

ELECTRONIC SYSTEM FOR THE ALICE HIGH RATE DETECTOR UPGRADE

By

JUBIN MITRA

ENGG04201304001

VARIABLE ENERGY CYCLOTRON CENTRE, KOLKATA

*A thesis Submitted to the
Board of Studies in Engineering Sciences*

*In partial fulfillment of requirements
for the Degree of*

DOCTOR OF PHILOSOPHY

of

HOMI BHABHA NATIONAL INSTITUTE



June, 2018

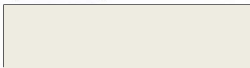


Homi Bhabha National Institute

Recommendations of the Viva Voce Committee

As members of the Viva Voce Committee, we certify that we have read the dissertation prepared by **Jubin Mitra** entitled “**Electronic system for the ALICE high rate detector upgrade**” and recommend that it may be accepted as fulfilling the thesis requirement for the award of Degree of Doctor of Philosophy.

Chairman - Dr. Sudhee Ranjan Banerjee (VECC)  Date: 27/6/2018

Guide / Convener - Dr. Tapan Kumar Nayak (VECC)  Date: 27.6.2018

Examiner – Dr. Sant Sharan Pathak (IIT Kharagpur)  Date: 27/6/2018

Member 1- Dr. Sarbajit Pal (VECC)  Date:

Member 2- Dr. Sukalyan Chattopadhyay (SINP)  Date: 27/6/18

Technology Adviser - Dr. Alex Kluge (CERN)  Date: 27.6.2018

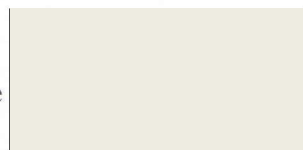
Final approval and acceptance of this thesis is contingent upon the candidate's submission of the final copies of the thesis to HBNI.

I hereby certify that I have read this thesis prepared under my direction and recommend that it may be accepted as fulfilling the thesis requirement.

Date: 27.6.2018

Place: KOLKATTA

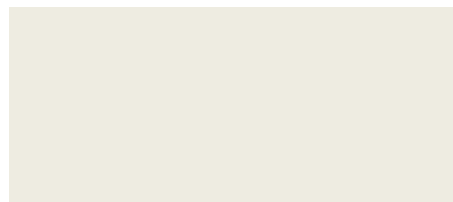
Guide



STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at Homi Bhabha National Institute (HBNI) and is deposited in the Library to be made available to borrowers under rules of the HBNI.

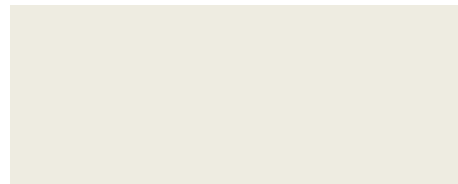
Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the Competent Authority of HBNI when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.



(**Jubin Mitra**)

DECLARATION

I, hereby declare that the investigation presented in the thesis has been carried out by me. The work is original and has not been submitted earlier as a whole or in part for a degree / diploma at this or any other Institution / University.



(**Jubin Mitra**)

List of Publications

Refereed Journal Publications

1. Mitra, Jubin, and Tapan K. Nayak. "[An FPGA-Based Phase Measurement System.](#)" **IEEE Transactions on Very Large Scale Integration (VLSI) Systems** vol. 26, no. 1, pp. 133-142, Jan. (2018).
2. Mitra, Jubin, and Tapan Nayak. "[Reconfigurable very high throughput low latency VLSI \(FPGA\) design architecture of CRC 32.](#)" **Integration, the VLSI Journal** 56 (2017).
3. Mitra, Jubin for ALICE Collaboration "[Common Readout Unit \(CRU\)-A new read-out architecture for the ALICE experiment.](#)" **Journal of Instrumentation** (2016).
(Conference related publication)
4. Mitra, Jubin, et al. "[GBT link testing and performance measurement on PCIe40 and AMC40 custom design FPGA boards.](#)" **Journal of Instrumentation** (2016).
(Conference related publication)
5. Mitra, Jubin, et al. "Fixed and deterministic phase study on Si5344 PLL with Temperature variation for LHC experiments" **Journal of Instrumentation** (under preparation)
6. Mitra, Jubin, et al. "CRU as Trigger and Timing distribution unit in ALICE experiment at RUN3 using TTC-PON GBT Bridge." **Nuclear Instruments and Methods in Physics Research A** (under review) ([arXiv:1806.01350](#))

Technical Note / Test Report (As Primary Author)

1. [Preliminary jitter measurement test setup to study the effect of CRU on LHC forward clock quality](#) (ALICE CRU Internal Technical Note)

2. [SI5344 PLL - Skew and Jitter Variation Study](#) (CERN Internal Technical Note)
3. [TTC PON'15 Implementation for Arria10 FPGA on CRU](#) (ALICE CRU Team Internal Note)
4. [PCIe40 communication testing with GBT versatile link demo board](#) (TEST REPORT)
5. [GBT Firmware Implementation for Arria10 FPGA on PCIe40 board](#) (User Manual)
6. [GBT BER measurement & Transceiver Optimization](#) (Internal Technical Note)
7. [GBT Firmware Implementation for Arria10 FPGA on PCIe40 board](#) (ALICE CRU Internal Document)
8. [Phase noise measurement of the forwarded clock from VLDB](#) (Test Report)

Conference Proceedings

1. J. Mitra and T. K. Nayak, “[Design of Data Aggregation Unit for High Energy Physics Experiment](#)”, **IEEE NSS Conference (Presenter)**
2. J. Mitra et.al, “[Error Resilient Secure Multi-Gigabit Optical Link Design for High Energy Physics Experiment](#)”, 29th IEEE International Conference on VLSI Design (**IEEE VLSID**), 2016, Kolkata, India (**Presenter**) (Acceptance Ratio: <12%)
3. J. Mitra, S. Mandal, S. A. Khan, J. Saini, S. Chattopadhyay and T. K. Nayak, “[Electronic Data Aggregation Architecture for High Energy Physics Big data taking Experiments](#)”, *In Proceedings of the DAE Symp. on Nucl. Phys* (Vol. 61, p. 1086), (2016) (**Presenter**)
4. J. Mitra and T. K. Nayak, “[Reconfigurable Concurrent VLSI \(FPGA\) Design Architecture of CRC-32 for high-speed data communication](#)”, The IEEE International

Symposium on Nanoelectronic and Information Systems (**IEEE INIS**), 2015 Indore, India (**Presenter**)

5. J. Mitra et. al, “[GBT Link Testing and Performance Measurement on PCIe40 and AMC40 Custom Design FPGA Boards](#)”, Tropical Workshop on Electronics for Particle Physics (**TWEPP**) 2015, Lisbon, Portugal (**Presenter**)
6. J. Mitra for ALICE-INDIA Collaboration, “[Common Readout Unit \(CRU\) - A new readout architecture for ALICE experiment](#)”, Tropical Workshop on Electronics for Particle Physics (**TWEPP**) 2015, Lisbon, Portugal (**Presenter**)
7. J. Mitra, S. A. Khan and T. K. Nayak, “[Common Readout System in ALICE](#)”, 7th International Conference on Physics and Astrophysics of Quark Gluon Plasma (**ICPAQGP**) 2015, Kolkata, India, **Proceeding of Science (PoS)** (**Presenter**)
8. J. Mitra , S. A. Khan, T. K. Nayak, T. Kiss and E. David, “[Testing and Performance Analysis of 10 Gigabit Ethernet on Altera Stratix[®] V FPGA](#)”, *In Proceedings of the DAE Symp. on Nucl. Phys* (Vol. 59, p. 884), (2014) (**Presenter**)

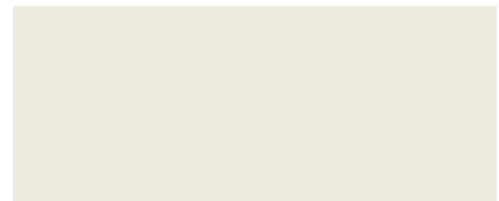
ALICE Technical Design Report (Collaborative Work)

1. “[Technical Design Report for the Upgrade of the Online - Offline Computing System](#)”
2. “[Technical Design Report for the Upgrade of the ALICE Read-out & Trigger System](#)”
3. “[Technical Design Report for the Upgrade of the ALICE Time Projection Chamber](#)”
4. “[Technical Design Report for the Muon Forward Tracker](#)”

Other Publications as Non-Primary Author

1. S. Mukherjee, F. Costa, R. Paul, A. Chakrabarti, S.A. Khan, **J. Mitra**, T. Nayak, “[An efficient approach to manage DMA descriptors and evaluate PCIe based DMA performance for ALICE Common Readout Unit \(CRU\)](#)”, Proceedings of the National conference on Advanced Detectors for Nuclear, High Energy and Astroparticle Physics
2. S. Mukherjee, F. Costa, R. Paul, A. Chakrabarti, S.A. Khan, **J. Mitra**, T. Nayak, “[Implementation and evaluation of custom logic for various configuration schemes based on I2C and HDLC protocol for ALICE Common Readout Unit\(CRU\)](#)”, Proceedings of the National conference on Advanced Detectors for Nuclear, High Energy and Astroparticle Physics
3. S. A. Khan, **J. Mitra**, “[Development of a high speed data acquisition system for the detectors at high luminosity LHC](#)”, *In Proceedings of the XXII DAE-BRNS SYMPOSIUM*, (2016)
4. R. Paul, A. Chakrabarti, **J. Mitra**, S. A. Khan, S. Mukherjee and T.K. Nayak, “[Channel Processor in 2D Cluster Finding Algorithm for High Energy Physics Application](#)”, *In Proceedings of the DAE Symp. on Nucl. Phys* (Vol. 61, p. 1046), (2016)
5. S. A. Khan, F. Costa, E. David, **J. Mitra**, T. Kiss, S. Mukherjee, R. Paul, T. K. Das, A. Chakrabarti and T. K. Nayak, “[Implementation of I2C bus master controller for CRU Slow Control in ALICE at LHC](#)”, *In Proceedings of the DAE Symp. on Nucl. Phys* (Vol. 61, p. 1070), (2016)
6. S. Mukherjee, F. Costa, R. Paul, A. Chakrabarti, S.A. Khan, **J. Mitra**, T. Nayak, “[An efficient approach to evaluate PCIe DMA design and DMA performance for](#)

- [Common Readout Unit\(CRU\)](#)”, *In Proceedings of the DAE Symp. on Nucl. Phys* (Vol. 61, p. 1078), (2016)
7. S. A. Khan, **J. Mitra**, T. K. Nayak, “[GBT link testing and performance measurement on Altera Stratix-V FPGA](#) ”, *In Proceedings of the DAE Symp. on Nucl. Phys* (Vol. 60, p. 1052), (2015) (Presenter)
8. S.A. Khan, **J. Mitra**, T. K. Nayak, T. Kiss, E. David, A. Kluge, “[Common Readout Unit - A New Readout Architecture for the ALICE experiment at the CERN-LHC](#)”, *In Proceedings of the DAE Symp. on Nucl. Phys* (Vol. 59, p. 884), (2014)
9. S.A. Khan, **J. Mitra**, T. K. Nayak, T. Kiss, E. David, “[Design and. Simulation of 10 Gigabit Ethernet on Altera FPGA](#)”, *In Proceedings of the DAE Symp. on Nucl. Phys* (Vol. 59, p. 884), (2014)
10. V. Jain, J. Saini, A.K. Dubey, S. Chattopadhyay, R. Ravishankar, T. Bandyopadhyay, S. Mandal, **J. Mitra**, A. Saha, “[Radiation test of active LVDB component with 60Co Gamma chamber and Neutron irradiation at VECC Cyclotron](#)” , *In Proceedings of the DAE Symp. on Nucl. Phys* (Vol. 59, p. 884), (2014)



(**Jubin Mitra**)

DEDICATED TO

My beloved parents

Mr. Dipendra Kali Mitra and Mrs. Bharati Mitra

ACKNOWLEDGMENT

My deep gratitude goes first to **(Prof.) Dr. Tapan Kumar Nayak**, who expertly guided me through my doctoral research and who shared the excitement of engineering design for 4 years. His unwavering enthusiasm for physics kept me constantly engaged with my research, and his personal generosity helped make my time at VECC enjoyable.

My sincere appreciation extends to my technical advisor **Dr. Alexander Kluge** and CRU design co-ordinator **Tivadar Kiss** for their mentoring, encouragement and early insights into the research work. I really appreciate the company of the VECC colleagues **Shuaib Ahmad Khan, Jogender Saini, Singaraju Rama Narayana** and **Partha Bhaskar** for nurturing my research acumen. I am extremely grateful to my entire CRU team members due to whom this journey is made possible within four years. As part of my research work, I have worked in multiple laboratories, without whom my research would not have matured. From CERN electronics team I would like to thank **Dr. Sophie Baron**; from CPPM lab **Jean Pierre Cachemiche**; from Wigner Research Institute **Erno David, Josef Imrek** and from CERN DAQ team **Filippo Costa**. I would also like to thank **Vaibhav Baid**, the director of Berylline Labs for untiring support. I am obliged to my committee members for letting my defense an enjoyable moment, and for their brilliant comments and suggestions.

A special thanks to my parents for all their sacrifices and prayers that made me sustained thus far. I would also like to thank all of my friends who supported me in writing, and giving impetus to strive towards my goal.

CONTENTS

Contents	Page No.
CONTENTS	xii
SYNOPSIS	xix
LIST OF ABBREVIATIONS	xxvii
LIST OF FIGURES	xxxiv
LIST OF TABLES	xlii
Chapter 1: Introduction	1
1.1 The Large Hadron Collider (LHC) of CERN	4
1.2 The ALICE experiment	9
1.3 ALICE Detector Upgrade	10
1.4 ALICE Technical Upgrade	12
1.5 Evolution of the ALICE Detector Data Readout Cards	13
1.6 Aim and scope of the thesis	17
1.7 Structure of the thesis	18
1.8 My Contributions	21
Chapter 2: Common Readout Unit in ALICE	23
2.1 Introduction	23
2.2 CRU Location in the ALICE experiment	24
2.3 CRU Readout configuration	26
2.4 CRU Usage	30
2.5 CRU Connectivity	31

CONTENTS

2.6	CRU as Trigger and Timing Distribution Unit	33
2.7	CRU as Data Aggregation Unit	34
2.8	CRU as Control Unit	36
2.9	CRU Firmware Architecture	36
2.10	CRU hardware architecture	38
2.11	Test Results	40
2.11.1	Minipod Performance	41
2.11.2	PCIe Performance	41
2.11.3	Resource Estimation	43
2.12	Conclusion	45
Chapter 3:	GBT Prototyping & Evaluation for the CRU Development	47
3.1	Introduction	47
3.2	Design Implementation	49
3.3	Resource Estimation	51
3.4	Latency Measurement	51
3.5	Bit Error Rate (BER) Analysis	54
3.6	Jitter and Eye Measurement	55
3.7	Calibration	57
3.7.1	Tx Latency Calibration (TLC)	58
3.7.2	Proposed Solution	59
3.7.3	Stratix V and Arria 10 FPGA	61
3.8	Conclusion	62
Chapter 4:	CRU as the Trigger and Timing Distribution Unit	63
4.1	Introduction	63
4.2	LHC Trigger Distribution	65
4.3	Triggered and triggerless architecture in ALICE	68
4.4	ALICE Clock Distribution strategy for RUN3	72

CONTENTS

4.5	High-Speed Serial Links in the TTC communication	73
4.6	GBT Specification	76
4.7	TTC PON Introduction	78
4.8	Use of TTC-PON GBT bridge in CRU for TTC communication	81
4.9	Design Resilience	84
4.10	Results	84
4.10.1	Latency Measurement	85
4.10.2	Jitter Measurement	87
4.10.3	BER Measurement	98
4.10.4	Transceiver Optimization	106
4.11	Discussion	115
4.12	Conclusion	116
Chapter 5:	Behaviour of the Jitter Cleaning PLL with Temp. Variation	117
5.1	Introduction	117
5.2	Measurement Setup	119
5.3	Results	120
5.4	Conclusion	127
Chapter 6:	Pico-second Resolution Phase Measurement System	129
6.1	Introduction	129
6.2	Design Implementation	132
6.3	Sampling methodology	134
6.4	Generating the sub-samples	136
6.4.1	Method of Cascaded PLL	136
6.4.2	Sampling Frequency for Sub-samples generation	137
6.5	Computation of Phase Magnitude	138
6.6	Measurement Error	142
6.7	Calibration	145

CONTENTS

6.8	Results	146
6.8.1	Measurement Setup	146
6.8.2	Comparison	149
6.9	Conclusion	152
Chapter 7:	Use of CRC-32 in the Detector Data Integrity Check	153
7.1	Introduction	153
7.2	Literature Survey	154
7.3	Common CRC-32 standards in use	156
7.4	Mathematical Model	156
7.5	Latency Optimization	162
7.5.1	Parallelism and concurrent architecture	163
7.6	CRC Logic Core Functional Description	164
7.7	Reconfigurable Architecture	166
7.7.1	Syndrome Table Synthesis	167
7.7.2	DESIGN 1: Compile Time reconfiguration	168
7.7.3	DESIGN 2: Run-time reconfiguration	169
7.8	Test Process	170
7.8.1	Test Setup 1: Throughput study with various modes of interfacing .	171
7.8.2	Test Setup 2: Resource utilization study with bus width variation .	176
7.9	Performance Analysis	176
7.9.1	For Test Setup 1 : Throughput study with various modes of interfacing	176
7.9.2	For Test Setup 2: Resource utilization study with bus width variation	178
7.10	Design Comparison	186
7.11	Future Scope	186
7.12	Conclusion	187
Chapter 8:	A Secure Error Resilient High-speed Link Standard	189
8.1	Introduction	189

CONTENTS

8.2	System Architecture	190
8.3	Scrambler	191
8.4	Advanced Encryption Standard (AES)	192
8.4.1	Algorithm Description	192
8.5	Golay Forward Error Correction Coder	195
8.6	Helical Interleaver	197
8.7	Frame Synchronization Block	198
8.7.1	Header and Trailer Addition Block	198
8.7.2	Rx Frame Aligner Block	199
8.8	Latency Optimized Gearbox	199
8.9	Transceiver Native PHY	200
8.10	Comparison	201
8.11	Results	203
8.11.1	Register resource utilization	203
8.11.2	Data Throughput	204
8.11.3	Optical MiniPOD Performance	204
8.12	Conclusion	206
Chapter 9: A Generalized Data Aggregation Unit (DAU) for HEP Ex-		
periments		207
9.1	Introduction	207
9.2	Comparison with other existing architectures	210
9.3	Design Constraints of the DAU	211
9.3.1	Types of Information and Constraints involved	211
9.3.2	Degrees of Synchrony	212
9.4	Choice of DAU Hardware and Interfaces	214
9.5	Implementation	214
9.5.1	Modes of Operation	217

CONTENTS

9.6 Queue model of DAU	217
9.7 Readout Architecture Transition from Hierarchy to IOT	221
9.8 Conclusion	223
Chapter 10: Summary and Future Scope	225
REFERENCES	231

CONTENTS

Synopsis

The primary objective of the thesis has been to study, design and test firmware for the electronic system of the ALICE (A Large Ion Collider Experiment) detector at the CERN Large Hadron Collider (LHC). The LHC is designed to collide proton on proton (pp) at the center of mass energy, $\sqrt{s} = 14$ TeV and lead on lead (Pb-Pb) at the center of mass energy per nucleon pair, $\sqrt{s_{NN}} = 5.5$ TeV at very high luminosities. By utilizing the data of the collision debris of pp and Pb-Pb collisions, the ALICE collaboration studies the formation of a new form of matter, called the quark-gluon plasma (QGP). ALICE has been recording unprecedented amount of data for pp, p-Pb and Pb-Pb collisions since 2010, the beginning of the LHC program. First major running period (Run1) is complete and the present run period (Run2) will be completed in 2018. From Run3, expected to start in 2021, the LHC will run at designed beam energies and the beam luminosities will increase by manifold. The luminosities for Pb-Pb collisions will increase by a factor of six to $6 \times 10^{27}/cm^2/sec$ and thus the data taking rate and data volume will increase in proportion. This will allow ALICE to inspect rare probes and further extend the scientific potential of the detector. Estimated combined data flow from the detector systems of ALICE to the online system is about 3.7 TB/sec. To cope with this large amount of data produced in these collisions and huge data flow, ALICE is going for a new electronic system in the form of the Common Readout Unit (CRU) which will be operational from the year 2021. A comprehensive study has been carried out in the thesis for the design architecture of the CRU and the associated firmware for data aggregation, multiplexing

and trigger distributions.

The CRU is at the heart of the ALICE electronic system, whose primary responsibility is to distribute trigger and timing information, to aggregate raw readout data from the sub-detectors on to several manageable links and to moderate the flow of control signals among various sub-systems. The CRU for its multi-functional role shares a common interface connection with the detector online readout systems, the Online-Offline (O2) computing system and the Central Trigger Processor (CTP). Different interface protocols have been explored for data flow at different stages. These include Gigabit Transceiver (GBT), Timing, Trigger and Control (TTC) Passive Optical Networks (PON), and PCI Express (PCIe). For CRU logic implementation, an FPGA based reconfigurable hardware is chosen, as it provides the advantage of faster development time with field re-programmability. PCIe40 DAQ Engine based on Intel[®] (formerly Altera) Arria[®] 10 FPGA has been chosen for the CRU firmware implementation.

From the run-time experience, it has been found that the detector dead times, busy signals and trigger decision time affect data taking rate. In the new upgraded architecture, the data readout is in continuous trigger-less mode, where the data is forwarded to the computing farms with dedicated time-markers called the Heart Beat (HB) trigger. The concept of HB has been introduced to allow throttling of the data throughput in case the data flow exceeds the instantaneous system bandwidth. Further processing is applied at a later stage for proper physics event selections.

Central Trigger Processor (CTP) is used to make trigger decisions using Local Trigger Unit (LTU) for communicating triggers and timing information to each sub-detector in ALICE. The Timing and Trigger Distribution Systems (TTS) of detectors are composed of two schemes namely TTS-FTL and TTS-TTC. For upgraded electronics, the TTS pathway is based on asynchronous Fast serial Trigger Links (FTL), and TTS-TTC pathway involves detectors with non-upgraded Front End Electronics (FEEs) or legacy detectors that continue to use CERN's present Trigger, Timing and Control (TTC) standards. The

latency critical trigger (LM) and busy signals will continue to use LVDS cables to connect to the detector front-end electronics. For non-critical latency triggers (L0, L1, heartbeat) and other control signals communication is through the TTS-FTL pathway. In TTS-FTL scheme, the trigger is routed from the CTP via the LTU through the CRU sitting in counting room to feed the sub-detectors in the cavern area. The pathway involves the transition of two asynchronous serial links TTC-PON (Timing Trigger and Control over Passive Optical Network) and GBT (Gigabit Transceiver). The downstream from the LTU to the CRU involves TTC-PON, which is time multiplexed trigger distribution topology. The downstream from the CRU to the FEE (Front-End Electronics) of sub-detectors involve GBT link standard, a radiation tolerant error resilient communication standard. The entire chain involves multiple regions of signal transitions that need detailed characterization and feasibility study. The major part of my research work deals with the efficient handling of these TTS-FTL links inside the CRU without compromising on the forwarded TTS signal quality.

Individual TTS links and electronic components are first tested, before conducting the performance study of the integrated chain. Initially started with the characterization of GBT TTS-FTL link. GBT is a radiation tolerant error resilient data communication standard having fixed latency support. GBT ecosystem is composed of GBT ASIC, GBT SCA chips, radiation hard optoelectronic components and FPGA based logic core. A single GBT channel link can support multiple information types like detector data, timing, trigger and control information. Radiation tolerant GBT chipset is used in detector front-end side for packaging of detector data and transmitting it in GBT standard. Current work involves GBT FPGA logic core implementation, testing and characterization on PCIe40 board. An initial study indicated that the use of 120 MHz (3 times of LHC Clock Frequency) word clock, transceiver needs calibration for maintaining deterministic latency. On further investigation, it was found that feedback compensated path is needed to avoid clock calibration. However, that functionality is supported when the transceiver

clock is driven at 240 MHz. This finding constraint the design to drive all the internal clock logic circuitry in 240 MHz domain. The higher frequency asserts stress on firmware routing and hence more pipelining is required to meet the timing requirement of the logic circuitry. So, for rapid prototyping of CRU with multiple GBT links connections, a topology is developed using Altera System Integration tool called QSYS that assist the developers in automating the synthesis process.

The other TTS-FTL link known as the TTC-PON is then characterized. It is a point-to-multipoint optical communication network that allows duplex data communication using two different wavelengths. A master node, called Optical Line Terminal (OLT) broadcasts the information to several slave nodes (downstream data transmission) and the slaves, called Optical Network Units (ONU), send information back to the OLT using a time division multiplexing (TDM) scheme. The downstream is used for trigger and timing information transmission while the upstream is used for busy and other control signal transmission. The initial implementation of the receiving side of the protocol is validated by connecting the ONU module on to the PCIe40 board. Before the transition of the recovered clock from the TTC-PON link to the GBT link for forwarding the LHC clock to the sub-detectors, a jitter cleaning is necessary. Si5344 PLL is used for jitter cleaning. After the performance measurement of individual GBT link, TTC-PON link and Si5344 PLL is conducted, next attempt is made for full chain characterization of the integrated TTS-FTL. In asynchronous serial links like GBT and TTC-PON, LHC clock is embedded in the serial data transmission. The embedded clock is recovered by the Front End Electronics in sub-detectors to generate Detector Specific Data Packet. The clock goes through multiple link transitions, making it susceptible to system and channel noises. It is necessary that the TTS-FTL channel maintains the quality and integrity of the forwarded clock signal. So, a preliminary study is conducted to see, when the LHC clock signal traverses TTC-PON GBT link connection bridge, how much jitter is added by the system.

During operational condition, there can be a fluctuation of temperature. Hence a study is done to evaluate whether the jitter cleaning PLL degrades in performance during temperature fluctuation. A fully automated setup is built to conduct the test. The critical element of the setup is the Climatic Test (CTS) Chambers in which temperature is remotely swept using ethernet interface. The device under test, i.e. Si5344 PLL is placed inside the CTS chamber, with an air-tight seal to isolate the PLL from the room temperature fluctuation. The characterization tests conducted for SI5344 PLL involves frequency combinations like 40 MHz - 40 MHz, 40 MHz - 120 MHz, 40 MHz - 240 MHz and 240 MHz - 240 MHz.

In the CRU firmware architecture, multiple clock transition zones occur. The timing signals from different channels need preservation of synchronicity with each round of power cycle, reset cycle or firmware upgrade. However, in the Intel[®](formerly Altera) Arria[®] 10 FPGA's, the recovered parallel clock from the high-speed serial transceivers do not maintain the same chip latency with each round of power cycle, firmware upgrade or reset cycle. It causes uncertainty of phase relationship between the received and the forwarded clocks. Traditional electronics systems used for time measurement designed using a classical mixed-signal approach are not suitable for this detection. CRU designers have opted for all-digital architecture for development of sensitive time measurement tools. To address the need to detect very minute phase shift changes inside FPGAs, a phase difference measurement technique is proposed as an FPGA logic core having resolution and precision in the range of few picoseconds. The working principle is based on sub-sample collection using systematic sampling technique, which gives an ability to measure phase difference even over high-frequency digital clocks. A mathematical model is formulated to describe the operating principle of the design. The VLSI architecture is then developed from the mathematical description. The performance of the design is justified in terms of accuracy, resolution and precision achieved.

For GBT and TTC-PON there exists only data frame protection. However, an addi-

tional layer of protection for data packet communication is needed for data integrity check. The popular choice is CRC32 as forward error detection scheme over data packet because of the low resource consumption overhead in the ultra-high-speed data communication. To tap the full potential of CRC algorithm in hardware level, it needs to be implemented in a hardware-friendly manner with proper user constraints. Hence, developed a very high throughput low latency VLSI design architecture design dedicated to CRC-32 with reconfigurable parameters. The high throughput is achieved by using expandable data bus line. While low latency is made possible by parallelizing the logic implementation. The uniqueness of the design lies in its ability to operate on the same clock cycle in which the code word is presented, with results produced immediately in the next clock cycle. The effects of variations in design parameters of CRC VLSI design on performance characteristics are also studied.

GBT being radiation hard error resilient protocol is used for linking electronic components from harsh radiation zone to non-radiation zone. However, this link does not provide data encryption and can lead to eavesdropping of the sensitive information if it is used in other critical infrastructures. The Hamming distance between the error vectors are also low and could lead to ambiguity during a long burst of error vector correction. The error detection capability provided in GBT is of the order of 32 bits. The new link architecture proposed is capable of multi-bit error detection to protect against radiation-induced soft-errors and prevents eavesdropping of classified information by encrypting the data packet using Advanced Encryption Standard (AES) while transmitting over the optical media. The work uses the same PCIe40 board for characterization of the link technology. The data communication scheme ensures proper DC-balance of the line and allows forward error correction (FEC) with encryption. It is implemented through concatenated blocks of Scrambler, Golay triple error correction coder, AES (Advanced Encryption Standard) cypher and Helical Interleaver. The custom link defined operates at a frequency of 8.192 Gbps, allowing 120 bit of user data frame with the capability to correct 15 bits of errors

and detect 35 bits of error or maximum 115 bits of odd bit error, with decidedly less ambiguity. The novelty of the design is justified through the performance measurement of the link standard.

Similar to CRU there exists data aggregation hardware solutions in other High Energy Physics (HEP) experiments, like TELL40 for LHCb, DPB for CBM and CODAC for ITER. A generalized solution is proposed that addresses the requirement for a design of a configurable Data Aggregation Unit (DAU) in HEP experiments. Apart from the primary function of aggregation of data, the DAU systems act as a data traffic manager for multi-modal data types in HEP and can able to autonomously restructure data distribution based on the current system stress conditions. An initial prototype of DAU is built to exhibit its promising functional role in the future upgrade of HEP readout architectures.

Keywords :

A Large Ion Collider Experiment (ALICE); Common Readout Unit (CRU); Data Aggregation Unit (DAU); Electronic system upgrade for RUN3; Field Programmable Gate Array (FPGA); Giga-Bit Transceiver (GBT) optical link; Large Hadron Collider (LHC); High-speed serial transceivers; High speed signal Integrity; Timing, Trigger and Control system based in Passive Optical Networks (TTC-PON); Peripheral Component Interconnect Express (PCIe); PCIe40 DAQ Engine; Phase measurement logic core; Very Large Scale Integration (VLSI);

SYNOPSIS

LIST OF ABBREVIATIONS

ACO ALICE Cosmic Ray Detector. 33

AES Advanced Encryption Standard. 189, 228

ALICE A Large Ion Collider Experiment. 8–11, 15, 23, 64, 68, 72, 84, 88, 115

ASIC Application-Specific Integrated Circuit. 14, 76

ATC Automated Transceiver Calibration. 58, 61

ATCA Advanced Telecommunications Computing Architecture. 27

ATX PLL Advanced Transmit PLL. 88, 191, 201

Avalon-MM Avalon[®] Memory-Mapped. 41, 171

BC Bunch Clock. 66, 68, 227

BER Bit Error Rate. 19, 64, 65, 85, 98, 99

C-RORC Common Read-Out Receiver Card. 15, 72

CDC Clock Domain Crossing. 51

CDR clock data recovery. 201

CERN The European Organization for Nuclear Research (French: Conseil Européen pour la Recherche Nucléaire). 1, 115

LIST OF ABBREVIATIONS

CMU PLL	Clock Multiplier Unit PLL. 88
COTS	Commercial Off-The-Shelf. 48
CR4	Counting room. 24
CRC	Cyclic Redundancy Check. 153
CRU	Common Readout Unit. 8, 10, 15, 17, 18, 24, 35, 63, 64, 69, 72, 74, 77, 82, 84, 88, 93, 102, 107, 115, 116, 169, 226, 229
CTP	Central Trigger Processor. 24, 64, 68, 69, 79, 88, 105, 226
D-RORC	DAQ Read-Out Receiver Card. 14
DAQ	Data Acquisition. 14, 15, 79, 102, 221
DASPOS	Data and Software Preservation for Open Science. 222
DAU	Data Aggregation Unit. 208, 228
DCS	Detector Control Subsystem. 36
DDL	Detector Data Link. 14, 15, 27
DDR	Double Data Rate. 58
DIC	Data Intensive Computing. 208
DIU	Destination Interface Unit. 14
DNL	Differential Non-linearity. 144
DPA	Dynamic Phase Alignment. 145
DPHEP	Data Preservation in Long Term Analysis in High Energy Physics. 222
DSDP	Detector Specific Data Packet. 36, 153

LIST OF ABBREVIATIONS

EMC Electro Magnetic Calorimeter. 33

EPN Event Processing Node. 35, 69

FAW Frame Alignment Word. 62

FEC forward error correction. 190

FEEs Front End Electronics. 14, 33

FIT First Interaction Trigger. 33, 68

FLP First Level Processor nodes. 35, 69

FPGA Field Programmable Gate Array. 13–15, 24, 48, 76, 77, 82, 84, 86, 88, 93, 101, 102, 105, 116

fPLL Fractional PLL. 88

FSM Finite State Machine. 82

FTL Fast serial Trigger Links. 19, 34

GBT Giga-Bit Transceiver optical link. 15, 18, 19, 26, 63, 64, 72, 74, 76–78, 81, 82, 85–88, 93, 98, 99, 101, 102, 107, 115, 129, 226

GBT-SCA GBT Slow Control. 36

GOL Gigabit Optical Link. 47

H-RORC High Level trigger (HLT) RORC. 14

HB Heartbeat. 69, 70

HBF Heartbeat Frames. 69, 70

HDLC High-level Data Link Control. 36

LIST OF ABBREVIATIONS

HEP High Energy Physics. 13, 20, 32, 129, 207, 228

HL-LHC High-Luminosity Large Hadron Collider. 7, 10

HSSI SerDes High Speed Serial Interface Serializer-Deserializer. 57

IBS Inter Beam Scattering. 65

IoT Internet of Things. 221, 222

IP Intellectual Property. 14

ISR Intersecting Storage Ring. 1

JTF Jitter Transfer Function. 96

LEP Large Electron-Positron Collider. 7

LHC Large Hadron Collider. 1, 11, 65, 66, 69, 74, 207, 227

LTU Local Trigger Unit. 33

LUT Lookup table. 174

MGT Multi-Gigabit Transceiver. 49, 57, 81

MPO Multifiber Push-On. 102

NIST National Institute of Standards and Technology. 192

NRE non-recurring expenses. 13

O2 Online and Offline. 24, 69

OLT Optical Line Terminal. 85, 102, 104

ONU Optical Network Unit. 85, 102, 104

LIST OF ABBREVIATIONS

PAR Place and Route. 142

PCI Peripheral Component Interconnect. 14

PCI-X Peripheral Component Interconnect eXtended. 14

PCIe Peripheral Component Interconnect Express. 15, 19, 27, 31, 226

PCS Program Coding Sublayer. 58

PD Phase Detector. 130

PDF Probability Distribution Function. 147

PFR cycle power on/off cycle, firmware upgrade and reset assertion/de-assertion cycle.
63, 115, 116, 227

PHO Photon Spectrometer. 33

PISO Parallel Input Serial Output. 58

PLD Programmable Logic Device. 14

PLL Phase Locked Loop. 19, 82, 88, 130

PMA Physical Medium Attachment. 77

PON Passive Optical Network. 104

PRBS pseudorandom binary sequence. 102, 104

PreSICE Precision Signal Integrity Calibration Engine. 61

PRORC PCI-based Readout Receiver Card. 14

PVT process, voltage and temperature. 57

QCD Quantum Chromodynamics. 23

LIST OF ABBREVIATIONS

QGP quark-gluon plasma. 2

RCUs Readout Controller Units. 14

RF Radio Frequency. 65, 66, 227

RISC Reduced Instruction Set Computer. 172

RLC Rx Latency Calibration. 58, 62

RMS Root Mean Square. 88

RORC Read-Out Receiver Card. 14

S-LINK Simple Link Interface. 47

SBT Software Build Tools. 174

SCCP SCA Channel Command-oriented Protocol. 36

SDA Serial Data Analyser. 55, 101

SEU Single Event Upset. 25

SFP small form-factor pluggable. 98, 102

SG-DMA Scatter-Gather Direct Memory Access. 173

SIU Source Interface Unit. 14

SPS Super Proton Synchrotron. 1

STF Sub-Time Frame. 69

TDM Time Division Multiplexing. 79

TF Time Frame. 69

LIST OF ABBREVIATIONS

TLC Tx Latency Calibration. 58, 61

TMR Triple Modular Redundancy. 25

TOF Time Of Flight detector. 33

TPC Time Projection Chamber. 33

TSD Temperature Sensing Diode. 60

TTC timing, trigger and control. 17, 19, 34, 47, 64, 68, 72, 74, 76, 78, 79

TTC-PON Timing, Trigger and Control system based on Passive Optical Networks. 19, 63, 64, 72, 74, 78, 79, 81, 82, 85, 87, 88, 93, 98, 99, 105, 115, 129, 226

TTK Transceiver Toolkit. 54

TTS Trigger and Timing System. 19, 33, 68–70, 72

VCO Voltage Controlled Oscillator. 201

VLDB Versatile Link Demo Board. 86, 88

WDM Wavelength Division Multiplexing. 79

ZDC Zero Degree Calorimeter. 33

LIST OF FIGURES

No.	Title	Page No.
Figure 1.1	History of the Universe	3
Figure 1.2	Accelerators at CERN complex and schematic layout of LHC Beam	5
Figure 1.3	Technical timeline of the LHC.	7
Figure 1.4	HEP experiments with varying Trigger Rate vs Event Size	8
Figure 1.5	Phase diagram of Quark-Gluon Plasma (QCD)	9
Figure 1.6	PHASE 1 major upgrade in ALICE to prepare for RUN3 and HL-LHC	10
Figure 1.7	ALICE Upgrade from 2021	11
Figure 1.8	ALICE Data Link Receiver Cards in use during different RUN . . .	17
Figure 2.1	CRU location in the ALICE experiment (not in scale)	26
Figure 2.2	CRU Readout Configurations	29
Figure 2.3	Showing the implementation of two protocol stack and its interface with user application layer	30
Figure 2.4	CRU interfaces with the neighbourhood blocks, which are (1) GBT; (2) TTC-PON; (3) PCIe [®] Gen 3	32
Figure 2.5	CRU as Trigger Distribution Unit	34
Figure 2.6	CRU as Data Aggregation and Processing Unit	35
Figure 2.7	CRU as Detector Control mediator (for detectors that will use GBT chipset in FEEs)	36
Figure 2.8	CRU internal logic block connections	37

LIST OF FIGURES

Figure 2.9	CRU development	38
Figure 2.10	A generalized diagram showing optical link connection with FPGA bank for PCIe40	39
Figure 2.11	MiniPOD TM performance with GBT protocol (4.8 Gbps) and 10 GbE protocol (10.312 Gbps)	42
Figure 3.1	Data format for GBT protocol specific data packet	49
Figure 3.2	GBT core design connectivity	50
Figure 3.3	Test Setup for 3 types of round trip measurement, each loopback arrangement are marked by different colours	52
Figure 3.4	Showing the Eye Diagram for Tx Optical using GBT encoded data (a) AMC40 and (b) PCIe40	56
Figure 3.5	Showing the Jitter Spectrum for (a) AMC40 and (b) PCIe40	57
Figure 3.6	Synchronization register based clock domain crossing with metasta- bility or phase error detection logic	60
Figure 3.7	Calibration trigger from manually tabulated phase error look-up table	60
Figure 3.8	Calibration trigger on data frame error or metastability	61
Figure 4.1	LHC Radio Frequency Swing with Beam Ramping	67
Figure 4.2	Role of Heart beat trigger in continuous readout mode	70
Figure 4.3	Heart beat intercepted with physics trigger to support both contin- uous and triggered readout mode	71
Figure 4.4	ALICE trigger distribution and detector read-out systems (The TTS distribution path is marked in green).	72
Figure 4.5	GBT ecosystem	76
Figure 4.6	Flow connection of a GBT Link	77
Figure 4.7	Showing how GBT Links are aggregated in 1 GBT Bank	78
Figure 4.8	Flow connection of TTC-PON link	79

LIST OF FIGURES

Figure 4.9	TTC-PON GBT bridge in star topology connection	81
Figure 4.10	Configuration-I: TTC-PON GBT Bridge	82
Figure 4.11	GBT Phase Uncertainty in Tx	83
Figure 4.12	Configuration-II: TTC-PON GBT Bridge	83
Figure 4.13	Latency of 151 ns is detected between Transmitting data and Re- ceiving Data	86
Figure 4.14	Test Setup for <i>three</i> types of round trip measurement, each loopback arrangement are marked by different colours	87
Figure 4.15	Jitter measurement test setup for the TTC-PON GBT bridge . . .	89
Figure 4.16	Oscillator phase noise in dBc/Hz vs. frequency offset [1]	91
Figure 4.17	Comparison of jitter cleaning performance between the two jitter cleaners	93
Figure 4.18	Phase Noise Analysis (Total Jitter) for VLDB clock output at 40 MHz frequency having the full TTC-PON – GBT bridge connection with the PLL configured at loop bandwidth 400 Hz and 4 kHz	94
Figure 4.19	Phase noise analysis (total jitter) for ONU uncleaned clock (40MHz), SI5344 jitter cleaned clock (120 MHz) and SI5344 jitter cleaned Clock (240 MHz)	95
Figure 4.20	Phase noise analysis (total jitter) for source clock (240MHz), ONU uncleaned clock (40 MHz) and SI5344 jitter cleaned clock (240 MHz) . . .	95
Figure 4.21	Jitter measurement test setup with GBT CHAIN and VLDB only .	96
Figure 4.22	Phase noise analysis (total jitter) for the VLDB clock output of : 40 MHz o/p with the GBT link connection, 80 MHz o/p with the GBT link connection, 40 MHz o/p with the TTC-PON – GBT bridge connection and 80 MHz o/p with the TTC-PON – GBT bridge connection	97
Figure 4.23	GBT Measurement Setup	99

LIST OF FIGURES

Figure 4.24 Time required to reach 10^{-12} BER vs. line rate. Showing the cases for GBT and TTC-PON.	100
Figure 4.25 Showing GBT BER measurement for GBT and widebus FEC scheme	101
Figure 4.26 Showing the Eye Diagram for Tx Optical using GBT encoded data for Arria [®] 10	102
Figure 4.27 Showing power Monitor of GBT Firmware during data communication mode in : (a) GBT Scheme and (b) Widebus Scheme	103
Figure 4.28 OM3 and OM4 BER curves and Eye Diagrams (Adapted the research article Schwemmer et. al. [2])	104
Figure 4.29 Downstream Bit Error Rate (BER) from the OLT to the ONU . . .	105
Figure 4.30 The TTC PON signal quality measurement setup.	106
Figure 4.31 TTC PON signal quality analysis in an Agilent Scope by plotting an eye diagram	107
Figure 4.32 TTC PON signal quality monitoring using Transceiver Toolkit (TTK) in serial loopback mode.	107
Figure 4.33 Choice of the optimized values from the complete range of configuration parameters	108
Figure 4.34 Transceiver parameter tuning plot with the VOD variation	110
Figure 4.35 Transceiver parameter tuning plot with the PRE-EMPHASIS 1st PRE TAP variation	111
Figure 4.36 Transceiver parameter tuning plot with the PRE-EMPHASIS 1st POST TAP variation	111
Figure 4.37 Transceiver parameter tuning plot with the PRE-EMPHASIS 2nd PRE TAP variation	112
Figure 4.38 Transceiver parameter tuning plot with the PRE-EMPHASIS 2nd POST TAP variation	112

LIST OF FIGURES

Figure 4.39 Transceiver parameter tuning plot with the EQUALIZATION variation	113
Figure 4.40 Transceiver parameter tuning plot with the VGA variation	113
Figure 4.41 Transceiver parameter tuned showing values for the default settings vs the best settings	114
Figure 4.42 Transceiver parameter performance with the PRBS variation	114
Figure 4.43 Eye Diagram (a) Default parameter settings; (b) Optimized parameter settings	115
Figure 5.1 Test setup of the Si5344 PLL.	119
Figure 5.2 Boxplot of skew distribution over 10000 reset cycle at $23^{\circ}C$ for different input-output frequency combinations	123
Figure 5.3 Skew Distribution over 10000 repetitive reset cycle at $23^{\circ}C$ for PLL configuration 240 MHz i/p and 240 Mhz o/p	123
Figure 5.4 The effect of temperature monitoring on the registered data points: (a) Hysteresis in reading data when the chamber temperature is only monitored; (b) Without any hysteresis in reading data when surface temperature of the PLL chip is also monitored	124
Figure 5.5 Skew Variation over 12 Temperature Cycles (10000 readings / print) for PLL configuration 240 MHz i/p and 240 Mhz o/p frequency.	124
Figure 5.6 Impact of thermal gradient on TIE clock jitter for different <i>input-output</i> frequency combinations : (a) 40 MHz i/p – 40 MHz o/p; (b) 40 MHz i/p – 120 MHz o/p; (c) 40 MHz i/p – 240 MHz o/p; (d) 240 MHz i/p – 240 MHz o/p.	125

