HT01537-Midsem_Report_Dissertation.docx#_Toc57549306)

[Figure 7 ZYNQ Architecture 12](file:///D:\mine\google_drive\notes\bits_wilp\sem_4\dissertation\final_report\2018HT01537-Midsem_Report_Dissertation.docx#_Toc57549307)

[Figure 8 eCPRI 13](file:///D:\mine\google_drive\notes\bits_wilp\sem_4\dissertation\final_report\2018HT01537-Midsem_Report_Dissertation.docx#_Toc57549308)

[Figure 9 Test Jig 13](file:///D:\mine\google_drive\notes\bits_wilp\sem_4\dissertation\final_report\2018HT01537-Midsem_Report_Dissertation.docx#_Toc57549309)

[Table 1 Abbrevation 6](#_Toc52129604)

[Table 2 Packet Identifier 10](#_Toc52129605)

Table 1 Abbreviation

|  |  |
| --- | --- |
| RRH | Remote Radio head |
| FTTA | Fiber To The Antenna |
| REC | Radio equipment controller |
| RE | Radio equipment |
| TDM | Time division Multiplexing |
| MME | Mobility Management Entity |
| S-GW | Signaling Gateway |
| PDN-GW | Packet Data Network Gateway |
| PCRF | Policy and Charging Rules Function |
| eNodeB | Enhanced Node Base station |
| UPF | User plane function |
| SMF | Session Management function |
| PCF | Policy control function |
| AMF | Access and Mobility function |
| CPRI | Common Public Radio Interface |
| eCPRI | Ethernet Common Public Radio Interface |
| GPS | Global Positioning system |
| PTP | Precision time protocol |
| LTE | Long Term Evolution |
| UMTS | Universal Module Telecommunication Service |
| GSM | Global System for Mobile Communication |
| IQ | In phase & Quadrature phase samples |
| C&M | Control & Management |
| AxC | Antenna Carrier |
| HDLC | High Level Data link Control |
| 4G | Fourth Generation network |
| 5G | Fifth Generation network |

# Introduction

## Mobile network

Mobile networks are complex and consist of many network entities. These entities then fall under the core network and radio part of the network. Together, the radio and the core networks connect the overall mobile network to the mobile phone and external networks.

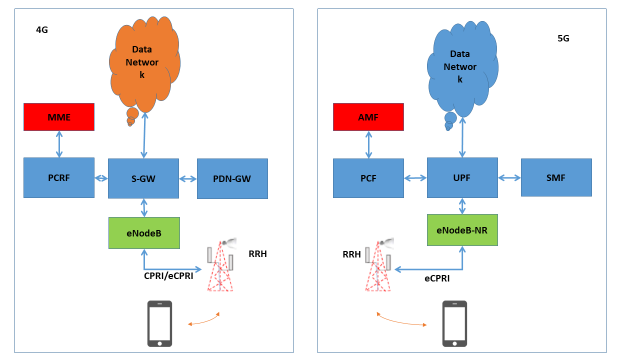


Figure 1 4G -5G Network

## Core network

**The mobile core network is a central part of the mobile network which makes it possible for subscribers to get access to the services they are entitled to. It is responsible for critical functions such as subscriber profile information, subscriber location, authentication of services and the necessary switching functions for voice and data session.**

## Radio head

A remote radio head (RRH), also called a remote radio unit (RRU) in wireless networks, is a remote radio transceiver that connects Base station with the UE (User equipment). Fourth generation (4G) and beyond infrastructure deployments will include the implementation of Fiber to the Antenna (FTTA) architecture.

FTTA architecture has enabled lower power requirements, distributed antenna sites, and a reduced base station footprint than conventional tower sites. The use of FTTA promotes the separation of power and signal components from the base station and their relocation to the top of the tower mast in a Remote Radio Head (RRH). The connection between the Base station is via CPRI cable

# CPRI

It is a digital interface standard to transport antenna samples between a Radio Equipment (RE) and a Radio Equipment Control (REC) base station performing the digital processing of these signals. Antennas signals are interleaved in a TDM fashion supported by a constant Bit Rate transport solution for each generation of the network. It provides time and synchronization information for the Radio Air Interface.

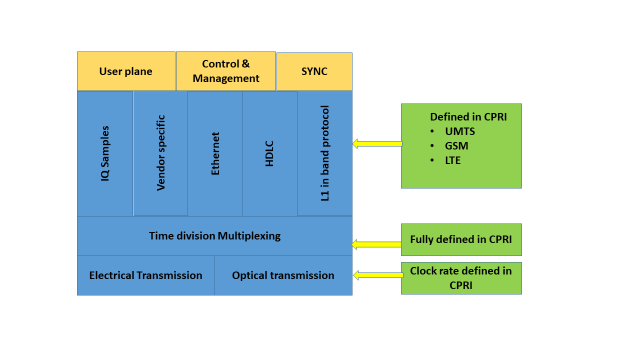


Figure 2 CPRI Specification

The initial version of specification was defined for point to point topology. The specification assumed no intermediate nodes for defining max latency. The network management was left to the application layer. Interoperability between different RRH was limited to the low layers which was covered by the specification. Hence it can be understood that CPRI specification defined how to exchange the radio signal data not the data content itself nor the associated management plane.