LIST OF FIGURES

Figure 5.7	Impact of thermal gradient on clock skew for different <i>input-output</i> frequency combinations : (a) 40 MHz i/p – 40 MHz o/p; (b) 40 MHz i/p – 120 MHz o/p; (c) 40 MHz i/p – 240 MHz o/p; (d) 240 MHz i/p – 240 MHz o/p.	126
Figure 5.8	PLL remains lock for different Offset in Frequency applied with respect to the LHC frequency of 40 MHz	126
Figure 6.1	The phase difference in the duty cycle of XORed waveforms is illustrated. Leading Phase Duty Cycle ($D_{\theta 1}$) and Lagging Phase Duty Cycle ($D_{\theta 2}$) are shown in (c-d) and (e-f) panels, respectively.	131
Figure 6.2	The scanning of the XORed waveform using the Sampling clock is illustrated for $F_s/F = 4/10$ or $\Delta/\Psi = 2/5$	135
Figure 6.3	Stepwise successive approximation of phase value computation for 56.25°	140
Figure 6.4	VLSI Architecture of Phase measurement instrumentation inside FPGA.	141
Figure 6.5	Effect of Phase polarity offset: (a) when no bank skew cancellation is in effect, the magnitude and polarity are mis-aligned leading to measurement ambiguity; (b) when bank skew cancellation is in effect the quantization steps are properly aligned.	142
Figure 6.6	Transfer Characteristics of Phase Measurement Logic Core when operating in <i>Active low Sample Counting Mode</i>	143
Figure 6.7	Histogram of Phase-Stepped Values.	143
Figure 6.8	PDF of Quantization Error (ϵ).	144
Figure 6.9	The effect of sampling on precision of the system.	147
Figure 6.10	Phase measurement accuracy, presented in ps and in degrees, as a function of sample population size (N).	149

LIST OF FIGURES

Figure 6.11 Precision Distribution for 50×10^3 sample population size operating in (a) Active High Sampling Counting Mode; (b) Active Low Sampling Counting Mode;	150
Figure 6.12 The dependence of resolution (ps) and <i>number of sub-samples</i> (Δ) with the frequency ratio (F_s/F).	151
Figure 6.13 Graphical bullseye representation of accuracy and precision (a) High accuracy and low precision; (b) High accuracy and high precision.	151
Figure 7.1 CRC-32 block diagram	165
Figure 7.2 CRC-32 Generator Timing diagram	165
Figure 7.3 CRC-32 Checker Timing diagram with dynamic <i>byteenable</i>	165
Figure 7.4 CRC-32 Checker Timing diagram with static <i>byteenable</i>	166
Figure 7.5 CRC-32 compile time reconfigurable architecture	169
Figure 7.6 CRC-32 run-time reconfigurable architecture	170
Figure 7.7 Showing the proposed CRC-32 architecture with Avalon MM connectivity	172
Figure 7.8 Showing the proposed CRC-32 architecture in custom instruction mode	173
Figure 7.9 Showing the proposed CRC-32 architecture in 4 different modes of connectivity	174
Figure 7.10 Showing the proposed CRC-32 architecture with Avalon ST Connectivity	175
Figure 7.11 Latency vs CRC bus Variation	179
Figure 7.12 Power vs CRC bus Width Variation	181
Figure 7.13 Power vs Frequency for 32 bit data bus	182
Figure 7.14 Power vs Frequency for 64 bit data bus	182
Figure 7.15 Power vs Frequency for 128 bit data bus	183

LIST OF FIGURES

Figure 7.16 Power vs Frequency for 256 bit data bus	183
Figure 7.17 Resources vs CRC bus Variation	185
Figure 7.18 Showing chip planner view of used resources (< 1%) in Stratix V FPGA	185
Figure 8.1 System Architecture Block Diagram	191
Figure 8.2 Showing AES Algorithm Flow	193
Figure 8.3 Showing multiple Golay encoder/decoder operating in-situ	197
Figure 8.4 Showing Helical Interleaving of 120 bit	197
Figure 8.5 Latency optimized gearbox	198
Figure 8.6 Data Frame format	198
Figure 8.7 Scanning window searching for FAW	199
Figure 8.8 Simplified diagram of transceiver data-path and clock connectivity .	200
Figure 8.9 Showing the Eye Diagram of 8.192 Gbps optical link	204
Figure 8.10 Showing Jitter measurement of 8.192Gbps optical link	205
Figure 9.1 Data Volume expansion in High Energy Physics Experiments in comparison to other digital data.	209
Figure 9.2 Architecture of DAU Implementation	216
Figure 9.3 Queue Model for the Data Aggregation Unit	219
Figure 9.4 Transition of Data readout Architecture in High Energy Physics Experiments	222

LIST OF TABLES

No.	Title	Page No.
Table 1.1	Allocation of LHC Intersection Points	6
Table 1.2	Running scenario for the LHC Runs 3 and 4	12
Table 2.1	Detector Specific CRU usage	31
Table 2.2	Comparison of three high speed interface protocol standards used in CRU	32
Table 2.3	Specification of the FPGA used in PCIe40 aka CRU card	40
Table 2.4	Shows the comparison of MiniPOD TM performance for GBT protocol and 10GbE Protocol	41
Table 2.5	An elementary integrated firmware design with all the Interface Links	43
Table 2.6	Individual logic blocks resource occupancy for variable number of links	44
Table 2.7	40 GBT Links resource occupancy for various type of combinations .	44
Table 3.1	Details of GBT protocol standards	48
Table 3.2	Reference Design configuration	51
Table 3.3	Shows the comparison of resource utilization in two boards for Stan- dard Reference Design	51
Table 3.4	Round Trip delay meaasurements for multi-level loopback in PCIe40	54
Table 3.5	GBT Serial Loopback Round Trip Measurement comparison for AMC40 and PCIe40	54
Table 3.6	BER Measurement	55

LIST OF TABLES

Table 3.7	Shows the comparison of Optical Minipod TM performance in two boards	56
Table 4.1	LHC Clock variation with LHC accelerator cycle	65
Table 4.2	LHC Radio Frequency Swing for Protons @ 7 TeV	67
Table 4.3	LHC Radio Frequency Swing for Ions @ 7 TeV	67
Table 4.4	List of Trigger Codes for Heartbeat Frames	71
Table 4.5	Degrees of Synchrony to behave as a fully synchronous system	74
Table 4.6	Showing the basic parametric difference between GBT and TTC-PON	75
Table 4.7	TTC-PON GENERATIONS	80
Table 4.8	Basic monitoring system to detect macroscopic behaviour	84
Table 4.9	The GBT round trip delay measurements for the multi-level loopback in the PCIe40	87
Table 4.10	SI5344 <i>Rev B</i> PLL Configuration	89
Table 4.11	VLDB Configuration	90
Table 4.12	Comparison of <i>RMS Jitter</i> results	97
Table 4.13	Comparison of the <i>RMS Jitter</i> at each intermediate points in CRU TTC-PON GBT bridge connection	98
Table 4.14	Power consumed by a single GBT link for different encoding scheme	102
Table 4.15	Variation of the received power with change in the attenuation . . .	105
Table 4.16	The default range of configuration parameters	108
Table 4.17	The optimized range of individual configuration parameters	109
Table 4.18	The optimized value of configuration parameters	110
Table 5.1	Skew & TIE measurement for various PLL configuration mode. . . .	122
Table 6.1	FPGA based phase detection methods	150
Table 7.1	CRC-32 TYPES	156

LIST OF TABLES

Table 7.2	Parameter value for CRC-32 Ethernet Standard	157
Table 7.3	Analogy between logic blocks and mathematics	168
Table 7.4	Performance of CRC-32 implementation in different architecture topology	177
Table 7.5	Speed up ratio for all hardware implementation	177
Table 7.6	Performance of CRC-32 in software implementation	178
Table 7.7	Timing Analysis	179
Table 7.8	Power Analysis	181
Table 7.9	Comparison of High-Speed CRC-32 Logic Implementation	186
Table 8.1	Trailer pattern meaning	199
Table 8.2	Frame Aligner State Machine	199
Table 8.3	Comparison between two data transmission protocol	202
Table 8.4	Register Resource Utilization	203
Table 8.5	Latency Measured	203
Table 9.1	New Readout Architecture in upgraded or future HEP experiments having the concept of Data Aggregation Unit	209
Table 9.2	Comparison of SerDes Speed among popular FPGA Vendors	213
Table 9.3	Comparison of popular Communication standards speed along with evolution timeline	215
Table 9.4	Comparison of three high speed interface protocol standards used in the design	216
Table 9.5	Queueing Model Performance comparison of some of the widely used models	220

Chapter 1

Introduction

CERN was created in 1954 with twelve European countries as the founding member states, with a common goal of carrying out nuclear research for pure scientific purpose in the benefit of mankind. Today, CERN is the world's largest research laboratory with twenty-two-member states and four associate members. India has a special observer status since 1991, and was actively involved in many scientific and technical cooperation with the CERN. Being invited by the CERN management team, India on November 2016 gained the status of associate member in the European organization.

In 1957, CERN commissioned its first high-energy accelerator, with a 600 MeV synchrocyclotron. Since then there is no looking back and CERN has always remained at the forefront of accelerator technology. In 1971, world's first proton-proton collider, called the Intersecting Storage Ring (ISR) came into operation. In 1974, the Super Proton Synchrotron (SPS) came into being, housed in a tunnel measuring 7 km in circumference. SPS has been the workhorse of the CERN's particle physics program and still continues to provide beams for the Large Hadron Collider (LHC). Presently, the CERN accelerator complex is a succession of particle accelerators, each accelerator boosts the speed of beam particles, before injecting to new one in the process. The last one in the chain is the LHC.

According to the theory of Big Bang, the Universe began about 13.7 billion years

ago by expanding from an infinitesimally small volume with extremely high density and temperature. Till about 10^{-43} second after the Big Bang, all the four fundamental forces (gravitational, electromagnetism, weak force and strong interaction) were united. Between 10^{-43} to 10^{-36} second, the gravity starts to separate from the other forces, and till about 10^{-12} second the electroweak force separates from the strong force. During the time, all the fundamental particles are believed to have acquired mass via a procedure known as the Higgs mechanism, in which Higgs boson acquires a vacuum expectation value. Till about 10^{-6} second after Big Bang, the Universe was filled with quarks, mediated by the gluons in a state of matter called the quark-gluon plasma (QGP). The QGP state is followed by nucleosynthesis with the formation of nuclei. The Figure 1.1 shows an impressive view of the evolution of present day matter after Big Bang.

The main purpose of building the world's most powerful accelerator LHC, commissioned in 2008, is to recreate the very energetic conditions that existed within a billionth of a second after the Big Bang, and allow scientists to study the fundamental ingredients of matter that the Universe was made of at the time. Let us go over the salient features of the LHC machine, the experiments and the physics topics that the scientists aim to address.

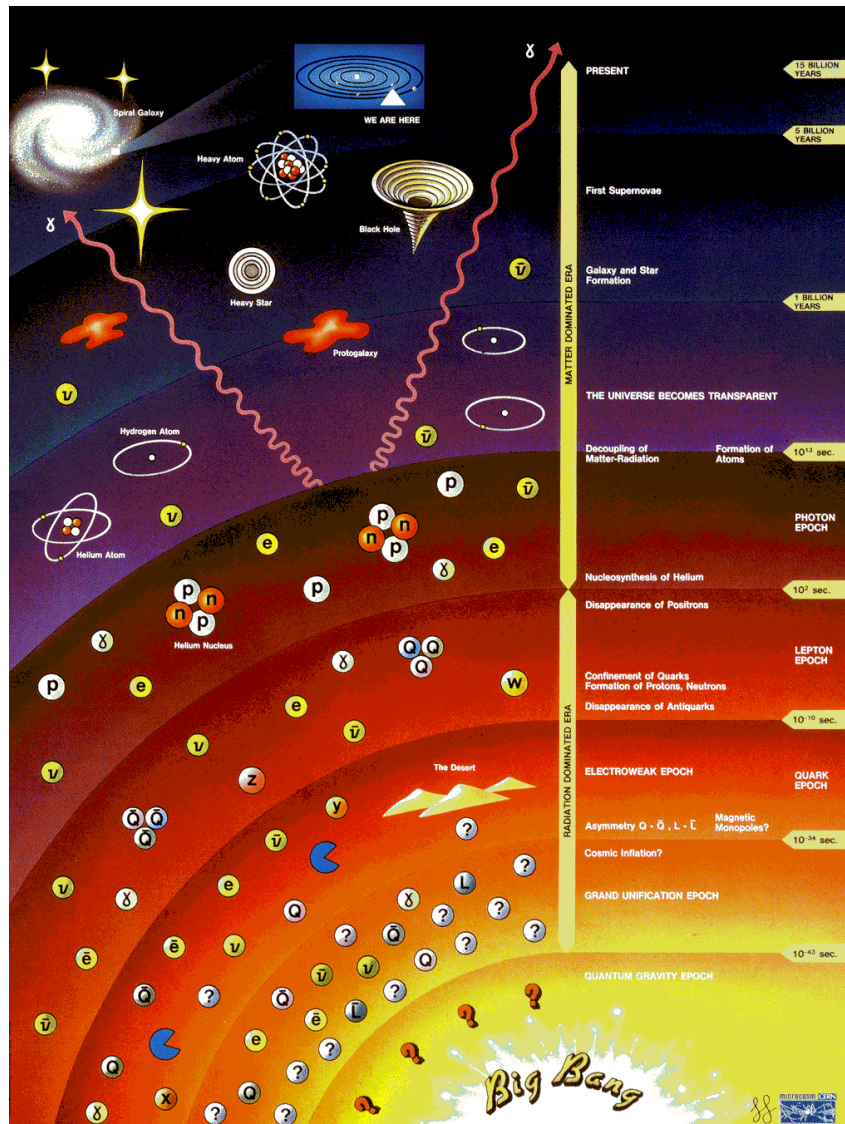


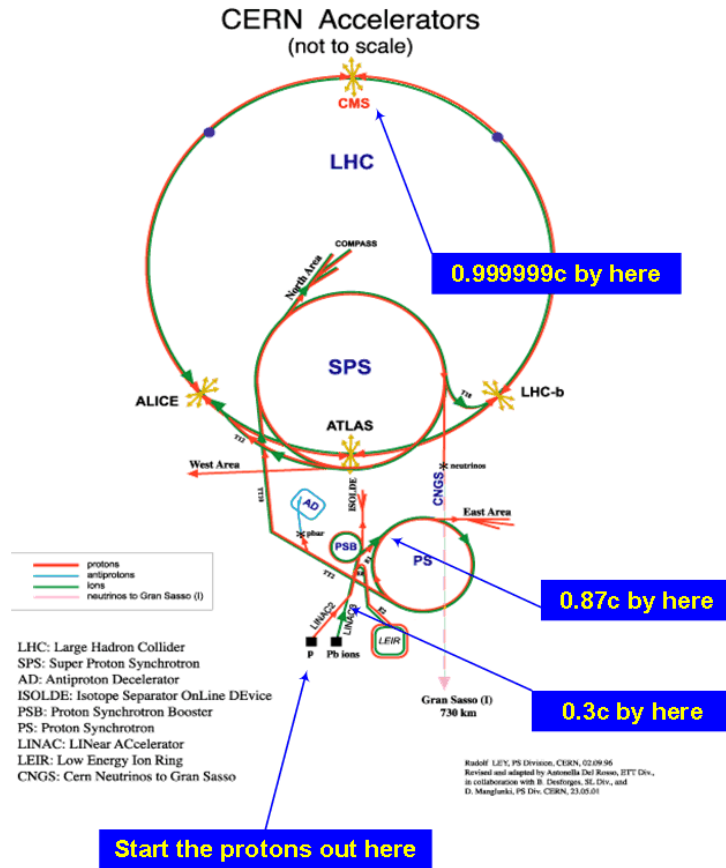
Figure 1.1: History of the Universe

1.1 The Large Hadron Collider (LHC) of CERN

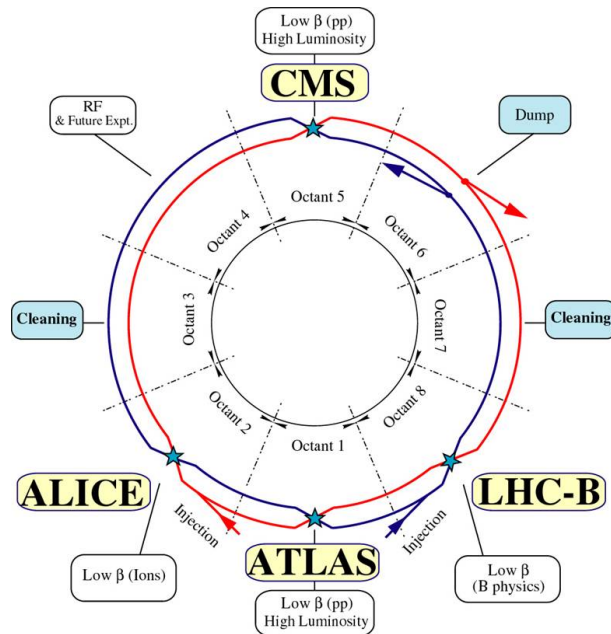
The LHC is constructed in a circular tunnel of 26.7 km circumference, situated at a depth of 170 m beneath the Franco-Swiss border near the Geneva, Switzerland. The tunnel plane is inclined towards the Léman lake at 1.4 % sloping, making one side of the tunnel deeper than the other. The depth ranges from 50 to 175 metres underground. The LHC encapsulates two adjacent parallel beamlines (or beam pipes). The beamlines contain particle beams that travel in opposite directions around the ring. The particle beams are formed of protons, ion nuclei, or proton and ion nuclei. A beam in the LHC is not a continuous string of particles but formed of hundreds of **bunches**.

The circulating beams in the LHC beam pipes are made to collide at four different locations. If the collision is head-on, there is no recoil and 100% of the kinetic energy is transformed to new particles. It is a great technological achievement to steer the beams with such a precision that although moving at almost the speed of light they collide exactly at the right place. To increase the chances of collision, the beam size is squeezed down as much as possible. In a beam bunch there are about 10^{11} protons. At the interaction points, these are squeezed to a size of 64 microns, about the width of a human hair. Still only about 20 collisions per beam crossings happen. The collision rate depends on the beam luminosity and the proton-proton cross section. The beam bunches cross every 25 ns, that allows around 600 million collisions per second. Most of the protons do not collide and continue to move. The beams are kept circulating for hours.

The **Lattice layout** of the LHC allows the geometrical arrangement to serve as a *beam collision point* or as a *utility point*. The LHC has eight arcs and eight straight sections, with each section approximately 528m long. The part of the ring between the two successive insertion point is referred as a *sector*. The portion of the arc from its center to the center of the next downstream arc is called as an *octant* [3]. Currently there are four cross points, where the four experiments are operational namely, the ATLAS experiment



(a)



(b)

Figure 1.2: (a) Description of CERN accelerator complex; (b) Location of four LHC experiments (Beam 1 – clockwise; Beam 2 – anticlockwise)

at Point 1, the ALICE experiment at Point 2, the CMS experiment at Point 5 and the LHCb experiment at Point 8 as shown in the Figure 1.2 (a). Details of each intersection point is tabulated in the Table 1.1.

Table 1.1: Allocation of LHC Intersection Points [4]

Intersection point	Place name	Tunnel		LHC
		Depth (m)	Slope (%)	
I	Meyrin	82.0	1.23	ATLAS
II	St. Genis	45.3	1.38	ALICE and Injection
III	Crozet	97.5	0.72	Cleaning
IV	Echenevex	137.6	0.36	RF
V	Cessy	86.6	1.23	CMS
VI	Versonnex	95.0	1.38	DUMP
VII	Ferney	94.0	0.72	Cleaning
VIII	Mategrin	98.8	0.36	LHCb and Injection

The **LHC accelerator Chain** is a succession of machines that accelerate the particles to attain an ultra-relativistic velocity with a very high energy as shown in the Figure 1.2 (b). Each machine in the chain boosts the energy higher before injecting it into the next sequence. The last element in the chain accelerates the particle at a record energy of 7 TeV. Beams filling in the LHC require several cycling time before the beam reaches the full energy. In the high-energy accelerators like the LHC, particles attain velocity very close to the speed of light. At the LHC particles move at 0.999999991 times the speed of light at top energy of 7000 GeV [5]. In these conditions as the energy increases, the increase in speed is minimal, instead there is a gain in the relativistic mass. Therefore, particle physicists think in terms of the particle's energy, rather than its speed. In a collider experiment when two beams collide the energy of the collision or *centre-of-mass energy* is the sum of the two beam energies, that makes 14 TeV ($= 7 \text{ TeV} + 7 \text{ TeV}$) for LHC.

Interactions at the crossing points are referred as events. The number of events gen-

erated per second in a LHC collision is given by

$$N_{event} = L\sigma_{event}, \quad (1.1)$$

where σ_{event} is the cross section for the event and L is the luminosity measured for gaussian profile beam. Luminosity is not constant over a physics run but decays with the degradation in the circulating beam. Hence, *Integrated Luminosity* parameter is used to sum up the luminosity over a run yield.

The LHC had replaced the Large Electron-Positron Collider (LEP) in the year 1998 and the LHC itself is proposed to be replaced by the High-Luminosity Large Hadron Collider (HL-LHC) in the year 2025. The upgrade aims at increasing luminosity by a factor of 10. Figure 1.3 portrays the timeline of the collider that shows how the LHC operated for over ten years is soon set to get a new life.

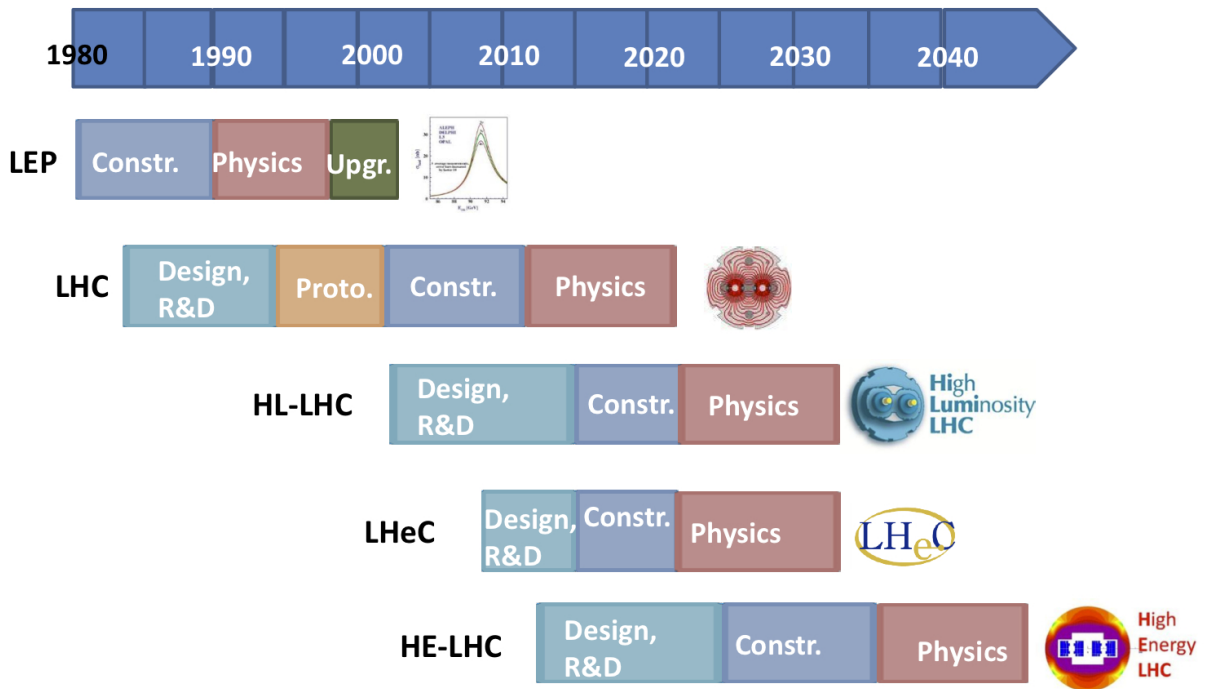


Figure 1.3: Technical timeline of the LHC.

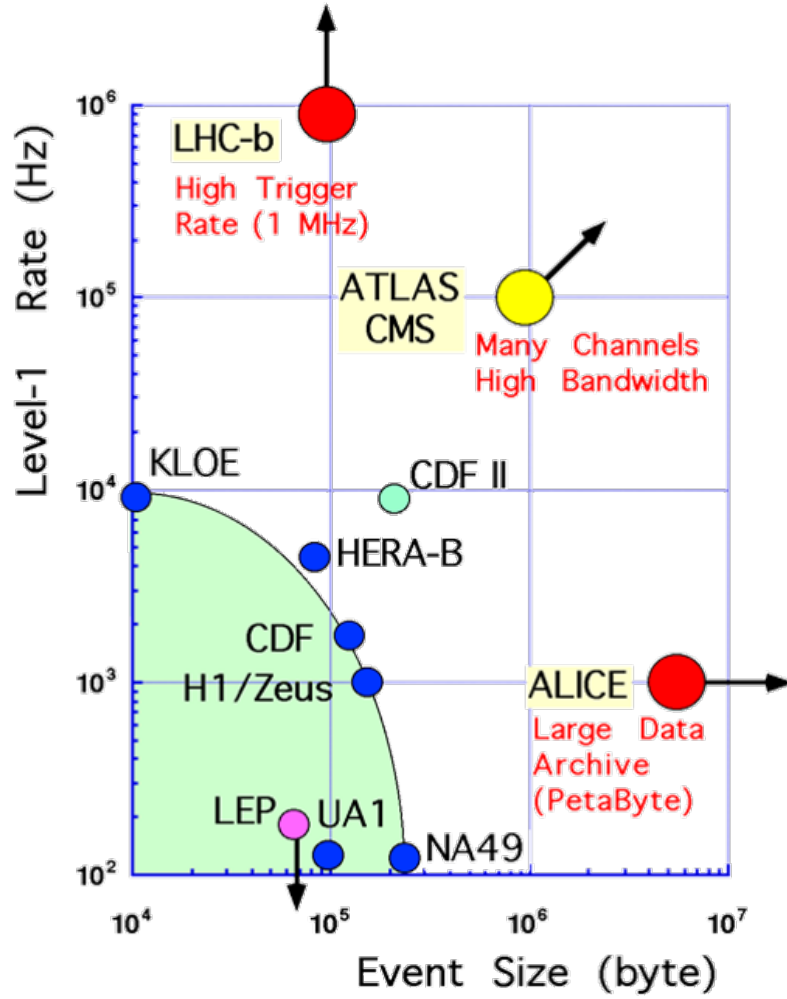


Figure 1.4: HEP experiments with varying Trigger Rate vs Event Size

The experiments at the LHC produce data that are soaring in volume, variety, velocity and varsiety. Figure 1.4 shows how different LHC experiments exhibit different data flow behaviour. Hence, each experiment needs a dedicated data balancing architecture to transfer the data from the detector readout electronics to the data storage space. The data management architecture needs new electronic systems and computing instrumentation techniques to control the data flow. *In the dissertation work the entire research is focused on the Common Readout Unit (CRU), a dedicated data balancing block required for an upgraded ALICE detector opting for triggerless architecture.*

1.2 The ALICE experiment

The very purpose of A Large Ion Collider Experiment (ALICE) is to study quark-gluon phase in the laboratory and learn more about the conditions that prevailed in the early Universe. In the year 1974, Prof. T.D. Lee (received Nobel Prize in Physics in 1957) had indeed realized that it might be possible to create a QGP like conditions in the laboratory by distributing very high energy, thus liberating quarks for a short time, over a relatively large volume. The QGP matter is formed in the laboratory by colliding heavy-ions at a speed close to that of light (at ultra-relativistic energies). In the process, it is possible to compress and heat the colliding nuclei in a way that the individual protons and the neutrons superimpose with each other. Hence, a region of high energy density could be created, consisting of regions where a relatively large number of de-confined quarks and gluons existed for a brief time. The QGP stage then reverts to the hadronic matter through expansion and cooling. The transition from the QGP state to that of the hadronic matter (and vice-versa) can be understood in terms of the phase diagram, as shown in the Figure 1.5. In the collider experiment of ALICE, the high-temperature phase is used to examine the properties of the QCD.

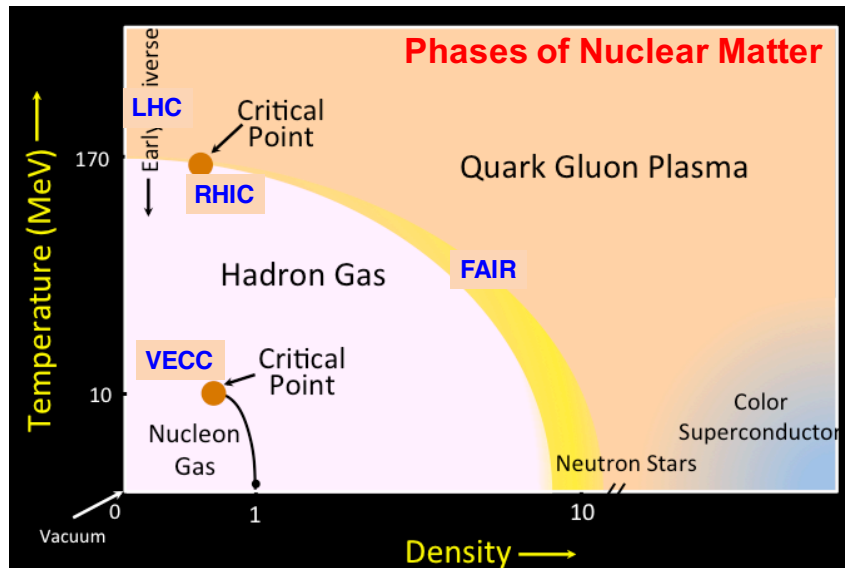


Figure 1.5: Phase diagram of Quark-Gluon Plasma (QCD)

1.3 ALICE Detector Upgrade

The LHC is going for its next major upgrade in 2020. The LHC has already attained the maximum energy of 13 TeV centre-of-mass energy in 2015 for proton-proton collisions and 5.0 TeV per nucleon in the case of Pb-Pb collisions. From the year 2020 onwards, High-Luminosity Large Hadron Collider (HL-LHC) will be operational whose main objective is to increase the luminosity of the machine by a large factor.

To fully exploit the physics potential provided by the machine, the ALICE experiment has decided to go for a major upgrade before the start of the third phase of LHC running (RUN3). Motivated by the successful physics results and past operation experiences, R&Ds for the ALICE upgrade have started. The manuscript presents how the change in physics objective has affected the data rate, that resulted in a new electronic block development called the **Common Readout Unit (CRU)** to act as a nodal point for data, control, and trigger distribution. Figure 1.6 shows how the ALICE major upgrade timeline is aligned with the LHC luminosity upgrade road-map.

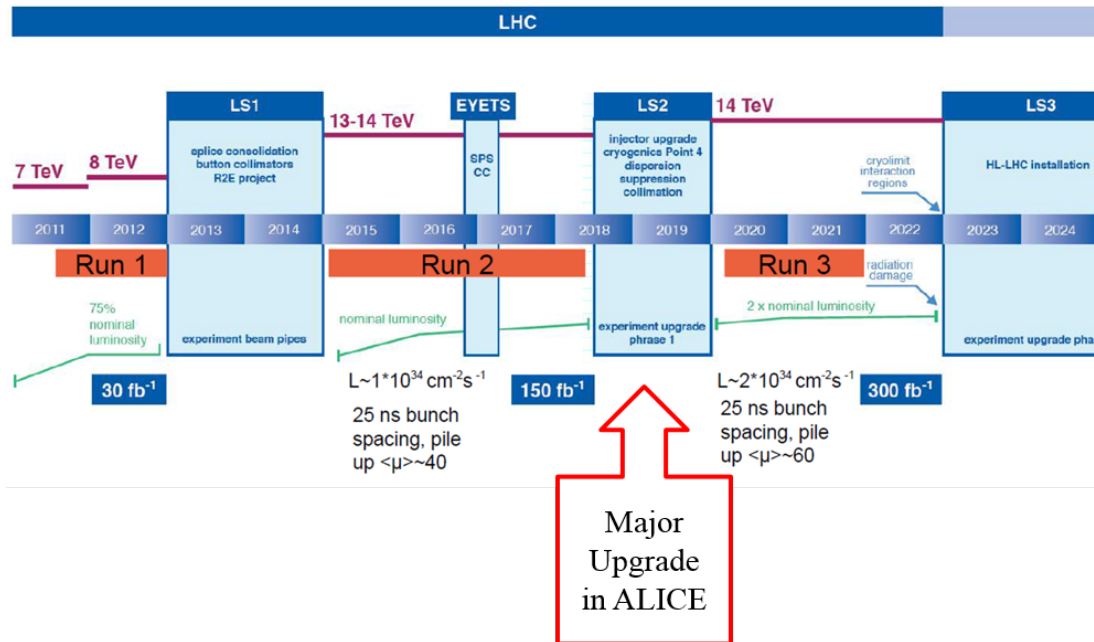


Figure 1.6: PHASE 1 major upgrade in ALICE to prepare for RUN3 and HL-LHC

For collider experiments, the instantaneous luminosity and integrated luminosity are important parameters to characterize its performance. As the LHC is aiming for higher luminosity, it means more number of events [6] will be generated over the experiment runtime as evident from the above expressions. Precision instrumentation of the ALICE detector is required for proper exploration of the high-intensity physics frontier. Exploration of the rare events require large event statistics with improved vertexing and tracking with optimum detector resolution. After the planned upgrade the readout will be capable of handling anticipated interaction rate of 50 kHz for Pb-Pb events and 200 kHz for pp and p-Pb events, resulting in a peak data flow traffic of about ~ 3.3 TB/s. Figure 1.7 shows the detectors that are going for the major upgrade as decided by ALICE collaboration.

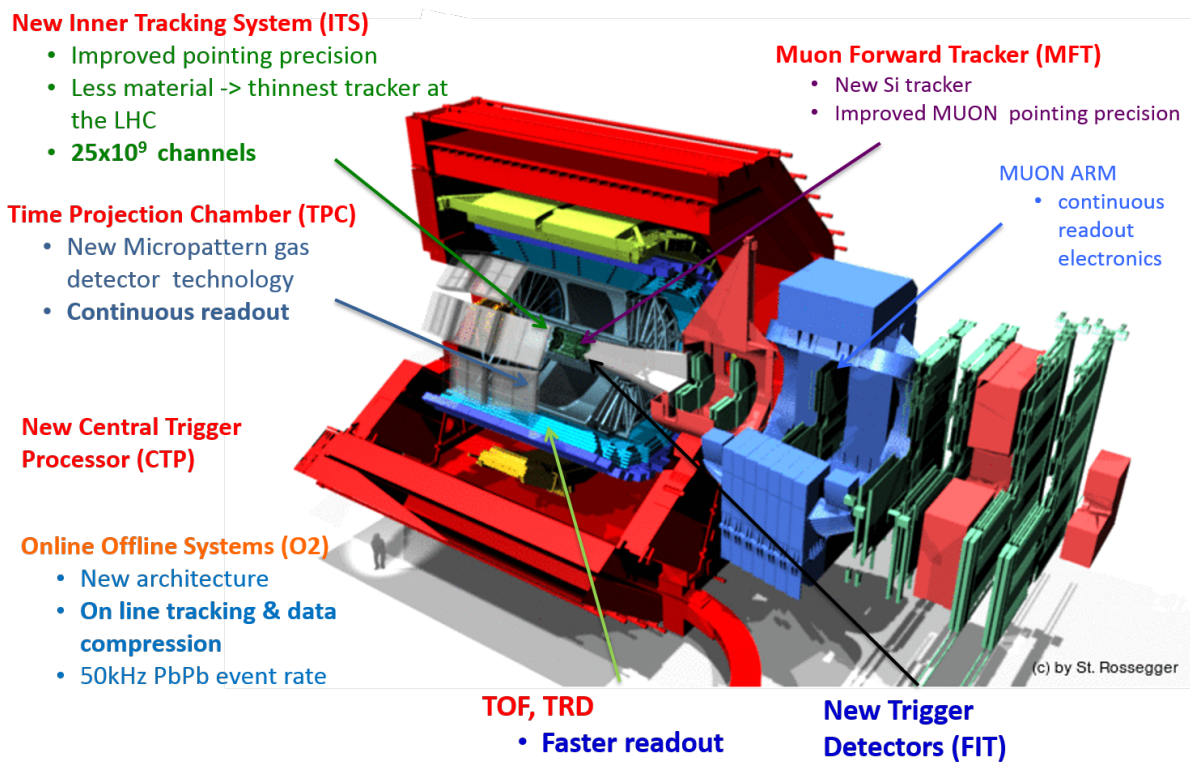


Figure 1.7: ALICE Upgrade from 2021

The ALICE [7] at the LHC is a multi-purpose detector designed to study the formation and evolution of nuclear matter at high temperatures and energy densities. Recording of data for pp, p-Pb and Pb-Pb collisions is ongoing by the ALICE since the beginning of

the LHC program (from the year 2010). The ALICE collaboration plans to extend its physics goals and technical capabilities, in order to fully exploit the scientific potential of the upcoming 2021 Run 3 and 2025 HL-LHC (High-Luminosity Large Hadron Collider) installations [8]. The road-map for the upgrade strategy is planned under the assumption that after the second long shutdown (LS2) in 2019, the LHC will progressively increase its luminosity with Pb beams eventually reaching an interaction rate of about 50 kHz from the existing 8 kHz, and instantaneous luminosities of $\mathcal{L} = 6 \times 10^{27} \text{ cm}^{-2}\text{s}^{-1}$, at a centre-of-mass energy $\sqrt{s_{\text{NN}}} = 5.5 \text{ TeV}$ for Pb-Pb beams. The experiment will then be in a position to accumulate 10 nb^{-1} of Pb-Pb collisions; corresponding to about 10^{11} interactions. Table 1.2 gives the LHC running scenario from the year 2021 onwards. For pp and Pb-Pb systems the table shows the top beam energies, peak luminosities, time between bunch collisions, collision rate and expected number of events.

Table 1.2: Running scenario for the LHC Runs 3 and 4

System	$\sqrt{s_{\text{NN}}}$ (TeV)	$\mathcal{L}$ ($\text{cm}^{-2}\text{s}^{-1}$)	Time between Bunch Collisions (ns)	Collision Rate (kHz)	Targeted record for Integrated Luminosity	Approximated N_{event}
<i>pp</i>	14	5×10^{34}	24.95	200	$\geq 6 \text{ pb}^{-1}$	1.4×10^{11}
<i>Pb - Pb</i>	5.5	6×10^{27}	49.9	50	$\geq 10 \text{ nb}^{-1}$	8×10^{10}

1.4 ALICE Technical Upgrade

A critical component of electronics and computing farm in the High Energy Physics (HEP) experiments is to decide on which data to store and what to discard. In such experiments, the rate at which detector data is sampled is much higher than the rate of physics interactions of primary interest. Here trigger decisions play an essential role in the decision on data taking. The detector dead time, the busy signal and the trigger decisions affect the data taking rate. In the upgraded architecture, it is decided to acquire data with a marked time-stamp in continuous mode and store on to the computing farms for online processing, where trigger decisions are applied for proper physics event selections.

The new process of data acquisition prevents data samples from being lost. However, there are provisions kept in the new design for the non-upgraded detectors to use the old technical links and readout architectures.

The paradigm shift in the readout strategy calls for the ALICE to develop a new readout architecture framework with more parallelism, compact layout, and balanced load distribution. The search for a new electronic system led to the proposal for the use of a data balancing block called the CRU. The CRU is dedicated to the trigger distribution, the data aggregation, and the detector control moderation. To keep pace with the future needs and demands in HEP experiments, there is a growing interest in the use of reconfigurable hardware like Field Programmable Gate Array (FPGA). The application of the reconfigurability feature allows faster development time with no upfront non-recurring expenses (NRE) for future upgrades having more predictable project cycle and field re-programmability. Hence, the DAQ boards that employ FPGA and meet the CRU firmware requirements are given preference.

1.5 Evolution of the ALICE Detector Data Readout Cards

In ALICE, the readout cards had been developed to cater to specific sub-detector requirements in the ALICE. D-RORC for the DAQ systems, H-RORC for the HLT trigger systems and RCU for the sub-detector readout systems. The ease of maintainability and development efficiency raised necessity to design a board that can serve the common interest of the multiple users. The aspiration led to the development of the 2nd generation of readout card called the C-RORC. The C-RORC can serve the purpose of both the DAQ and the HLT systems. However, the sub-detector systems used separate custom boards.

ALICE during RUN1 used the Detector Data Link (DDL) to transfer data between the detectors and the DAQ system. First version of the DDL referred as DDL1 [9], operated at the link speed of 2.125 Gb/s in full duplex mode. Communication channel of DDL1 consists of the Source Interface Unit (SIU), a mezzanine card embedded in the Front End Electronics (FEEs) of the detector; the Destination Interface Unit (DIU), is a firmware programmed design implemented on an FPGA based Read-Out Receiver Card (RORC) [10] installed in the DAQ farm. The physical medium is formed of multi-mode OM2 optical fibre having variable length that can run up to 200m long. Two versions of the RORC were developed for the RUN1. The first version is called the PCI-based Read-out Receiver Card (PRORC) [11] and the second version is called the DAQ Read-Out Receiver Card (D-RORC) [10]. The PRORC got 32 bit/33 MHz Peripheral Component Interconnect (PCI) interface realized using a commercial ASIC of type AMCC S5935, having ALTERA APEX EP20K200 as Programmable Logic Device (PLD) type. The second version, D-RORC got 64 bit/66 MHz PCI interface realized using an Intellectual Property (IP) implemented on the APEX-E PLD [12].

After several R&Ds, two types of D-RORCs that found application in the experiment [10]:

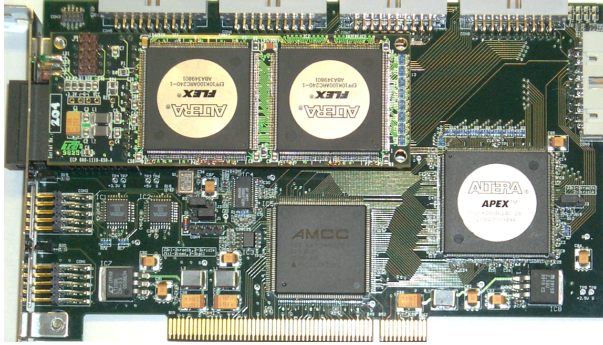
- Dual channel PCI-X D-RORC with 64bit / 100MHz PCI-X bus interface having ALTERA STRATIX 2 FPGA.
- Dual channel PCI-X D-RORC with x4 PCI (Gen. 1) bus interface having ALTERA STRATIX 2 Gx FPGA.

Apart from D-RORCs, the FPGA based co-processor architectures are also used in trigger distribution like the High Level trigger (HLT) RORC (H-RORC) and for raw data readout from detector is the Readout Controller Units (RCUs) in the RUN1. Figure 1.8 illustrates the lists of readout cards used during all the RUNs.

ALICE during RUN2 used the Common Read-Out Receiver Card (C-RORC) [13]. The C-RORC uses Xilinx Virtex-6 FPGA having support for a Peripheral Component Interconnect Express (PCIe) Gen2 (40 Gbps) and 12 links of DDL1 (2.125 Gb/s) or DDL2 (4.25 Gb/s up to 5.3125 Gb/s) [9]. The purpose for the C-RORC development is to increase the processing capabilities in the RUN2, and to develop a common readout card for the ALICE DAQ and the ALICE high level trigger systems to meet the common necessity. However, the detector readout cards continued to be a separate development project, called as the RCU2 [14].

ALICE from RUN3 to further increase in the processing capabilities of the link rate, have lead to the development of the Common Readout Unit (CRU) [15]. From the third generation of the readout cards, a huge leap forward has been taken in the design of the board such that the design can simultaneously meet the requirement of multiple experiments. The generic hardware developed is termed as the PCIe40 DAQ Engine [16]. The PCIe40 DAQ Engine is programmed with the CRU firmware, to act as common data distribution and balancing unit between the DAQ systems, the trigger systems and the multiple sub-detector systems. The CRU is a merged version of two readout boards used in the RUN2, those are the C-RORC and the RCU. The hardware development shares a common effort between the ALICE and the LHCb experiment. The readout board PCIe40 that is based on x16 PCIe Gen3 standard gets directly plugged into the DAQ server. Data from the front end are transmitted using the GBT links (4.8 Gbps link speed) to the PCIe40 and then forwarded to the DAQ systems. The maximum number of GBT links supported in a PCIe40 DAQ Engine is *forty-eight*.

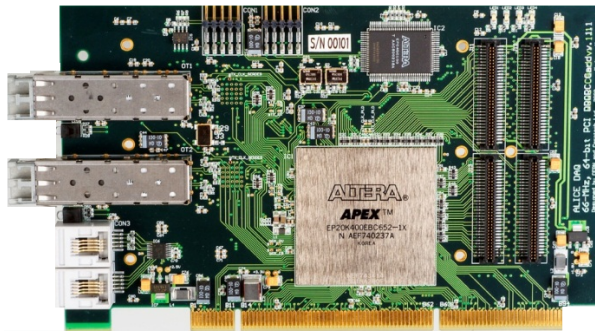
1.5. EVOLUTION OF THE ALICE DETECTOR DATA READOUT CARDS



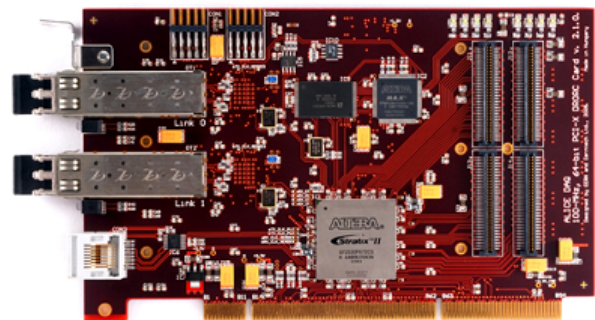
ALICE RUN 1
pRORC
(PCIe Read-Out Receiver Card)



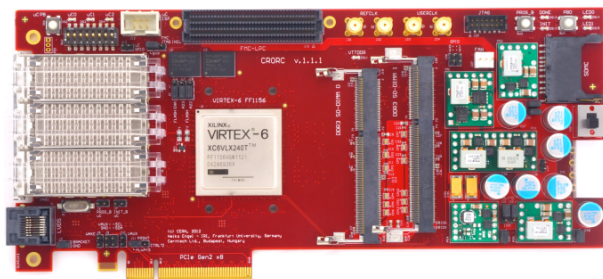
ALICE RUN 1
H-RORC
(HLT Read-Out Receiver Card)



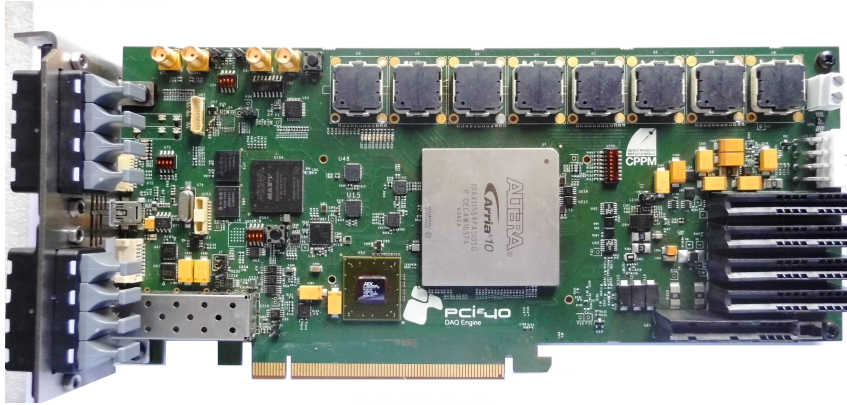
ALICE RUN 1
D-RORC
(DAQ Read-Out Receiver Card type 1)
2.125 Gb/s custom DDL
64bit / 100MHz PCI-X



ALICE RUN 1
D-RORC
(DAQ Read-Out Receiver Card type 2)
2.125 Gb/s custom DDL
x4, Gen1 PCIE



ALICE RUN 2
C-RORC
(Common Read-Out Receiver Card)
4.25 Gb/s custom DDL2
x8, Gen2 PCIE



**ALICE RUN 3
CRU**
(Common Readout Unit)
Board name : PCIe40
4.8 Gb/s GBT
x16, Gen3 PCIE

Figure 1.8: ALICE Data Link Receiver Cards in use during different RUN

1.6 Aim and scope of the thesis

The thesis aims to discuss the research and engineering work involved in the context of the CRU firmware development. Particular emphasis is attached to the study of trigger and timing signal communication through the CRU module. Other engineering solutions developed during the R&D phase of the CRU are also covered. The study objectives include:

- (i) Description of the design purpose for a new electronic system for the ALICE high rate detector upgrade called the Common Readout Unit (CRU).
- (ii) Elaborates on the selection of candidate board for the CRU firmware and resource utilisation of sub-modules.
- (iii) Development, characterization and evaluation of the CRU interface links; the GBT and the TTC-PON.
- (iv) Investigates the combined effect of the TTC-PON and the GBT interface links on the timing, trigger and control (TTC) communication in the context of the CRU hardware.
- (v) Studies the effect of thermal variation on the performance of the jitter cleaning PLL used in the CRU PCIe40 DAQ Engine.

- (vi) Delineates the engineering solutions needed for a phase measurement system within a CRU card having resolution and accuracy in the range of few picoseconds.
- (vii) Give details of the design principles needed to develop a cyclic redundancy checksum for data integrity check over a very high throughput data with a constraint of low processing latency.
- (viii) Highlights the need for defining a generalized standard of data balancing blocks for the HEP experiments.

1.7 Structure of the thesis

The dissertation is structured into *ten* chapters to discuss the development work related to the Common Readout Unit (CRU). Conceptually the work has been stratified into *three* parts.

The first part involves the CRU design abstraction, theoretical explanation, related firmware development and the qualification tests.

- *Chapter 1* discusses the motivation for the ALICE upgrade and the paradigm shift towards triggerless readout architecture from RUN3 onwards. The evolution of ALICE electronic system is explained with illustrative figures.
- *Chapter 2* introduces the concept of Common Readout Unit (CRU) and gives an overview of the interfacing details with the neighbouring blocks. A top-level view of the firmware architecture with major connection details is given. The strategy for the hardware selection to house the CRU firmware logic is presented. Moreover, a coarse resource estimation is done apriori to estimate the percentage of resources available to the detector specific logic with the periphery logics implemented.
- *Chapter 3* presents the main features and implementation details of the Giga-Bit Transceiver optical link (GBT) which is one of the interfacing links in the CRU

hardware PCIe40 DAQ Engine. The number of GBT links needed per CRU board is 24 while maximum occupancy supported is 48 links. A resource estimation with multiple links in different configurations is done to predict the percentage of logic resources utilised in the fitting of the firmware. The GBT link is responsible for timing, trigger and control (TTC) communication. Latency measurements are done to confirm that it remains constant throughout the design run-time. The signal integrity of the GBT is monitored using Bit Error Rate (BER) analysis and eye diagram. The initial version of the GBT implementation showed uncertainty in recovered clock phase latency that required calibration logic for error mitigation. However, in the final version, the calibration is not required provided the GBT operated at 240 MHz word clock that allows feedback compensated mode, details are discussed in chapter 4.

- *Chapter 4* gives an extensive overview of tools and techniques needed for characterizing the trigger and timing information distribution through CRU using TTC-PON GBT bridge. TTC information for communication uses the concept of Trigger and Timing System (TTS) over Fast serial Trigger Links (FTL). Apart from Peripheral Component Interconnect Express (PCIe), CRU got two interface links namely TTC-PON and GBT that supports TTS-FTL communication. Elaborate evaluation tests are included and the suitability of the standards to meet the criticality of TTC information transmission is discussed.
- *Chapter 5* further elaborates on the role of jitter cleaning Phase Locked Loop (PLL) in ALICE experiment as a necessary aid to meet the timing requirement. A study of the behaviour of the PLL with variation in temperature on the performance and the effect of multiple soft/hard resets on phase consistency are examined.

The second part of the thesis discusses the auxiliary toolkit developed as a supporting framework for the CRU functionality.

- *Chapter 6* is devoted to the sophisticated technical principles and instrumentation technique involved in building an ultra-sensitive phase measurement system inside an FPGA. A detailed analytical study is included, to show that resolution in the range of few picoseconds is achievable using the design technique. The module finds application as one of the monitoring tools of clock phases inside the CRU FPGA board.
- *Chapter 7* presents a configurable VLSI design architecture for the CRC-32 checksum used in the CRU as the second level of data integrity checker for the data packets. The throughput of incoming/outgoing data in CRU is the range of gigabytes and the tolerable latency for processing is low. Keeping the constraints in view, the body of the chapter describes the development architecture of a CRC-32 design having an ability to operate on 10-4.8 Gbps links with a unit operational latency.

The third part of the dissertation highlights the need for development in the domain of high speed link standards and the inter-connectivity of readout unit architectures. Two works are presented in the discourse of the study that addresses the requirement and provides a new skeleton of development.

- *Chapter 8* focuses on the security vulnerability involved in local interface connections within ALICE and the need for better error detection capability over error correction. There happens to be a trade off between error detection capability and channel bandwidth. To address the mentioned necessities, a secure error resilient high speed link standard is developed.
- *Chapter 9* highlights the importance of understanding the data flow model in High Energy Physics (HEP) readout architectures. Typically the data type is of three kinds, namely trigger and timing information, detector data information and monitoring, control and configuration information. Multiple HEP experiments have similar requirements as ALICE and build a customised data balancing block identical to

the CRU. The scope of the chapter is to bring forward the need for a common design specification and standards that needed to be followed. The concept is illustrated with a custom design block called the Data Aggregation Unit (DAU).

The thesis addresses a wide spectrum of techniques and solutions for application of reconfigurable hardware in HEP. Most of the techniques can also be used or adapted for other domains. An overall summary of the thesis is provided in *Chapter 10*.

1.8 My Contributions

Considering the volume and versatility of the engineering work involved in the CRU design, the work embodied in the study solely discusses the development work where I have participated as a lead researcher.

Chapter 1 is the introductory part, where I have discussed the specification set by the ALICE collaboration for the CRU module.

From Chapter 2 to 5, I have elaborated on the engineering design and testing of the CRU hardware that involves the joint effort of the CRU team members. The assignments where I have acted as a lead researcher and as a primary author of the documentation/publications are only included in these chapters.

Chapter 6 to 9 is my individual development and credit of complete originality of the designs solely lies with me.

As a remark, it is important for me to acknowledge the constant inspiration and motivation provided while working for the ALICE collaboration, without which the work would not have achieved the current stature.

Chapter 2

Common Readout Unit in ALICE

2.1 Introduction

Precision instrumentation of the ALICE detector is required for proper exploration of high-intensity physics frontiers like QCD evolution, jet measurements, and search for heavy nuclear states [8]. These rare event studies require large event statistics. Improved vertexing and tracking capability require increased detector resolution. Together this demands higher interaction rate and large event size data payload. Combined data flow from the detector to the online system will exceed $\sim 3\text{TB/s}$. The current trigger readout system limits the efficiency of data taking at a high rate. The new design must have the option for continuous readout. However, there will be some detectors that will stick to triggered readout mode. So, provisions for triggered readout must also be kept. Care has been taken such that under nominal conditions the detector do not get busy. In the new architecture the detector only gets busy in case of an error condition or during operation under non-nominal conditions.

The paradigm shift in read-out strategy calls for ALICE to develop a new design framework with more parallelism, compact layout and balanced load distribution. For efficient load balancing a special common data processing block is partitioned out from

Online-Offline computing system called Common Readout Unit (CRU). It is dedicated for trigger distribution, data aggregation, and detector control moderation. CRU uses reconfigurable hardware Field Programmable Gate Array (FPGA) for its development. The compactness of the design is maintained by limited system granularity, equally distributed functionality and limiting the cable involvement by usage of array based optical connections, time-multiplexed serial transmission or cable free interface scheme.

In this chapter, I elaborate and discuss the CRU idea from its inception to its initial resource estimation results. Section 2.2 elaborates on the location of CRU in ALICE experiment. A details about the readout configuration is explained in Section 2.3. The number of CRU modules needed for different sub-detector systems is highlighted in Section 2.4. Section 2.5 deals with CRU placement in the ALICE Experiment and the interface standards used for communication with neighbourhood functional blocks. Sections 2.6, 2.7, and 2.8 discusses in brief the three types of CRU functional roles. An integrated firmware design architecture for CRU is proposed in Section 2.9. The selection and survey of CRU hardware suitable to the CRU firmware requirement are done in Section 2.10. Finally, initial test results of the CRU unit functional modules are discussed in Section 2.11.

2.2 CRU Location in the ALICE experiment

CRU acts as a common interface between the ALICE on-detector electronic system, the Online and Offline (O2) computing system and the Central Trigger Processor (CTP). Being the central element, CRU has to handle three types of data traffic which include detector data, trigger and timing information and control instructions. There has been an option to keep the CRU either in the cavern or the counting room as shown in Figure 2.1a and 2.1b. Location 1 shows CRU placed in Cavern at critical radiation zone, whereas Location 2 shows CRU placed in Counting room (CR4) in the PX24 shaft at controlled

radiation zone. The location choice depends on three parameters: the amount of cabling required, radiation hardness of FPGA boards needed and scope for future maintenance.

Consider the 1st location site for CRU. Here, because of the proximity to radiation zone, the CRU DAQ board need to be radiation hard. It means use of radiation hard FPGAs are required. These radiation-hardened FPGA processing technology are still many generations behind state-of-the-art commercial IC processes. For example, the rad-hard FPGAs are in the 65-nm or less-dense process nodes, whereas commercial grade FPGAs have gone down to 14-nm FINFET Technology. Now these carries a drawback, that number of logic cells available for programming are much lower than that of commercial grade FPGAs. Besides popularly used digital Single Event Upset (SEU) mitigation technique is Triple Modular Redundancy (TMR) circuits or voting logic, which further lowers the available logic resources. For these reasons the total resource available for user logic development is lower than that of the commercial grade FPGAs. The location-1, however, got some advantages over location-2, like minimum cable length required between Detector to CRU and CTP to CRU.

Now, consider the 2nd location site for CRU. Here in controlled radiation zone it is possible to use the latest and the most advanced FPGA chip available in the market and build firmware on it. The location point also provides an easy hardware access for maintenance. However, the site also has a drawback. The length of cabling required from cavern to the counting room is roughly 75 m and involves cabling of 8344 links from sub-detectors digitized readout channels. For each optical fibre cable there involves a transmission latency of ~ 367 ns or 15 ($= 367 / 25$) LHC clock cycles. So, it clearly means the trigger information pathway between CTP-CRU-Detector are suitable for triggers whose allowed *latency* $> 2 \times (367 \text{ ns} + \text{Asynchronous Serial Protocol Serialization/De-serialization latency})$. It is multiplied by factor 2 to account for traversal time of the signal from CTP-CRU and back to CRU-Detector. Hence, to communicate those fast critical triggers they need to be connected directly from CTP to the sub-detectors. Altogether, cable needed is much

more than location 1.

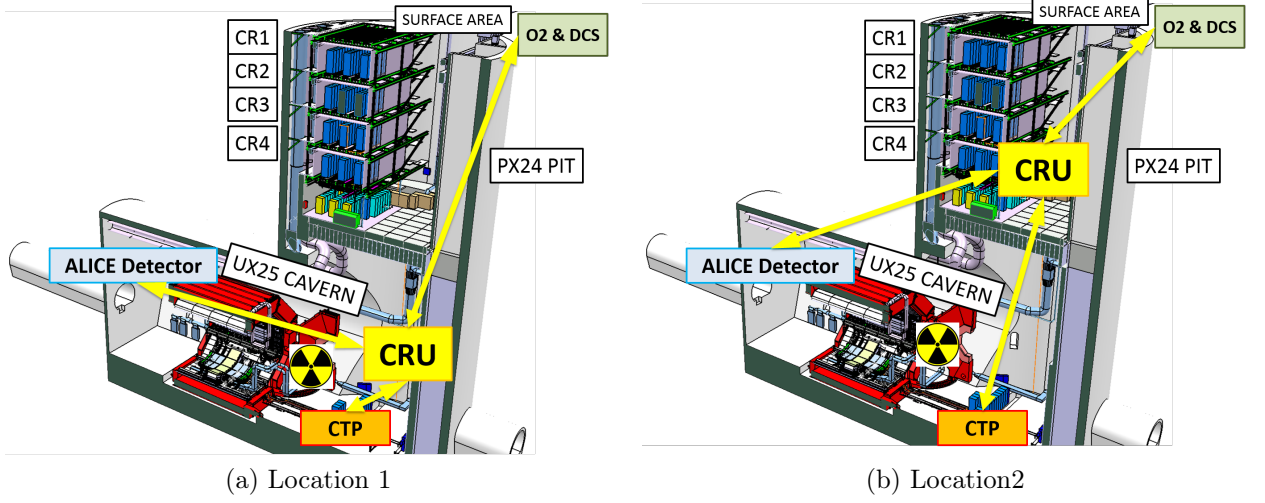


Figure 2.1: CRU location in the ALICE experiment (not in scale)

From Run3 the ALICE experiment is moving towards continuous readout architecture. In that case trigger and timing information will not be latency critical, and long asynchronous links (like GBT [17], PON [18]) can be used for trigger transmission. However, it would remain critical for sub-detectors that still depend on trigger based architecture like legacy sub-detectors or upgraded detectors operating in triggered mode for commissioning. The majority of the detector decided to operate in trigger-less architecture, based on the heartbeat software trigger that is used to designate the time frame boundaries for event building at the online computing system. For easy maintenance, the future firmware upgrades and debugging, easy accessibility is required, sometimes even in between experiment run-time data taking. Weighing all the pros and cons for both the location sites, location 2 has been decided as the suitable position for the CRU.

2.3 CRU Readout configuration

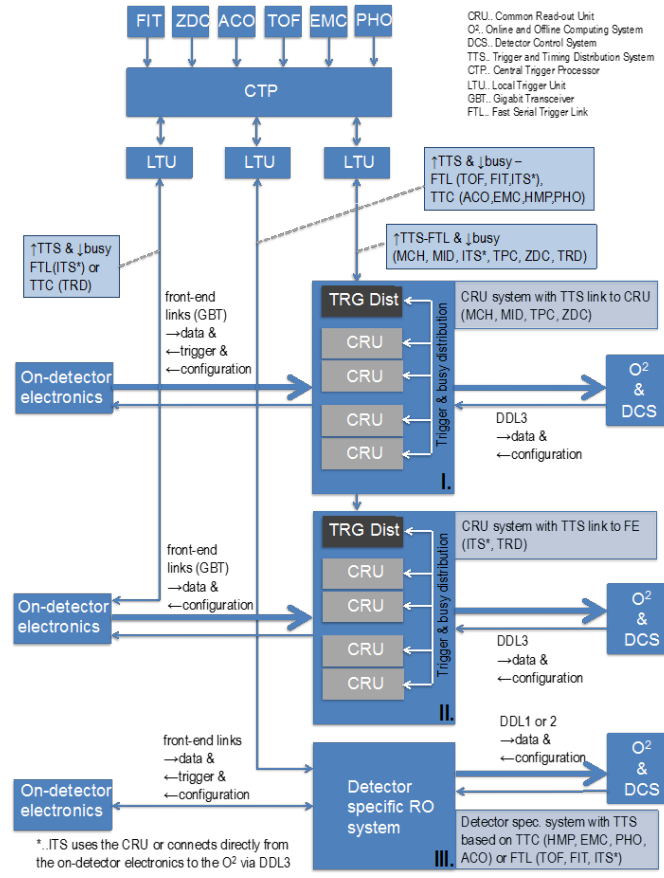
The major task of CRU functionality is to aggregate the incoming detector readout data over the Giga-Bit Transceiver optical link (GBT) links [19, 20] and forward it on to a

limited number of Detector Data Link (DDL) as compatible to the computing group requirement. HEP experiments are moving towards reconfigurable hardware like FPGAs. It has many advantages like faster development time, no upfront non-recurring expenses (NRE), more predictable project cycle and field re-programmability. A survey is conducted on FPGA-based DAQ boards that provide solutions to connect multiple optical channels for incoming data traffic and high bandwidth output channels for pushing the data to the computing systems. Two candidate boards are found to be suitable to match the CRU system requirement, namely the PCIe40 and the AMC40. PCIe40 is based on latest 20 nm Intel[®](formerly Altera) Arria[®] 10 FPGA, having provision for 48 bidirectional GBT links and 16 lanes PCIe channel lanes. AMC40 is based on 28 nm Altera Stratix V FPGA, having provision for 24 bidirectional GBT links and 12 bidirectional 10 Gbps links. As can be seen from Table 2.1, total ~3 TB/s of incoming data need to be pushed to the online system. For ease of maintainability, separate CRU's for each sub-detectors are employed. Each sub-detector will not use complete CRU hardware resources at its full occupancy for proper load distribution. Load distribution among CRU boards is critical, as it controls the amount of heat generated, system failure due to overload and efficient aggregation of events at the event builder of the online computing system. Therefore, a single CRU at maximum uses 24 GBT links even though the hardware has provision for 48 links.

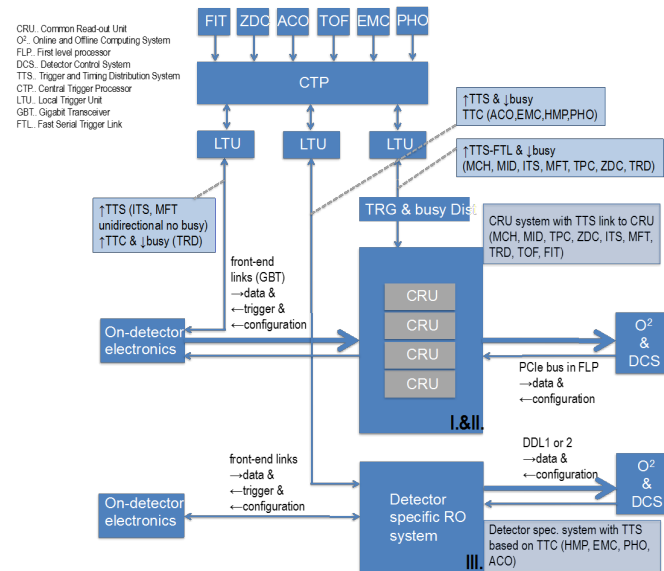
Both the FPGA boards are found to be suitable for ALICE. The choice depends on whether to go for Advanced Telecommunications Computing Architecture (ATCA) or Peripheral Component Interconnect Express (PCIe) based architecture. ATCA based architecture provides modularity for design framework and high-speed backplane for trigger and control information distribution among the CRU boards. The PCIe form factor needs no DDL link as it directly connects to the PCIe bus of the CPU system. However, there exists a risk, as PCs have very fast up-gradation cycle and whether presently selected PCIe Gen 3 slots would be supported in future is unclear. However, assurance has been

given by PCI-SIG community that PCIe Gen 3 provides legacy support for upcoming next 2 generations of PCIe. So, based on two form-factor of CRUs, there can be two types of readout configuration as shown in Figure 2.2. For details refer to ALICE Electronics Technical Design Report [21].

The major decision parameter is to select the FPGA board that has sufficient logic resources for detector data sorting, clustering and compressing. For Arria[®] 10 FPGA (in PCIe40) the number of logic resources is roughly double that of Stratix V FPGA (in AMC40). After implementing the periphery logic a comparative analysis is done to note which board is left over with more resources for detector core logic development as shown in Figure 2.3. Since Arria[®] 10 has PCIe hard IP whereas in Stratix V there is no hard IP for 10 Gigabit Ethernet IP, more logic building blocks are utilized in case of Stratix V. Clearly Arria[®] 10 is a good fit and hence PCIe40 has been adopted. Intel (Formerly Altera) also provides vertical migration for Arria[®] 10, which means when more advanced Stratix 10 FPGA will be available on the market same firmware and hardware board can be used over again, without any recurring development cost.



(a) Configuration 1



(b) Configuration 2

Figure 2.2: CRU Readout Configurations

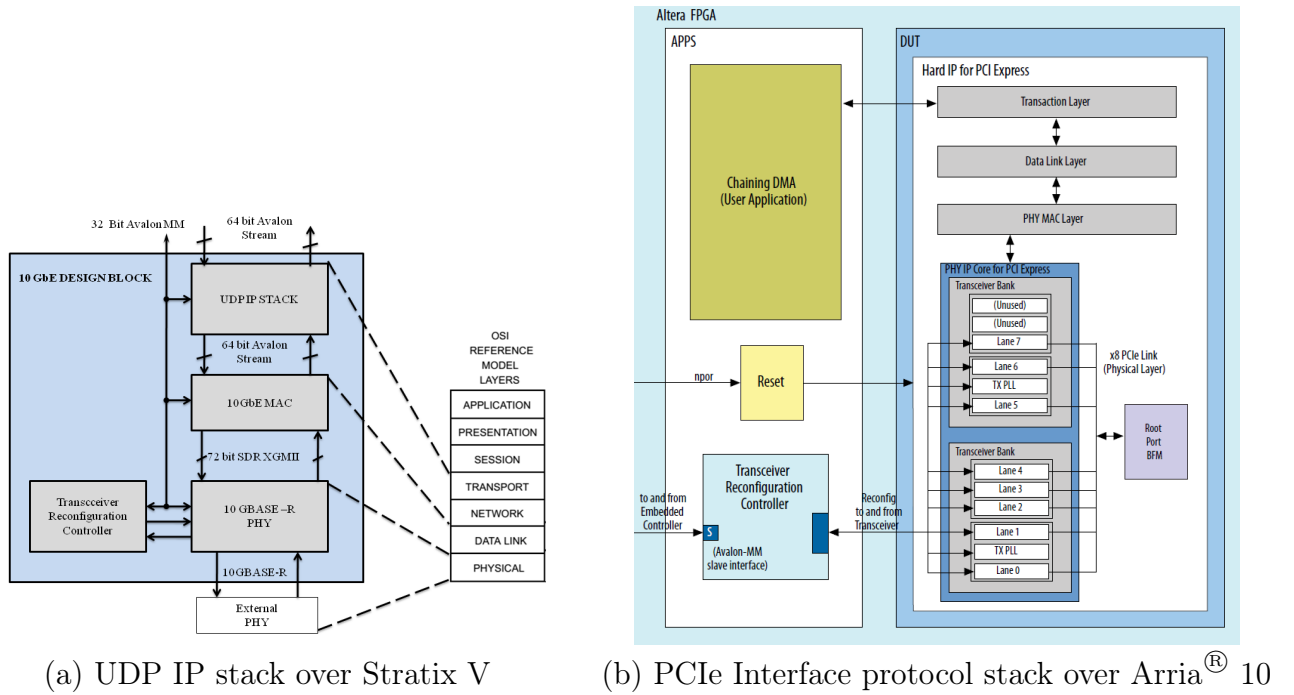


Figure 2.3: Showing the implementation of two protocol stack and its interface with user application layer

2.4 CRU Usage

Detectors that use the CRU are listed in Table 2.1. Other detectors not using CRU in their readout architecture is not included in the list. The table summarizes the link usage for each detector along with the number of CRU boards needed. Moreover, the link count includes CRU-FE links that carry hit data from the on-detector electronics to the CRU and the TTS-FE links that carry trigger data from the CRU to the on-detector electronics. In addition to those details, the table also includes the board type, the maximum readout rate and the data rate (for Pb-Pb collision) for each sub-detector.

Table 2.1: Detector Specific CRU usage

User Groups	FEE / Readout Boards	Maximum Readout Rate (kHz)	Data Rate for Pb-Pb (GB/s)	Link Type	Total number of Links	Number of CRU boards
CTP (Central Trigger Processor)	FPGA (Kintex 7)	200	0.02	GBT & TTC-PON	19+ 1	1
FIT (Fast Interaction Trigger)	FPGA (Virtex 6)	100	0.115	GBT	26	1
ITS (Inner Tracking System)	FPGA (Kintex 7)	100	45	GBT	192	8
MCH (Muon Chamber)	ASIC (SAMPA)	100	2.2	GBT	575	24
MFT (Muon Forward Tracker)	FPGA (Kintex 7)	100	10	GBT	80	4
MID (Muon Identifier)	FPGA (8x Max10, 2x Cyclone V)	100	0.3	GBT	32	2
TOF (Time Of Flight)	FPGA (IGLOO2)	200	2.5	GBT	72	3
TPC (Time Projection Chamber)	ASIC (SAMPA)	50	3670	GBT (Widebus Mode)	6552	360
TRD (Transition Radiation Detector)	FPGA	200	20	Custom	1044	29
ZDC (Zero Degree Calorimeter)	FPGA (Vertex 5,6)	100	0.06	GBT	1	1
Total			3750.135		8593+1	433

2.5 CRU Connectivity

The CRU is a common functional block between the on-detector systems, the Online - Offline computing system (O^2) and the Central Trigger Processor (CTP) as shown in the Figure 2.4. The link to the computing system requires an interface standard that has a very high throughput large bandwidth multi-channel support with guaranteed transmission capability. Since CRU will be connected to the server machine, it is obvious to select a commercial standard that has reasonable cost and has a large ecosystem with ample solutions. Hence, *PCI-Express*[®] (*PCIe*) bus is a viable solution to interface with the O^2 . The CTP deals with the timing and trigger information transmission. It needs a data pathway with high split ratio and fixed latency. *TTC-PON* standard [18] is considered for this interface. For on-detector electronic systems to transmit readout data with very high reliability from detector situated in a radiation zone to CRU situated in a non-radiation zone needs a new interface standard. The same link will also be used for

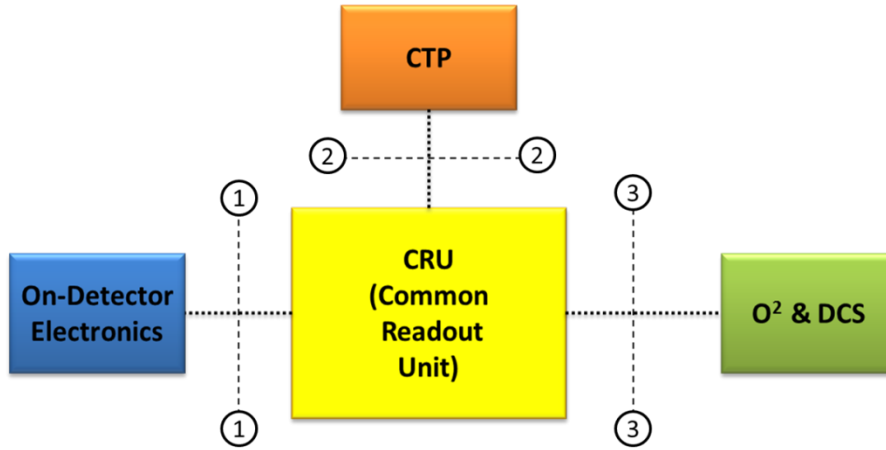


Figure 2.4: CRU interfaces with the neighbourhood blocks, which are (1) GBT; (2) TTC-PON; (3) PCIe[®] Gen 3

trigger distribution and control of detectors from the CRU boards. The *GBT* interface standard [19], a custom made asynchronous protocol for High Energy Physics (HEP) is used to communicate the different types of data traffic (detector data, trigger and timing information and control information) over the same channel. The comparison of all three standards are shown in Table 4.6.

Table 2.2: Comparison of three high speed interface protocol standards used in CRU

Parameters	GBT [20, 19, 22]	TTC-PON [18]	PCIe [®] Gen3 [23]
Technology Specification	Custom	XGPON1 with modifications	PCI Express [®] Base Specification Revision 3.1
Designer Group	CERN	ITU-T with CERN modifications	PCI-SIG
Line Rate	4.8 Gbps	Downstream: 9.6 Gbps Upstream: 2.4 Gbps	8 Gbps per lane
Payload Rate	3.2 Gbps	Downstream: ~7.68 Gbps Upstream: 640 Mbps	985 MB/s per lane
Payload Size	120 bits@40 MHz	Downstream: 192 bits@40 MHz Upstream: ~16 bits@40 MHz	256 bits@250 MHz for 8 lanes
Wavelength	850 nm (Multi-mode Tx) 1310 nm (Single-mode Tx)	Downstream: 1577 nm Upstream: 1270 nm	Not Applicable
Network Topology	Point-to-Point	Point-to-Multipoint	Point-to-Point
Encoding	RS Encoding with Block Interleaver	8b/10b	128b/130b
Synchronous Trigger Support	Yes	Yes	No
Trigger Latency	Optical loop-back Round-trip: 150 ns	Downstream: ~100 ns Upstream: 1.6 μ s	Not Applicable
Commercially Available	No	No	Yes

2.6 CRU as Trigger and Timing Distribution Unit

In RUN3, the upgraded detectors will be read-out in continuous, triggerless mode. The strategy is to select and readout all interactions and apply an online reduction in the computing systems. However, in order to provide backward compatibility to the non-upgraded detectors, a trigger system must exist to cope with detector dead-time during the read-out. For continuous readout detectors, the heartbeat software trigger is used to designate time frame boundaries for event building at the online computing system. It is also used for monitoring of the detector health status. The time frame duration of the heartbeat trigger is comparable to the sub-detector Time Projection Chamber (TPC) drift time, i.e. 100 ms. For non-upgraded detectors with trigger based readout, receives any one of the trigger levels (LM, L0, L1) based on the detector trigger latency requirement. The main trigger generates an interaction trigger provided by the First Interaction Trigger (FIT) [21], an LM signal which is complemented by trigger pulses generated from ALICE Cosmic Ray Detector (ACO), Electro Magnetic Calorimeter (EMC), Photon Spectrometer (PHO), Time Of Flight detector (TOF), Zero Degree Calorimeter (ZDC) to generate an L0 signal and also a delayed copy from EMC and ZDC is used to generate an L1 signal. Trigger inputs are collected to satisfy three different latencies, denoted by three different levels as LM, L0 and L1.

All these trigger decisions are provided to the ALICE sub-detectors by CTP. CTP uses the Local Trigger Unit (LTU) for communicating triggers and timing information to each sub-detectors, using Trigger and Timing System (TTS). In the new architecture, CTP uses 24 LTUs. Each LTU supports three types of trigger and timing distribution schemes to detector Front End Electronics (FEEs). The latency critical trigger (LM) and busy signals use LVDS cables to connect to the detector FEEs. For non-critical latency triggers (L0, L1, heartbeat) and other control signals different communication pathways are used that involves more latency. Two communication pathways are available: TTS-FTL and

TTS-TTC. For upgraded electronics the TTS pathway is based on asynchronous Fast serial Trigger Links (FTL). Detectors with non-upgraded FEEs or legacy detectors will keep using CERN's timing, trigger and control (TTC) standard [24] based equipments. In TTS-FTL mode, trigger is routed via CRU sitting in counting room to feed the detectors in the cavern. The pathway involves two asynchronous serial link transitions (TTC-PON and GBT), each operating in fixed latency mode. The downstream from CTP to CRU involves TTC-PON, which is time multiplexed trigger distribution topology. The downstream from CRU to LTU involves GBT protocol standard. The total routed path delay is assumed to be $1.5 \mu s$ for 150 m ($= 75 \text{ m} \times 2$) cable length. Figure 2.5 shows in detail three TTS distribution schemes and role of CRU for TTS-FTL communication.

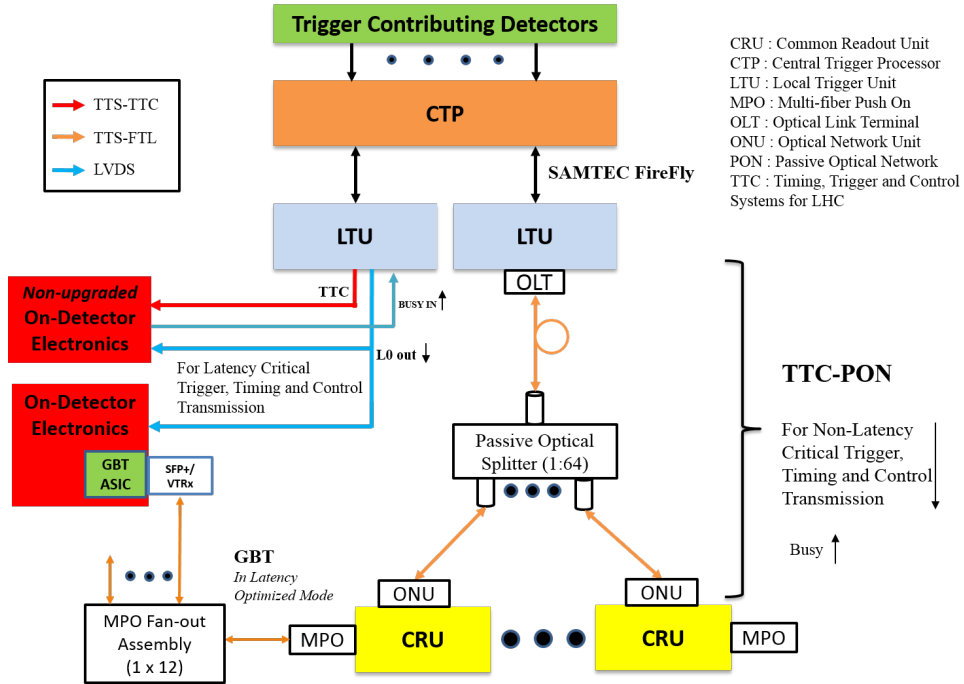


Figure 2.5: CRU as Trigger Distribution Unit

2.7 CRU as Data Aggregation Unit

The O^2 computing farm together with readout electronics involves a number of data processing steps. For uninterrupted error free dataflow from detector systems to data storage

2.8 CRU as Control Unit

The GBT Slow Control (GBT-SCA) ASIC 130 nm CMOS technology, part of the GBT chipset, has the purpose to distribute control and monitoring signals to the on-detector front-end electronics and perform monitoring operations of detector environmental parameters [26]. Detector Control Subsystem (DCS) will access the CRU through O² software layers. GBT-SCA ASIC uses GBTx ASIC to communicate with over GBT link standard in serial bit stream mode. The data link layer is based on two protocol layers: e-link transport protocol layer which uses bi-directional packet oriented High-level Data Link Control (HDLC) standard and SCA Channel Command-oriented Protocol (SCCP) for channel coupling/de-coupling from e-links. Figure 2.7 shows an example of a single link communication between detector side FEE slow control and DCS via CRU boards.

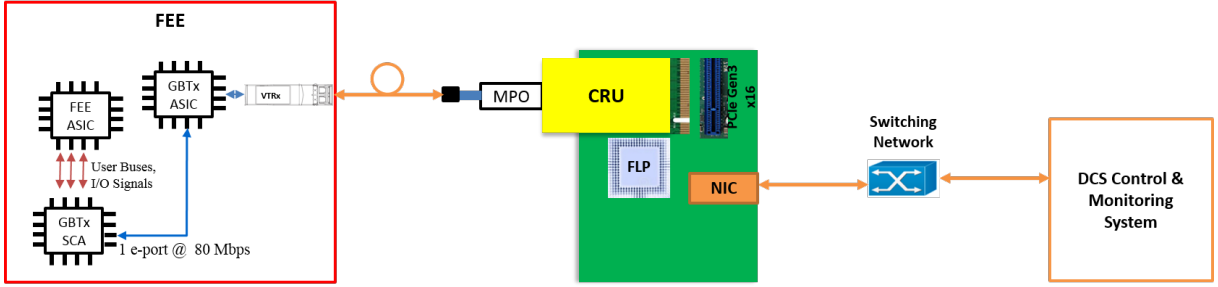


Figure 2.7: CRU as Detector Control mediator (for detectors that will use GBT chipset in FEEs)

2.9 CRU Firmware Architecture

CRU firmware architecture is divided into multiple functional modules as depicted in Figure 2.8. Broadly it can be sub-divided into two major groups: *Periphery Logic* and *Core Logic*. The Periphery Logic includes all the logic related to interface links (GBT, TTC-PON and PCIe[®]). The interface logic comprises physical sublayer (Native PHY x6, PCIe Hard IP, GPON PHY), coding sublayer (GBT PCS, TTC PON PCS, DMA Controller), application sublayer (HDLC, DSDP, SCCP, Flow Control), buffer memory blocks and

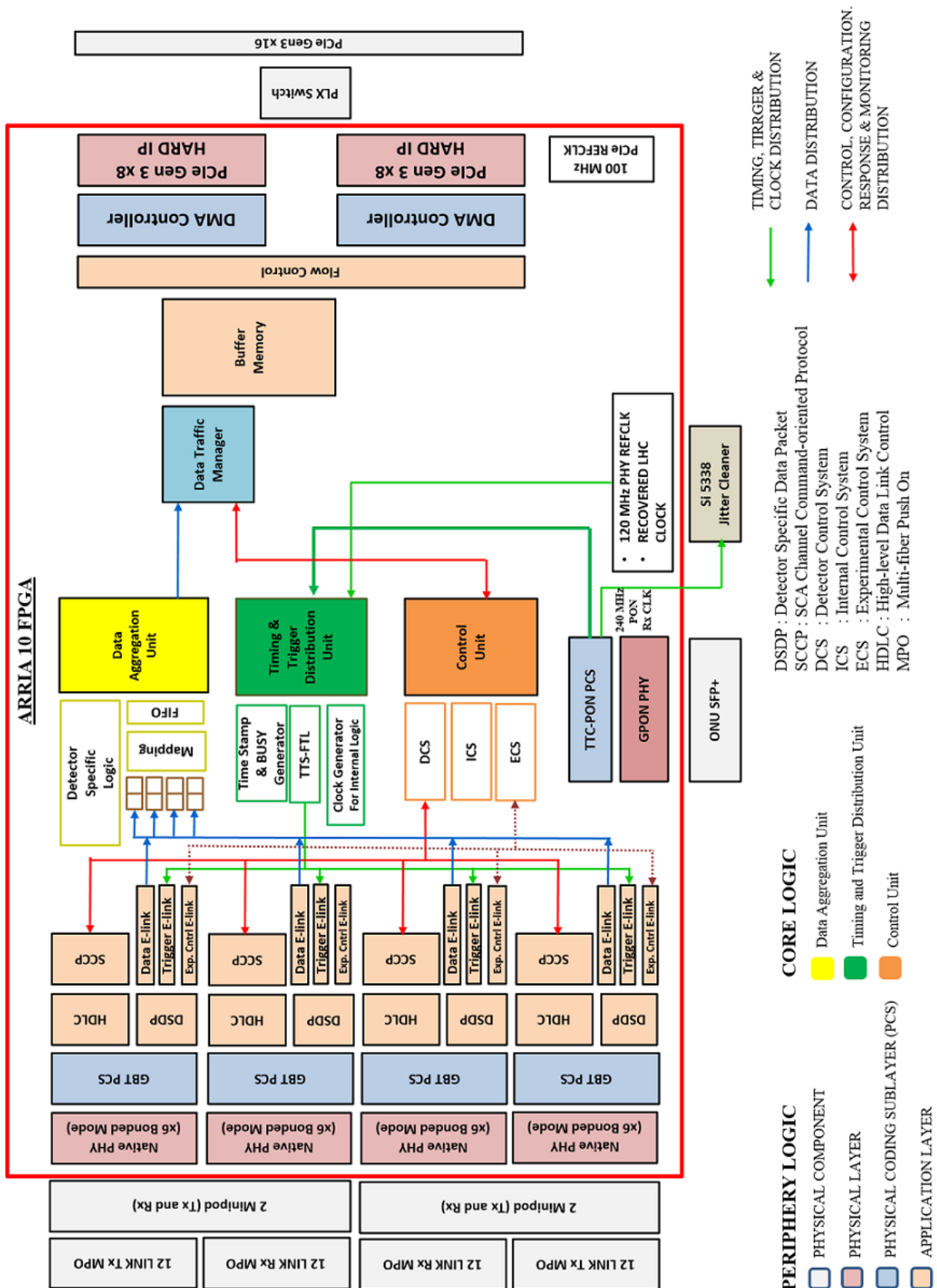


Figure 2.8: CRU internal logic block connections

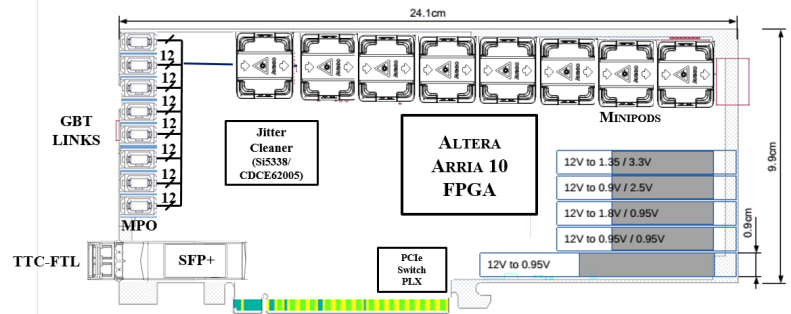
clocking resources. While the Core Logic group includes all the logics related to functional blocks namely Timing and Trigger Distribution Unit, Data Aggregation Unit, Control Unit and others. The Timing and Trigger Distribution Unit mainly focus on the TTS-FTL trigger forward, clock generation for driving internal logics and time-stamp marker for data packets. The Data Aggregation Unit decouples the data coming from respective data channels, buffers the data for detector specific logic to operate on and then forwards it to interface link wrapping in a specific packet format the aggregated data. The Control Unit acts as an access provider for detector control instructions, experimental control instructions and internal CRU control instructions to the DCS subsystem.

Apart from these design perspectives, CRU firmware development is also classified into two major groups: (1) Generic Firmware; (2) Detector Specific Firmware. The Generic Firmware includes all the Periphery Logic and Core Logic blocks that are common to all the sub-detector groups. Detector specific firmware restricts the Periphery Logic usage and the Core Logic needed to format, process and multiplex data according to the specific sub-detector requirements.