The CPRI frame support different type of data as payload and the Ethernet data is part of the fast C&M payload.

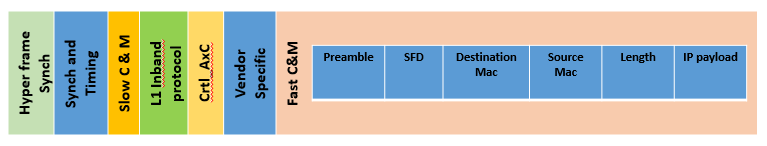


Figure 3 CPRI IP payload

The CPRI frame support at the C&M has a complicate method to support higher layer. The clock synchronization between REC and RE has to be sent in the CPRI payload, it is sent as part of the synchronization header via 8b10b high speed transceiver coding, therefore CPRI is a fully synchronized system.

The clock can be obtained via GPS or via PTP. The GPS requires clear sky for good signal reception, since CPRI requires very high resolution the REC unit must be near a good GPS source.

In addition to the above mentioned point, there is no security for the IQ sample being transported over the CPRI cable. Hence in the future deployment of the 4G and 5G network, eCPRI specification will be used to transport IQ samples. It uses Ethernet transport so can be mixed with other traffic and doesn’t need to be fully synchronized. CPRI works as point to point, eCPRI is switchable and routable. The disadvantage is there is no guaranteed delivery or bandwidth.

# eCPRI

eCPRI is packet based front haul interface developed by CPRI forum for the 4G and 5G network. It has the same level of interoperability as CPRI. Ethernet/IP networking, synchronization and security uses existing standards for end to end connection between REC and RE.

eCPRI does not constrain the use of specific network and data link layer protocols to form the network. Any type of network can be used for eCPRI, provided eCPRI requirements are fulfilled. The standard is defined to comply with the requirements associated with stringent radio technologies features in terms of:

* Timing and frequency accuracy
* Bandwidth capacity,
* Latency,
* Packet loss,

In case of eCPRI, messages shall be transmitted in standard Ethernet frames. There are two tyes of the packets format. In the first type the eCPRI is udp payload and there is no unique identifier. The application will have a port which is mapped for eCPRI payload. In the second type, there will be unique identifier in the Ethernet packet type filed, eCPRI Ethertype (AEFE). Current implementation will focus on the second type of packets.

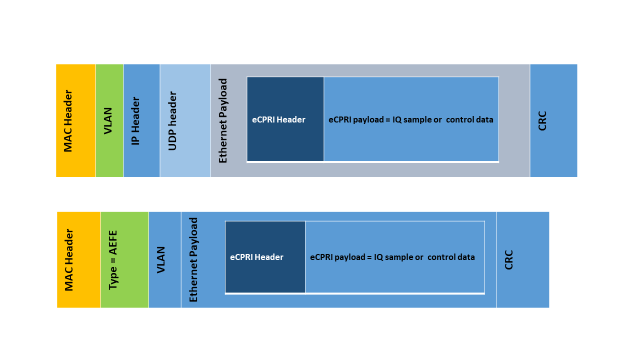


Figure 4 eCPRI Packet Payload

## Packet format:

The eCPRI packet consist of header and a associate payload. The header length is of 4 bytes and the payload can be of length 65534 bytes as show in the diagram below

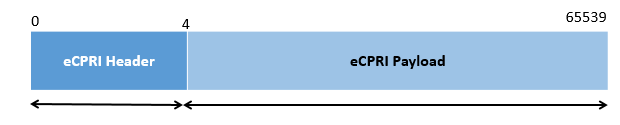


Figure 5 eCPRI Packet

The header consists of the eCPRI protocol version, message type and payload size as show in the diagram below

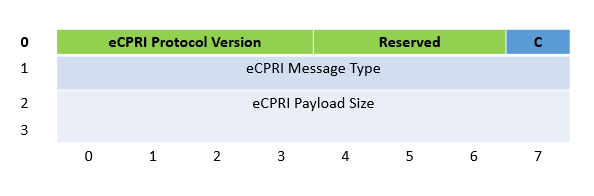


Figure 6 eCPRI header

The following are different message types that are defined in the specification

|  |  |
| --- | --- |
| **Message Type #** | **Name** |
| 0 | IQ Data |
| 1 | Bit Sequence |
| 2 | Real-Time Control Data |
| 3 | Generic Data Transfer |
| 4 | Remote Memory Access |
| 5 | One-way Delay Measurement |
| 6 | Remote Reset |
| 7 | Event Indication |
| 8–63 | Reserved |
| 64 –255 | Vendor Specific |

Table 2 Packet Identifier

## Message Type #4: Remote Memory Access

For the project the message type 4 will be implemented which is remote memory access procedure.