2.10 CRU hardware architecture



(a) PCIe40



(b) PCIe40 Schematic

Figure 2.9: CRU development

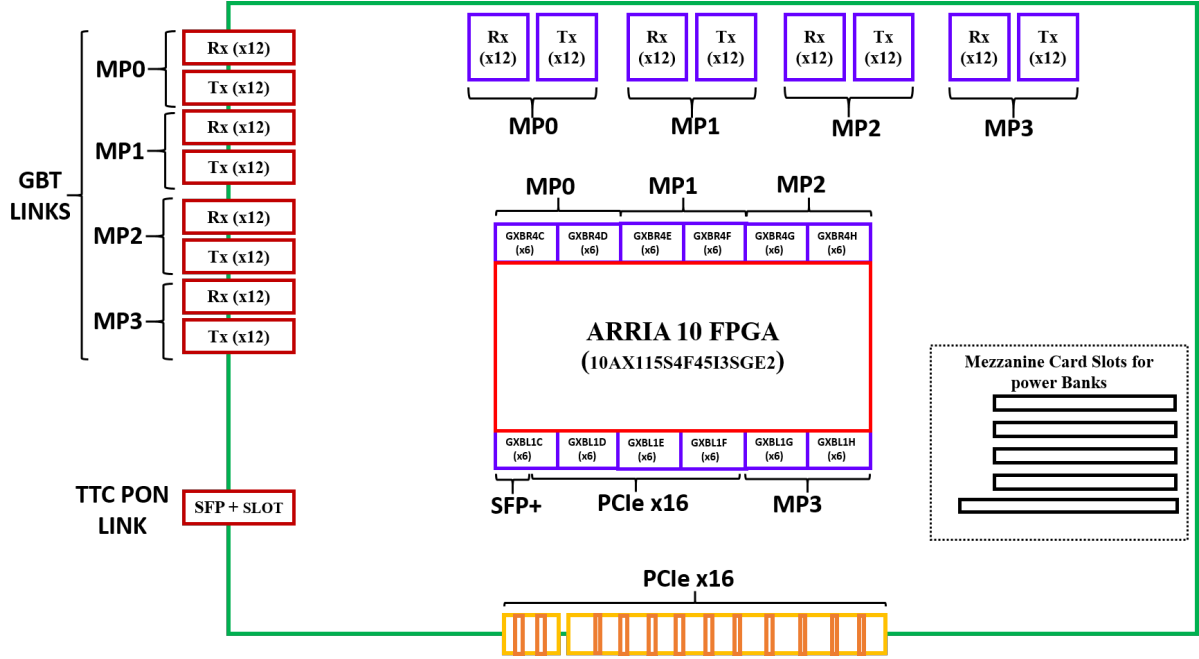


Figure 2.10: A generalized diagram showing optical link connection with FPGA bank for PCIe40

The CRU firmware is loaded on the PCIe40 DAQ Engine [16]. PCIe40 DAQ Engine is a PCI Express[®] Gen 3 based readout board. It is equipped with 49 high-speed bidirectional optical communication links, of which 48 links use MiniPod optical modules, and one link uses SFP+ form-factor optical module for establishing bidirectional optical communication. It directly connects to the motherboard through PCIe 16-lanes edge-connector. The board uses latest 20nm Arria[®] 10 midrange FPGA from Intel (Formerly Altera). PCIe40 has high-performance, low jitter clock generator CDCE62005 and the Si5338 chip embedded in it, useful for LHC clock recovery. Figure 2.9 shows a view of PCIe40 DAQ Engine along with its basic schematic. In Figure 2.9 (a) and 2.10, the PCIe40 DAQ Engine and its generalized architecture are shown respectively.

In the schematic shown in Figure 2.10, for MiniPOD pair 0 (MP0) detector links #01 – 12 are connected, for MiniPOD pair 1 (MP1) detector links #13–24 are connected (optional for certain sub-detectors), for MiniPOD pair 2 (MP2) detector links #25–35 are connected (optional for certain sub-detectors) and for MiniPOD pair 3 (MP3) detector links #36–47 are connected (optional for certain sub-detectors).

are connected (not needed in CRU application). The details about the FPGA is given in Table 2.3. As a hardware board PCIe40 got 14 layer High Density Interconnect (HDI) PCB that are laser drilled blind, buried and stacked vias. Altogether, PCIe40 houses 1750 electronic components. The PCB has high glass transition temperature (Tg), high Decomposition Temperature (Td) and low dielectric constant (Dk) to suit for high speed signal integrity. For PCB lamination, FR408HR from Isola Group whose Tg is 190, Td is 360 and Dk is 3.64@5 GHz. Since the board would directly be plugged in PCIe Gen 3 interface socket, strict restriction applies on its thickness that is $0.57\text{ mm} + / - 10\%$ and has track requirements of 3 mil thick.

Table 2.3: Specification of the FPGA used in PCIe40 aka CRU card

FPGA Parameters	Values
FPGA Type	Arria [®] 10 FPGA
Part No.	10AX115S4F45I3SGE2
BGA package	1932 pins
Adaptive logic module (ALM)	427200
General Purpose I/Os	960
High Speed Serial Interfaces	72
PCIe Hard IP Blocks	4
Memory Bits	55562240
M20K	2713
DSP Blocks	1518
Fractional PLLs	32
I/O PLLs	16
Global Clocks	32
Hard Processor CPU cores	0

2.11 Test Results

Complex FPGA logic design needs testing performed at various levels before final deployment. The most common levels of testing are: System, Integration and Unit. Basic unit test results are shown in the chapter. These include Avago MiniPODTM performance as shown in Figure 2.11, and throughput measurement of PCIe Gen2. The Arria[®] 10 En-

gineering Sample (ES)-1 supports upto PCIe Gen2, from ES2 and production chips Gen3 is supported. A preliminary resource estimation report is presented in Table 2.5 and 2.6, as part of integration test.

2.11.1 Minipod Performance

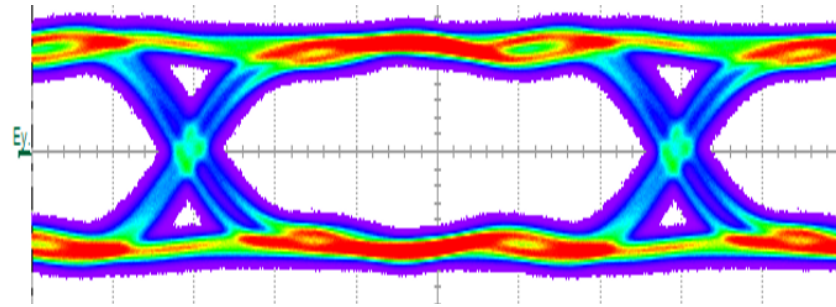
High-speed electrical signals progressively degrade with increasing trace length in PCB material. There is also a chance of EMI containment with 12 optical channels interfaced with single optical module, or MiniPOD [27]. A comparative signal integrity measurement is done using 4.8 Gbps GBT protocol and 10.3125 Gbps 10 GbE protocol as shown in Table 2.4.

Table 2.4: Shows the comparison of MiniPODTM performance for GBT protocol and 10GbE Protocol

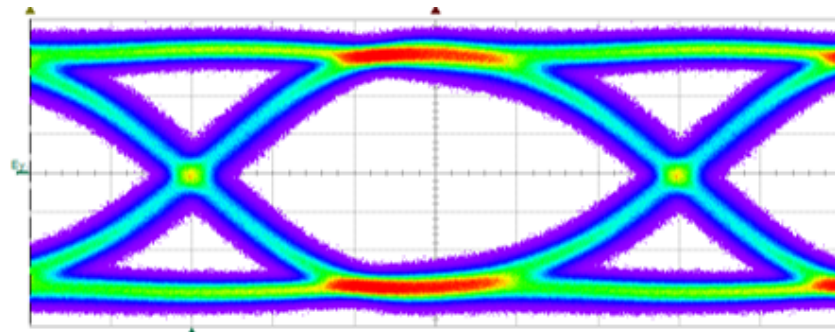
Parameters	GBT Protocol	10 GbE Protocol
Link Speed	4.8 Gbps	10.312 Gbps
Eye height	373 mV	373 mV
Eye Width	176.8 ps	79.4 ps
Deterministic Jitter (Dj)	13.125 ps	5.172 ps
Periodic Jitter (Pj)	7.240 ps	0.675 ps
Data Dependant Jitter (DDj)	21.647 ps	9.158 ps
Inter Symbol Interference (ISI)	21.657 ps	7.399 ps
Duty Cycle Jitter (DCD)	1.912 ps	2.062 ps
Random Jitter(Rj)	3.204 ps	2.204 ps
Total Jitter (Tj)	58.185 ps	36.180 ps
Standard Deviation(σ)	2.898 ps	2.045 ps

2.11.2 PCIe Performance

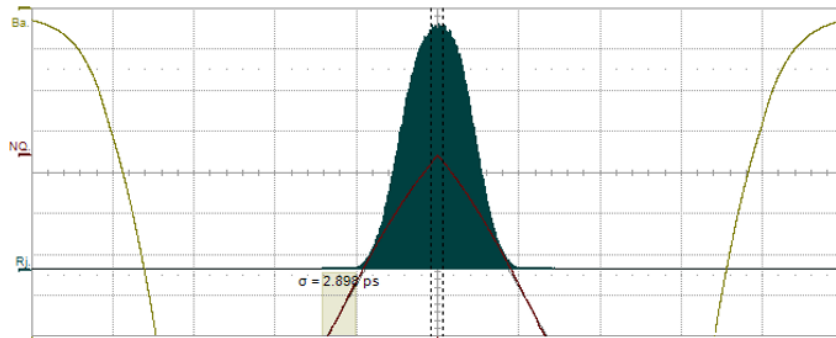
The Avalon[®] Memory-Mapped (Avalon-MM) DMA PCI Express [28] for Arria[®] 10 is used for bench marking PCIe[®] Gen2 x8 performance in PCIe40. It is interfaced with application layer using QSYS interconnect with 256 bit interface. The results from Arria[®] 10 ES1 are currently available with us. Thus, PCIe[®] Gen2 x8 measured performance is



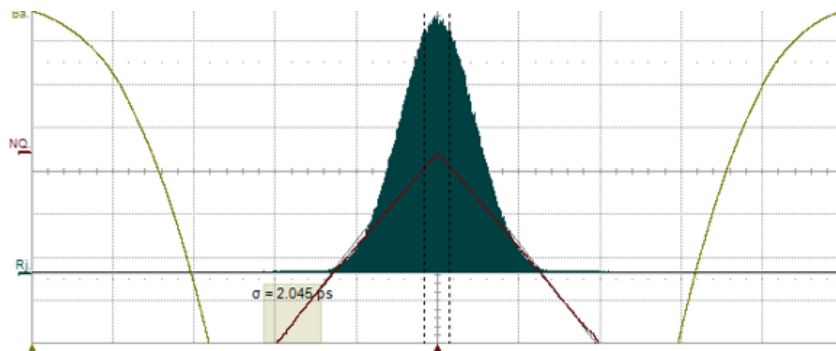
(a) GBT Tx Eye Diagram



(b) 10 GbE Tx Eye Diagram Schematic



(c) GBT Tx Jitter Spectrum



(d) 10 GbE Jitter Spectrum

Figure 2.11: MiniPODTM performance with GBT protocol (4.8 Gbps) and 10 GbE protocol (10.312 Gbps)

shown.

$$\text{Data Signalling Rate} = 5 \text{ Gbps per lane} \times 8 = 40 \text{ Gbps}$$

$$\text{Measured Data Throughput} = 32 \text{ Gbps}$$

PCIe[®] Gen2 uses 8B/10B encoding which introduces a 20% overhead on data payload, which accounts for lesser useful data throughput. The test is repeated for both the PCIe banks.

2.11.3 Resource Estimation

Before full implementation of generic firmware, an elementary integrated firmware design is constituted. It includes all the Interface Links with individual data pattern generator and checker, without **any interconnect glue logic**. The result in Table 2.5 is likely to evolve with firmware development. It also indicates that plenty of resources are available for detector specific firmware development. The break up of logic resources used in each part of periphery logic core (GBT, PCIe, TTK) is given in Table 2.6. For the estimation Intel[®](formerly Altera) Arria[®] 10 FPGA family is used with device part number 10AX115S4F45I3SGE2 [29].

Table 2.5: An elementary integrated firmware design with all the Interface Links

Aggregated Links without inter-connect glue logic	Logic Utilization	HSSI SERDES Utilization
48 GBT link + x16 PCIe Gen 3 + SFP+ (Transceiver Toolkit design)	101,351 / 427,200 (23.72 %)	65 / 72 (90 %)
36 GBT link + x16 PCIe Gen 3 + SFP+ (Transceiver Toolkit design)	83,012 / 427,200 (19.43 %)	53 / 72 (74 %)
24 GBT link + x16 PCIe Gen 3 + SFP+ (Transceiver Toolkit design)	64,956 / 427,200 (15.20 %)	41 / 72 (57 %)

Configuration Wise

For the TPC detector, the requirement is 40 GBT link having Widebus coding working in a standard mode. In the architecture x6 PMA bonded mode is used, that allows GBT links to be clubbed in groups of 6. The configuration saves on the clocking resources. The

Table 2.6: Individual logic blocks resource occupancy for variable number of links

Resource Type	Total Available	Resource Usage				
		x24 GBT	x36 GBT	x48 GBT	x16 PCIe+DMA	TTK SFP+
Logic utilization (in ALMs)	427,200	34,381 (8.05 %)	51,086 (11.95 %)	67,362 (15.76 %)	28,051 (6.56 %)	2,755 (0.64 %)
Total registers	1,708,800	54,946 (3.21 %)	81,247 (4.75 %)	105,322 (6.16 %)	45,461 (2.66 %)	5,526 (0.32 %)
Total block memory bits	55,562,240	70,144 (0.12 %)	94,400 (0.17 %)	118,656 (0.21 %)	1,491,044 (2.68 %)	23,168 (0.04 %)
Total RAM Blocks (M20K)	2,713	153 (5.64 %)	227 (8.37 %)	301 (11.09 %)	111 (4.09 %)	8 (0.29 %)
Total DSP Blocks	1,518	0 (0.00 %)	0 (0.00 %)	0 (0.00 %)	0 (0.00 %)	0 (0.00 %)
IOPLL	16	8 (50.00 %)	12 (75.00 %)	16 (100.00 %)	0 (0.00 %)	1 (6.25 %)
FPLL	32	0 (0.00 %)	0 (0.00 %)	0 (0.00 %)	2 (6.25 %)	0 (0.00 %)
HSSI ATX PLL	32	4 (12.5 %)	6 (18.75 %)	8 (25 %)	0 (0.00 %)	1 (6.25 %)
Total HSSI RX channels	72	24 (33.33 %)	36 (50.00 %)	48 (66.67 %)	16 (22.22 %)	1 (1.38 %)
Total HSSI TX channels	72	24 (33.33 %)	36 (50.00 %)	48 (66.67 %)	16 (22.22 %)	1 (1.38 %)
PCIe Hard IPs	4	0 (0.00 %)	0 (0.00 %)	0 (0.00 %)	2 (50.00 %)	0 (0.00 %)

collected links of six are referred as the GBT Bank. However, the choice of a GBT link arrangement selection is based on different occupancy and accordingly there can be three different types of combinations.

$$\text{Combination 1} :: 40 \text{ GBT Links} = 6 \text{ GBT Links} \times 6 + 2 \text{ GBT Links} \times 2 \quad (2.1)$$

$$\text{Combination 2} :: 40 \text{ GBT Links} = 6 \text{ GBT Links} \times 6 + 4 \text{ GBT Links} \quad (2.2)$$

$$\text{Combination 3} :: 40 \text{ GBT Links} = 5 \text{ GBT Links} \times 8 \quad (2.3)$$

Table 2.7: 40 GBT Links resource occupancy for various type of combinations

Resource Type	Total Available	Resource Usage		
		Combination 1	Combination 2	Combination 3
Logic utilization (in ALMs)	427,200	59,251 (13.87 %)	58,696 (13.74 %)	61,358 (14.36 %)
Total registers	1,708,800	94,498 (5.53 %)	93,724 (5.48 %)	93,849 (5.49 %)
Total block memory bits	55,562,240	60,416 (0.11 %)	59,904 (0.11 %)	55,296 (0.10 %)
Total RAM Blocks (M20K)	2,713	184 (6.78 %)	183 (6.74 %)	168 (6.19 %)
Total DSP Blocks	1,518	0 (0.00 %)	0 (0.00 %)	0 (0.00 %)
IOPLL	16	8 (50.00 %)	7 (43.75 %)	8 (50.00 %)
FPLL	32	0 (0.00 %)	0 (0.00 %)	0 (0.00 %)
HSSI ATX PLL	32	8 (25.00 %)	7 (21.87 %)	8 (25.00 %)
Total HSSI RX channels	72	40 (55.56 %)	40 (55.56 %)	40 (55.56 %)
Total HSSI TX channels	72	40 (55.56 %)	40 (55.56 %)	40 (55.56 %)
PCIe Hard IPs	4	0 (0.00 %)	0 (0.00 %)	0 (0.00 %)

Here Rx frame clk uses reference fabric clock to drive the receiving logic, so saves 1 PLL per GBT Bank

2.12 Conclusion

In order to exploit the full physics potential offered by the ALICE upgrade from 2021 RUN3 and HL-LHC upgrade from 2025, the ALICE experiment proposes to implement a dedicated data, trigger and control distribution unit, called Common Readout Unit. The chapter covers the functional role and location of the CRU design unit in the ALICE experiment. The choice of CRU hardware and development of a generic firmware architecture is elaborated in detail. GBT, TTC-PON and PCIe are used as interface links. The firmware design uses the LHCb PCIe40 FPGA board. Preliminary test results that are presented is used in initial estimation of firmware logic resources needed per CRU hardware.

In future if an increased demand occurs for more computing power from detector side, then a hierarchy of CRU boards can be used to meet the computational need. Intel[®] (formerly Altera) FPGA chip supports vertical migration to latest generations. For Arria[®] 10 FPGA family possibility to migrate to Stratix 10 FPGA keeping the same pin configuration exists. Clearly all the factors aid in the persistence of the present design to cope with unanticipated future needs in the long run.

Chapter 3

GBT Prototyping & Evaluation for the CRU Development

3.1 Introduction

In High Energy Physics experiment one of the major challenges is to transfer data with very high reliability between the different sub-detectors situated in the harsh radiation zone to the data acquisition electronics located in the non-radiation area. The Simple Link Interface (S-LINK) [30] specification defined in 1995 at CERN, describes a data link for moving data or control word between front-end electronics to the read-out electronics, via Gigabit Optical Link (GOL) serializer chip. The sub-detectors at the same time must be able to receive trigger and timing information maintaining a constant latency. A unified approach developed by RD12 for the broadcasting of timing, trigger and control (TTC) [31] signals from the RF generators of the LHC machine to the outputs of the timing receiver ASICs (TTCrx + QPLL) at the experiment and beam instrumentation destinations. Since there wasn't one single chip doing both the functionalities, so the CERN team decided to build one single chip combining both. This initiated the GBT (Gigabit Transceiver) [20] and the versatile link [32] project. GBT is a radiation tolerant

error resilient data communication standard with fixed latency support. A single GBT channel link can handle detector data, timing, trigger and control information traffic. Radiation tolerant GBT chipset used for packaging of detector data and transmitting it in GBT standard. While the optoelectronic components and point to point optical link connecting the GBT ASIC with the Field Programmable Gate Array (FPGA) / Commercial Off-The-Shelf (COTS) is qualified by versatile link project group. A short summary of the GBT protocol is tabulated in Table 3.1 and Figure 3.1 gives the details of GBT frame.

Current work describes *GBT FPGA* CORE firmware [19, 33] implementation on FPGA, inter-FPGA firmware migration, clock calibration strategy and comparative performance study. Two custom DAQ boards AMC40 [34] and PCIe40 [16, 35] based on latest Intel[®] (formerly Altera) Stratix[®] V [36] and Arria 10 [37] FPGA were used for the detailed comparative study of GBT FPGA Core firmware implementation.

Table 3.1: Details of GBT protocol standards

Parameters	GBT Data Transmission Protocol [20, 19, 33, 26]
Security	Not Applicable
Channel Data Throughput	4.8 Gbps
Raw data Throughput	3.2 Gbps
Bandwidth Utilization (Data Efficiency)	66.67% (80 / 120)
Bandwidth Utilization (Coding Efficiency)	73.33% (88 / 120)
Forward Error Correction	RS Encoding (15,11) with symbol size of 4 bits
Number of FEC block used	2
Error Detection	32 bits
Error Correction	16 bits
Burst Error Correction	16 bits
Latency optimized data transmission	Supported
Interleaver	Block
Supported Data Type	Idle/Control and Data

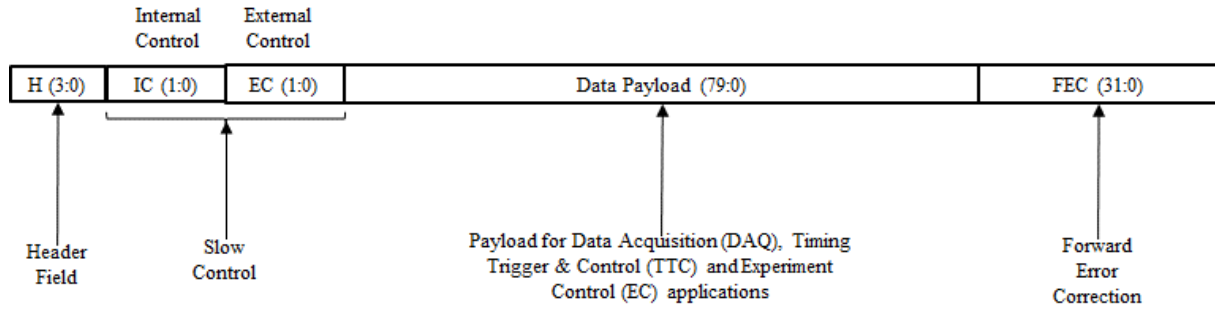
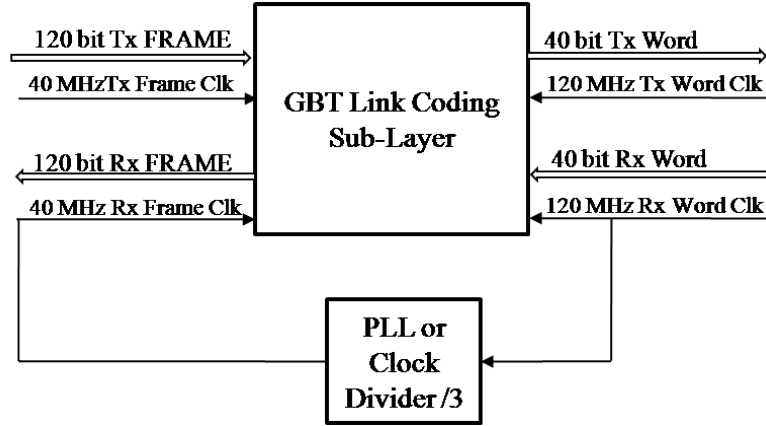


Figure 3.1: Data format for GBT protocol specific data packet

3.2 Design Implementation

AMC40 and PCIe40 are based on Intel[®] (formerly Altera) Stratix[®] V and Arria[®] 10 FPGA respectively. GBT FPGA core is composed of two sections: (1) GBT Coding Sub-layer; (2) Multi-Gigabit Transceiver (MGT). GBT coding sub-layer is independent of FPGA architecture chosen. While, Multi-Gigabit Transceiver (MGT) is FPGA dependant. Figures 3.2a, 3.2b and 3.2c shows for GBT firmware implementation only transceiver connectivity need to be modified based on device migration guideline, rest of the core coding logic can be kept intact.

Both the designs use **PMA** (Physical Media Attachment) **bonded mode** [36, 37] configuration, to get same latency through each bonded data-path. Accordingly in Stratix[®] V and Arria[®] 10, GBT PMA links are grouped into 3 and 6 links respectively to form a single **GBT Bank**.



(a) Basic signal block for GBT Coding Sub-layer

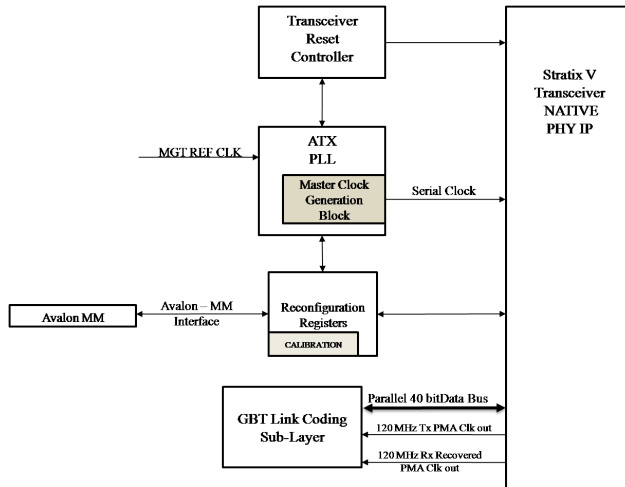
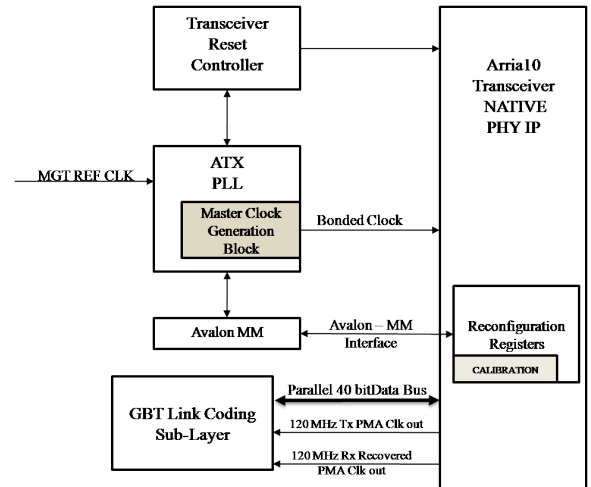

(b) Stratix[®] V Transceiver Design Connectivity

(c) Arria[®] 10 Transceiver Design Connectivity

Figure 3.2: GBT core design connectivity

3.3 Resource Estimation

For resource estimation, a standard reference design from CERN GBT team is used having same configurations on both boards. The reference design altogether contains 4 links operating in two different modes as shown in Table 3.2. Comparison of resource utilization for GBT reference design implemented in AMC40 vs PCIe40 boards are shown in Table 3.3.

Table 3.2: Reference Design configuration

Configuration Parameter	Value
No of GBT Bank	2
Bank 1	<i>No of Links</i> = 1 <i>Tx Mode</i> = <i>STANDARD</i> <i>Rx Mode</i> = <i>LATENCY OPTIMIZED</i>
Bank 2	<i>No of Links</i> = 3 <i>Tx Mode</i> = <i>LATENCY OPTIMIZED</i> <i>Rx Mode</i> = <i>STANDARD</i>

Table 3.3: Shows the comparison of resource utilization in two boards for Standard Reference Design

Parameters	AMC40		PCIe40	
<i>FPGA Family</i>	Stratix [®] V		Arria [®] 10	
<i>Device Part Number</i>	5SGXEA7N2F45C3		10AX115S4F45I3SGE2	
	Occupancy Ratio	Percentage of Occupancy	Occupancy Ratio	Percentage of Occupancy
Logic utilization (in ALMs)	10,542 / 234,720	4%	9,055 / 427,200	2 %
Total registers	20060		18338	
Total block memory bits	202,752 / 52,428,800	< 1 %	126,464 / 55,562,240	< 1 %
Total RAM Blocks	56 / 2,560	2 %	40 / 2,713	1 %
Total HSSI PMA TX/RX (Serializers/Deserializers)	4 / 48	8 %	4 / 72	6 %
Total PLLs	11 / 92	12 %	10 / 176	6 %

3.4 Latency Measurement

GBT has got two operational modes namely; Standard mode that uses an elastic buffer or FIFO for Clock Domain Crossing (CDC) and Latency Optimized mode which uses an inelastic buffer or register bank for CDC. Latency measurement is very critical for GBT. In GBT, information content can be data, timing, and control format, each comes with

its latency boundary condition. Most stringent latency boundary conditions apply for Timing and Trigger distribution.

Latency

Latency occurs in both transmission and receiving directions, and is different in each direction, depending on the media and path involved. Latency gives information of the logic path delay, and whether the path contains an elastic or an inelastic buffer. It directly contributes to the calculation of round trip delay. It is observed that round trip delay can be used to get a quick estimation of aggregate latency.

Roundtrip Delay

The round trip delay corresponds to the length of time it takes for a signal to be sent plus the length of time it takes for an acknowledgment of that signal to be received. It includes serialization, deserialization time along with propagation delay.

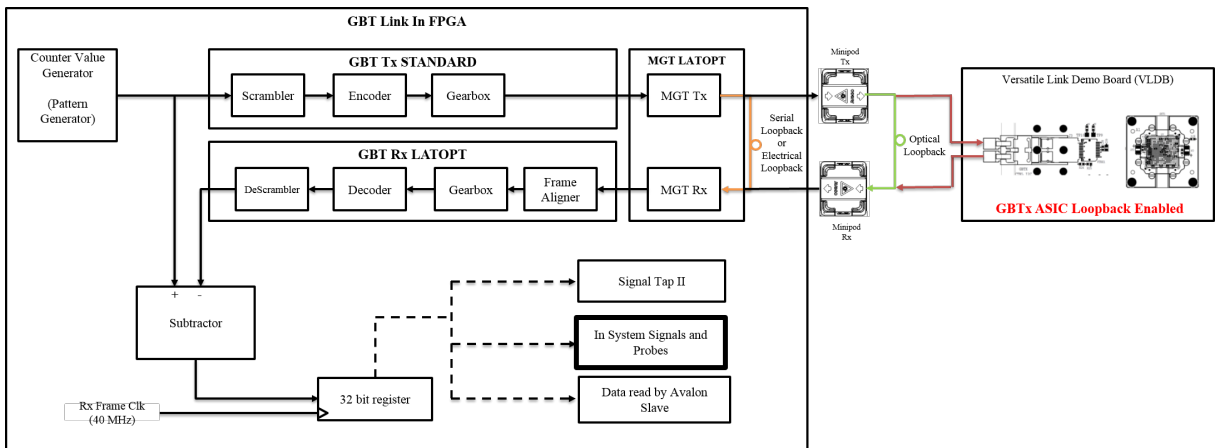


Figure 3.3: Test Setup for 3 types of round trip measurement, each loopback arrangement are marked by different colours

The round trip delay are formulated in these following given equations for easy understanding of the path delay or logic delay (D) involved.

1. Serial Loopback or Electrical Loopback

$$D_{Serial\ Loopback\ Round-Trip} = (D_{GBT\ Tx\ Encoder} + D_{Serialization} + D_{Deserialization} + D_{GBT\ Rx\ Decoder})^{FPGA}$$

2. Optical Loopback

$$D_{Optical\ Loopback\ Round-Trip} = D_{Serial\ Loopback\ Round-Trip} + D_{Optical\ Fibre\ Length}$$

3. GBTx ASIC Loopback

$$D_{GBTx\ ASIC\ Loopback\ Round-Trip} = D_{Optical\ Loopback\ Round-Trip} + (D_{GBT\ Tx\ Encoder} + D_{Serialization} + D_{Deserialization} + D_{GBT\ Rx\ Decoder})^{ASIC}$$

The round-trip delay can be considered as round-trip latency ($\mathcal{L}$), provided there are no hold ups in the path, like elastic buffer, queuing, congestion control. Such conditions exists only under special constraints. GBT ASIC by default operates in latency optimized mode on both Tx and Rx side, while GBT FPGA core needs to be forced to operate in such mode.

$$\mathcal{L}_{GBTx\ ASIC\ Round-Trip} = D_{GBTx\ ASIC\ Loopback\ Round-Trip} - D_{Optical\ Loopback\ Round-Trip} \quad (3.1)$$

$$\mathcal{L}_{GBT\ FPGA\ Core\ Round-Trip} = D_{Serial\ Loopback\ Round-Trip} : \text{when } Tx \text{ and } Rx \text{ latency optimized} \quad (3.2)$$

$$\Delta\mathcal{L} = (\mathcal{L}_{GBT\ FPGA\ Core\ Round-Trip} - \mathcal{L}_{GBTx\ ASIC\ Round-Trip}) \quad (3.3)$$

By design GBT ASIC takes lowest round-trip latency. So, lower the $\Delta\mathcal{L}$ better the GBT FPGA Core performance.

Resolution

Measurement of latency within FPGA using firmware is always dependent on the data rate, and at which step of data parallelism it is sampled. In this case, it is sampled at the

3.5. BIT ERROR RATE (BER) ANALYSIS

beginning of sending data frame (120 bit)@40 MHz. So, the resolution is 25ns, which is the rate of LHC bunch crossing.

Table 3.4: Round Trip delay measurements for multi-level loopback in PCIe40

GBT Protocol		ROUND TRIP DELAY		
(In terms of LHC bunch crossing Rate = 25ns)				
Transmission Side	Receiver Side	Serial or Electrical Loopback	Optical Loopback	GBT ASIC Loopback
<i>Latency Optimized</i>	<i>Latency Optimized</i>	6	7	11
<i>Latency Optimized</i>	<i>Standard</i>	13	14	18
<i>Standard</i>	<i>Latency Optimized</i>	8	9	13
<i>Standard</i>	<i>Standard</i>	18	18	22

$$\mathcal{L}_{GBTx\ ASIC\ Round-Trip} = 4$$

$$\text{From the above table it can be inferred: } \mathcal{L}_{GBT\ FPGA\ Core\ Round-Trip} = 6$$

$$\Delta\mathcal{L} = 2$$

Table 3.5: GBT Serial Loopback Round Trip Measurement comparison for AMC40 and PCIe40

	Different Mode of operations			
Tx Side →	Standard	Standard	Latency Optimized	Latency Optimized
Rx Side →	Standard	Latency Optimized	Standard	Latency Optimized
AMC40	600 ns	350 ns	350 ns	150 ns
PCIe40	450 ns	350 ns	200 ns	150 ns

3.5 Bit Error Rate (BER) Analysis

The Transceiver Toolkit (TTK), an on-chip debugging tool provided by Altera for BER monitoring is used for this analysis. The test values are listed in Table 3.6. TTK full

functionality is not available for Arria[®] 10 ES1. So, different test setup is used for two boards. The results for PCIe40 1st version DAQ engine are preliminary.

Table 3.6: BER Measurement

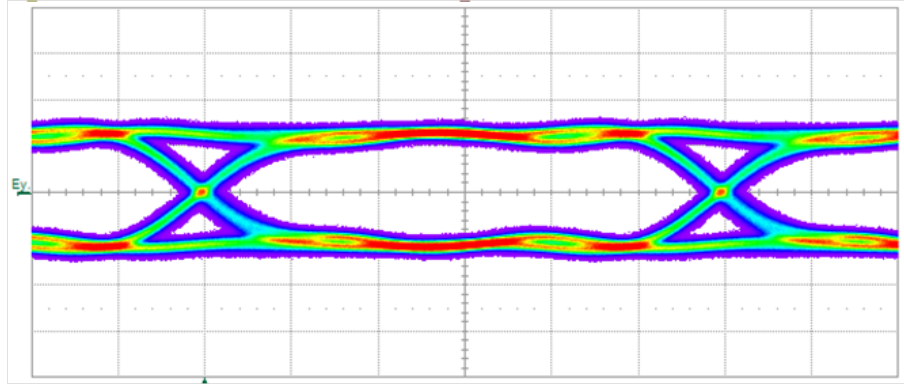
Parameters	AMC40	PCIe40
Test Setup	AMC40 (Tx) - optical cable - AMC40 (Rx)	PCIe40 (Tx) - optical cable - AMC40 (Rx)
<i>Tx configuration Parameter</i>		
<i>V_{OD} control</i>	50	29
<i>Pre – emphasis 1st post – tap</i>	0	0
<i>Pre – emphasis pre – tap</i>	0	0
<i>Pre – emphasis 2nd post – tap</i>	0	0
<i>Rx configuration Parameter</i>		
<i>DC Gain</i>	1	1
<i>Equalization Control factor</i>	0	0
<i>DFEEnabled</i>	off	off
PLL reference Clock Frequency	120 MHz	120 MHz
Test Pattern	PRBS31	PRBS31
Results of TTK		
<i>BER (Bit Error Rate)</i>	1.115E-12	5.0723E-10 [#]
<i>EyeWidth/EyeHeight</i>	26/127	23/89

[#] This result is preliminary, will subject to change from new versions

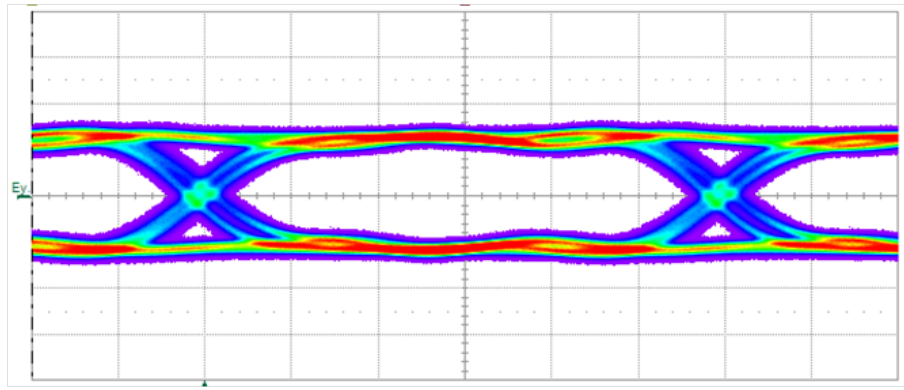
3.6 Jitter and Eye Measurement

Data from the FPGA transceivers are transmitted to the onboard MiniPODTM [27] optical modules. They are coupled with flat ribbon cable using a Prizm connector and terminated on the other side with industry standard MTP connector. A Lecroy Serial Data Analyser (SDA) oscilloscope is used for analyzing MiniPODTM signal quality.

An eye diagram is a common indicator of the quality of signals in high-speed digital transmissions. The signal to noise ratio of this high-speed data signal is directly indicated by the amount of eye closure or Eye Height.



(a)



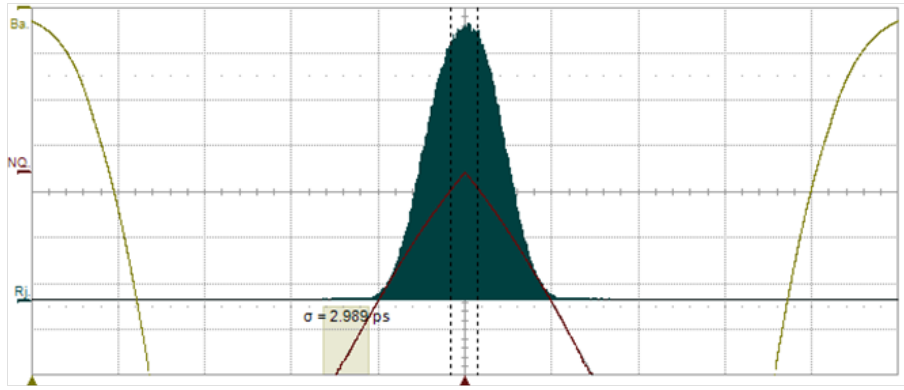
(b)

Figure 3.4: Showing the Eye Diagram for Tx Optical using GBT encoded data (a) AMC40 and (b) PCIe40

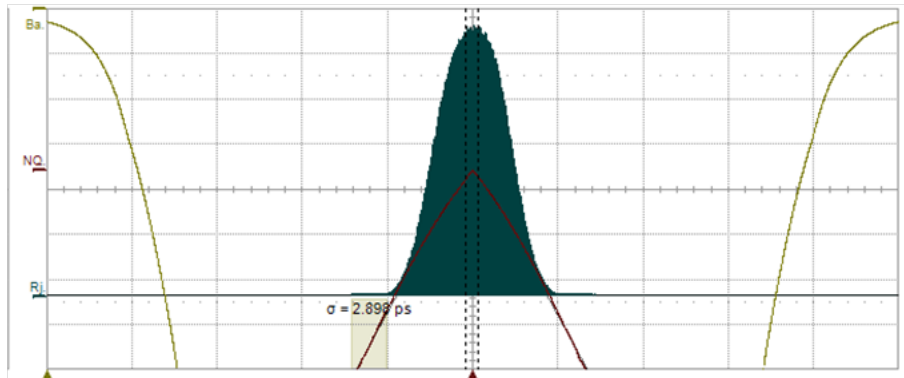
Table 3.7: Shows the comparison of Optical MinipodTM performance in two boards

	AMC40	PCIe40
Deterministic Jitter (Dj)	5.503 ps	13.125 ps
Periodic Jitter (Pj)	6.75 ps	7.24 ps
Data Dependant Jitter (DDj)	11.228 ps	21.647 ps
Inter Symbol Interference (ISI)	11.095 ps	21.657 ps
Duty Cycle Jitter (DCD)	2.000 ps	1.912 ps
Random Jitter(Rj)	3.245 ps	3.204 ps
Total Jitter (Tj)	51.148 ps	58.185 ps
Standard Deviation(σ)	2.989 ps	2.898 ps

Jitter spectrum is the measurement of timing variation of a signal edge from its ideal values. Contributing factors include power load variation, thermal noise, line attenuation and interference coupled from nearby devices. It is very important that the clock jitter is within a tolerable hold time of the pipeline registers for error free data transmission.



(a)



(b)

Figure 3.5: Showing the Jitter Spectrum for (a) AMC40 and (b) PCIe40

3.7 Calibration

High Speed Serial Interface Serializer-Deserializer (HSSI SerDes) or Multi-Gigabit Transceiver (MGT) as well as internal PLLs include both analog and digital blocks that require calibration to compensate for process, voltage and temperature (PVT) variations [37]. MGT

in fixed-latency mode do not have the circuitry to maintain same latency for data path with each power or reset cycle [38]. In time synchronization protocol, like *GBT in Latency optimized mode*, **Automated Transceiver Calibration** (ATC) must be followed by **Tx Latency Calibration** (TLC) and **Rx Latency Calibration** (RLC) during both power and reset cycle. In the case of incomplete or improper calibration, ATC contributes to random jitter while TLC and RLC are responsible for deterministic jitter. The latency error measured in Unit Intervals (UI) which corresponds to one pulse duration of GBT data stream ($1 \text{ UI} = 208.33\text{ps}@4.8\text{Gbps}$).

3.7.1 Tx Latency Calibration (TLC)

The fabric-transceiver clock interface in GBT design consists of signals: input reference clocks (refclk) and transceiver data path interface clocks (tx_clkout, rx_clkout) [36, 37]. Transceiver forwards *tx_clkout* to be used in FPGA fabric as *tx parallel word clock*. This clock is used to drive the user logic data into the transmitter Double Data Rate (DDR) Parallel Input Serial Output (PISO). This *tx_clkout* (120 Mhz) is obtained from the serial clock (2400 MHz or 2 UIs) by dividing it with serialization factor of 20. However, this clock divider used to obtain *tx parallel word clock* (tx_clkout) introduces a phase difference uncertainty ($\delta\phi_p$) with *external parallel word clock* (refclk). Equation (3.4) shows $\delta\phi_p$ can take any value within a set of 20 elements in $\Delta\Phi_p$.

$$\Delta\Phi_p = \{2n \text{ UI} : n \in \mathbb{I}, 0 \leq n \leq 19\}, \text{ where } 1 \text{ UI} = 208.33\text{ps}@4.8\text{Gbps} \quad (3.4)$$

$$\text{Max}[\delta\phi_p] = 38 \text{ UI} = 7.916 \text{ ns} \approx 8 \text{ ns} \quad (3.5)$$

$$\text{Phase Difference Error or Latency Error} = \pm \text{Max}[\delta\phi_p]/2 = \pm 4 \text{ ns} \quad (3.6)$$

This phase difference error goes completely undetected by GBT Program Coding Sub-layer (PCS) unless it falls in metastability region and flags data frame error. The phase

(ϕ_p) transition from one value to other with each reset cycle is 2 UIs, which is sequential, predictable and has exhaustive number of states. This property is used to develop the proposed calibration solution to get always the same value for $\delta\phi_p$ and therefore get a reproducible fixed latency or just find a value that avoids metastability.

3.7.2 Proposed Solution

Forwarded Tx Word Clk takes some initialization time after each reset cycle, so for proper alignment between the frame and the word in (1:3) Gearbox, the gearbox sync reset is delayed by few clock cycles from the global reset. Synchronization register chain or synchronizer is used to minimize the failures due to metastability [39]. The calibration trigger can be based on metastability detection (Logic 1) or phase error detection (Logic 2). If the requirement of the design allows a latency error tolerance margin of $\pm 4ns$ then metastability detection logic is suitable because of its lower resource usage. Otherwise, the phase error detection logic is more suitable for better phase predictability design. The entire solution is implemented by modification in the latency optimized gearbox in GBT PCS.

Phase Error Detection Logic

Tx FrameClk has a synchronous phase relationship with reference clock (refclk). So, the phase calculation is computed between the Tx word and Tx frame clock. This digital calculation of phase difference within FPGA fabric is not very accurate, yet sufficient to detect a $\delta\phi_p$ phase difference. A look-up table is prepared manually, and a particular phase value is pre-defined with a safe margin of the metastable zone. Figure 3.7 shows the detected phase difference is compared against phase calibration table values to assert global reset.

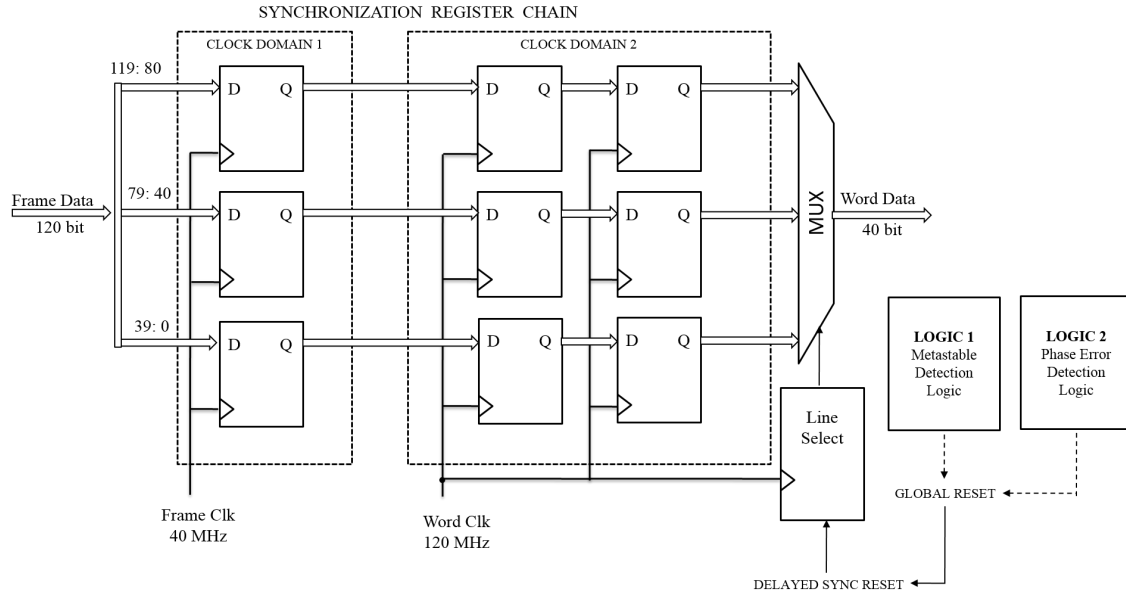


Figure 3.6: Synchronization register based clock domain crossing with metastability or phase error detection logic

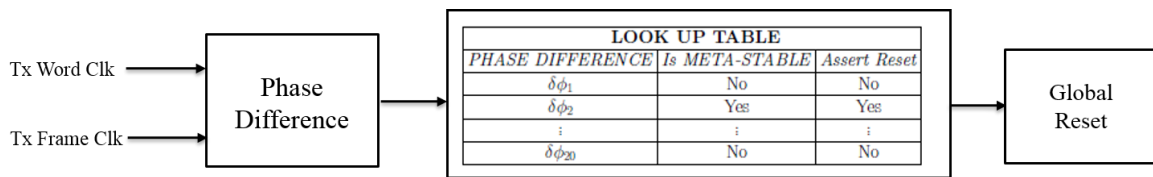


Figure 3.7: Calibration trigger from manually tabulated phase error look-up table

Phase selection for temperature variation tolerance Proper phase synchronization between frame clock and word clock is necessary to avoid metastability during data transmission. However, with temperature variation phase drift happens. If the synchronized region lies close to the metastable region, phase drift due to temperature causes it to get unstable. So, the calibrated phase need to be chosen wisely. The temperature is varied by controlling the on-board cooling system and noted for any data frame errors. Temperature Sensing Diode (TSD) internal to FPGA is used for temperature reading.

Metastability Detection Logic

To ensure data write reliability the input to a register must be stable for a minimum time before the clock edge (register setup time) and for data read reliability, the output of a register must be stable a minimum time after the clock edge (register hold time) [39]. For proper data frame to word conversion, it must avoid metastable regions. However, due to the phase transition of forwarded tx_clkout , it sometimes falls into this zone and causes data error. Figure 3.8 shows with flowchart 3 types of data error detection logic.

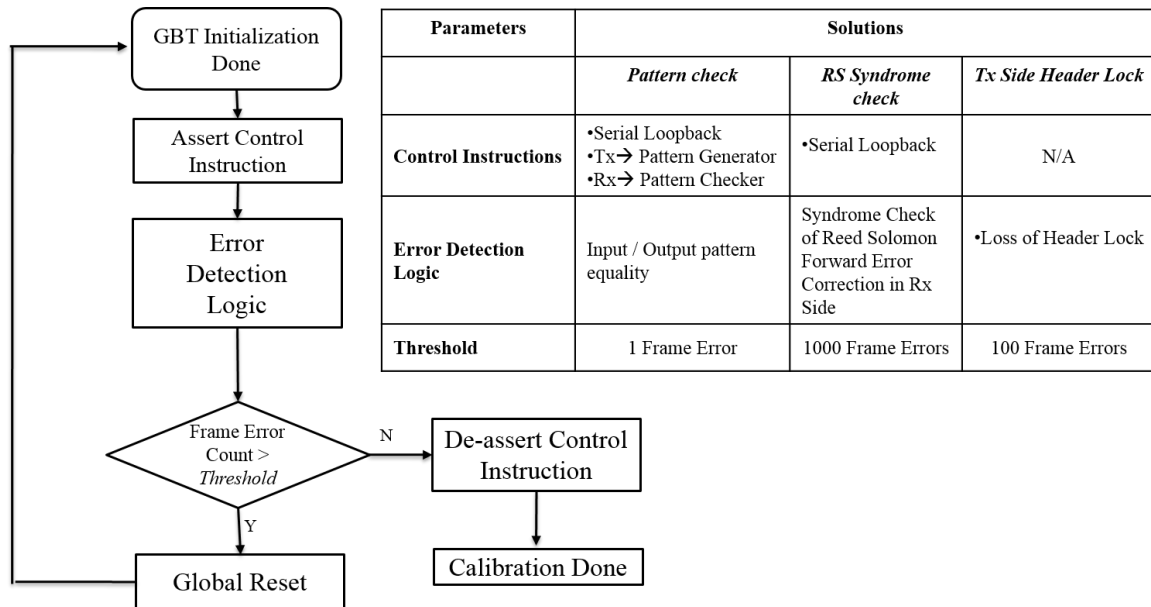


Figure 3.8: Calibration trigger on data frame error or metastability

3.7.3 Stratix V and Arria 10 FPGA

A user recalibration of the FPGA is required after every power cycle showing a loss of lock of the MGT reference clock during Automated Transceiver Calibration (ATC) [36, 37]. Arria[®] 10 FPGA uses hardened Precision Signal Integrity Calibration Engine (PreSICE) to perform calibration routines at every power-up before entering user-mode. After transceiver calibration is over, latency calibration logic is followed. Tx Latency Calibration (TLC) is done as discussed in section 3.7.1. The Arria 10 device used in this

test procedure can work without TLC. Rx Latency Calibration (RLC) uses a new type of barrel shift logic that does bit slip scanning to lock Frame Alignment Word (FAW) at first hit [20]. The maximum number of bits slipped is equal to the FPGA fabric-to-transceiver interface width minus 1, i.e. 19 ($= 20 - 1$). After completing the calibration sequences, the control is transferred to user logic.

3.8 Conclusion

With the detailed analysis it is proved that the GBT protocol can indeed be implemented with success in both the 28 nm Stratix[®] V and 20 nm Arria[®] 10 Intel[®] (formerly Altera) FPGAs. The source code of developed firmware are archived in CERN espace, and more resources are available to users on requests. The calibration logic is needed if operated in 120 MHz word clock. If the GBT link uses 240 MHz word clock then it can be operated in a feedback compensated mode and hence no calibration logic is needed to mitigate the phase uncertainty of the recovered clock. The discussion of the final GBT version is continued in Chapter 4.

Chapter 4

CRU as the Trigger and Timing Distribution Unit

4.1 Introduction

The timing signal integrity forms an important service requirement of the Common Read-out Unit (CRU) system. A detailed study is done to ascertain that the multiple high-speed communication links act together by being synchronous to the LHC clock signal to transmit the detector readout data and the timing signals with a constant latency. Moreover, the embedded clock phase information is to be preserved and the jitter introduced due to the effect of channel noise also to be retained at a low level. Tests are conducted to confirm the ability of the CRU system to preserve the same behaviour with each power on/off cycle, firmware upgrade and reset assertion/de-assertion cycle (PFR cycle).

The trigger distribution system using the CRU is an amalgamation of multi-link technologies involving different protocol standards. For an entire system to operate synchronously and efficiently, the individual designs are optimized to be able to work in coherence to the neighbouring blocks. In essence, the Giga-Bit Transceiver optical link (GBT) and the Timing, Trigger and Control system based on Passive Optical Networks

(TTC-PON) link work in conjunction to communicate the TTC information from the Central Trigger Processor (CTP) to a detector through the CRU electronics system. The propagation path involves multiple transition points that involves different protocol conversions, however their concurrent executions might interact subtly. These interactions and their inter-dependencies at juncture points are prone to stochastic fluctuations, and hence proper characterization is needed to affirm the behaviour is deterministic. Hence, piece-wise qualification tests are done, before the final goal to integrate, implement and deploy the design elements. The Intel[®] (formerly Altera) Arria[®] 10 development board is chosen as the test board, which carries the same FPGA chip that will be used in the final CRU development card of PCIe40 [40]. The advantage of having a development board is that it provides easy access to the pins and ports to conduct the signal integrity analysis.

The chapter is organized as per the following. Section 4.2 discusses on the LHC trigger distribution. Section 4.3 gives an overview of ALICE trigger and the reasoning for triggerless architecture for Run3. The data flow of the triggerless detector readout raises the need for the use of a new online data frame marker called the heartbeat (HB) trigger, which is also examined in the same section. The role of the TTC systems and the timing distribution to different parts of the ALICE experiment are outlined in the Section 4.4. High-speed serial links in the TTC communication is discussed in Section 4.5. The flow of detector raw data to the CRU using the GBT link is discussed in Section 4.6. In the Section 4.7, the use of a single CTP link to communicate with the multiple CRUs using a TTC-PON in a time multiplexing manner is elaborated. The symbiosis of the TTC-PON and the GBT link technology to form a TTC-PON and GBT link bridge made by the CRU firmware is discussed in brief in Section 4.8. The design integrates multiple links to detect any unwarranted behaviour in the system and to prevent a failure of cascaded type. Intrinsic system monitoring tools are built to gain statistics about the macroscopic behaviour of the system, which are explained in brief in Section 4.9. Section 4.10 covers the results related to the latency measurement, the jitter measurement, the Bit Error

Rate (BER) measurement and the optimization of transceiver parameters. A discussion of the results is given in Section 4.11. Finally, a summary of the present studies and future outlook are presented in the last section.

4.2 LHC Trigger Distribution

Table 4.1: LHC Clock variation with LHC accelerator cycle [41]

NAME	DESCRIPTION	MAGNITUDE**	RATE**	BEAM MODES
RF resync				
RF/DUMP check		+ 100 Hz	22 Hz/s	SETUP (before resync)
RAMP	RF goes from 450 GeV to 7 Tev (or 3.5 TeV)	+ 87 Hz for protons + 550 Hz for ions	< 0.5 Hz/s	RAMP
RAMP DOWN	RF goes from 7 Tev (or 3.5 TeV) to 450 GeV	- 87 Hz for protons - 550 Hz for ions	< 0.5 Hz/s	RAMP DOWN
INJECTION test	Injection tests with offset energy	+/- 240 Hz	22 Hz/s	
DUMP PROTECTION test	+/-1000 Hz for loss maps	+/- 100 Hz	22 Hz/s	ADJUST
CHROMATICITY and/or DISPERSION measurements	Typically, done manually, both beam, at flat top or flat bottom, but could also be automated for any time (even in ramp)	+/- 5 Hz	22 Hz/s	INJECTION MODES (flat bottom), FLAT TOP
TIDAL adjustment	After reaching flat top, value to be set according to the tide of the moment	+/- 1.5 Hz	22 Hz/s	FLAT TOP
Other ORBIT adjustments		+/- 1.5 Hz		FLAT TOP

The Radio Frequency (RF) of the LHC is one of the crucial parameters that determines the bunch length. The RF directly affects the luminosity and also ensures the longitudinal emittance is maintained larger than transverse emittance that occurs due to the Inter Beam Scattering (IBS) during the beam injection and collision [4]. To satisfy such opposing requirement the RF is chosen to operate at 400MHz and also must be an integral multiple of revolution frequency (f_{rev}) as shown in Eq. 4.1.

$$f_{RF} = h \cdot f_{rev} \quad , \quad (4.1)$$

where $h \approx 35640$ and is known as harmonic number.

A particle synchronized to the LHC RF frequency is called a *synchronous particle*. Under the influence of the RF system, the particles instead of being distributed uniformly, oscillates longitudinally around the synchronous particle to form a *bunch*. The particle bunch is trapped in a RF *bucket* that arises from varying the RF voltage. The number of possible buckets in circular accelerator corresponds to the lowest harmonic number, which corresponds to 35640 buckets for the LHC. Each bucket is spaced by one RF period and is numbered from 1 to 35640. To give an overall optimum performance of the LHC machine for two high luminosity experiments taking data at the same time, bunch separation is required. Hence, every tenth bucket is filled with a bunch having a separation time of 25 ns. The LHC in nominal mode fills only 2808 out of 35640 buckets with bunches.

To synchronize the experiments taking data using the LHC machine, the RF timing signals are defined. These signals are the *Bunch Clock (BC)* and the *Orbit*. The *BC* is a square wave derived from the RF (f_{RF}) by a frequency division of 10. The *Orbit* is a train of 5ns long pulses operating at revolution frequency of f_{rev} and points to single bucket. For each ring, the Bunch Clock (BC1, BC2) and the Orbit (Orb1, Orb2) signals are assigned. Apart from the mentioned signal, there also exists a reference Bunch Clock (BCref) signal not related to the beam frequency and is fixed at the 40.078900 MHz. BCref is locked to the beam during collisions only.

During the beam ramping or in the absence of stable beam, the Bunch Clock (BC) signals swing from the reference value with respect to the beam magnet current. Details of the variations are shown in the Table 4.1 and the graphical representation of the variation is given by the plot in the Figure 4.1. The radio frequency swing for the proton beam and the Ion beam at 7 TeV energy is tabulated in the Table 4.2 and the Table 4.3 respectively.

In the detectors, everything are in synchronous to the Bunch Clock (BC), namely the collision events, the signal sampling from analogue to digital conversion, the time detection, the trigger transmission, the data storage, the data reduction and the data

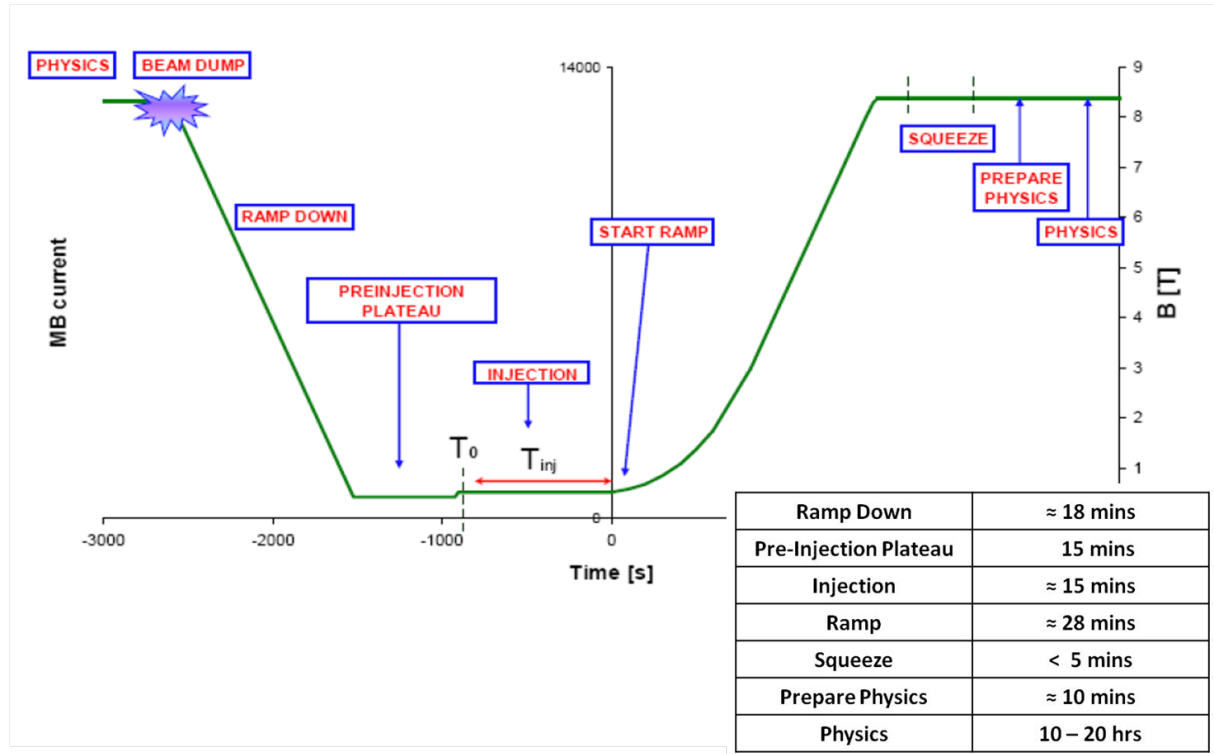


Figure 4.1: LHC Radio Frequency Swing with Beam Ramping

Table 4.2: LHC Radio Frequency Swing for Protons @ 7 TeV

Signal Type		Injection 450 GeV	Physics 7 TeV	Δ
BC	Frequency	40.0788790 MHz	40.0789658 MHz	87 Hz
	Period	24.9507980 ns	24.9507430 ns	0.054 ps
Orbit	Frequency	11.245477 kHz	11.245501 kHz	0.024 Hz
	Period	88.924643 μ s	88.924450 μ s	193 ps

Table 4.3: LHC Radio Frequency Swing for Ions @ 7 TeV

Signal Type		Injection 450 GeV	Physics 7 TeV	Δ
BC	Frequency	40.0784139 MHz	40.0789639 MHz	550 Hz
	Period	24.9510870 ns	24.9507450 ns	0.054 ps
Orbit	Frequency	11.245346 kHz	11.245501 kHz	0.155 Hz
	Period	88.925676 μ s	88.924451 μ s	1.225 ns

transmission. The BC has to be delivered everywhere, anytime, and with an excellent quality. To ensure the timing, trigger and control (TTC) signal integrity during the distribution of the clocks in the experimental sites the TTC systems are being developed [42].

4.3 Triggered and triggerless architecture in ALICE

The Central Trigger Processor (CTP) in ALICE manages the trigger decisions globally and supervise the production of trigger requests by combining the inputs from a system of trigger generating detectors. The CTP plays a pivotal role in identifying rare events, which are recorded for later analysis. In the Run2, the ALICE uses a hardware trigger strategy, where the signals from minimum bias data sample are selected using thresholds on event multiplicity, transverse momenta of tracks and other such observables combining several detectors [43, 44]. In Run2, the maximum readout rate was limited to 500 Hz for Pb-Pb events. In case the trigger rate exceeds a sub-detector read-out capability, the system saturates and asserts a busy signal.

In RUN3, the ALICE would operate at *six* times the current peak luminosity of $10^{27} \text{ cm}^{-2} \text{ s}^{-1}$, collecting over a *ten* times the targeted integrated luminosity of 1 nb^{-1} for the allocated runtime and operating at the collision rate 50 kHz for Pb-Pb ions instead of 8 kHz [8]. The physics objective of the upgrade is to improve the precision of the measurement of QGP signatures. The QGP physics processes do not exhibit signatures that can be selected by hardware triggers directly. In the triggerless readout scheme, all events are readout. The upgraded event selection strategy uses a combination of triggerless readout scheme and the minimum bias trigger generated from the First Interaction Trigger (FIT) detector system [45]. The new readout architecture for timing and/or trigger distribution topology in ALICE is briefly explained in the Section 4.4.

The principle of the ALICE upgrade read-out architecture relies on the Trigger and Timing System (TTS) ability to efficiently distribute the critical TTC signals with con-

stant latency over optical links to the read-out front-end cards and to receive the busy signal to throttle the trigger distribution when needed. All the data packets originating from the sub-detectors are time tagged. The transmission delay of the read-out data path is not stringent and can use non-constant latency links to the on-line farm. The triggerless data acquisition allows readout of multiple sub-detectors without stressing the trigger decision system when a sub-detector gets busy or faulty. In this manner, the continuous triggerless readout mode increases the event selectivity and allows sampling of the full luminosity. The only drawback that the upgraded system faces is to cope with the massive amount of total generated data that is approximately about 3.6 TB/s. The data flow before the final storage gets reduced by the combined effort of the Online and Offline (O2) computing systems. To delimit the overflow of the assembled events across the time frame boundary during data packet formation in the O2 system, a new trigger called *heartbeat* [46, 47] is defined, as explained in the next paragraph.

Heartbeat (HB) in a continuous readout mode are asserted periodically to delimit a stream of readout data [48]. As illustrated in Figure 4.2, HB trigger is used to generate a manageable flow of the Heartbeat Frames (HBF). The HBFs are forwarded from the readout electronics to the CRU over the TTS links, where it is processed and forwarded to the First Level Processor nodes (FLP) and then to the Event Processing Node (EPN), as shown in Figure 4.4. For an each successful HBF delivery to the FLP, the CRU sends an HB acknowledge message to the CTP along with the information about the CRU data buffer. For the entire operation the LHC clock is used as a reference signal to synchronize the data flow. Under a nominal condition, the HBs rate correspond to one LHC orbit period of $89.4 \mu s$ or $10 kHz$. One FLP accumulates 256 HBFs to generate a Sub-Time Frame (STF) every ~ 22 ms, giving a rate of $\sim 50 Hz$. Within the EPN the STFs coming from all the FLPs are aggregated over the same time period, that includes both triggered or continuously readout detectors to form a complete Time Frame (TF). Keeping the anticipated customization needed, both the HBF length and number of the HBF in a TF

4.3. TRIGGERED AND TRIGGERLESS ARCHITECTURE IN ALICE

are programmable. The HBF header, trailer and other IDs are defined to aid the flow management for the TTS links. One hot encoding is used for the 16 bit trigger codes of the HB trigger.

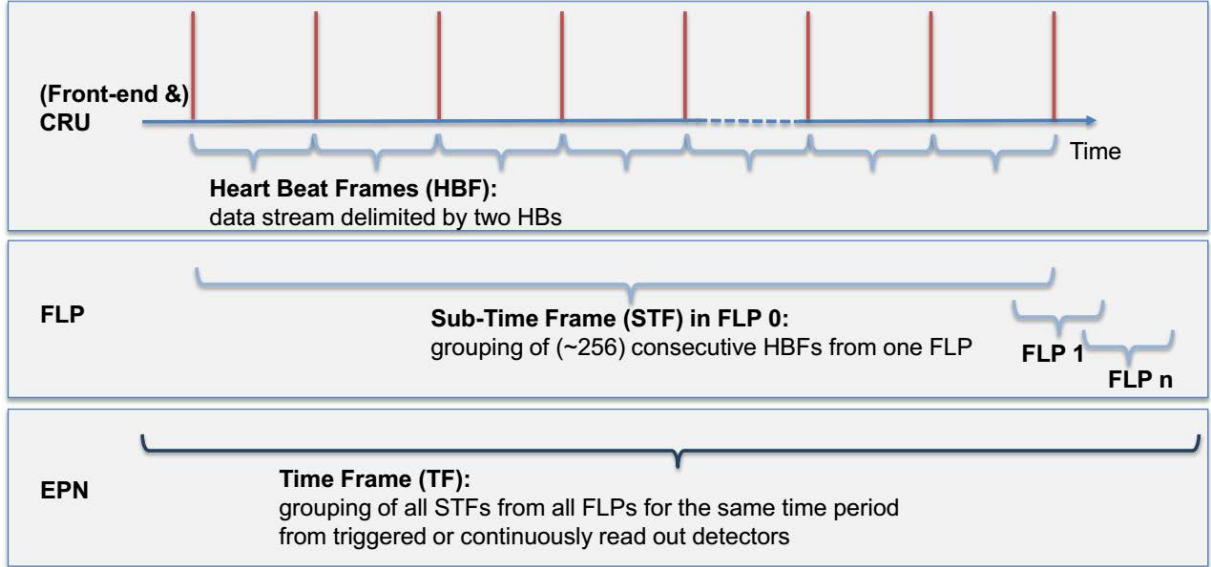


Figure 4.2: Role of Heart beat trigger in continuous readout mode

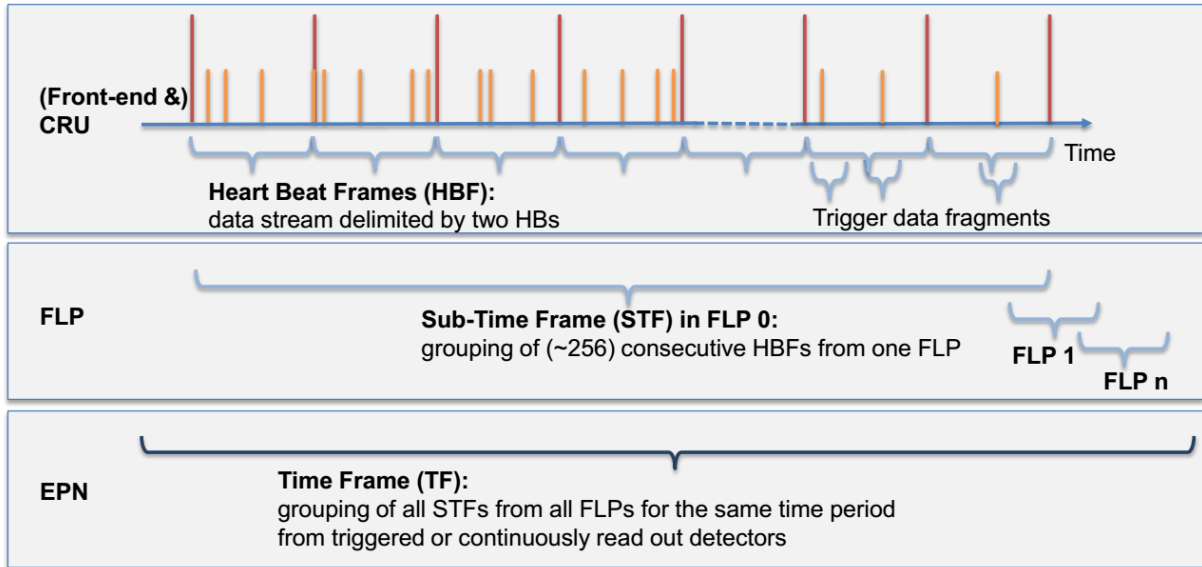


Figure 4.3: Heart beat intercepted with physics trigger to support both continuous and triggered readout mode

Table 4.4: List of Trigger Codes for Heartbeat Frames

TRIGGER CODE (<i>SENT BY CRU TO FEE</i>)	KEYWORD	PURPOSE
0x80	SOR	Start-Of-Run for assembling events for a new HB event
0x40	EOR	End-Of-Run Sequence
0x20	PAUSE	Stop assembling events
0x10	RESUME	Resume assembling events
0x08	CALIBRATE	Initiate Calibration Cycle
0x04	DUMP	Dump statistics/status registers for the associated HB event
0x02	RESET	Reset Front-End Electronics
0x01	SYNC	Synchronize detector Clocks

4.4 ALICE Clock Distribution strategy for RUN3

The ALICE detector read-out system has three configuration modes to receive the TTS information [45] : I. Detectors with non-trigger latency critical systems use the CRU connectivity only, such as the Time Projection Chamber (TPC) and Muon Tracking Chambers (MCH) systems; II. Detectors with latency critical trigger information connects directly to the CTP, such as the Inner Tracking System (ITS) ; and III. Detectors that would not upgrade to new readout architecture use the C-RORC (Run2 readout card) to receive the TTS information on the on-detector electronics via the TTC protocol. The details of the connectivity of the three modes are highlighted in the Figure 4.4.

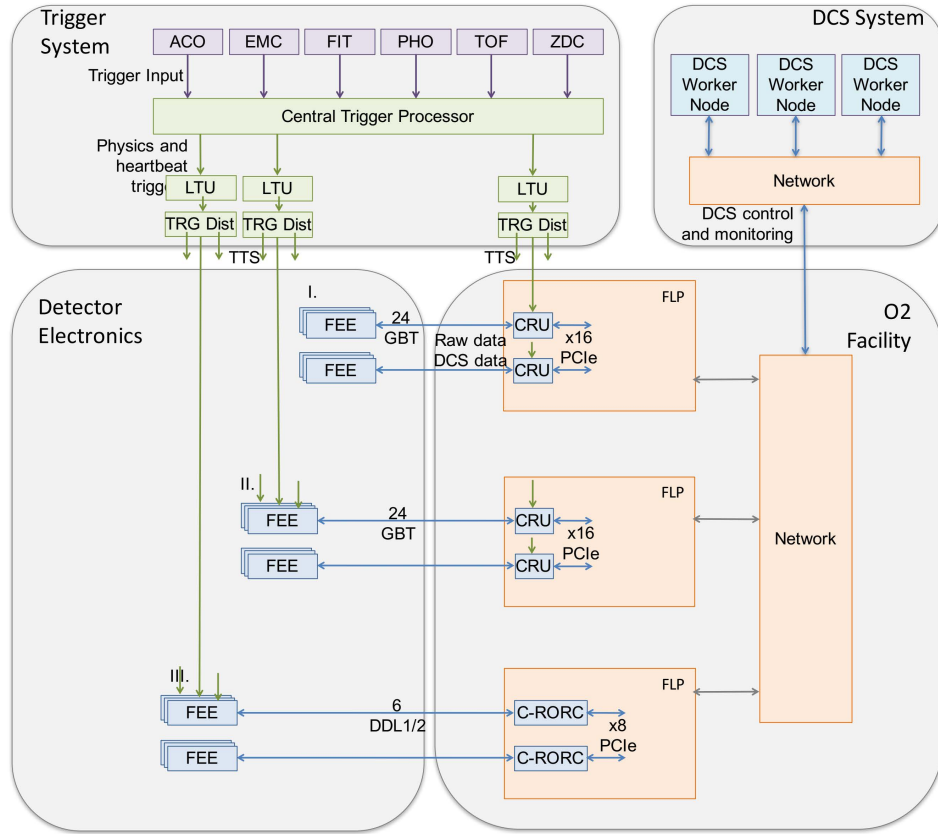


Figure 4.4: ALICE trigger distribution and detector read-out systems (The TTS distribution path is marked in green).

The detectors which operate in Type I mode, use the TTC-PON and GBT bridge connection to forward the timing information as illustrated in the Figure 2.5. The detectors

can operate in a continuous or a triggered readout mode. Depending on the configuration mode, the heartbeat trigger or physics trigger is employed.

4.5 High-Speed Serial Links in the TTC communication

For a communication system to operate reliably, one of the four classes of the clocking methods are employed, namely the asynchronous (uses no clock signal) scheme, the synchronous (uses same clock frequency and known phase) scheme, the meso-chronous (uses same clock frequency and unknown phase) scheme and the plesiochronous (uses same clock frequency but with drifting phase) scheme [49]. In the implementation of the asynchronous serial links the clock is embedded within the data stream and behaves in the same manner as the synchronous communication system. The use of embedded clock removes the need for a separate dedicated connection for a clock communication apart from the data stream. However, for the receiver side to recover the embedded clock with efficiency there need to be sufficient transition density in the transmitted bits. Common approach used to maintain the bit transition density is with the help of scrambling algorithm.

Use of elastic buffers are not preferable to prevent an anachronistic behaviour in the data packet formation. This approach helps to maintain the constant latency in the link that is necessary for arranging the data types having no time-stamp. Synchronous relationship of flow of events between the source and the receiver over an asynchronous link are preserved by maintaining a certain timing relationship at the physical level of the communication chain. An analogy has been given with the distributed systems that behave in a fully synchronous manner only after abiding certain degrees of synchrony [50]. Similarly an asynchronous system can act as a synchronous system provided it abides by certain constraints. In the comparison Table 4.5 an attempt has been made to correlate

the constraints.

Table 4.5: Degrees of Synchrony to behave as a fully synchronous system

For Distributed systems (Bounded and known)	For Asynchronous systems (Design Constraints)
Processing speed	Frequency locked
Message delivery delay	Fixed latency
Local clock rate drift	Low Jitter
Load pattern	Data Time-stamped
Difference between local clocks	Phase locked

Plenty of commercially viable high-speed asynchronous link standards are available. However, those are not suitable for application in the LHC environment. The reason being the LHC operates at a unique frequency of 40 MHz that is not compatible with the standards of the other commercial links. The use of unconventional clock frequency for data payload communication led the CERN electronics team to develop a custom solution referred as the TTC interface link standards. The TTC standards used in CRU are the GBT and the TTC-PON. Comparison between the specification of the two standards are listed in the Table 4.6. For embedding the clock in the serial link and maintaining the bit transition density, the GBT uses the block interleaver while the TTC-PON uses the 8b/10b as channel encoding scheme.

The causal relationship of the events are not preserved at the physical level of the protocol stack. Joint use of the clock synchronization and the syntonization for the recovery of the embedded clock is employed at the physical level. The approach ensures frequency stability with low jitter of the recovered phase locked clock. If the links are operated in latency optimized mode [33] then the latency or the delay path of the data lines remains constant, and is needed for the timing, trigger and control (TTC) data communication. While other levels of complex synchronous information handling like the time-stamp and the trigger management are preserved at the higher level of the CRU

Table 4.6: Showing the basic parametric difference between GBT and TTC-PON

Parameters	GBT [51, 19, 22]	TTC-PON [18]
Technology Specification	Custom	XGPON1 with modifications
Designer Group	CERN	ITU-T with CERN modifications
Line Rate	4.8 Gbps	Downstream: 9.6 Gbps Upstream: 2.4 Gbps
Payload Rate	3.2 Gbps	Downstream: ~7.68 Gbps Upstream: 640 Mbps
Payload Size	120 bits@40 MHz	Downstream: 192 bits@40 MHz Upstream: ~16 bits@40 MHz
Wavelength	850 nm (Multi-mode Tx) 1310 nm (Single-mode Tx)	Downstream: 1577 nm Upstream: 1270 nm
Network Topology	Point-to-Point	Point-to-Multipoint
Encoding	RS Encoding with Block Interleaver	8b/10b
Synchronous Trigger Support	Yes	Yes
Trigger Latency	Optical loop-back Round-trip: 150 ns	Downstream: ~100 ns Upstream: 1.6 μ s
Commercially Available	No	No

firmware logic stack.

4.6 GBT Specification

The outgoing protocol standard used for TTC communication from the CRU to the detector electronics is GBT. The GBT framework [19] defines the technology standards necessary to allow high-speed time critical data communication with high error resilience to communicate reliably from LHC radiation zone to readout electronics situated remotely. The GBT ecosystem, shown in the Figure 4.5 is composed of three parts, namely, the GBT ASIC that houses the Versatile Link chip along with the GBT slow control ASIC, Optical Fibre connection operating in a single mode (1310 nm) or a multi-mode (850 nm), GBT the Slow Control ASIC and an FPGA programmed with the GBT logic core.

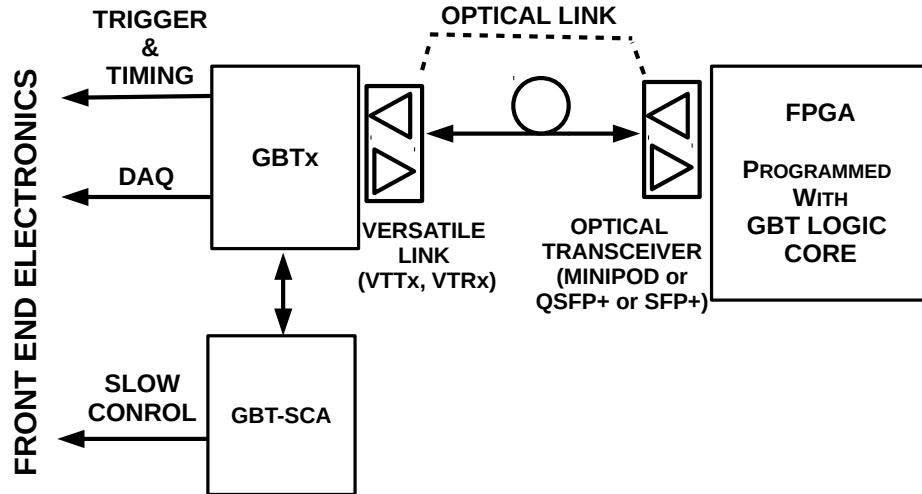


Figure 4.5: GBT ecosystem

The GBT link supports two modes of operation, the standard mode and the latency optimized mode. The latency optimized mode is needed for a time critical applications that requires constant latency. The GBT link supports two types of data frame formats, namely the GBT frame format or the Widebus format. The GBT frame format appends an error correction code formed from the Reed-Solomon algorithm cascaded with the interleaver and the scrambler. While the Widebus frame format implies that the entire bus is available for the user data and no redundant error checking bits are appended. The

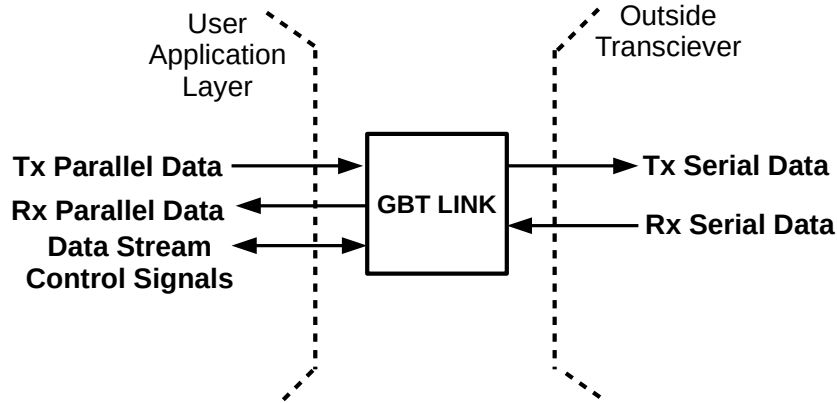


Figure 4.6: Flow connection of a GBT Link

line rate of the GBT link is 4.8 Gb/s . For the GBT scheme the effective data transfer rate is of 3.36 Gbps and for the Wide-Bus scheme the effective data transfer rate is of 4.64 Gbps .

In ALICE, maximum 24 GBT links are required per CRU board. Most part of the CRU FPGA resources are needed for the detector specific logic. To consider a way to save on the GBT specific periphery logic resources a new design approach is required. The effect of optimization on saving the logic resources is studied by Baron et. al. [52] by sharing one decoder block for several links. Another level of optimization solution possible in the Arria[®] 10 FPGA that saves on clocking resources and reduces intra-link clock skew is by using a x6 Physical Medium Attachment (PMA) bonded mode [17]. The PMA bonded mode allows *six* GBT links to be packed closely as illustrated in the Figure 4.7. Together the links are referred as the *GBT Bank*. In other words, the GBT Bank is that largest common group formed of the GBT links that are bonded or clubbed together for an FPGA resource optimization, which is six in this case. The bonded architecture comes with a constraint that all the links need to follow the same standards for a particular GBT link. Individually the settings for the GBT banks are completely configurable. For example, if a designer needs to have 20 links per CRU board then one can split it as three GBT

banks of six links and one GBT bank with two links.

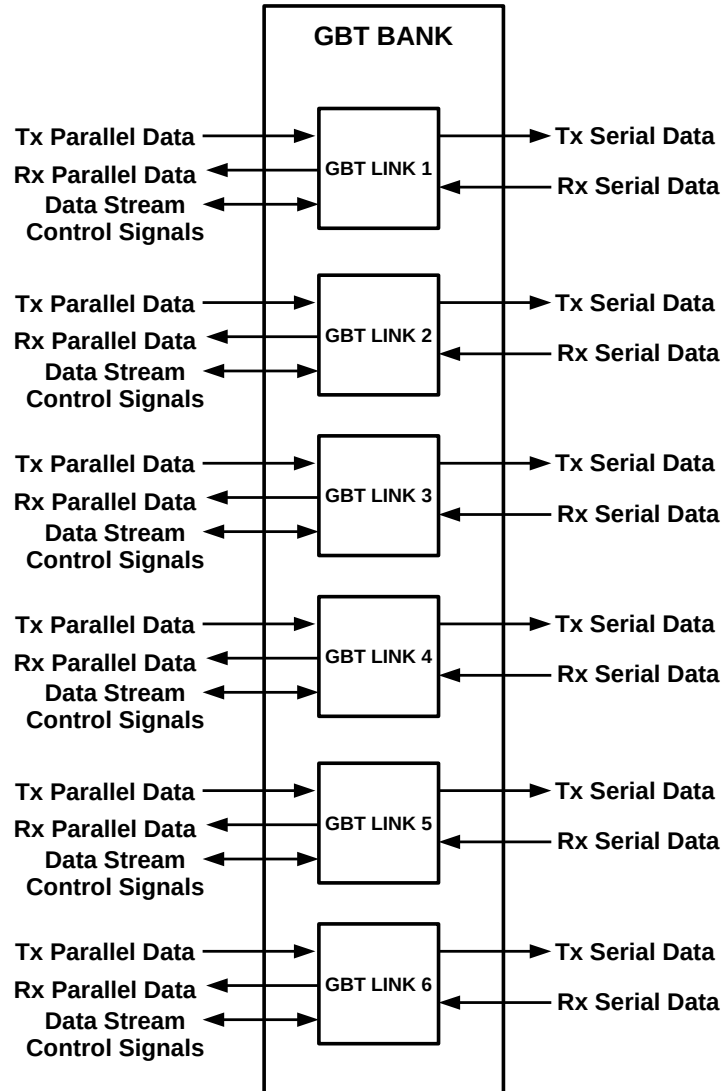


Figure 4.7: Showing how GBT Links are aggregated in 1 GBT Bank

4.7 TTC PON Introduction

The incoming protocol standard used for TTC communication between the CTP and the CRU is TTC-PON. Passive Optical Networks for the particle physics applications at CERN was first proposed in 2009 [53]. It was later extended for higher speed in 2013

[54]. The article by Papakonstantinou et. al. (2011) suggested the future use of Time Division Multiplexing (TDM) in PONs for the TTC information and the possibility of using Wavelength Division Multiplexing (WDM) PONs for the Data Acquisition (DAQ) applications. Since then the protocol has went through much up-gradation as shown in the Table 4.7. In our study 2016 development version of TTC-PON is used as shown in Table 4.6.

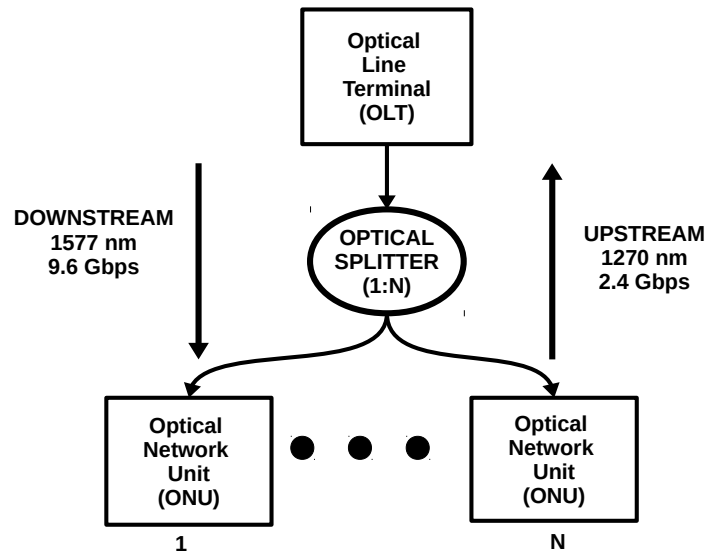


Figure 4.8: Flow connection of TTC-PON link

The TTC-PON architecture is based on PON technology that finds application in Fiber To The Home (FTTH, FTTx) networks. TTC-PON is a single fibre, bi-directional, point-to-multipoint network architecture that uses optical splitters to enable a master node or Optical Line Terminal (OLT) to communicate with multiple slave nodes or Optical Network Units (ONUs) [18], as illustrated in Figure 4.8. The downstream (from OLT to ONUs) runs at 9.6 Gbps at operating wavelength band of 1577 nm, while the upstream (from ONU to OLT) runs at 2.4 Gbps operating in wavelength window of 1270 nm. Using the TTC-PON technology, the timing, trigger and control (TTC) information from the CTP are communicated over an optical link in a time multiplexed fashion that allows a single link to transmit the TTC information to be splitted among the multiple CRUs

[15, 55], as shown in Figure 2.5. The link topology reduces the number of links to be used and hence minimizes the hardware costs involved significantly.