The message type ‘Remote Memory Access’ allows reading or writing from/to a specific memory address on the opposite eCPRI node. The service is symmetric i.e. any “side” of the interface can initiate the service.

A read or write request/response sequence is an atomic procedure, a requester needs to wait for the response from the receiver before sending a new request to the same receiver. A write request without response is also defined, this procedure is a one message procedure.

## FPGA architecture

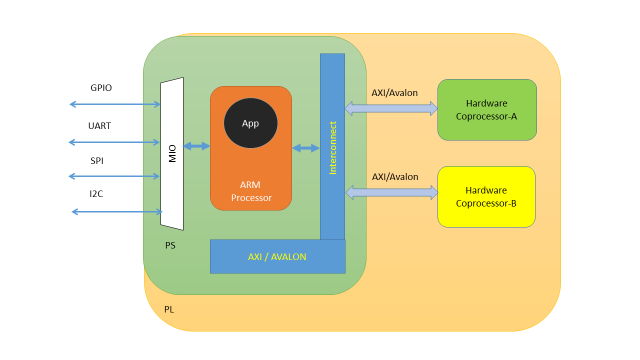
Xilinx Zynq processor is the hardware on which eCPRI logic will be tested for simulating the real time to understand the additional complexity. This processor has PS and PL section, where PL is the user programmable logic section and PS is the processing system which is an ARM processor. This architecture helps in fast development of the required design since the hardware interfacing between processor and FPGA is not required hence reducing the development lifecycle and board foot print.

Figure 7 ZYNQ Architecture

## Hardware Component

The hardware components will be made implemented in the PL programmable logic unit. The implementation will be done in Verilog language, which is a hardware description language. The implementation will involve the logic to parse the eCPRI packet and existing hardware IP with which the implemented logic will be integrated. The logic will have FIFO & RAM buffer for buffering the incoming IP packet, eCPRI logic to separate the user defined data and provide AXI bus interface to allow access from the processor.

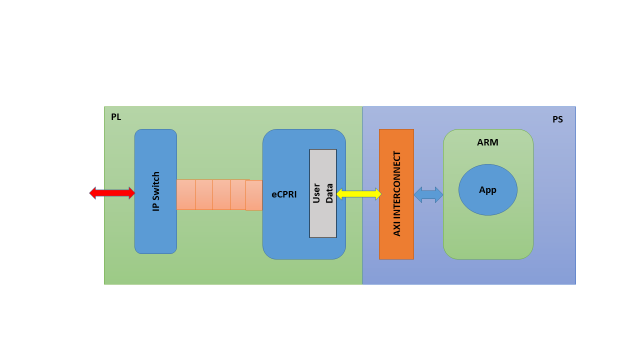


Figure 8 eCPRI

## Software component

The software component will a single application which reads from the memory region mapped PL region. The application will be implemented in ‘c’ language and it will be running on Linux OS.

# Test bed

## Simulation Test bench

The test bench for testing the eCPRI logic will be implemented in Verilog and it will not be tested on the processor. PCAP file will be used as input test vector, file operation support has to be added to the test bench. Different pcap file can be be generated using software packet generator.

## Block diagram of the test jig

The keysight eCPRI packet generator will be used for generating packet of message ID 4. The packet payload parsing will be tested but the payload parameter will not be key parameter since it does not have any significant benefit for this proof of concept implementation