Table 4.7: TTC-PON GENERATIONS [56]

General Features				
Parameters	TTC-SYSTEM	TTC-PON (2011)	TTC-PON (2014)	TTC-PON (2015)
Technology	SONET/SDH OC-3	1G-EPON	10G-EPON	XGPON1
Line Rate (D/U)	80 Mbps	1.6/0.8 Gbps	11.2/2.8 Gbps	9.6/2.4 Gbps
Bidirectional	No	Yes	Yes	Yes
Fiber Length	10 - 300 m	10 - 1000 m	10 - 1000 m	10 - 1000 m
Split Ratio	1 : 32	1 : 64	1 : 64	1 : 128
BC Synchronous	Yes	1 : 64	1 : 64	1 : 128
Error Detection	Channel B Only	Yes	Yes	Yes
Downstream Features				
Line Rate	80 Mbps	1.6 Gbps	11.2 Gbps	9.6 Gbps
Wavelength	1310 nm	1490 nm	1577 nm	1577 nm
Encoding	BPM, Hamming	8b/10b	8b/10b	8b/10b
Payload	~80 Mbps	~590 Mbps	~8.64 Gbps	~7.68 Gbps
Payload / BC	< 2 bits / BC	32 bits / BC	216 bits / BC	192 bits / BC
Trigger Rate	< 40 MHz	40 MHz	40 MHz	40 MHz
- Synchronous Trigger Type	No	Yes (16 bits)	Yes (> 32 bits)	Yes (> 32 bits)
Trigger Type	~100 ns	~250 ns	~80 ns	~100 ns
	Fixed & Deterministic			
Upstream Features				
Line Rate	x	0.8 Gbps	2.8 Gbps	2.4 Gbps
Wavelength	x	1310 nm	1270 nm	1270 nm
Encoding	x	8b/10b	8b/10b	8b/10b
Total Payload (All ONUs)	x	2.2 Mbps	640 Mbps	640 Mbps
Total Payload / BC	x	3.5 bits / BC	~16 bits / BC	~16 bits / BC
ONU Burst Length	x	~ 1.6 μs	25 ns (16 bits)	25 ns (16 bits)
Max Latency for 1 : 64	x	14 μs	25 ns (16 bits)	25 ns (16 bits)
Max Latency for 1 : 128	x	x	3.2 μs	3.2 μs

4.8 Use of TTC-PON GBT bridge in CRU for TTC communication

The TTC-PON and GBT bridge connection is the interconnection between the two mutually independent GBT and TTC-PON links connected using a firmware defined logic. The bridge connection is dedicated for the delivery of the TTC payload. Different types of topology for the bridge connection are possible. For CRU design, the star topology is used for interconnection, where the TTC-PON forms the central nodal hub for forwarding the TTC information to 24 GBT links, as shown in the Figure 4.9.

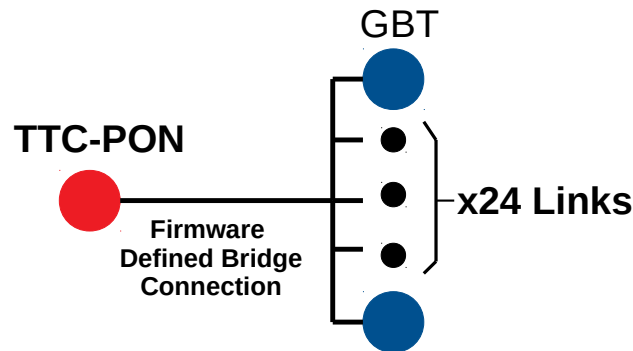


Figure 4.9: TTC-PON GBT bridge in star topology connection

The link connection between a TTC-PON to the multiple GBT is elaborated. Initial implementation and testing was based on the scheme where the 240 MHz clock is recovered from the TTC-PON'15 protocol and then fed into the jitter cleaner before forwarding it to the GBT at 120 MHz as shown in the Figure 4.10 as the configuration-I. However, the implementation on the GBT side suffered from phase inconsistency of the forward clock with each power cycle or reset cycle. The issue arised because the divider of the Multi-Gigabit Transceiver (MGT) locks the recovered fabric clock at any of the rising edges of the serial clock. The span of the GBT clock phase uncertainty lies within -4 ns to 4 ns. In the Figure 4.11 it is exhibited that for the 10,000 soft reset cycles of the firmware, the phase variation exhibits a uniform distribution over the range of [-4 ns,4 ns]. This has

been solved by calibration logic that slips the GBT Tx clock to align with the phase of the recovered clock using a Finite State Machine (FSM) [17] .

An improved design option evolved in the version upgrade of the Intel FPGA technology that has provides a provision to enable a feedback compensation mode in the transceiver PLL to ensure the deterministic nature of the clock. The feature constraints the design solution to operate 240 MHz frequency only [37]. Hence, the latest CRU firmware design uses the frequency of 240 MHz to cross the entire link chain, instead of stepping down the frequency and stepping up again. The development led to the TTC-PON and the GBT connection configuration-II as shown in the Figure 4.12. In both the configurations given in the Figures 4.10 and 4.12 the trigger data path has to cross two clock domains. Even if the clocks are phase locked and of the same frequency the FPGA sees it coming from two different sources. Hence to avoid the metastability issue in the design synchronizers are added [39]. Moreover, the proper fitter constraints are applied in the firmware logic to lock the logic placements to reduce intra-links skew variation with each firmware upgrade.

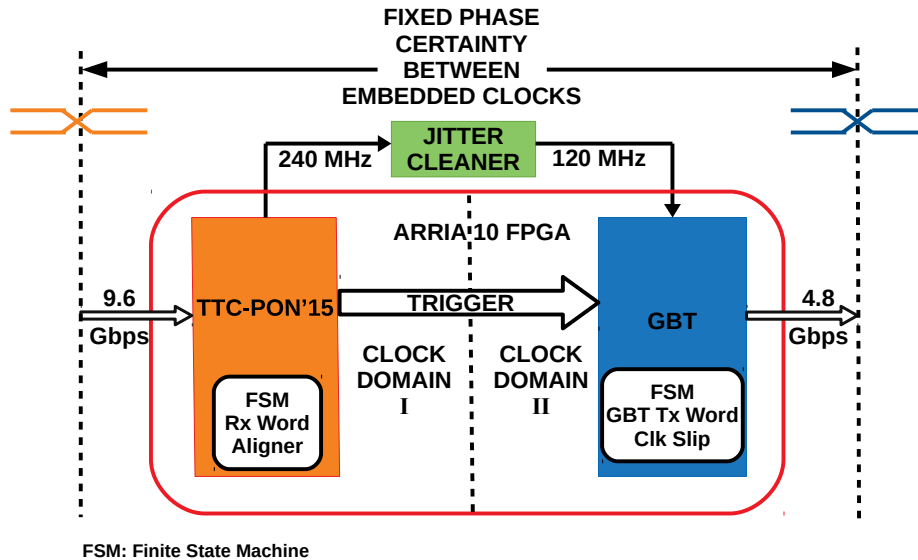


Figure 4.10: Configuration-I: TTC-PON GBT Bridge

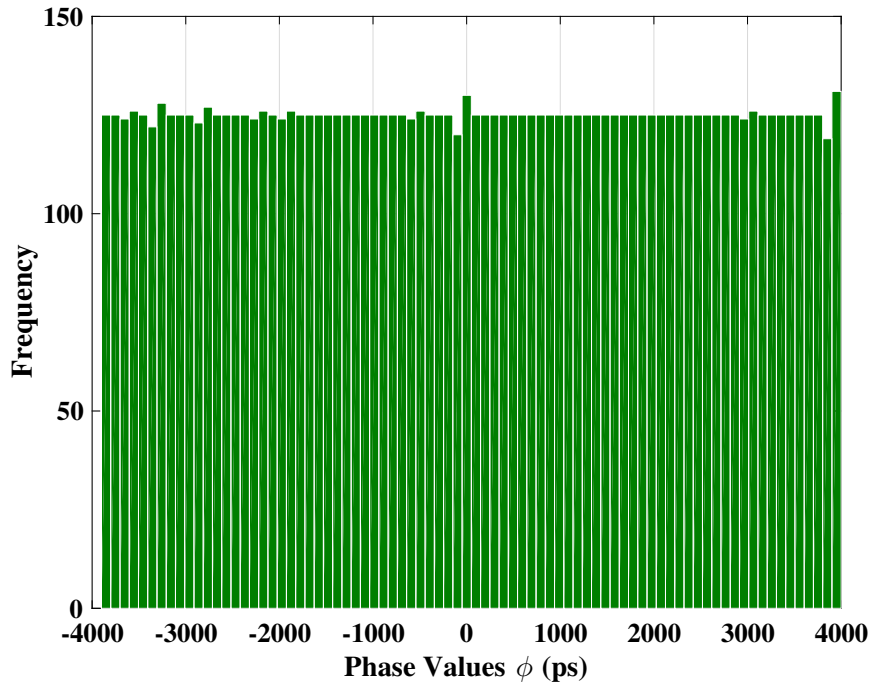


Figure 4.11: GBT Phase Uncertainty in Tx

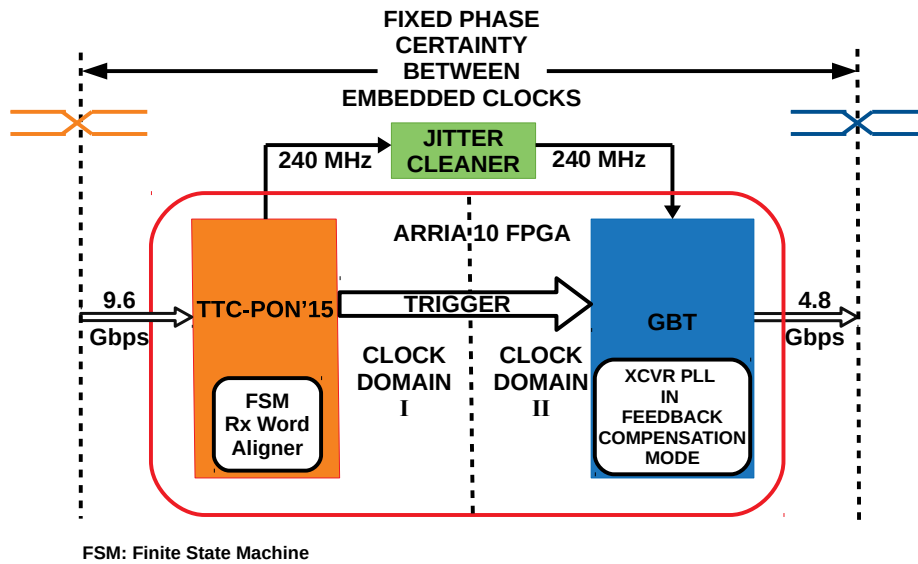


Figure 4.12: Configuration-II: TTC-PON GBT Bridge

4.9 Design Resilience

The CRU being a complex heterogeneous system has to deal with multiple links of different communication standards. During a stressful run-time scenario the stochastic fluctuations in the data link pathways might go outside the tolerable zone. The stochastic behaviour is associated with uncertainties and can trigger a chain of cascading upsets in the link chains. Hence, a quantifiable autonomous acquisition system is required to monitor and trace for any unwarranted behaviour.

As a fall back solution the house keeping tools are included with the main CRU firmware to act as a caretaker to predict any errors and disruption by tracking the macroscopic behaviours of the CRU system. Any deviation of the system behaviour if registered then flags it as a warning or an error to the system management console at the online computing system of the ALICE. The inclusive monitoring system involves the three main tools to detect any aberration in the system behavior, as shown in the Table 4.8. The monitoring system aids in the increase of the resilience and reliability of the system.

Table 4.8: Basic monitoring system to detect macroscopic behaviour

Measurement Type	Monitoring Signal	To Detect
Frequency Monitor	Derived or recovered clock	Presence of Clock
Phase Monitor	Pico-seconds resolution phase measurement between the related clocks [57]	PLL and xcvr PHY working correctly
Temperature Monitor	The FPGA chip & the CRU board	Cooling system is functional

4.10 Results

The entire trigger related logic involves role and functioning of multiple blocks. Each sub-blocks are treated individually, tested, characterized and then integrated in the design

system. The test systems are buffeted with various stress scenarios, before compiling the final result in optimum environment condition.

Since the work deals with timing information transmission, hence tests that give information about the clock quality during a link transition are included. Tests that are entailed in the following sub-sections, are the latency measurement, the jitter measurement, the BER measurement and the optimization of transceiver parameters.

4.10.1 Latency Measurement

The latency measurement gives an estimation of the logic path delay involved and also senses whether the path traverses through an elastic or an inelastic buffer. Lower the latency is more suitable it is for communication of time-sensitive information, such that service response can be delivered in the shortest period. However, variable latency means the path is ideal for data payload but not for time-critical payloads like in the case of timing and trigger. So, it is the challenge for both the protocols, the TTC-PON and the GBT, to meet the low latency with the high throughput and yet be able to preserve the same latency without any variation over the entire run period. Since the serial links traverse through the multiple clock multiplication and the subsequent division zones, the links are subjected to the risk of sudden major variations in the latencies. In the latency measurement also checked for any significant deviations that can perturb the entire time-critical pathway.

A special comma is used to measure latency in the TTC-PON. The comma character is sent every $8\ \mu s$ (K28.1) and a flag at the Optical Line Terminal (OLT) level is created to indicate the value is send. A match-flag is created when the Optical Network Unit (ONU) received the special character (just after 8b10b decoder). The latency between those two flags are then measured using the oscilloscope. Several ONU_RX resets were performed and the position between those flags was deterministic, as shown in the Figure 4.13.

For the GBT measurement, an initial version of the PCIe40 DAQ Engine with the

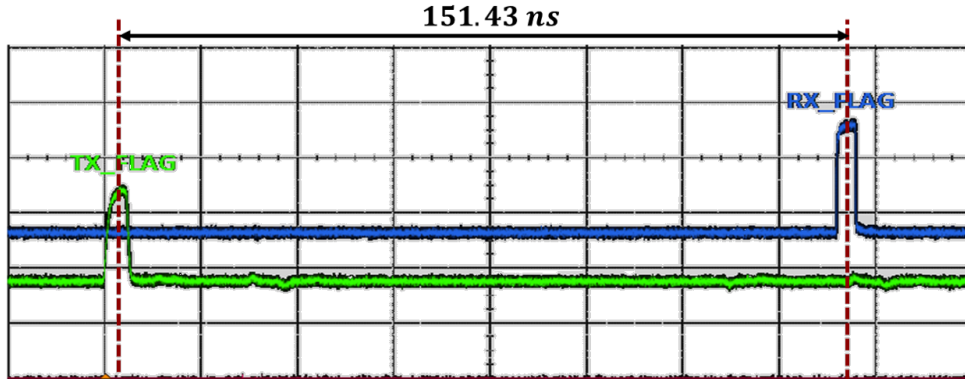


Figure 4.13: Latency of 151 ns is detected between Transmitting data and Receiving Data

Arria[®] 10 FPGA engineering sample is used. Since the setup does not allow to probe at the individual points, an ingenious solution to give a coarse estimate of the latency using firmware based measurement logic is defined. For the measurement a 32-bit ripple counter is used as the generated pattern to communicate over the GBT stream. The principle of the measurement is to enable the loopback, receive the packet, unwrap from the received GBT payload and then compare the received counter value with the sender's to estimate the round trip delay. As the GBT frame arrival rate is of 40 MHz, hence the course measurement of the round trip delay achieved is of 25 ns resolution. The round trip delay corresponds to the length of time a signal takes to be sent plus the length of time it takes for the reflected echo of that signal from the receiver to be registered. It includes the serialization and the deserialization time along with the propagation delay. For the measurement as can be seen in the Figure 3.3, *three* loopback points are chosen, those are : (1) Electrical loopback within the FPGA; (2) Optical loopback; (3) Loopback enabled at the Versatile Link Demo Board (VLDB) side. In the Table 4.9 the latency of the GBT protocol with Tx and Rx configured in the latency optimized mode or the standard mode are tabulated. However, the measurement for the Wide-Bus mode is skipped, as there was no requirement from any ALICE detector group at the time of measurement. The GBT firmware used for this measurement is the development version used in the year 2015-16 that has got 120 MHz as word clock.

Both the links exhibited stable latency even when presented with multiple soft or hard resets and power cycles. No unwanted significant deviations are registered. However, the addition of jitter is present which is a standard behaviour due to the unwanted channel interference and the system noises involved. Details about the jitter measurement of the recovered clock are covered in the sub-section 4.10.2.

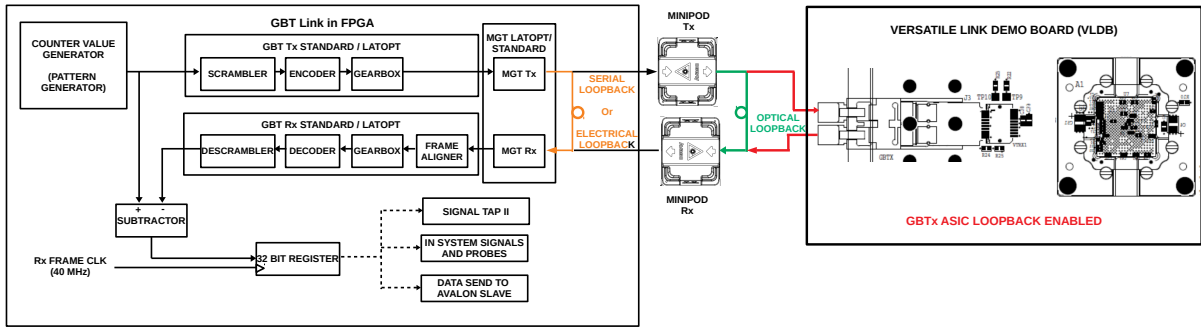


Figure 4.14: Test Setup for *three* types of round trip measurement, each loopback arrangement are marked by different colours

Table 4.9: The GBT round trip delay measurements for the multi-level loopback in the PCIe40

GBT Protocol		ROUND TRIP DELAY (Resolution of one LHC Bunch clock cycle i.e. 25ns)		
Transmission Side	Receiver Side	Serial or Electrical Loopback	Optical Loopback	GBT ASIC Loopback
<i>Latency Optimized</i>	<i>Latency Optimized</i>	150 ns	175 ns	275 ns
<i>Latency Optimized</i>	<i>Standard</i>	325 ns	350 ns	450 ns
<i>Standard</i>	<i>Latency Optimized</i>	200 ns	225 ns	325 ns
<i>Standard</i>	<i>Standard</i>	450 ns	450 ns	550 ns

4.10.2 Jitter Measurement

The asynchronous fast serial trigger links (in the CRU application the GBT and the TTC-PON) embed clock signal in the serial data transmission line. The deterministic latency of the timing information transmission is maintained by embedding the synchronization information in the serial data transmission. The embedded clock is recovered by the front end electronics of the detector to generate the *detector specific data packet*. The clock

goes through the multiple link transitions, and becomes susceptible to the system and the channel noises. The permissible range of the Root Mean Square (RMS) value of the clock jitter is specific to each sub-detectors requirement and varies with the criticality of the timing of the communication needed. The range of the RMS jitter inclusive of the specifications for all the sub-detectors of the ALICE, typically lies within the range of $300\text{ ps} - 20\text{ ps}$. A preliminary study is conducted to determine if the RMS jitter value of the forwarded clock is less than the lowest jitter requirement of 20 ps . The test uses the TTC-PON'15 version and the GBT version operating at 240 MHz clock domain. In the Intel[®] Arria[®] 10 FPGA devices three types of transmitting PLLs are available: (a) the Advanced Transmit PLL (ATX PLL); (b) the Fractional PLL (fPLL); (c) the Clock Multiplier Unit PLL (CMU PLL) or the channel PLL. Since the design uses bonded application, hence the ATX PLL and the fPLL can only be used. The best jitter performance is seen using the ATX PLL over the fPLL. However, the recommendation based on the data rates from the Intel is to use the fPLL for the transmit PLL [37], hence in the design fPLL is used.

During the pre-validation test the ideal test scenarios are prototyped with the FPGA development boards having the same family of FPGAs as the final production version. To illustrate for the CTP and for the CRU, the Kintex Ultrascale and the Arria[®] 10 development boards are used respectively. For rapid prototyping of the test design, a *split hardware* setup is used to model the behaviour of the CRU. It means that the GBT protocol is implemented in one Arria[®] 10 FPGA board and the TTC-PON in other Arria[®] 10 FPGA board, while the clock is transmitted from one board to the other after jitter cleaning it in the SI5344 PLL module. The split hardware model allows an easy access to the *Test Points (TP)* for the jitter measurement test setup as can be seen in the Figure 4.15. The settings of the configuration parameters of the SI5344 PLL module and the Versatile Link Demo Board (VLDB) used for the test is given in the Table 4.10 and the Table 4.11 respectively.

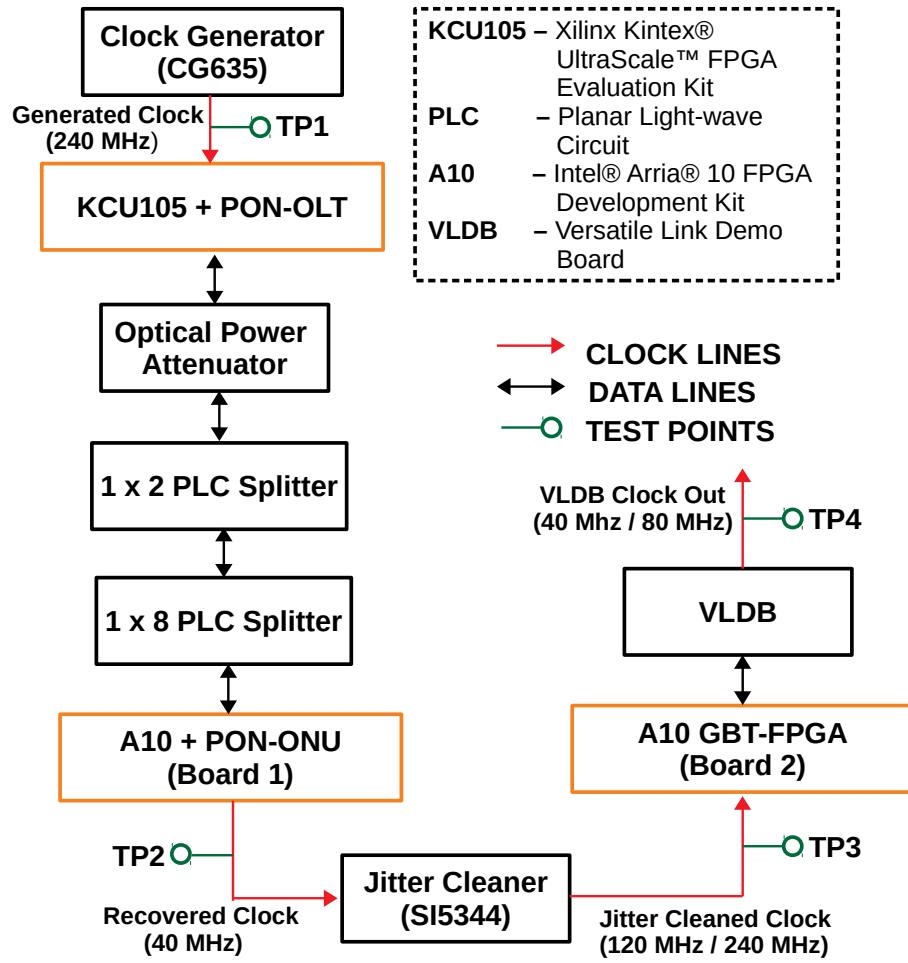


Figure 4.15: Jitter measurement test setup for the TTC-PON GBT bridge

Table 4.10: SI5344 *Rev B* PLL Configuration

Parameters	Status
Free Run Only Mode	: Not Enabled
Zero Delay Mode	: Not Enabled
Loop Bandwidth	: 200 Hz
Fastlock Enable	: ON
Fastlock Loop Bandwidth	: 1 kHz
Hitless Switching	: Not Enabled
HoldOff Mode	: Disabled ¹
Termination	: LVCMOS In-Phase 1.8 V 31 Ω

¹ REG: 0x052C[0] = 0x0

Table 4.11: VLDB Configuration

Parameters	Status
Elink Number	: 0
Channel Number	: 0
Data Transmission	: Off

The *Phase noise* representation used for the jitter measurement gives an accurate estimation of the phase fluctuations in the frequency domain analysis. Phase noise is determined as the ratio of the noise in a 1-Hz bandwidth at a specified frequency offset, f_m , to the oscillator signal amplitude at frequency f_O . The unit used is dBc/Hz , where dBc (decibels relative to the carrier) is the power ratio of a signal to a carrier signal, expressed in decibels. It is conventional to characterize an oscillator in terms of its single-sideband phase noise as shown in the Figure 4.16, where the phase noise is in dBc/Hz plotted as a function of frequency offset, f_m , with the frequency axis on a log scale. The RMS jitter (in linear terms not dB) is calculated from a piecewise linear integration of the single sideband phase noise data points. The Equation 4.2 used for calculation is adapted from the Tutorial MT-008 Analog Devices [1]. The results are correlated with the Phase Noise Analyzer software generated values.

$$A = \text{Area} = \text{Integrated Phase Noise Power (dBc)},$$

$$RMS \text{ Jitter} \approx \frac{\sqrt{2 \times 10^{A/10}}}{2\pi f_O}, \quad (4.2)$$

where f_O is the oscillator frequency.

For performing the integration on the *phase noise power values*, the *trapezoidal rule* [58] is used over a defined bandwidth given by the Equation 4.3.

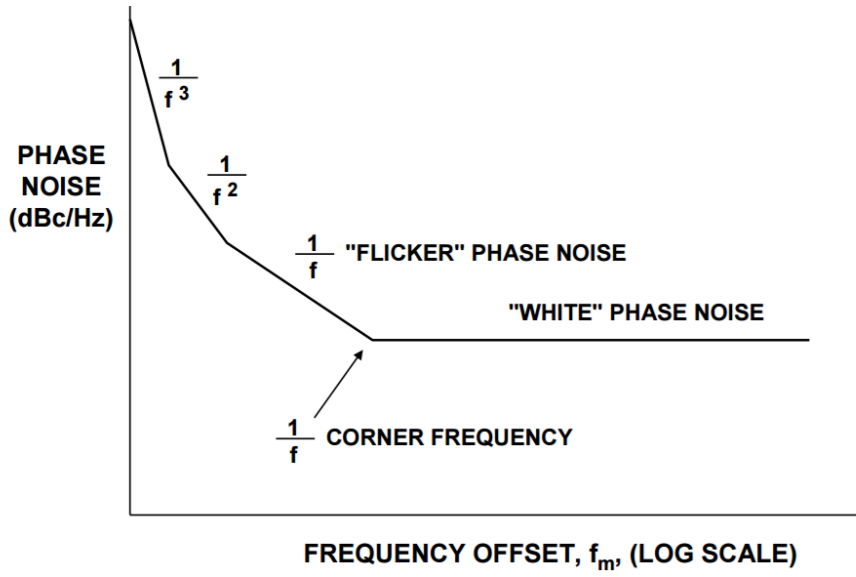


Figure 4.16: Oscillator phase noise in dBc/Hz vs. frequency offset [1]

$$\int_a^b f(x) dx \approx (b - a) \frac{f(b) + f(a)}{2} \quad (4.3)$$

The experiment calculates the RMS Jitter value from the phase noise power within the bandwidth of **10 Hz to 20 MHz**. As the zone of operation is in the high frequency range, hence the effect of the lower frequency phase noise is neglected [59]. This implies that even though the integrated jitter value computed over the plot A is evaluated higher in relative to the other curve B does not implies that the curve A is better than the curve B. However, if the phase noise curve at the high frequency region of interest for the curve A is below the curve B, then the curve A is considered to be better in performance than the curve B. Such concept is applied in the interpretation of the phase noise measurement plots shown in the Figures 4.17, 4.18, 4.19, 4.20 and 4.22. The oscilloscope settings are kept same for the different measurements for the sake of consistency.

The results contain the measurement of phase noise of clock output from the different test points (TP) in the link chain. The Arria[®] 10 transceiver specific phase noise data

points [60] relative to the reference clock phase noise are also included. The Arria[®] 10 transceivers in the Intel data-sheet [60] gives the phase noise points for the reference clock operating at 622 *MHz* frequency. The REFCLK phase noise requirement at frequencies other than the 622 *MHz* is calculated using the Equation 4.4 adapted from the Intel data-sheet [60].

$$\begin{aligned} &\text{REFCLK phase noise at } f \text{ (MHz)} \\ &= \text{REFCLK phase noise at 622 MHz} + 20 * \log(f/622) \end{aligned} \quad (4.4)$$

The details of the tapping points used during the measurement is given as illustration in the Figure 4.15. Following measurements are conducted to evaluate the performance and to achieve the best jitter cleaning effect.

Performance comparison between SI53XX PLL family

The initial purpose for the study of phase noise measurement is to determine the PLL family that fulfills the CRU requirement. PLLs from the different vendors are characterized by the CERN electronic team members, out of which the PLLs from Silicon Labs SI53XX family found to be suitable for the jitter cleaning requirement for the LHC timing signal. Figure 4.17 gives comparative result of the jitter cleaning performance of PLLs belonging to the SI53XX family namely the SI5338 and the SI5344. The test demonstrates that the SI5344 PLL jitter cleaning is a better match for the phase noise requirement at 240 MHz reference clock frequency for the Arria[®] 10 FPGA SerDes. The test points used for the measurements are TP1 and TP3. The study plays a significant role in deciding the PLL type to be installed in the CRU hardware PCIe40 DAQ Engine. The SI5345 PLL having 10 output nodes, is a variant of SI5344 PLL family [61], which is chosen as the onboard jitter cleaner for the CRU PCIe40 DAQ Engine [16].

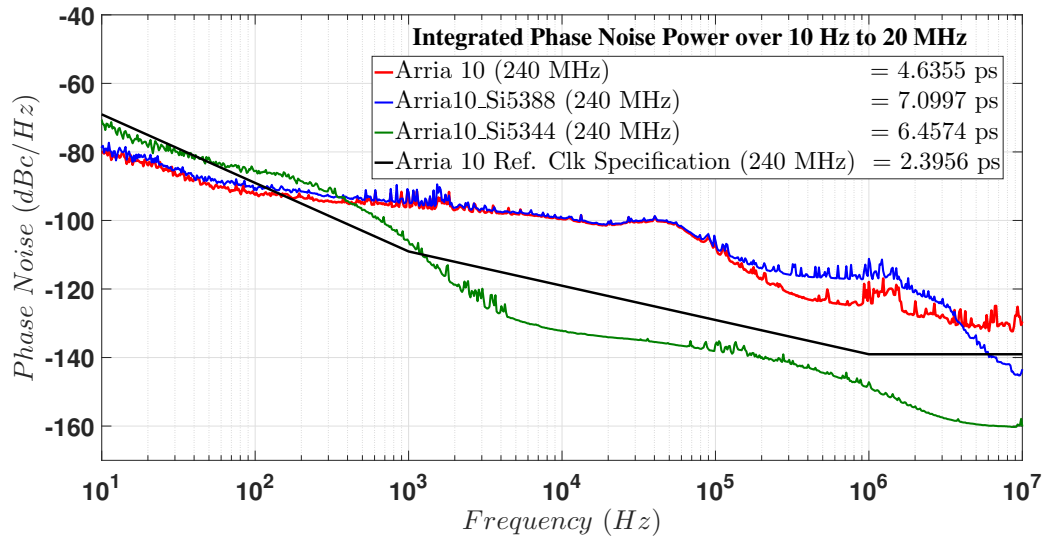


Figure 4.17: Comparison of jitter cleaning performance between the two jitter cleaners

Performance comparison with PLL bandwidth variation

Following test is to determine at which bandwidth configuration the PLL performs at its best. The Figure 4.18 shows the phase noise study done on the clock signal tapped at the test point TP3. The test is to evaluate the effect of the bandwidth variation on the integrated RMS jitter. From the plot it can be inferred that the PLL gives best jitter cleaning performance for the 200 Hz bandwidth configuration mode.

Performance comparison with variation in output clock frequency

After investigating the effect of the operational bandwidth on the jitter cleaning PLL, the other study that requires attention is the choice of the output clock frequency of the PLL that gives a better jitter attenuation. The CRU firmware design can work at two operational frequencies namely 120 MHz and 240 MHz. The tests with the two frequencies at test point TP3 is shown in the Figure 4.19. The CRU firmware with the current specification uses 240 MHz frequency to make transit of the timing signal from the TTC-PON to the GBT without stepping down of frequency at intermediate points. The result shows that the jitter cleaner is able to satisfy the requirement of the Arria®

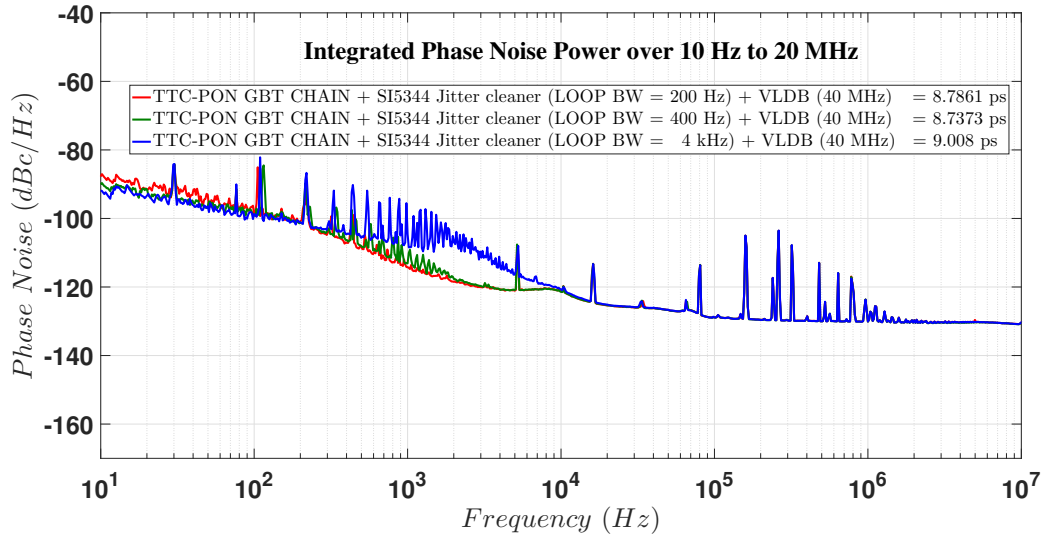


Figure 4.18: Phase Noise Analysis (Total Jitter) for VLDB clock output at 40 MHz frequency having the full TTC-PON – GBT bridge connection with the PLL configured at loop bandwidth 400 Hz and 4 kHz

10 FPGA jitter specification of SerDes at 240 MHz reference clock frequency.

Effect of jitter cleaning performance on integrated GBT and TTC-PON chain

The subsequent test for the SI5344 PLL performance is conducted while fitted in the integrated system. The test points TP1, TP2 and TP3 are used to derive the phase noise plot as shown in the Figure 4.20. The test result validates that the jitter cleaning performance of the PLL keeps the jitter within the tolerable level as specified for the Arria[®] 10 FPGA.

Comparison of jitter cleaner performance against an ideal test case scenario

The purpose of the test is to compare the presence of jitter in an ideal experimental condition against the practical scenario with jitter cleaner in use. The terminal destination of the embedded clock signal in the link chain is the delivery to the GBT chipset. In our test case VLDB is used, as it houses the GBT chipset. The quality of the embedded clock received by the end-point VLDB is studied, where the recovered output clock frequency is

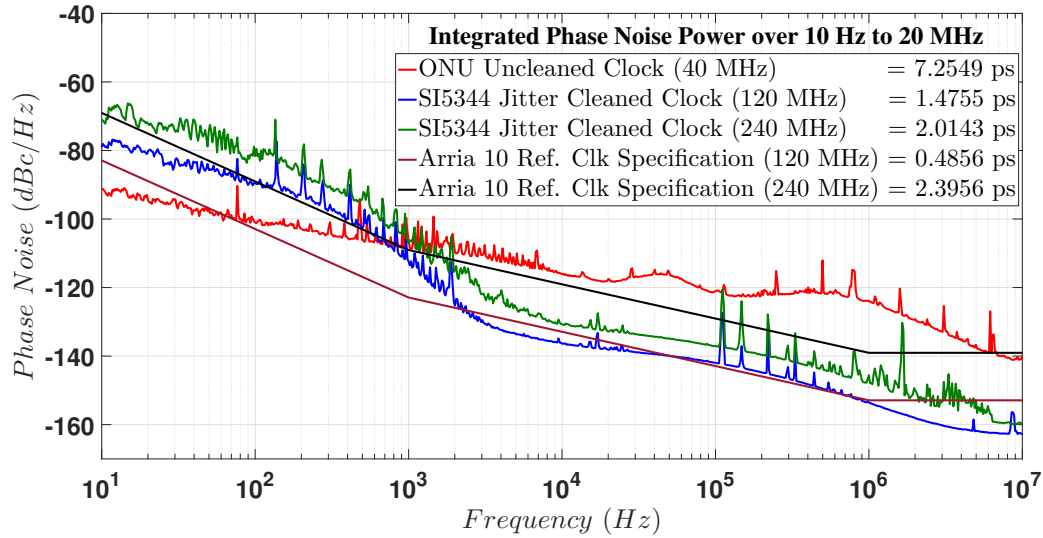


Figure 4.19: Phase noise analysis (total jitter) for ONU uncleaned clock (40MHz), SI5344 jitter cleaned clock (120 MHz) and SI5344 jitter cleaned Clock (240 MHz)

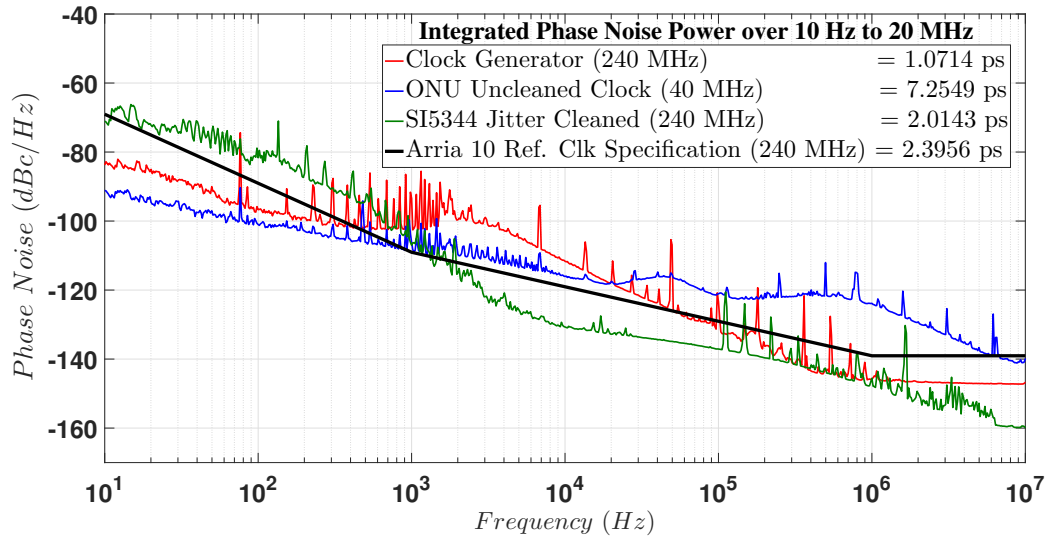


Figure 4.20: Phase noise analysis (total jitter) for source clock (240MHz), ONU uncleaned clock (40 MHz) and SI5344 jitter cleaned clock (240 MHz)

set at 40 MHz and 80 MHz respectively. Two types of connection chains are constituted for the test. For the ideal test scenario, all the noisy source points are dropped and the transition points are minimized. It is composed of clock signal originating directly from the clock generator, that gets embedded using the CRU firmware to GBT payload and finally getting communicated to VLDB to be recovered in 40/80 MHz frequency. The setup connection is shown in the Figure 4.21. For the practical test scenario, the formerly used experimental setup is utilized as shown in the Figure 4.15. The results are plotted in the Figure 4.22 and test result are referred as "VLDB CLK OUT WITH GBT" and "VLDB CLK OUT WITH TTC-PON GBT BRIDGE" respectively. The results of the two test setups show strong positive correlation, hence it can be concluded that the SI5344 jitter cleaner can successfully be employed for the cleaning of the embedded clock during traversal of the TTC-PON and GBT bridge connection.

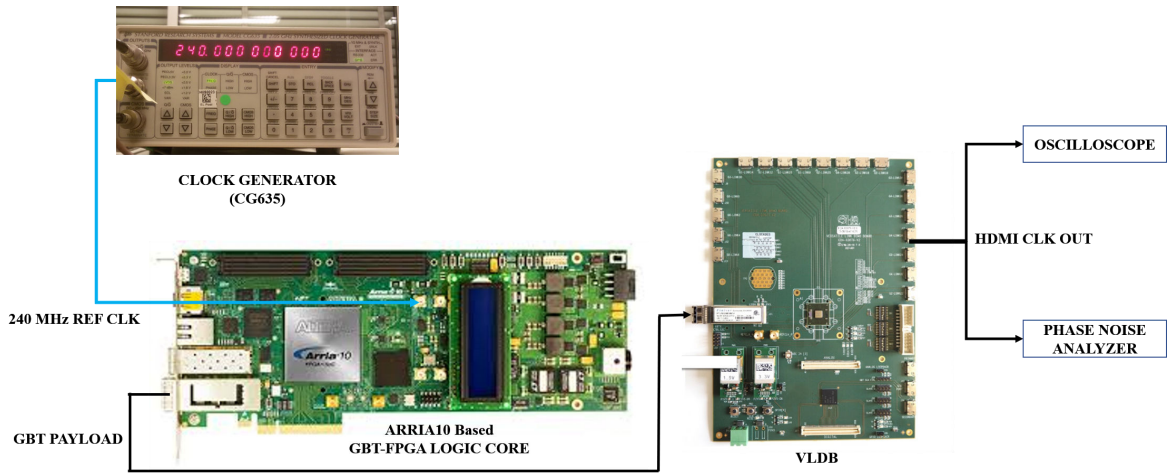


Figure 4.21: Jitter measurement test setup with GBT CHAIN and VLDB only

Comparison of jitter value at intermediate points

A comparison of all the configurations discussed in the phase noise plots for the integrated RMS jitter is tabulated in the Table 4.12. The table information is used to derive the Jitter Transfer Function (JTF) of the individual composite link elements in the link chain. The JTF [62] is the ratio of the output jitter to the applied jitter on the reference clock,

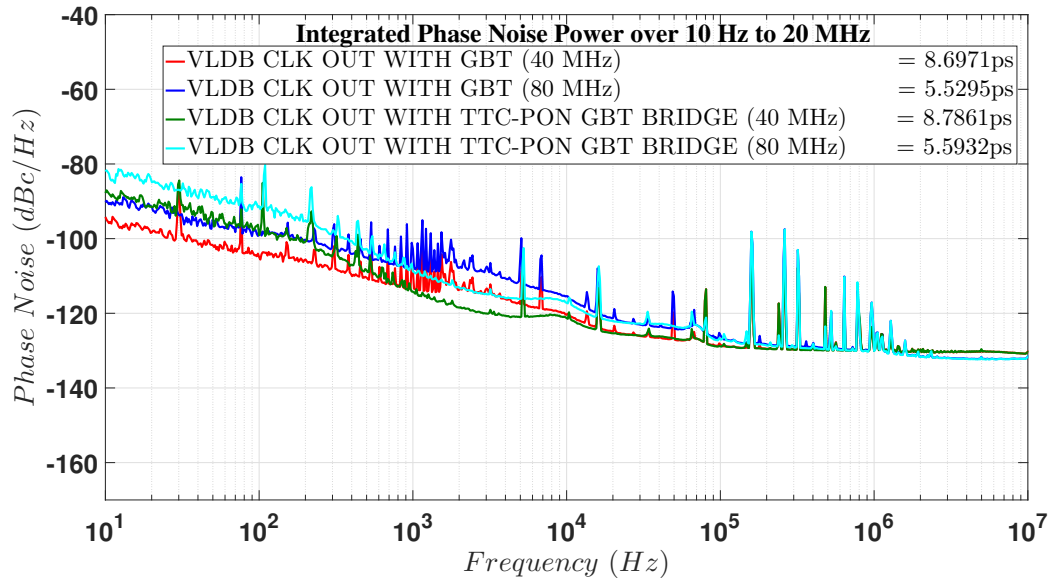


Figure 4.22: Phase noise analysis (total jitter) for the VLDB clock output of : 40 MHz o/p with the GBT link connection, 80 MHz o/p with the GBT link connection, 40 MHz o/p with the TTC-PON – GBT bridge connection and 80 MHz o/p with the TTC-PON – GBT bridge connection

Table 4.12: Comparison of *RMS Jitter* results

CONNECTION TYPE	RANDOM JITTER (ps)	PERIODIC JITTER (ps)	TOTAL JITTER (ps)
Clock generator (240 MHz)	0.967	1.071	1.071
ONU Uncleaned Clock (40 MHz)	6.481	7.298	7.255
Si5344 Jitter Cleaned Clock (120 MHz)	1.327	1.447	1.475
Si5344 Jitter Cleaned Clock (240 MHz)	1.265	1.407	2.014
VLDB CLK OUT WITH GBT (40 MHz)	7.409	8.697	8.697
VLDB CLK OUT WITH GBT (80 MHz)	3.501	3.884	5.529
VLDB CLK OUT WITH TTC-PON GBT BRIDGE (40 MHz)	7.507	8.799	8.786
VLDB CLK OUT WITH TTC-PON GBT BRIDGE (80 MHz)	3.615	4.048	5.593
VLDB CLK OUT WITH TTC-PON GBT BRIDGE (40 MHz) HAVING SI5344 LOOP BW 400 Hz	7.439	8.738	8.737
VLDB CLK OUT WITH TTC-PON GBT BRIDGE (40 MHz) HAVING SI5344 LOOP BW 4 KHz	7.727	9.018	9.008

where both the signals are measured as a function of the frequency. The calculated values of the JTF are given in the Table 4.13. To summarize the SI5344 PLL satisfies the jitter cleaning requirement of the clock needed in the TTC-PON and GBT bridge connection transition.

Table 4.13: Comparison of the *RMS Jitter* at each intermediate points in CRU TTC-PON GBT bridge connection

CONNECTION TYPE	INPUT TOTAL RMS JITTER (ps)	OUTPUT TOTAL RMS JITTER (ps)	JITTER TRANSFER (dB)
Clock generator (240 MHz) to ONU Uncleaned Clock (40MHz)	1.071	7.255	16.616
ONU Uncleaned Clock (40MHz) to Jitter Cleaned Clock (240 MHz)	7.255	2.014	-11.132
Jitter Cleaned Clock (240 MHz) to VLDB Clock Out (40 MHz)	2.014	8.697	12.706

4.10.3 BER Measurement

The Intel[®] Arria[®] 10 development board is used for conducting the test. A 10G 850nm Multimode Datacom SFP+ transceiver is used for the optical signal transmission in the GBT protocol at 4.8 Gbps line rate. A customized variable fiber optic solution having in-line attenuator capability within the range of 0-60 dB is used for providing attenuation to the signal. For optical power measurement, a hand held power meter with a SC-ST connector is used. The attenuator cable adds an additional *Insertion loss* of ≤ 3 dB to the entire test chain. Individual snapshot of the measurement setup is provided in the Figure 4.23. Interested users can use the firmware uploaded at the CERN Gitlab link [63] to reproduce the results in other hardware conditions. For the BER measurement default transceiver configuration is used.

BER is evaluated from the ratio of the number of errors received to the total number

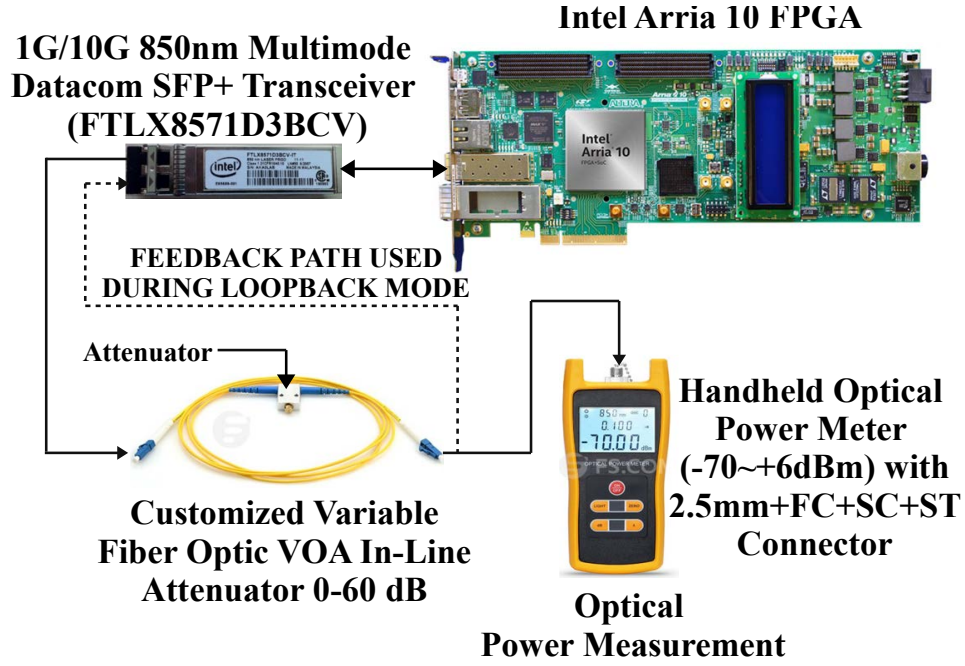


Figure 4.23: GBT Measurement Setup

of bits transmitted. Ideally as the number of transmitted bits approaches infinity, the BER becomes a perfect estimate. However, for practical tests there is a need for test procedure that allows to measure BER with a high confidence level. J. Rudd [64] has documented a method for reducing the test time for stressing a system, by calculating the number of bits needed to be transmitted to estimate error probability with a particular statistical confidence level. Equation (4.5) shows the trade-off between test time (T) and confidence level (CL).

$$\left. \begin{aligned} n &= -\frac{\ln(1 - CL)}{BER} + \frac{\ln\left(\sum_{k=0}^N \frac{(n * BER)^k}{k!}\right)}{BER} \\ T &= n/R \end{aligned} \right\} \quad (4.5)$$

where n is the total number bits transmitted, N is the number of errors that occurred during the transmission and R is the line rate. For $N = 0$ the solution is trivial. In the

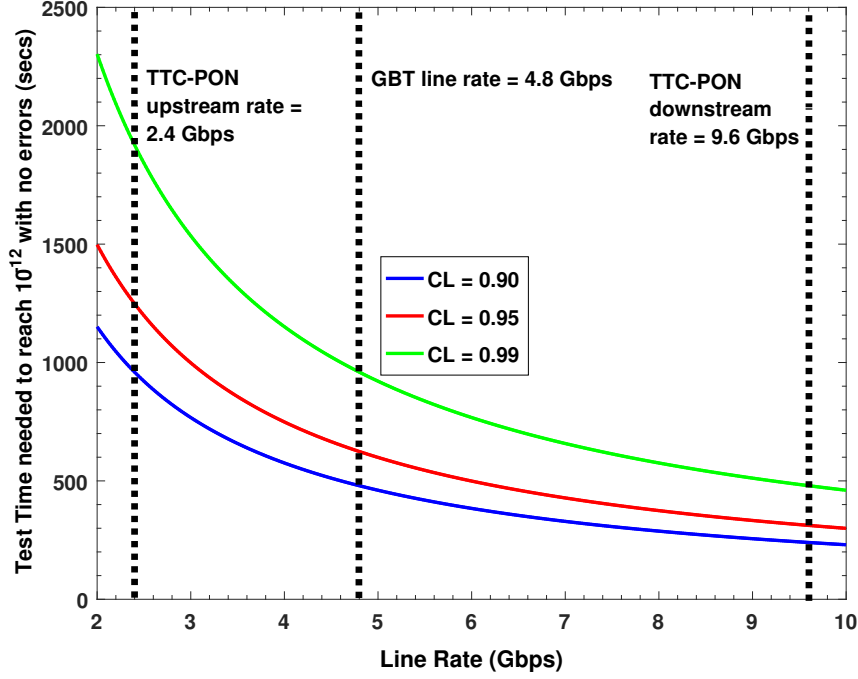


Figure 4.24: Time required to reach 10^{-12} BER vs. line rate. Showing the cases for GBT and TTC-PON.

work of Detraz et. al. [65] an effort has been made to define the minimum experiment time required for GBT BER measurement with different level of confidence as derived from the Eq. (4.5). Further the concept is extended and marked for the minimum measurement time needed for TTC-PON link also, as shown in Figure 4.24.

BER Measurement for the GBT

The GBT BER measurement for the GBT encoding scheme operating in the GBT mode and the Widebus mode is plotted in the Figure 4.25. An exponential fit to the readout data is done. Below -17 dBm receiver sensitivity, due to loss of clock, further BER measurement cannot be pursued. However, the plot can be extrapolated based on standard

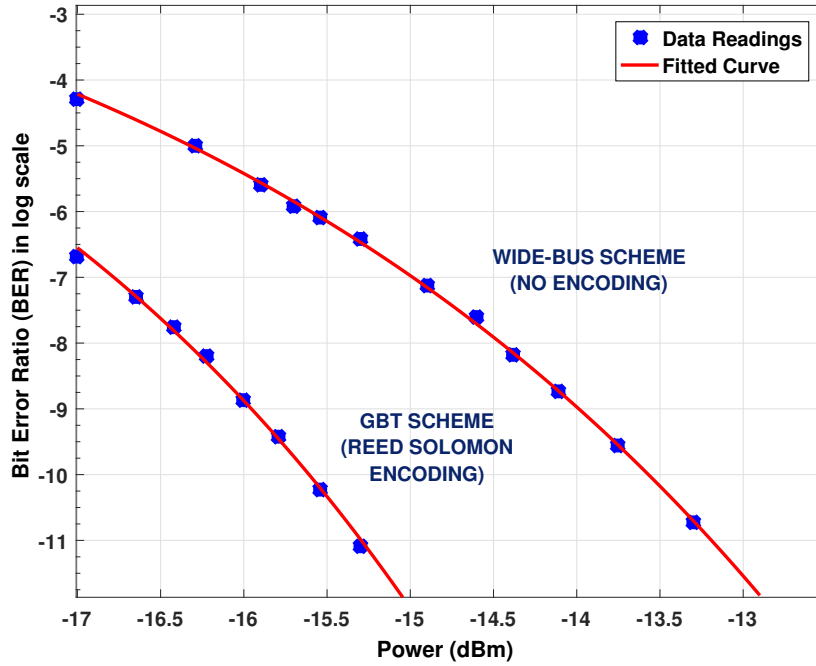


Figure 4.25: Showing GBT BER measurement for GBT and widebus FEC scheme

‘erfc’ based nature of the curve, assuming Gaussian noise.

$$\begin{aligned}
 & \text{Margin of Receiver Sensitivity for targetted BER } 10^{-12} \\
 & \text{between both the scheme} = (15 - 12.9) \text{ dBm} = 2.1 \text{ dBm}
 \end{aligned} \tag{4.6}$$

The difference measured is 2.1 *dBm* as given in Eq. (4.6). The result is in close agreement to the measurement conducted by Csaba Soos for GBT protocol implementation implemented on Xilinx FPGA [66], that is around 2.5 *dBm*.

The GBT link Signal Quality Data from the FPGA transceivers are transmitted using QSFP+ transceiver modules to convert the electrical signals to the optical signals for communication over a single mode fibre. A Lecroy Serial Data Analyser (SDA) oscilloscope is used for analyzing the signal quality. An eye diagram is used as an indicator to measure the quality of the optical transmission signals at the GBT line rate of 4.8

Gbps. The signal to noise ratio of the high-speed data signal is directly indicated by the amount of eye closure or Eye Height. For the GBT transmission signal using a QSFP+ transceiver module, an *eye height* of 406.6 mV and an *eye width* of 173.3 ps is achieved as shown in the Figure 3.2a.

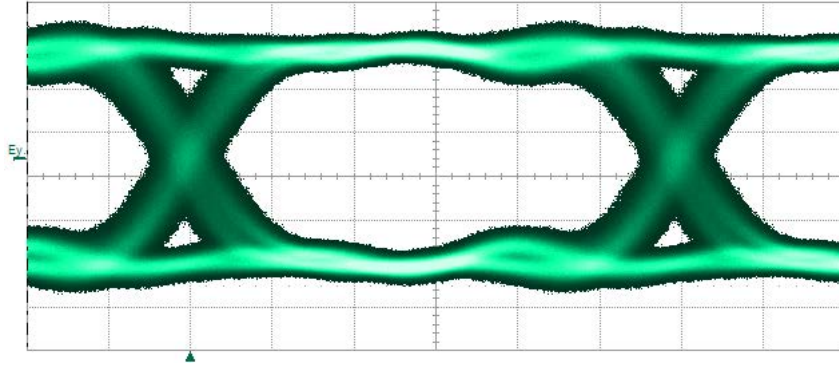


Figure 4.26: Showing the Eye Diagram for Tx Optical using GBT encoded data for Arria[®] 10

Power Measurement Intel internal monitoring tools are used to register the power consumed and the temperature of the Arria[®] 10 FPGA chip during the test measurement. Figure 4.27 shows the power variation plot as monitored using the tool and Table 4.14

Table 4.14: Power consumed by a single GBT link for different encoding scheme

Encoding Scheme	Board Temp. °C	FPGA Temp. °C	RMS Current (I_{RMS}) (mA)	Voltage (V) (mV)	Power ($P = V \times I_{RMS}$) (mW)
GBT	37	42	4911.42	914.77	4492.82
Widebus	37	42	4463.20	914.66	4082.31

summarizes the results collected when a single GBT link under the different encoding scheme consumes different power.

In the CRU project, the link connecting the radiation hard component to the non-radiation hard component, is based on the GBT link technology operating at 4.8 Gbit/s using a 850 nm multimode optical fibre. The link is connected between the versatile link transceiver to a Multifiber Push-On (MPO) optical connector at the CRU PCIe40 DAQ



(a)



(b)

Figure 4.27: Showing power Monitor of GBT Firmware during data communication mode in : (a) GBT Scheme and (b) Widebus Scheme

Engine using an optical fibre splitter. A study has already been conducted by Schwemmer et. al. [2] to note the performance on 400 m long OM3 and OM4 cables. So, further test on optical cable characterization is not pursued.

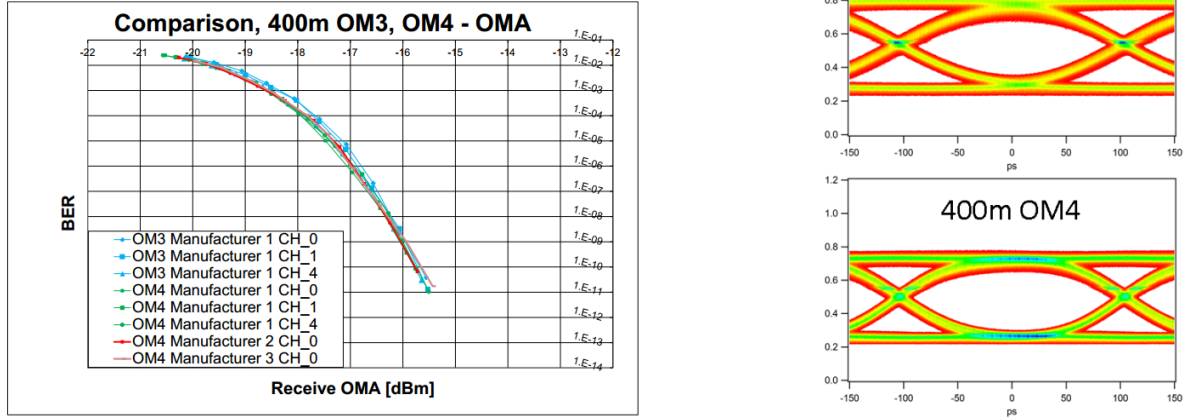


Figure 4.28: OM3 and OM4 BER curves and Eye Diagrams (Adapted the research article Schwemmer et. al. [2])

BER Measurement for the TTC-PON

The test setup involves an optical connection from the Kintex[®] UltrascaleTM FPGA having a Optical Line Terminal (OLT) module to the Intel[®] Arria[®] 10 FPGA having a Optical Network Unit (ONU) module. For emulating the real experiment conditions an inline power attenuator and an optical splitter is used. Details of the experimental setup is illustrated graphically in the Figure 4.30. The transmission part on the ONU-SFP on the Intel FPGA board was disabled for the upstream path as it was not implemented during the pre-validation measurement test. The ONU just acted as a receiver. A pseudorandom binary sequence (PRBS) generator was incorporate in the OLT Kintex[®] UltrascaleTM design side while a PRBS checker was used in the ONU Arria[®] 10 design side in order to perform the test. Attenuation was changed in the range of 10.0 dB : 1.0 dB : 20.0 dB. For each attenuation value, the average optical received power at the ONU level was measured

using the JDSU OLP-87 PON Power Meter. The results are tabulated in the Table 4.15. Transmitted power of the OLT is +3.67 dBm. For each change in the attenuation value, 5×10^{11} bits are transmitted to measure the quantity of errors.

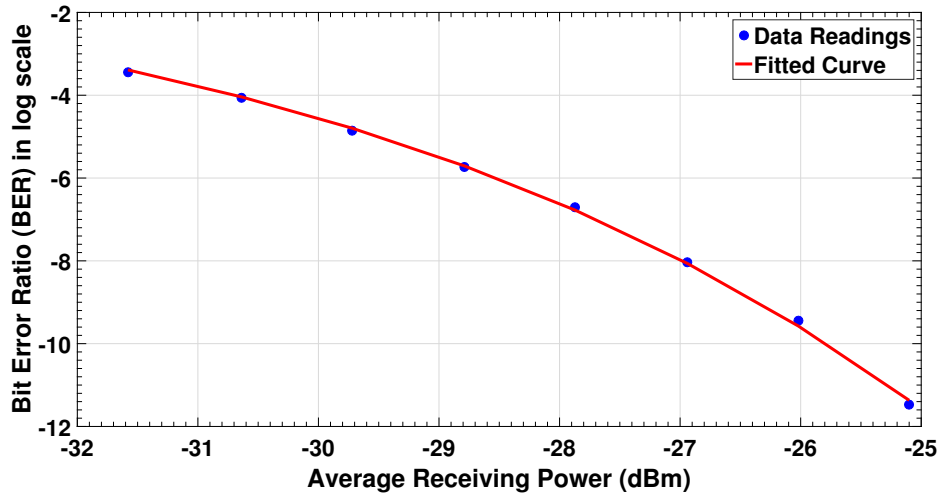


Figure 4.29: Downstream Bit Error Rate (BER) from the OLT to the ONU

Table 4.15: Variation of the received power with change in the attenuation

Attenuation (dB)	Received Power (dBm)
20	-31.58
19	-30.64
18	-29.72
17	-28.79
16	-27.87
15	-26.94
14	-26.02
13	-25.10
12	-24.19
11	-23.25
10	-22.32
0	-13.13

The TTC PON Arria[®] 10 Signal Quality Figure 4.31 shows the transmission quality of the optical signal of the TTC-PON from the CTP prototype board, that uses a

Xilinx Ultrascale FPGA. The Figure 4.32 shows the receiving signal quality of the TTC-PON as monitored within the Arria[®] 10 FPGA using a transceiver toolkit (TTK). The Eye Width to the Eye height ratio of 53/39 is registered in the TTK for the test bits of 1.1E12 using the PRBS31 stress pattern.

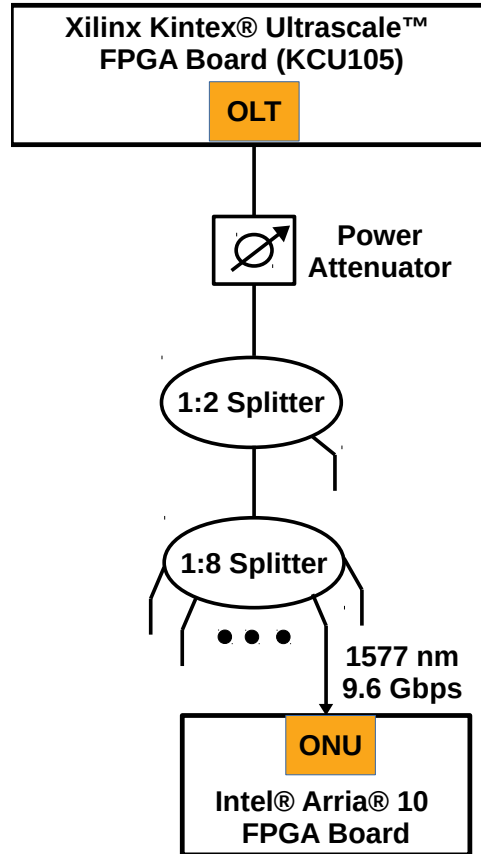


Figure 4.30: The TTC PON signal quality measurement setup.

4.10.4 Transceiver Optimization

A test procedure is developed to tune the high-speed link using the signal conditioning circuitry provided in the Arria[®] 10 transceivers. Quartus v15.1 transceiver testing toolkit [67] is used to monitor the signal characteristics. Several articles are mentioned in the Altera (now Intel) literature the need for proper optimization of transceiver for a maximum performance [68] [67] [69] [70]. All those articles are dedicated to the old generation

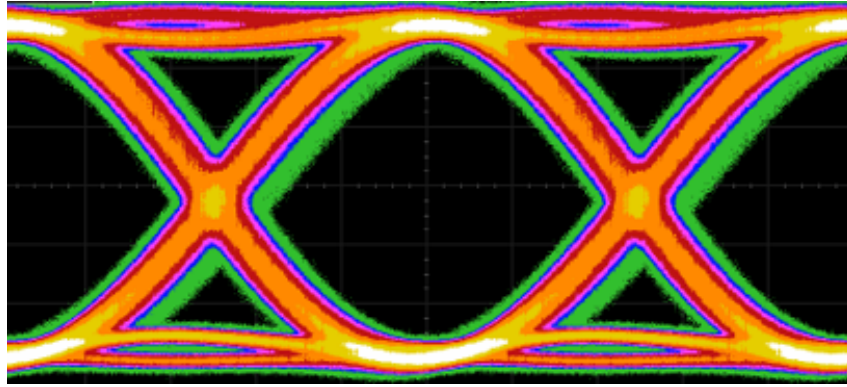


Figure 4.31: TTC PON signal quality analysis in an Agilent Scope by plotting an eye diagram

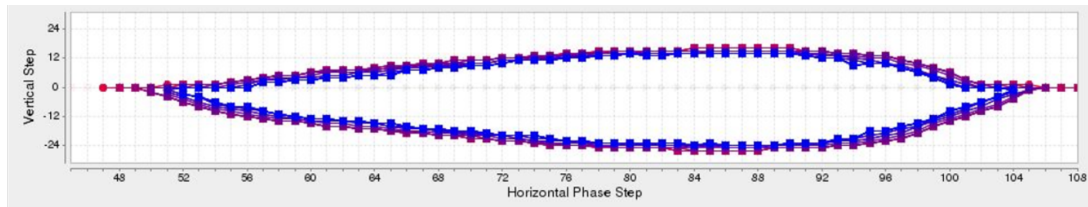


Figure 4.32: TTC PON signal quality monitoring using Transceiver Toolkit (TTK) in serial loopback mode.

FPGAs like Stratix IV, Cyclone and others, hence a study was necessary to have a first-hand result of the effect of the transceiver tuning on the Arria[®] 10 FPGAs [37]. For the transceiver tests, the line rate of the GBT that is 4.8 Gbps is used in an optical loopback mode. The majority of the links used in the CRU are of GBT standard hence the GBT link rate is selected to carry out the test.

The transceiver optimization by testing for all the combinations would be an inefficient approach as the time required would be very long. A short calculation is given to show the exact measurement time needed to scan for all the configurations. Consider the reading time for each test configurations is 10 secs. With the allowed modifiable transceiver properties in Arria[®] 10, the configuration range possible for each parameter is listed in the Table 4.16. The total number of configuration cases comes down to 70 ($= 4 + (3 \times 4) + (6 \times 3) + (9 \times 4)$). The total time needed is of 11.66 mins ($= 70 \times 10 \text{ secs} = 700 \text{ secs}$). The major reason to opt for the quick estimation is to characterize for more

Table 4.16: The default range of configuration parameters

Parameters	Range	No. of cases
V_{OD}	00 – 31	32
Pre-emphasis 1st post-tap	- 31 – 31	63
Pre-emphasis 1st pre-tap	- 31 – 31	63
Pre-emphasis 2nd post-tap	- 15 – 15	31
Pre-emphasis 1st post-tap	- 07 – 07	15
Equalization	00 – 15	16
VGA	00 – 07	07

than 24 links per CRU board and repeat it for over 100 CRU boards in a short period of time where the result of one board cannot be applied to the other board. Hence, this *heuristic approach* is developed.

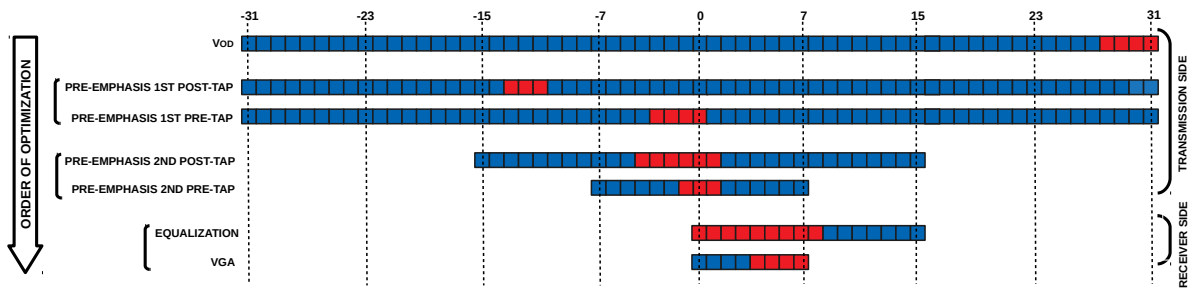


Figure 4.33: Choice of the optimized values from the complete range of configuration parameters

Instead of spending huge computational time to look for an optimal solution, a good enough workaround is to find a potential suboptimal solution. Individual parameter configuration range are scanned by keeping all other parameters fixed at Intel default values, and obtained a range of optimized values determined from the eye-width to the eye-height ratio in the EyeQ signal monitoring tool. The variation of each parameter against the eye height and the eye width using a spider chart is plotted in the Figures 4.34, 4.35, 4.36, 4.37, 4.38, 4.39 and 4.40. Due to a linear nature of the configuration parameters, the chosen optimized values are found in a closed sequence as depicted in the Figure 4.33.

The interesting behaviour about the parameters is that it could be grouped to optimize

Table 4.17: The optimized range of individual configuration parameters

Parameters	Range	No. of cases
V _{OD}	28 – 31	4
Pre-emphasis 1st post-tap	- 13 – - 11	3
Pre-emphasis 1st pre-tap	- 03 – 00	4
Pre-emphasis 2nd post-tap	- 04 – 01	6
Pre-emphasis 1st post-tap	- 01 – 01	3
Equalization	00 – 08	9
VGA	04 – 07	4

separately without having any notable interference effect on the adjacent parameters. Like {VOD}, {Pre-Emphasis 1st Post Tap, Pre-Emphasis 1st Pre Tap}, {Pre-Emphasis 2nd Post Tap, Pre-Emphasis 2nd Pre Tap} and {VGA, EQUALIZATION} can be grouped separately. However, for the fast approximation, the order of optimization is kept same as in the order mentioned. The pictorial representation is illustrated in the Figure 4.33. Users can change the order for further optimisation. By this method, the time taken reduced significantly. The total number of configuration cases comes down to 70 ($= 4 + (3 \times 4) + (6 \times 3) + (9 \times 4)$). The total time needed is of 11.66 mins ($= 70 \times 10 \text{ secs} = 700 \text{ secs}$). The major reason to opt for the quick estimation is to characterize for more than 24 links per CRU board and repeat it for over 100 CRU boards in a short period of time where the result of one board cannot be applied to the other board. Hence, a plausible quick approach is developed.

After the parameters are optimized, the values for the default and the best conditions are shown in the Figure 4.41. For the entire test, PRBS31 is used to stress the system as it shows the maximum bit transitions among the other available patterns. The eye diagram against the default parameter and the optimized parameter are shown in the Figure 4.43. The different parameter configurations of the transceiver can share the same eye diagram values or the performance metrics. The set of those values or the configuration parameter tuples are referred as the *solution space*. The obtained best configuration parameters of the device under test is tabulated in the Table 4.18. The optimized configuration values are

Table 4.18: The optimized value of configuration parameters

Parameters	Optimized Value
V_{OD}	28
Pre-emphasis 1st post-tap	-13
Pre-emphasis 1st pre-tap	-03
Pre-emphasis 2nd post-tap	-04
Pre-emphasis 1st post-tap	01
Equalization	00
VGA	05

highly sensitive and depend on the FPGA process technology, the system temperature and the optical transceiver used. Hence, even for a minor hardware modification, the configuration parameter values need to be re-evaluated.

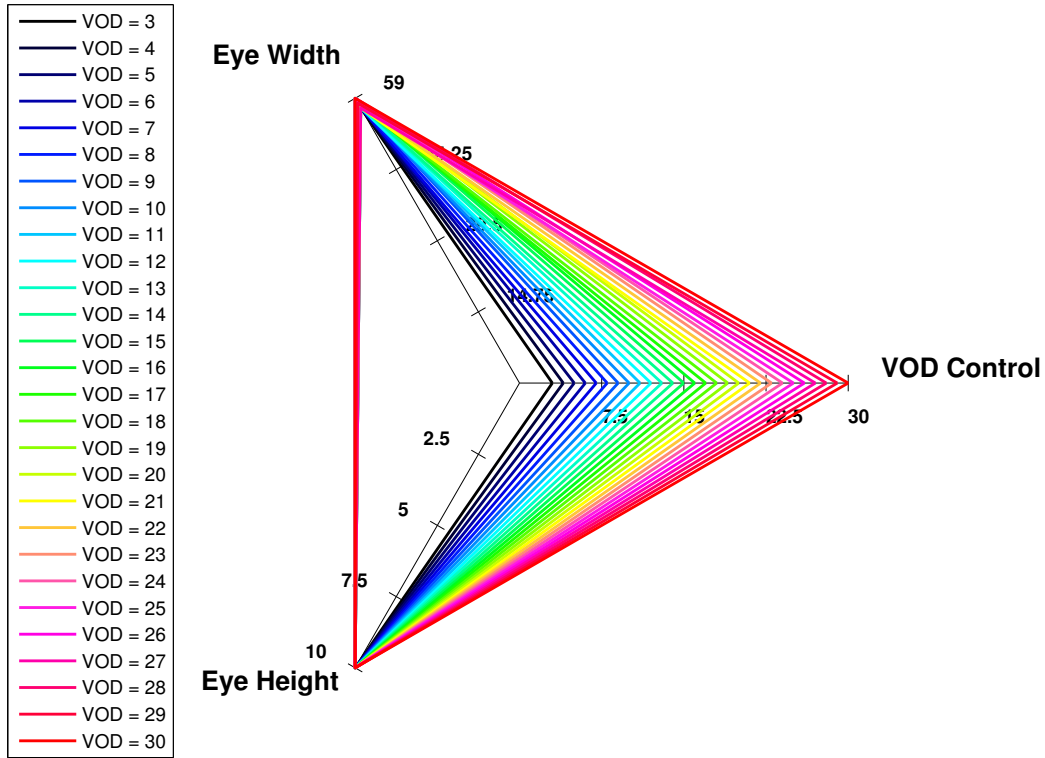


Figure 4.34: Transceiver parameter tuning plot with the VOD variation

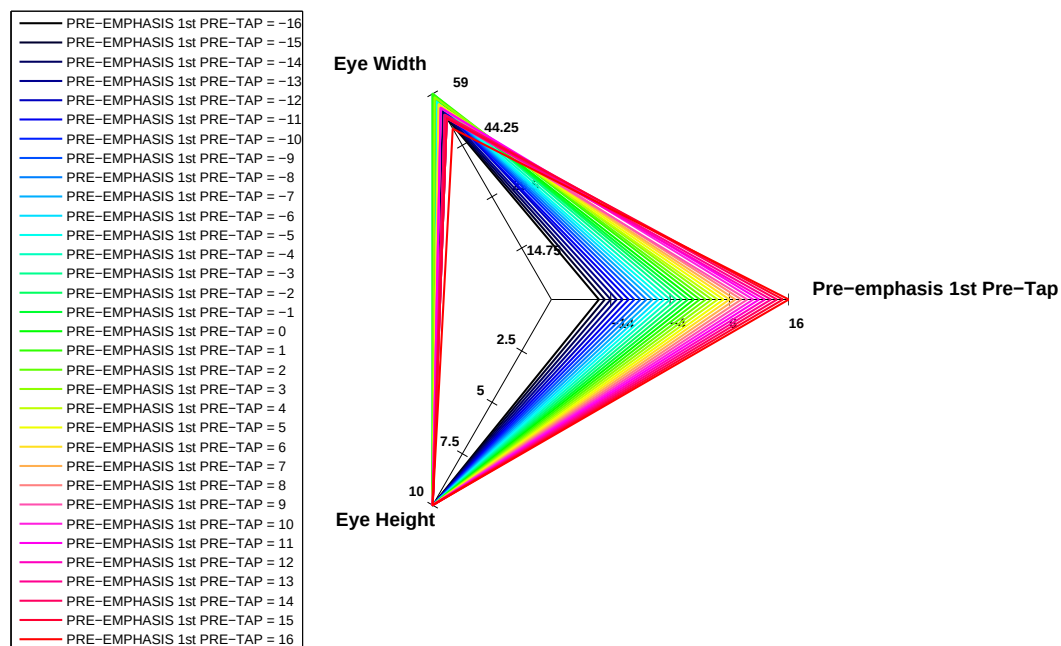


Figure 4.35: Transceiver parameter tuning plot with the PRE-EMPHASIS 1st PRE TAP variation

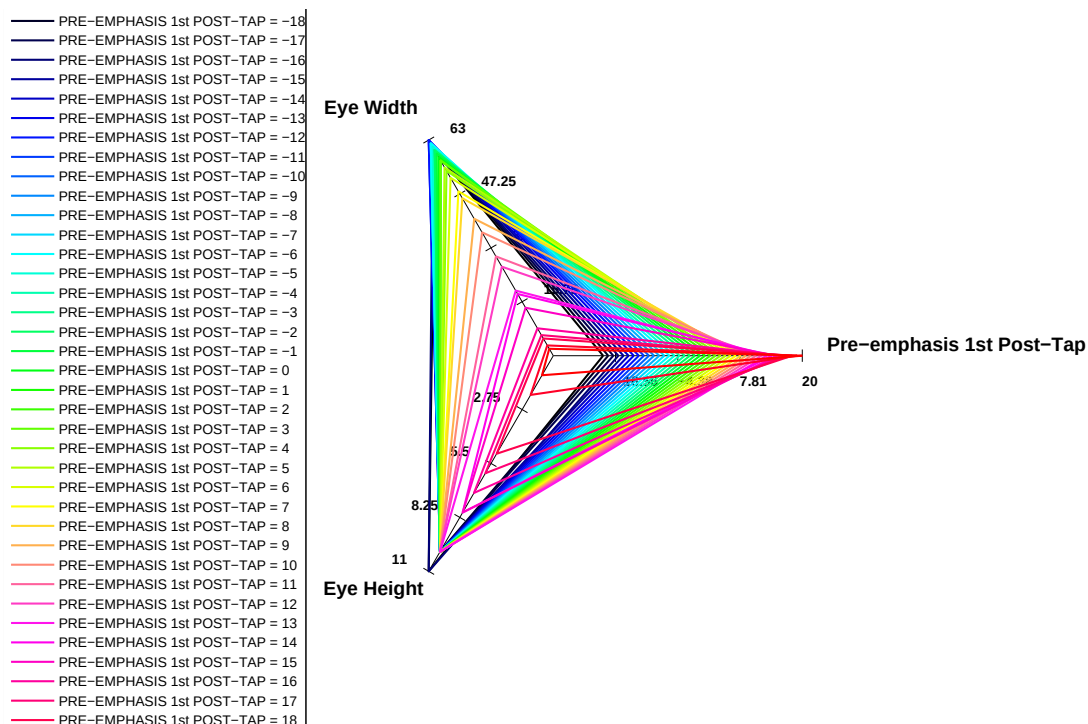


Figure 4.36: Transceiver parameter tuning plot with the PRE-EMPHASIS 1st POST TAP variation

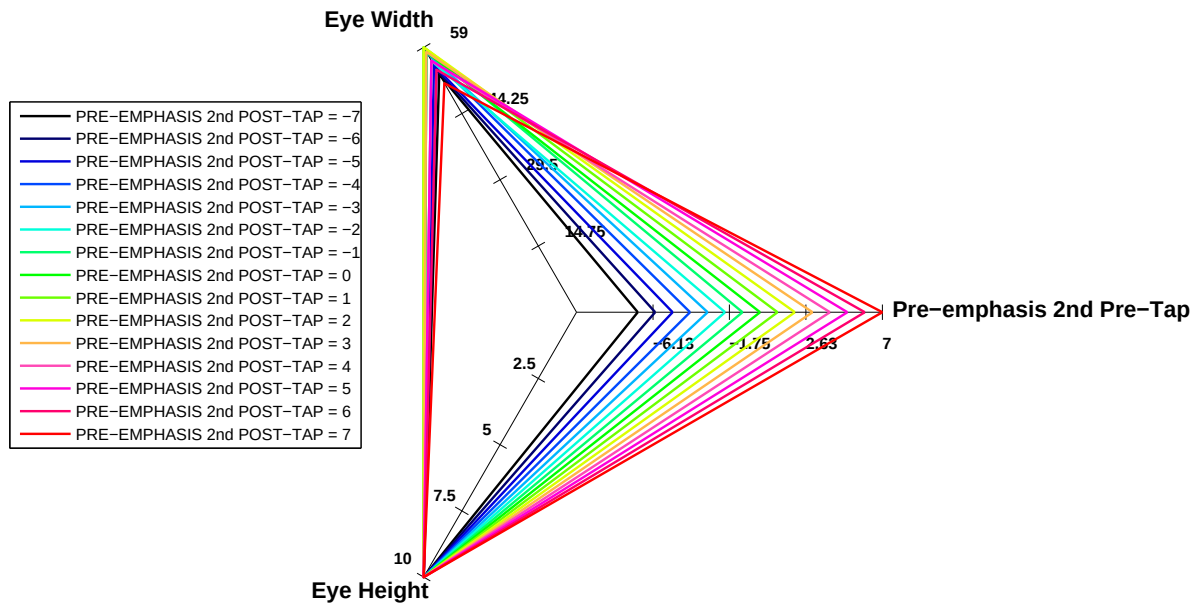


Figure 4.37: Transceiver parameter tuning plot with the PRE-EMPHASIS 2nd PRE TAP variation

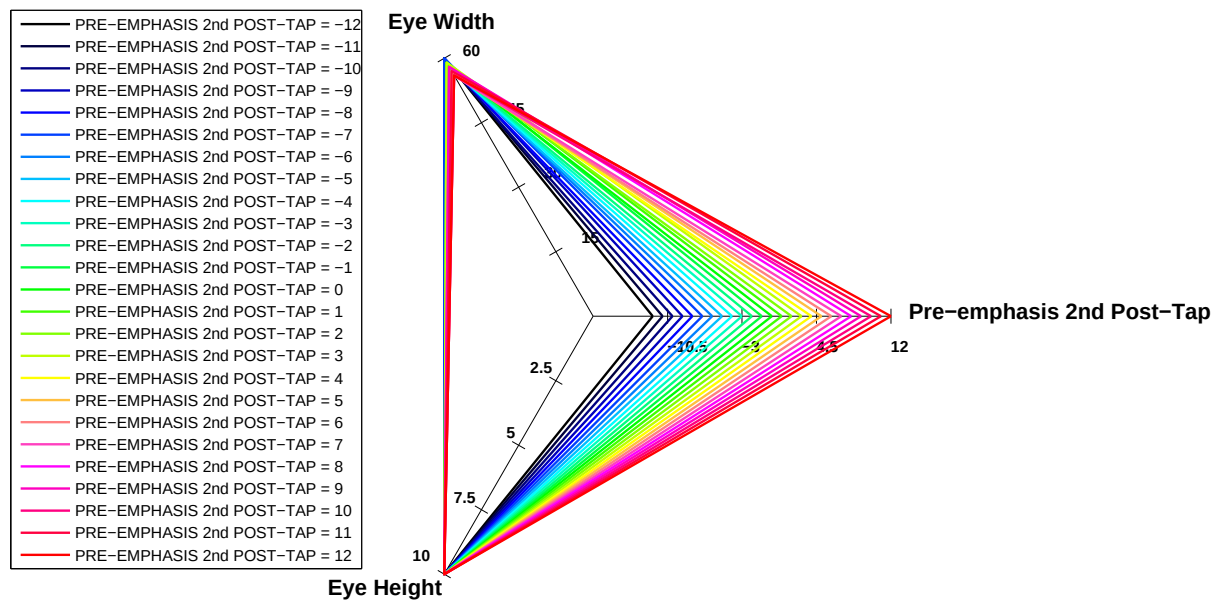


Figure 4.38: Transceiver parameter tuning plot with the PRE-EMPHASIS 2nd POST TAP variation

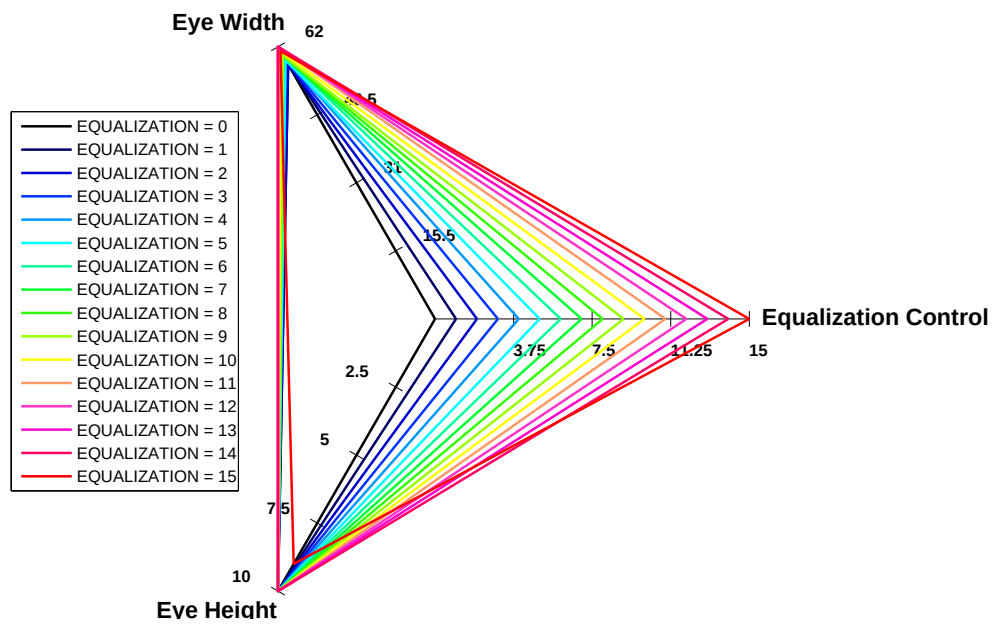


Figure 4.39: Transceiver parameter tuning plot with the EQUALIZATION variation

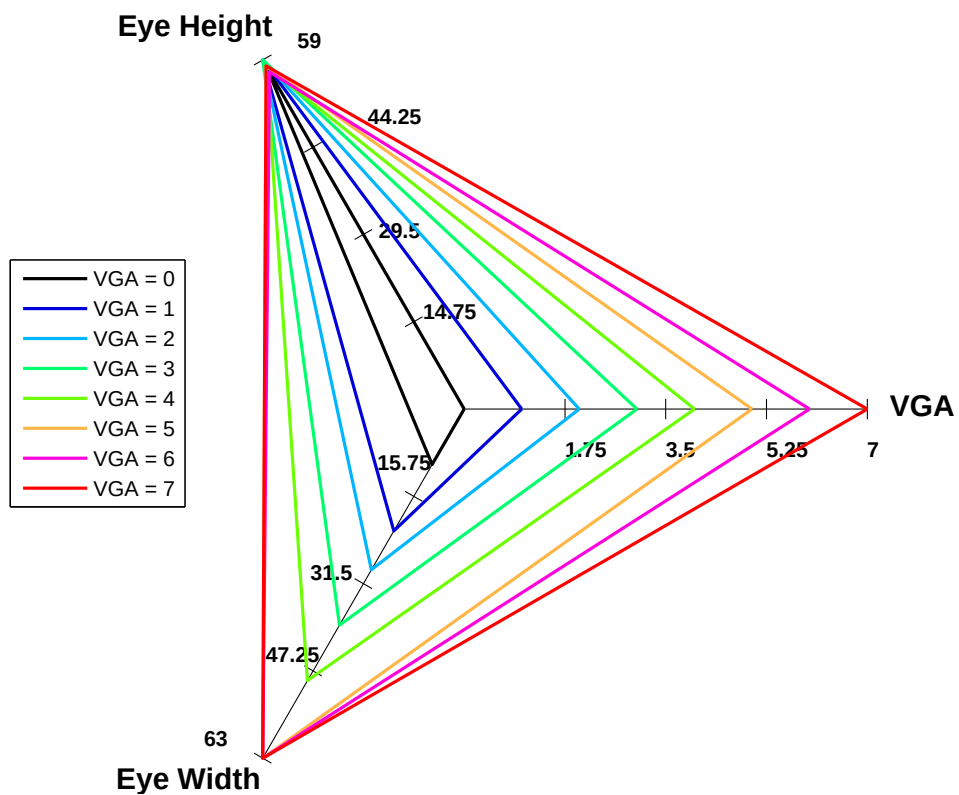


Figure 4.40: Transceiver parameter tuning plot with the VGA variation

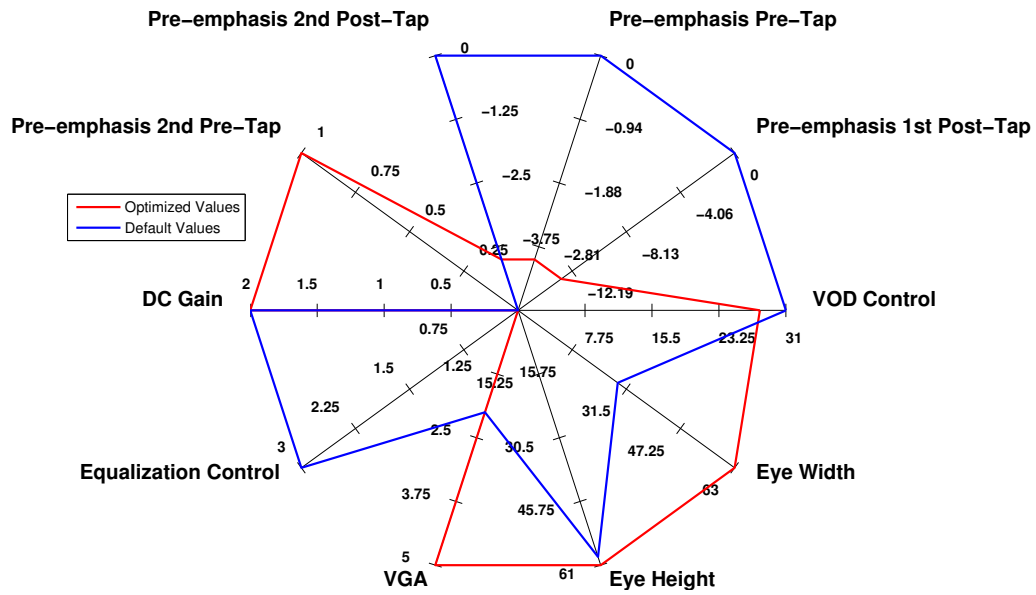


Figure 4.41: Transceiver parameter tuned showing values for the default settings vs the best settings

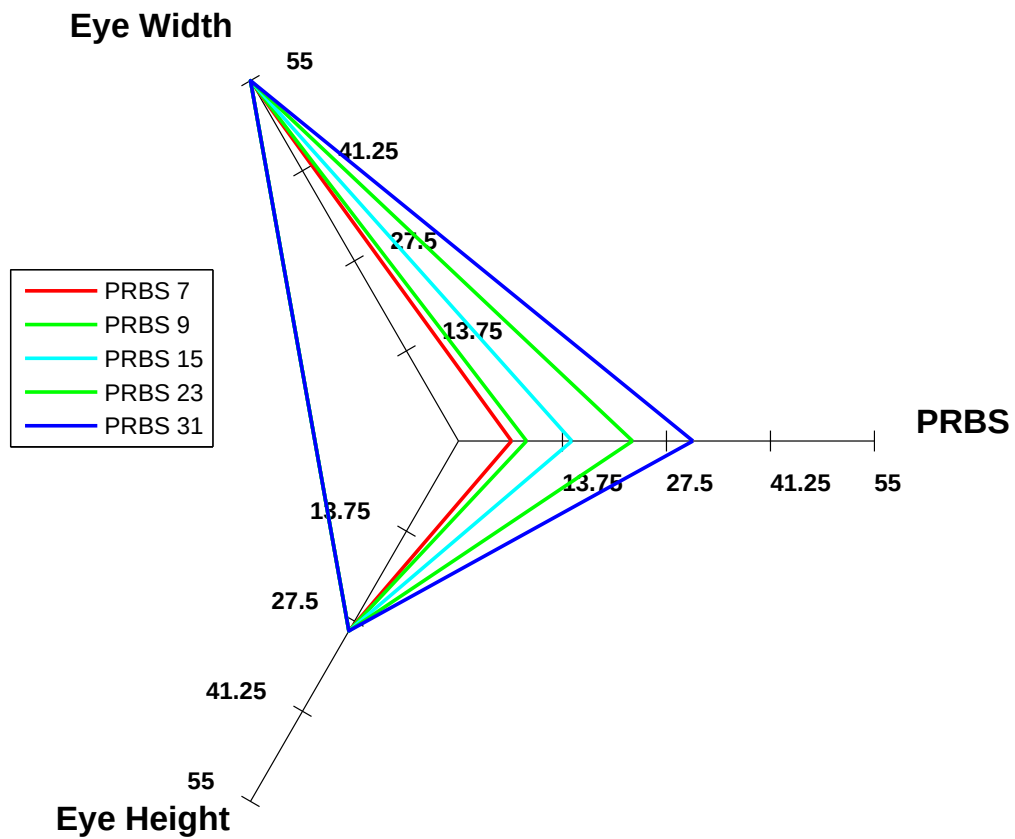


Figure 4.42: Transceiver parameter performance with the PRBS variation

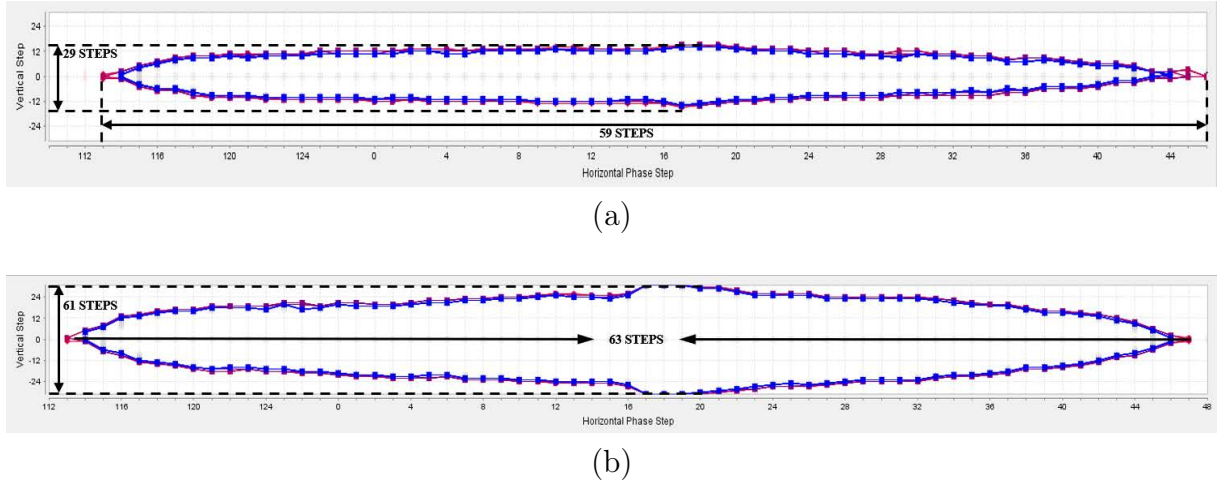


Figure 4.43: Eye Diagram (a) Default parameter settings; (b) Optimized parameter settings

4.11 Discussion

The behaviour of all the composite elements in the TTC-PON and GBT bridge connection is analyzed for compatibility regarding interconnection and interoperability. A detailed characterization test on the integrated prototype design is conducted to check for any unanticipated design faults before the final commissioning in CRU firmware. Four performance measuring metrics are used in the characterization: (1) Latency; (2) Jitter; (3) BER and (4) Transceiver parameter settings. The testing phase of the firmware has passed through several iterations of power on/off cycle, firmware upgrade and reset assertion/de-assertion cycle (PFR cycle) for the reliability test of the communication bridge. During the entire study, large sets of empirical results are collected for analysis. The analysis of the statistics confirms that the TTC communication bridge connection with its present design configuration showed complete deterministic behaviour with no points of uncertainty over the several rounds of PFR cycle.