Figure 9 Test Jig

# Risk

The following are the known risk

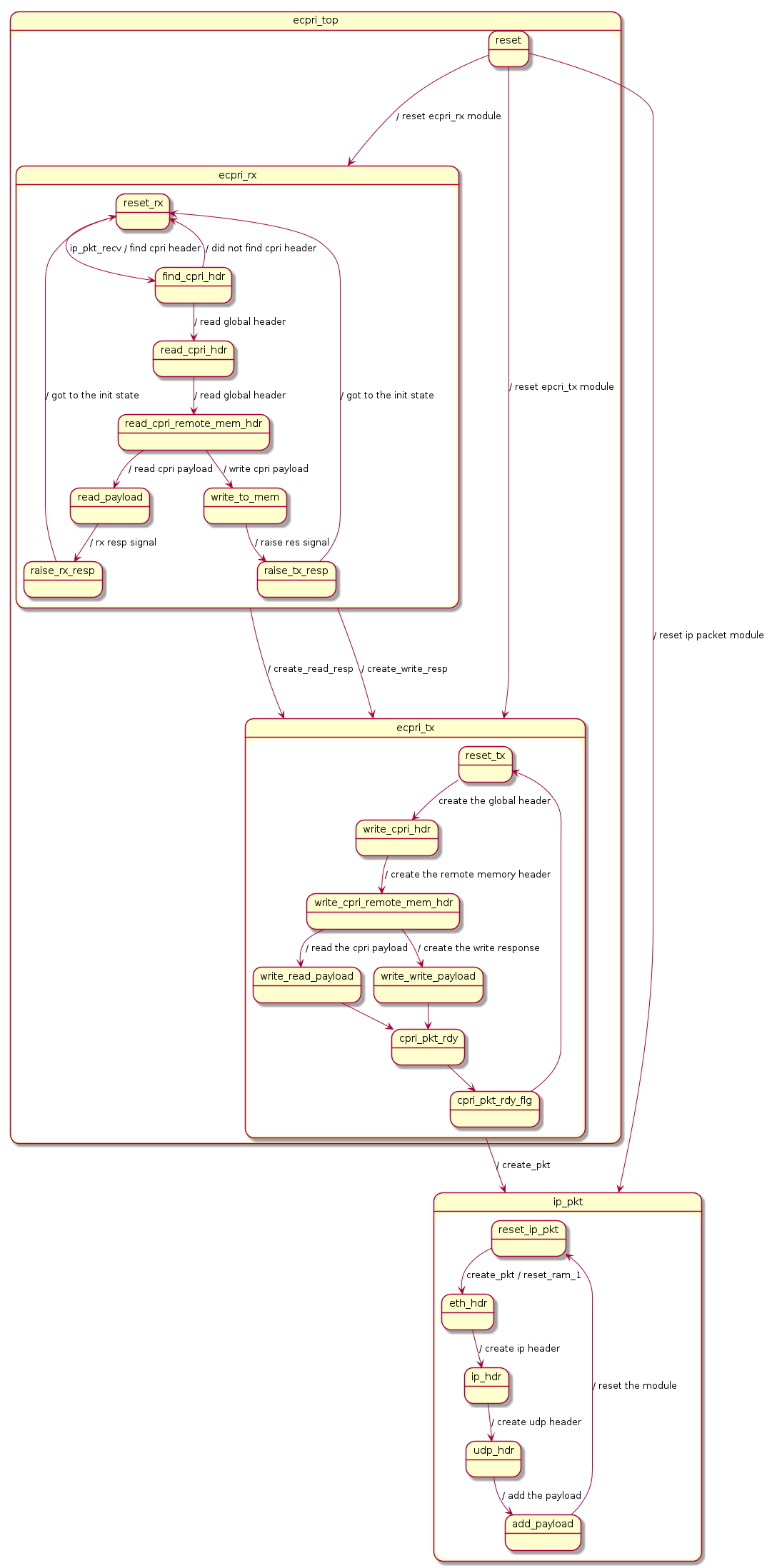
1. Complexity of the AXI bus integration is not known
2. Generating packets for simulation
3. Test Jig configuration is expected to be done in few steps
4. OS bring up on the Zynq platform for the software application to run

# FSM

The FSM is design to two primary segments as follows

1. ecpri\_rx : The FSM supports
   1. Initial parser which validates the incoming packet
   2. Provide trigger point for raising signals to the ecpri\_tx for sending read response or write response
   3. Define the trigger point for writing copies the payload to the memory location which can be accessed by the software application
2. ecpri\_tx: The FSM supports
   1. Action to be taken when signals are received the ecpri\_rx module
   2. Define the sequence of the operation for creation of the headers for the creation of the response packet
   3. Define the trigger point for the read operation

A secondary FSM “ip\_pkt” has been added for creation of the IP packet. This FSM is not required with the second type of the packet.



# Block diagram

The block represents various connections between the logic fabric and the storage buffer.

There are four RAM modules whose functions are as below

1. ram\_recv\_eth\_packet : This ram block is used for storing the received packets
2. ram\_cpri\_payload : This ram is for writing the payload when write requesst is received and used for reading the payload when read request is received
3. ram\_cpri\_packet : This ram block is received for the creation of the cpri response packet which include storing the header and and the payload.
4. ram\_eth\_packet\_hdr : This ram block is used for creation of the ip packet. This module will be simulated in the test bench.

There are two primary logic block

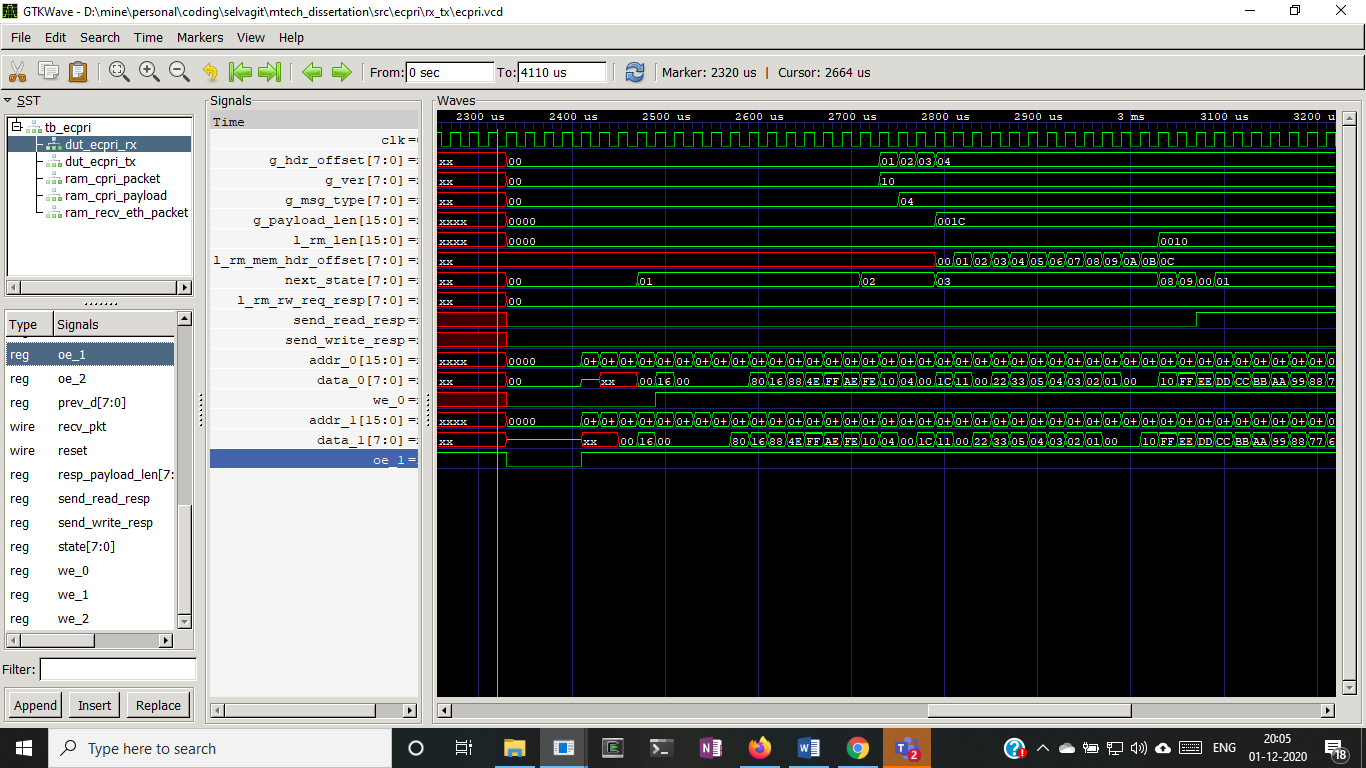
1. ecpri\_rx :
   1. It receives the packets from the switch
   2. Parser the packets for eCPRI header
   3. Find the request type
   4. Copy the payloads to the user accessible memory “ram\_cpri\_payload” for write request
   5. Raise signal to the “ecpri\_tx” for creating the response packet
2. ecpri\_tx:
   1. Receives read and write response signal from ecpri\_rx module
   2. Create ecpri response packet header
   3. Copy the payload for the write response

The secondary ip blocks are simulated via the test bench

1. IP switch : This block read the the pcap file and write the Ethernet packet to the ram\_recv\_eth\_packets
2. IP\_pkt : This is abstracted for duming the cpri packets.

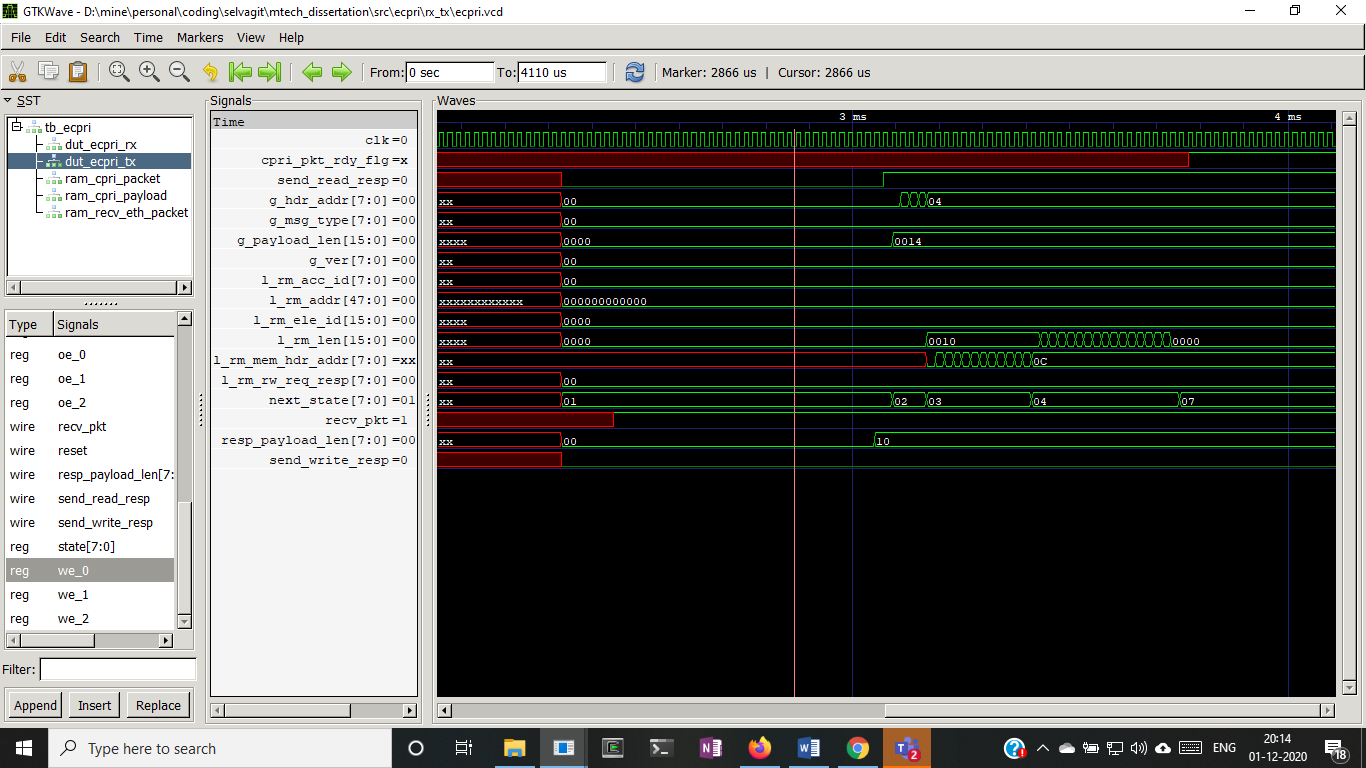
# Simulator output

The internal of the each modules are presented in the waveform captures for each modules

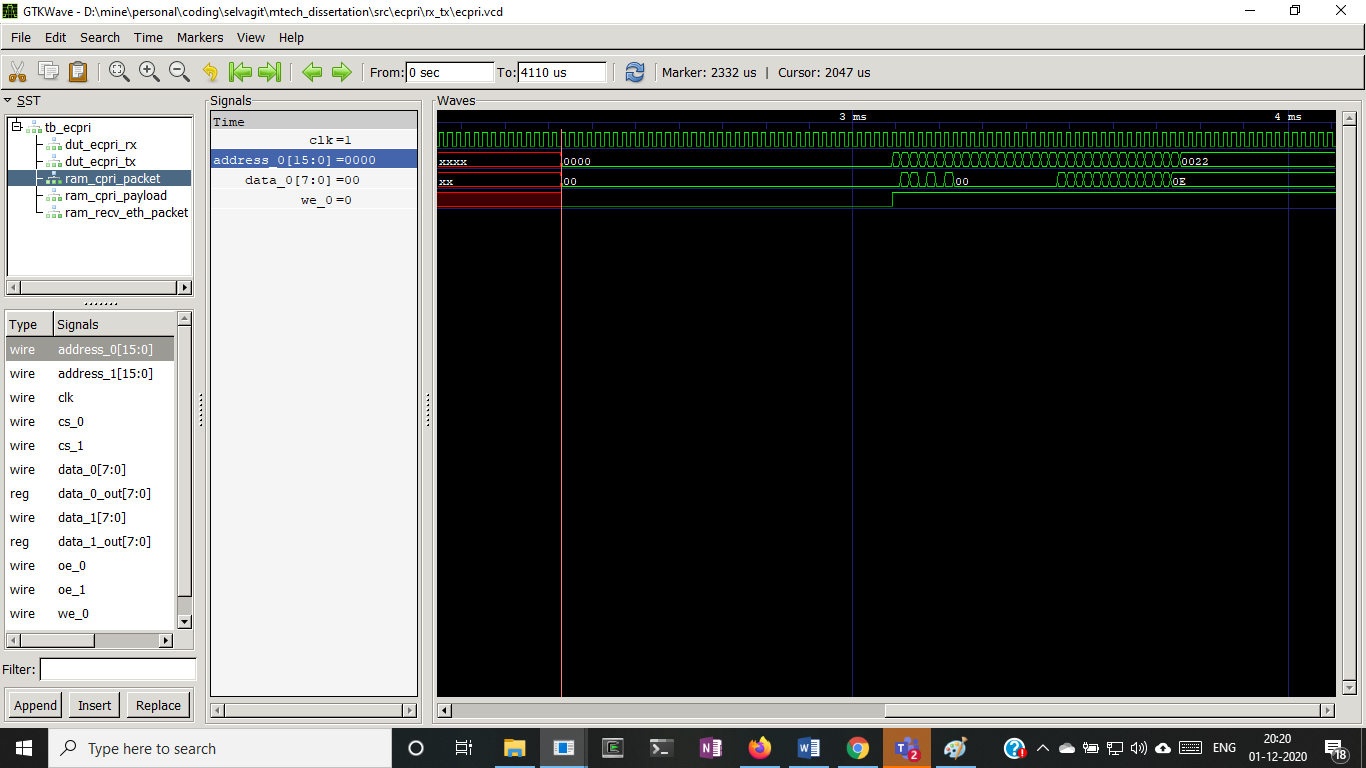
1. ****

## epcri\_rx

## ecpri\_tx

****

## ram\_cpri\_packet

****

## ram\_cpri\_payload

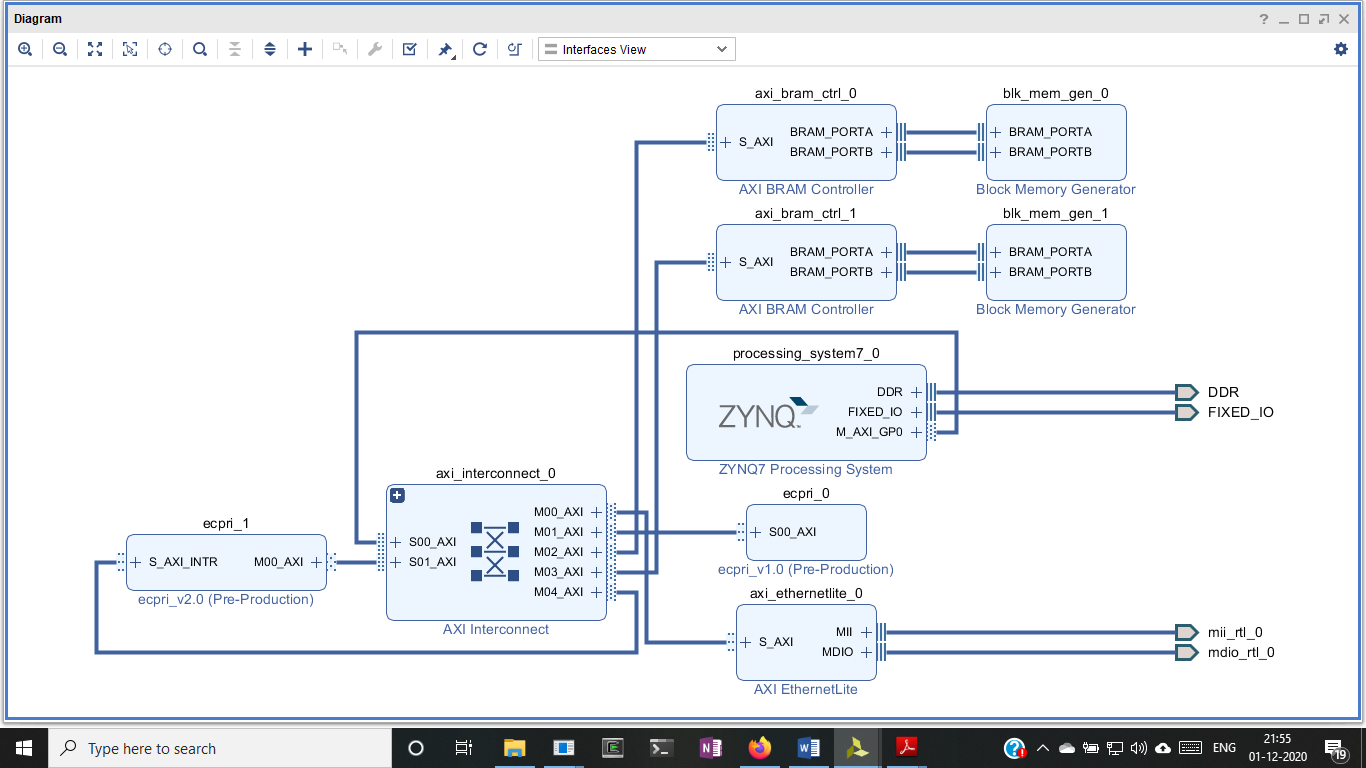
## 

## ram\_recv\_eth\_packet

# 

# Platform specific changes

Zynq is the hardware platform which has been choosen for the testing the eCPRI logic. AXI bus support has to be added the eCPRI module. AXI lite version has been choosen and it support single word read operation but does not support burst mode of operation.



# Critical decisions

The following are the design decision working on the projects

1. Verilog is a hardware description language, does not extensive support for the file operation. Pcap files was used as input and it is a file format, it will have it own header. The file header is not required for the ecpri verilog logic. The logic to parse the header was implemented via system Verilog which has support new data type similar to type struct used in “c”. This effort in learning system Verilog was not considered during the initial stage of the project.
2. The test platform for generating packets was not available during the initial stages of the project, therefore the packet generator was written using the python scapp library.
3. Iverilog was an open source Verilog compiler used for compiling. Since the compiler does not support all the syntax of the system Verilog, test bench took more time to write using the basic feature
4. To reduce the development effort, during design using company’s ip code was considered. However on further study on the scope of the activity following significant challenges was identified
   1. The company code base was written in VHDL, while the current design was implemented using the Verilog (Sytem Verilog). Therefore the design could not be tested with open source tools since the tools did not support mix language testing.
   2. The IP code base design of the company uses Avalon bus. However Avalon bus was not support in Vivado tool.
   3. The based on the above observation, this design approach was not taken
5. The during the integration of the modules on the following changes the following design changes were made
   1. Vivado tool supports AXI interconnect; the axi bus support was added. This added axi bus verification effort
   2. The modules where written to support 8 bit data but the axi supports only 32 bit data. Since the 32 bit data support was required the internal of the parsing, the packet has to be completely re-written.
   3. The code designed in the simulator had three port for memory access to three different ram modules. However with the AXI modules single port has to be used for accessing memory across two RAM modules
   4. During the simulator based development it was assumed the eth packets will be forward to the eCPRI modules ram without any software component. With the inclusion of the AXI bus the packets from the Ethernet ip has to be read by the software application and then put into the eCPRI ram.
   5. With AXI bus AXI Ethlite module was added. The module receives and transmits Ethernet packet. To enable the capture of the packets software application must reconfigure the modules.
   6. With AXI interface the software complexity has increased to support configuration of the Ethernet interface, reading the ethernet packet and putting on the ecpri ram, reading the ecpri packet and pushing it to eth module

# Conclusion

The project was an R&D project to design of the eCPRI modules. The design for the eCPRI was proven in the simulator world. However the design had to under go significant changes to support the hardware platform. The design changes based on the inter connect was significantly under estimated.

Even though the primary goal of the eCPRI logic implementation was completed, because of the design changes made in the later part of the schedule, the secondary goal of the implementing the eCPRI support on the FPGA could not be completed.

# Future activity

The following are the possible options for improvement in the projects

1. Interrupt mapping to each modules can be added to increase the performance of the FPGA logic
2. Direct transfer to the from the Ethernet module to the eCPRI can be done by adding DMA support to the AXI bus.
3. Strong verification test bench has to be built for the eCPRI modules

# Reference

The literature below represents the initial base for the implementation of eCPRI functional. Since the project is implementation of the eCPRI specification, it is also presented below

1. Antonio de la Oliva, Jos´e Alberto Hern´andez, David Larrabeiti, Arturo Azcorra. An overview of the CPRI specification and its application to C-RAN based LTE scenarios
2. Divya Chitimalla1, Koteswararao Kondepu2, 3, Luca Valcarenghi2, Massimo Tornatore1and Biswanath Mukherjee. 5G Fronthaul – Latency and Jitter Studies of CPRI over Ethernet
3. Juan Camilo Vega, Qianfeng (Clark) Shen, Alberto Leon-Garcia, Paul Chow. Introducing ReCPRI: A Field Re-Configurable Protocol for Backhaul Communication in a Radio Access Network
4. eCPRI Transport Network V1.2 (Common Public Radio Interface: Requirements for the eCPRI Transport Network: 2018-06-25)
5. eCPRI Specification V2.0 (Common Public Radio Interface: eCPRI Interface Specification: 2019-05-10)
6. CPRI Specification V7.0 (2015-10-09), Common Public Radio Interface (CPRI); Interface Specification
7. Glen Gibb, George Varghese, Mark Horowitz, Nick McKeown Stanford University Microsoft Research {grg, horowitz, nickm}@stanford.edu [varghese@microsoft.com](mailto:varghese@microsoft.com); Design Principles for Packet Parsers

# Software repository link

The following git hub link will provide the history of the software development activities for the dissertation <https://github.com/selvagit/mtech_dessertation>.