The results meet the specified trigger and timing communication standards hence no further compensatory measures are needed. To avoid any unprecedented failure during the data taking; a set of monitoring logic is integrated along with the CRU firmware logic

core to register the macroscopic behaviour of the system, as a preventive measure, as mentioned in Section 4.9. The intrinsic details related to the CRU firmware designs are available at the ALICE-CERN CRU TWiki page [71].

4.12 Conclusion

I have carried out a detailed study of the trigger and timing distributions using the TTC-PON and GBT bridge connection in ALICE. The study is carried out using the CRU development boards for rapid dissemination of performance metrics. The results show that the TTC-PON and GBT can work in synergy to communicate successfully the timing and trigger information and can effectively be deployed. The study confirmed that the system behaviour is completely deterministic with multiple rounds of PFR cycle. The FPGA used in the CRU board is 20 *nm* Intel[®] Arria[®] 10 . The CRU firmware logic uses static placement configuration, hence the stress points remain fixed over the operation runtime. Future scope is to do a reliability study by accelerated stress scenario to mimic the effect of degradation in the timing circuits and wearability in the logic/memory cells [72]. These would identify the stress hotspots and allow us to overcome the system faults by applying the mitigation solutions accordingly. Another study that is equally important is to have a data flow analysis of the spatiotemporal behaviour of the data traffic [73] for each sub-detector system to arrange and reallocate the CRU peripheral logic resources in an optimized manner.

Chapter 5

Behaviour of the Jitter Cleaning PLL with Temp. Variation

5.1 Introduction

The experiments at the CERN LHC compose of multiple detector systems, each with a large number of sub-detectors. Each of the sub-detectors receives the Bunch Clock (BC) signal derived from the RF frequency as a synchronization reference. The Timing-Trigger and Control (TTC) system [42] is dedicated for synchronizing the electronics of the experiments to the LHC beam. The upgraded TTC system uses Passive Optical Network (PON) technology [74] and CERN GigaBit Transceiver (GBT) [17] optical solutions to broadcast TTC information. FPGA based readout boards like CRU are used in the back-end system of the LHC experiment for TTC communication and lead the designers to opt for a programmable PLL solution that can be mounted on the readout card itself. The design parameters are hard to change once mounted on the PCIe 40 PCB, the CRU card. Hence, proper identification of LHC timing related requirement is necessary before choosing the required PLL.

During serial link transition or inter/intra channel interference unwanted jitter gets

introduced in the TTC communication chain. The first requirement is that the chosen PLL have the ability to substantially clean the recovered clock and reduce cycle-to-cycle jitter, TIE jitter and intra clock skew jitter [42]. It is necessary to maintain a constant phase of the recovered clock during frequency drift in LHC beam dump and filling operation. LHC master or Bunch Clock (BC) is of 40.0786 MHz frequency. The official values for LHC BC frequency variation are 87 Hz and 548 Hz that occurs in a period of ≈ 10 mins, during the proton and ion beam energy ramp respectively. Hence, preventing the loss of lock in PLL during frequency ramp is the second requirement. The third requirement is to ensure no sudden jumps of phase occurs in the recovered clock in between electronic equipment power cycles. The fourth requirement is to ensure the drift in PLL jitter lies within tolerable margin during temperature fluctuation, to prevent significant drift in the phase that can bring the system out of the locking range.

PLLs are widely used throughout the clock distribution tree in LHC to filter out jitter while tracking the slow variations of the clock signal. For clocking multiple FPGA transceiver banks require a PLL that provides multi-output jitter attenuating clocks. Need for hitless switching feature also arises such that it can absorb the phase difference between two input clocks during a switchover and minimizes the risk that downstream low-bandwidth PLLs can suffer from loss of lock due to a clock rearrangement. Further holdover reference clock function is required, to ensure the transceivers maintain proper operation in the event the reference clock is not available. A wide range of PLL technologies are available. A detailed characterization of PLL is needed to select the best PLL based oscillator for the timing application. Si5344 PLL from Silicon Labs is chosen as one of the candidate PLL for testing in the CERN Electronics lab.

The Si5345/44/42 PLL [75] chip comes with fourth-generation DSPLL and Multi-Synth[™] technologies combined with jitter attenuating clock multipliers that enable synthesis of any high-frequency clock that requires the highest levels of jitter performance. Si5345/44/42 product family supports 4 number of inputs, with fan-out of 10/4/2 respec-

tively. In the test procedure Si5344 evaluation board is used for characterizing the PLL. The main objective of the tests is to validate that the PLL meets all the stated LHC requirements.

5.2 Measurement Setup

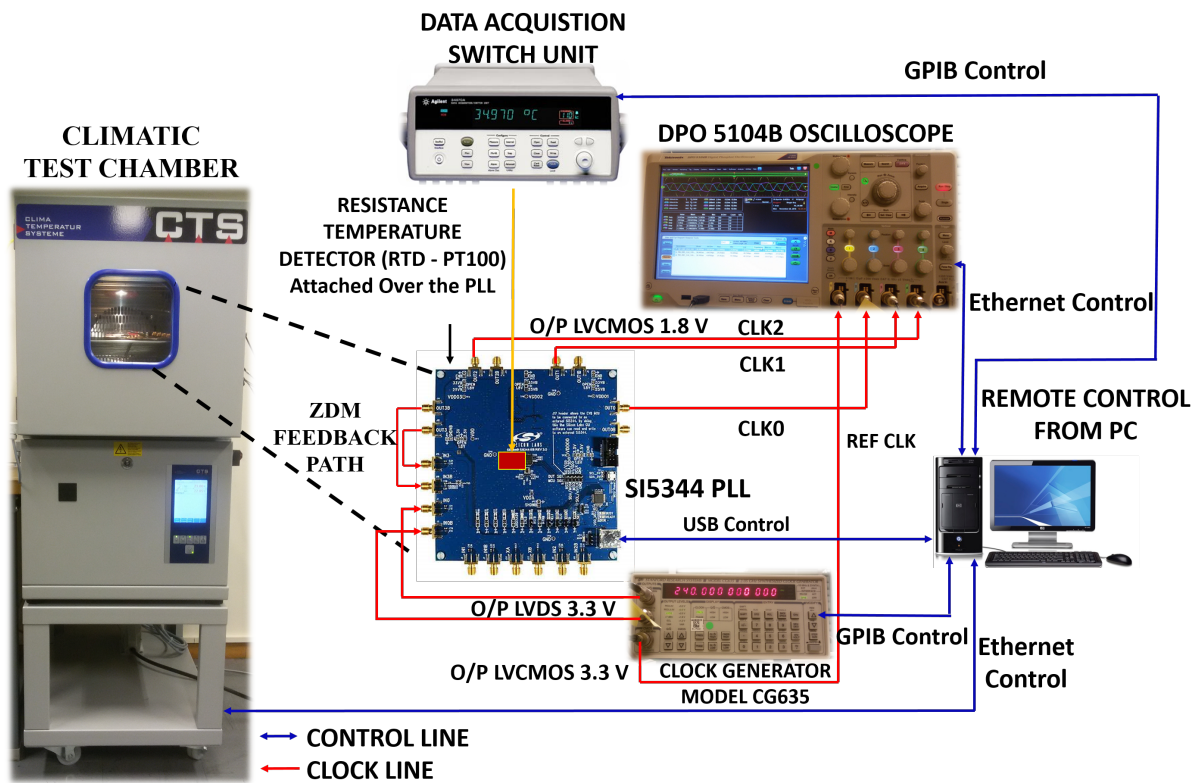


Figure 5.1: Test setup of the Si5344 PLL.

A fully automated test setup is built to characterize the PLL. The key element of the setup is the CTS (Climatic Test Chambers) that is used to apply controlled variation of temperature on the Si5344 PLL. For precise measurement, PT100 RTD (Resistance Temperature Detectors) is used to monitor the surface temperature of the Si5344 PLL chip. The sensor is mounted to the surface using an epoxy adhesive. The phase of the output clocks has been tracked with respect to the clean reference clock provided by a CG635 generator. Si5344 EVB is connected to PC using USB and programmed over I2C

protocol using ClockBuilder Pro software. It is configured at loop bandwidth of 200 Hz with output driver set at LVCMOS mode. The output signals are applied to an oscilloscope and jitter measurements are recorded using DPOJET software tools. Fig. 5.1 shows the arrangement of test setup. Reset cycle test is conducted at an ambient temperature of 23°C and frequency drift test against temperature ramping rate of $1.9^{\circ}\text{C}/\text{min}$ over 12-14 thermal cycles. Temperature range of 10 to 60 deg C is chosen as two extremes of the thermal cycle to operate. To maintain statistical integrity, 10000 readings are taken for a particular PLL combination using digital oscilloscope DPO-5104B working in single shot capture mode with statistical population set in the range of thousands.

5.3 Results

The results are recorded for four PLL configurations as needed by different experimental groups at CERN, namely 40 MHz (i/p) – 40 MHz (o/p), 40 MHz (i/p) – 120 MHz (o/p), 40 MHz (i/p) – 240 MHz (o/p) and 240 MHz (i/p) – 240 MHz (o/p). The results are further sub-divided into two groups based on modes of operation, namely ZDM (Zero Delay Mode) and ‘No ZDM’. A zero-delay mode is used in applications that needs a constant minimum delay between the required input and outputs. ZDM helps to cancel out the internal delay introduced by PLL, by feeding back any one of the outputs back to the FB_IN pins. For both the modes the PLL is operated in ‘HOLDOVER OFF’ mode. Upon power up or reset the PLL will temporarily go into holdover and then when it comes out of holdover it does an internal phase adjustment which will shift the output phase. To get fixed IO delay HOLDOVER is turned off by setting proper configuration registers in the settings file. Possible side effect is that, if the input signal is lost, the output clock would drift in frequency. The rev. B. of this PLL (the one used for the test) does not use fast lock when holdover is disabled. In the test system the reading for skew jitter and Time Interval Error (TIE) is noted, as these two parameters play a dominant role in LHC

experiment [42]. The usage of the parameters in the context of the experiment is defined in the following paragraphs.

PLL Tracking Skew Jitter [76] [42] The phase difference between the reference clock and the PLL clock of a output pin is denoted by $\Delta\theta = (\theta_{REF_CLK} - \theta_{CLK0})$. Reading of phase difference value at average room temperature for 23 °C in both heating and cooling temperature cycle is represented by $\Delta\theta^{23+}$ and $\Delta\theta^{23-}$ respectively. The mean value is calculated by taking the average of the two readings obtained, $\Delta\bar{\theta}^{23} = Avg\{\theta^{23+}, \Delta\theta^{23-}\}$.

Time Interval Error (TIE) [42] Time Interval Error (TIE) is the difference in time between the actual threshold crossing and the expected transition point (or derived clock edge). TIE is given by the discrete time domain representation of phase noise in the units of pico-seconds. In the experiment scope settings having 1 shot trigger is used with statistical population >10000, to acquire TIE measurement.

The readings for hard reset cycle and temperature drifting of the PLL in 8 configuration modes are summarized in Table 5.1. In Figures 5.3, 5.5, 5.7 and 5.6, the performance of one of the configuration mode of the PLL is illustrated, where in it is set at ‘No ZDM’+‘HOLDOVER OFF’ mode having input and output clock frequency set at 240 MHz. The results for all the configurations exhibits the phase drift occurs linearly with temperature variation having no abrupt phase jump. As can be inferred from Table 5.1, typically deviation lies within 102-86 ps for No ZDM mode and 51-37 ps for ZDM mode, with TIE variation in the range of 0.59-0.69 ps. No phase inconsistency has been detected during 10,000 repetitive reset cycles conducted over all combinations. However, a gaussian behavior of phase skew jitter with a span of 17 to 20 ps is noted having std. dev. of 2.7-2.9 ps. A frequency drift of ± 500 Hz centering over frequency of 40 MHz is applied, having rate drift of 1 Hz per 10 secs to check if a loss of lock occurs in the PLL. The frequency drift test is repeated 10 times to conclude that the PLL locking range is suitable to withstand

Table 5.1: Skew & TIE measurement for various PLL configuration mode.

PLL MODE	NO ZDM	ZDM	NO ZDM	ZDM	NO ZDM	ZDM	NO ZDM	ZDM
I/P Clk (MHz)	40	40	40	40	40	40	240	240
O/P Clk (MHz)	40	40	120	120	240	240	240	240
Temp. Cycle	14	14	14	14	14	14	12	12
Temp. Ramping Rate (°C/min)	1.9	1.9	1.9	1.9	1.9	1.9	1.9	1.9
Skew Variation (ps)	101.81	51.4	92.31	42.81	86.22	37.26	88.45	47.75
TIE Variation (ps)	0.684	0.624	0.610	0.594	0.690	0.624	0.601	0.637
10000 Reset Cycle keeping constant temperature of 23 °C								
Skew Variation (ps)	19.26	16.11	17.13	17.28	18.19	20.11	20.34	20.64
Std. Dev. of Skew (ps)	3.168	2.718	2.712	2.770	2.764	2.705	2.727	2.933

No of readings taken per configuration = 10000, Statistical population used for Skew and TIE measurement are ~ 2000 and ~ 20000 respectively.

the LHC frequency drift requirement.

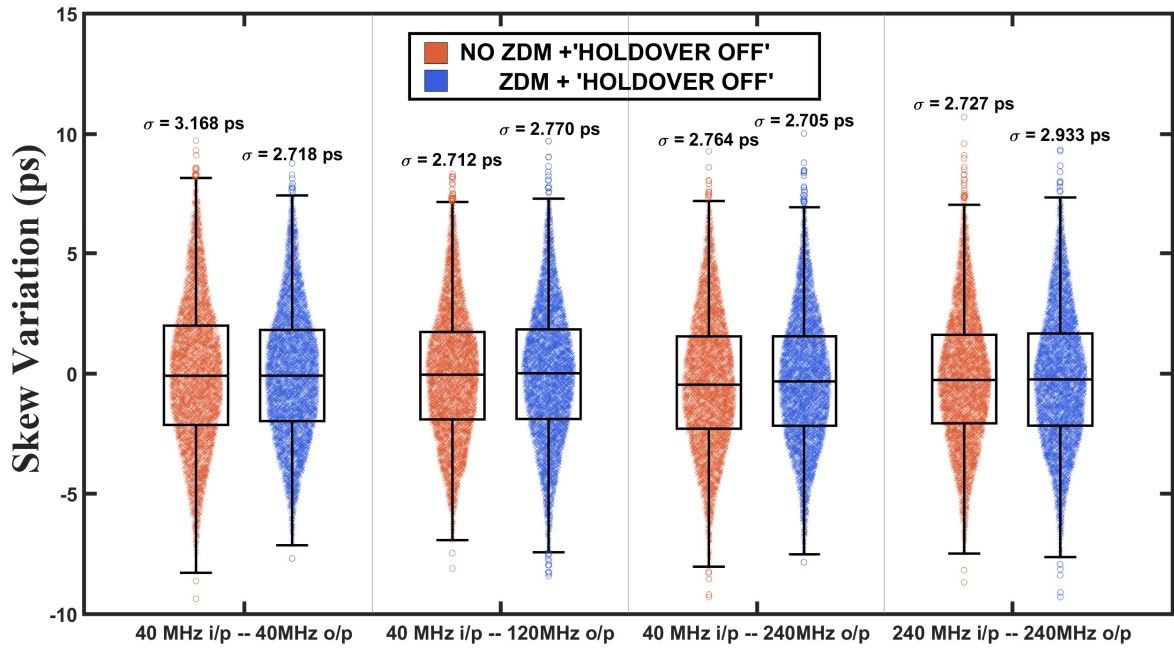


Figure 5.2: Boxplot of skew distribution over 10000 reset cycle at $23^{\circ}C$ for different input-output frequency combinations

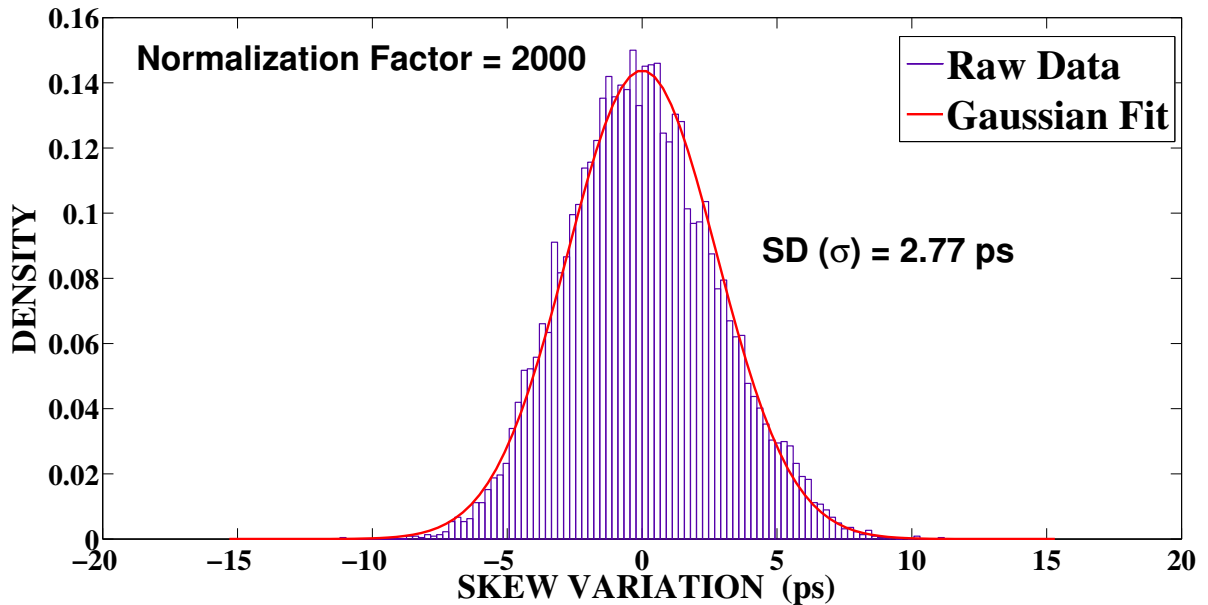


Figure 5.3: Skew Distribution over 10000 repetitive reset cycle at $23^{\circ}C$ for PLL configuration 240 MHz i/p and 240 Mhz o/p

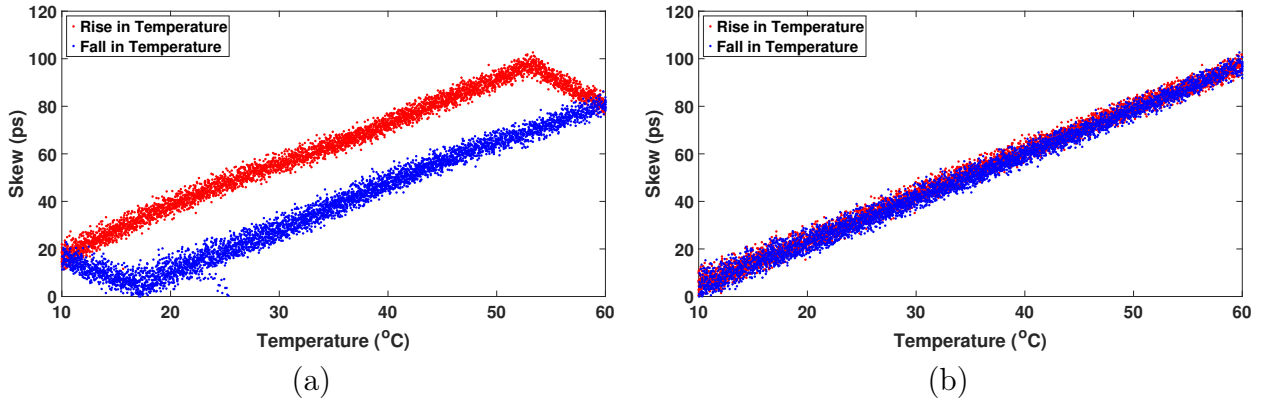


Figure 5.4: The effect of temperature monitoring on the registered data points: (a) Hysteresis in reading data when the chamber temperature is only monitored; (b) Without any hysteresis in reading data when surface temperature of the PLL chip is also monitored

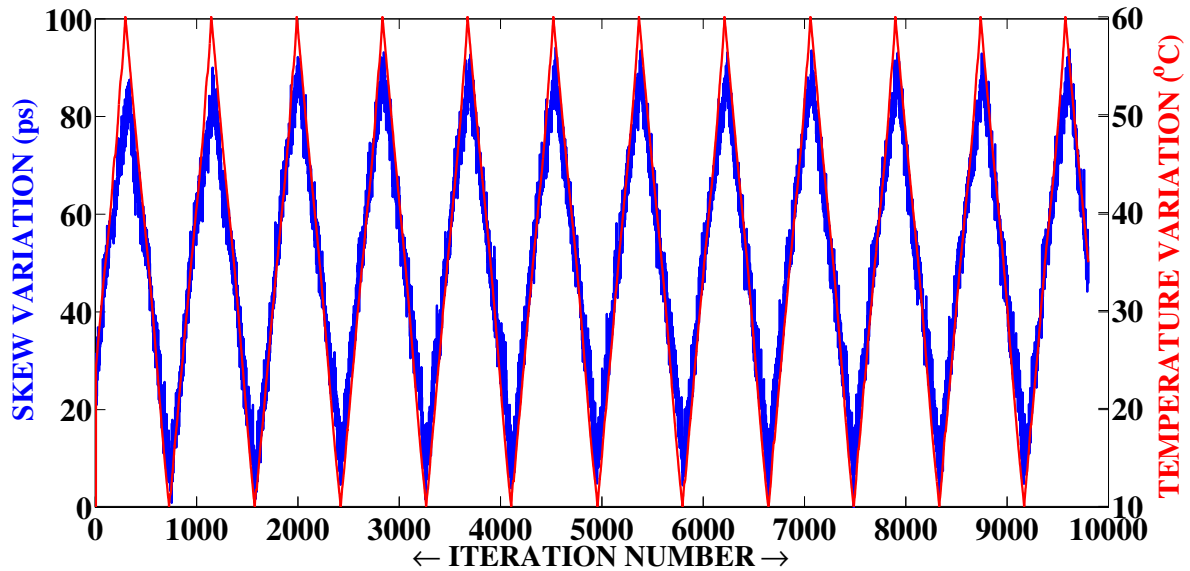


Figure 5.5: Skew Variation over 12 Temperature Cycles (10000 readings / print) for PLL configuration 240 MHz i/p and 240 Mhz o/p frequency.

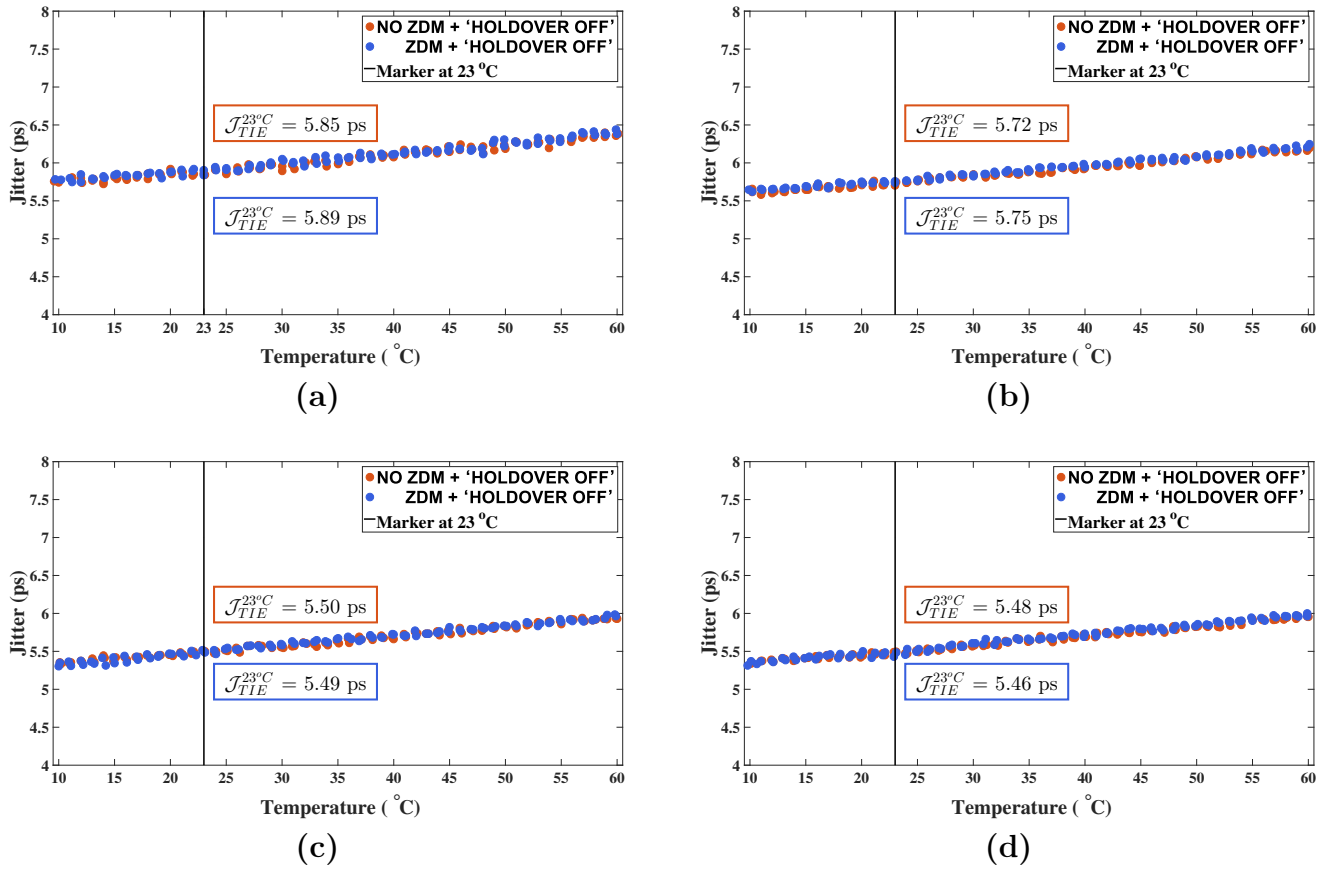


Figure 5.6: Impact of thermal gradient on TIE clock jitter for different *input-output* frequency combinations : (a) 40 MHz i/p – 40 MHz o/p; (b) 40 MHz i/p – 120 MHz o/p; (c) 40 MHz i/p – 240 MHz o/p; (d) 240 MHz i/p – 240 MHz o/p.

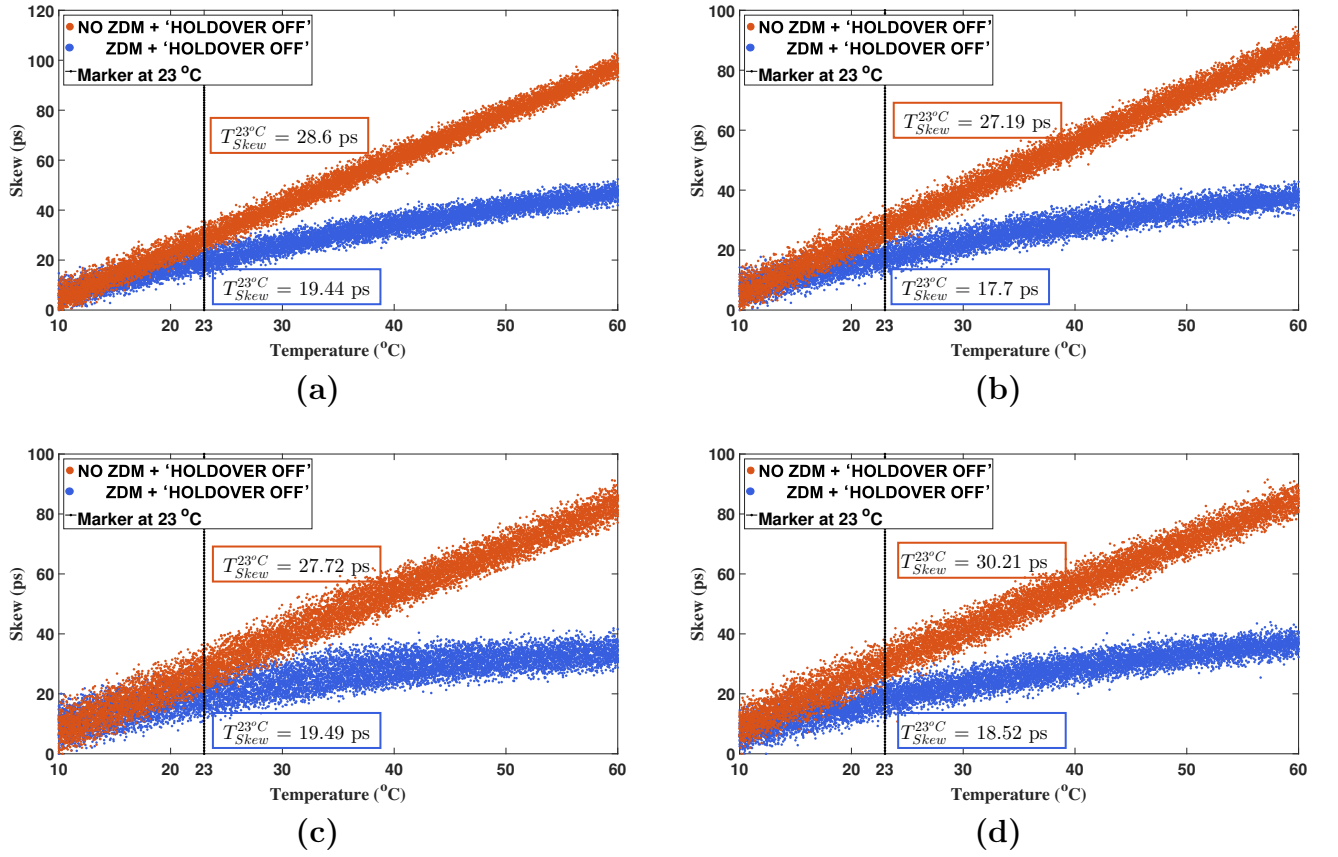


Figure 5.7: Impact of thermal gradient on clock skew for different *input-output* frequency combinations : (a) 40 MHz i/p – 40 MHz o/p; (b) 40 MHz i/p – 120 MHz o/p; (c) 40 MHz i/p – 240 MHz o/p; (d) 240 MHz i/p – 240 MHz o/p.

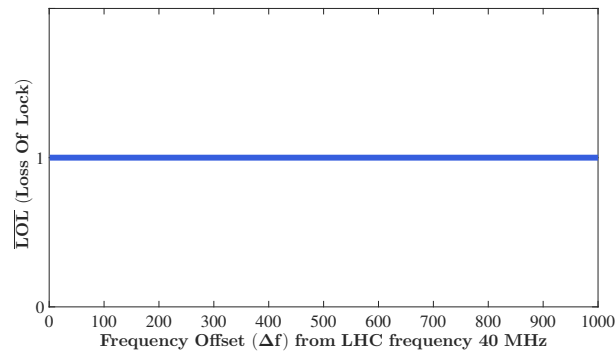


Figure 5.8: PLL remains lock for different Offset in Frequency applied with respect to the LHC frequency of 40 MHz

5.4 Conclusion

The results indicate that the chosen Si5344/45 PLL satisfies major LHC TTC quality standards. However, the PLL is unable to provide the solution to the holdover mode with the chosen configuration. This feature is required when the reference clock is missing. The discussion in the thesis is limited to the clock related jitter measurement with temperature variation. The jitter cleaning ability of SI5344 PLL is not discussed. Selected PLL is found to be suitable after several benchmark tests conducted by central CERN electronics team against other PLL candidates. The version of the PLL chip has found application in both ALICE and LHCb first production version of the readout cards for the LHC RUN3 upgrade.

Chapter 6

Pico-second Resolution Phase Measurement System

6.1 Introduction

Experiments use phase information to calibrate and synchronize signals between different circuit elements. In certain experiments like in High Energy Physics (HEP), preservation of phase relationship between critical signals throughout the experiment runtime is a necessary condition. In recent trend for compact implementation of full digital architectures, reconfigurable hardware technologies like FPGAs play a very dominant role. Popular latency critical communication link standards used in HEP experiments like Giga-Bit Transceiver optical link (GBT) and Timing, Trigger and Control system based on Passive Optical Networks (TTC-PON) technology [15, 17], are implemented in FPGAs directly. These links carry trigger and timing information needed for time-stamp generation and event building. It is essential that the latency critical protocols maintain constant phase differences in the recovered signals for the entire experiment's run-time. High speed serial transceivers of FPGA's do not maintain constant phase shift with each round of power cycle, reset cycle, loss of lock in the transceiver [77], firmware upgrade or aging of clock

circuitry in Phase Locked Loop (PLL). *A detailed description of this erratic behaviour of FPGA circuitry related to GBT is mentioned in Section 3.7.1 and re-discussed in Section 4.8.* A logic design for phase monitoring capability to register phase shift changes in the range of 20-100 picoseconds is needed inside the FPGA circuitry. This would allow to extract the relative phase information and re-calibrate the system when needed, to maintain the constant phase relationships.

Several approaches for phase measurement have been discussed in the literature. The classical principle of using the over-sampling technique is inadequate to measure a relative phase difference between the two high frequency clocks inside an FPGA fabric, whose frequency exceeds the maximum limit supported by the fabric (< 500 MHz). One of the solutions as mentioned in the works of Lu *et. al* [78], Trebbels *et. al* [79] and Rong *et. al* [80], is to sample it externally using an analogue to digital converter (ADC) and then feed it back to the FPGA for computation. However, the technique needs an additional hardware to measure the phase difference of the internal digital clocks. Without the use of additional hardware, a phase measurement approach had been proposed by Hidvegi *et. al* [81] by using the dynamic phase alignment feature of the FPGA PLLs. The drawback of the method is the achieved resolution, which is limited to the 1/8th of the voltage controlled oscillator (VCO) frequency.

In this chapter a new approach is proposed for an accurate phase measurement in an FPGA using sub-samples collected by the systematic sampling over XOR based Phase Detector (PD) signal. The XOR based PD introduces the least timing jitter because of the simplicity of its design [82]. It is known that use of sub-sampling technique causes spectral leakage. However, in the application, the use of averaging method for the phase measurement makes it less susceptible to interfere with the results [83]. A mathematical model is formulated to explain the operating principle of the phase measurement technique. The proposed technique is suitable to construct a logic core for relative phase measurement between clocks having very low to very high frequencies with precision,

accuracy and resolution in the range of few picoseconds.

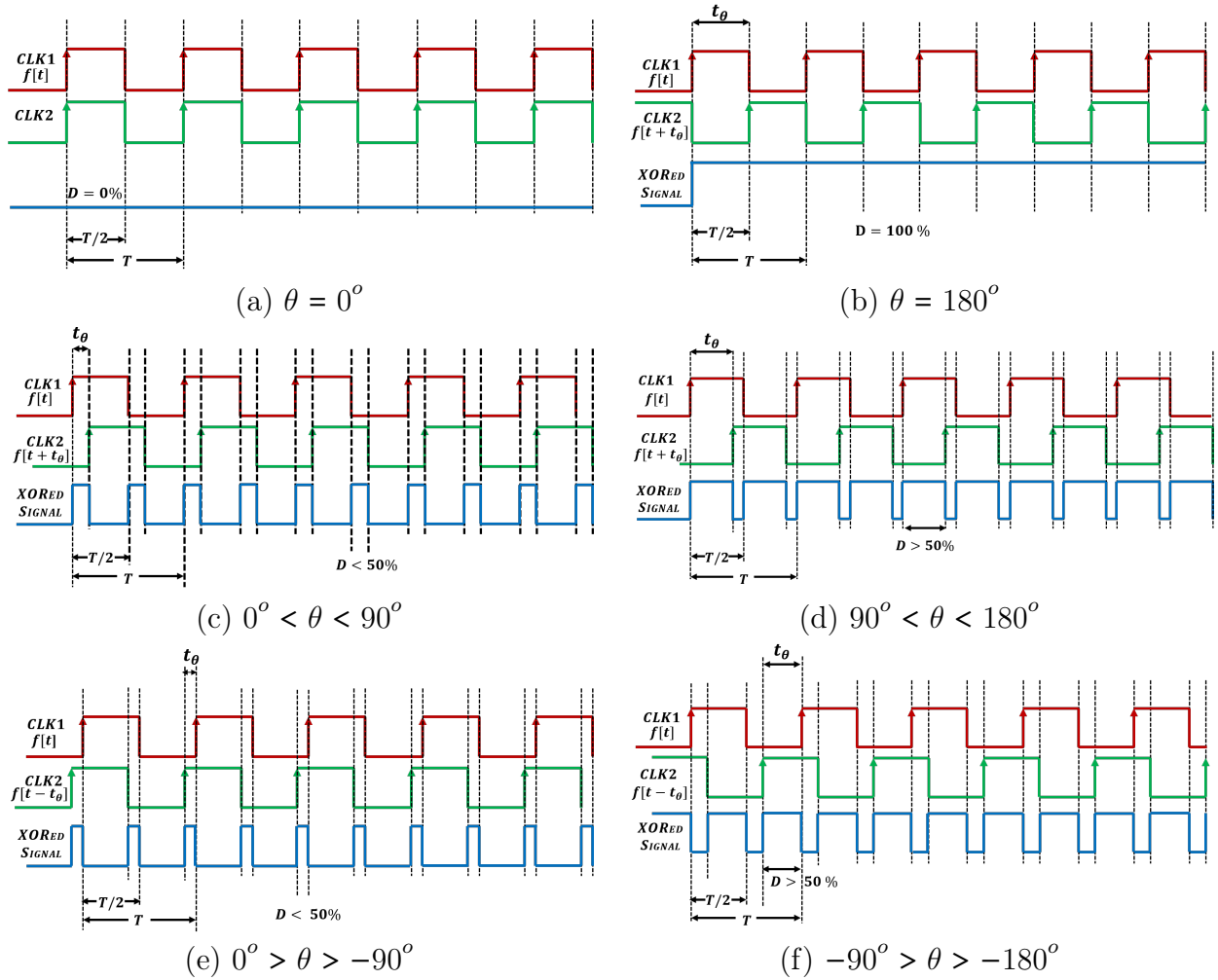


Figure 6.1: The phase difference in the duty cycle of XORed waveforms is illustrated. Leading Phase Duty Cycle ($D_{\theta 1}$) and Lagging Phase Duty Cycle ($D_{\theta 2}$) are shown in (c-d) and (e-f) panels, respectively.

The chapter is organised as follows. Section 6.2 discusses the principal components of the architecture, which include the phase detector, duty cycle computation and the phase polarity detector. Sections 6.3 and 6.4 present the details of sub-sampling methodology. The procedure for phase value computation is outlined in section 6.5. In section 6.6 the instrumentation errors are described. This is followed by the calibration procedure in section 6.7. The results are presented and discussed in section 6.8. Comparison to other relevant work is also given in this section. Finally the chapter is concluded with a

discussion and an outlook.

6.2 Design Implementation

The digital clock signals are defined by the continuous-time Heaviside step function [84], as denoted by $u(t)$. The digital clock (CLK1) having square waveform with periodicity T and pulse width DT centered at nT , is represented by Eq. (6.1).

$$x[t] = \sum_{n=-\infty}^{+\infty} \left(u \left[t - nT + \frac{DT}{2} \right] - u \left[t - nT - \frac{DT}{2} \right] \right), \quad (6.1)$$

where Duty Cycle, $D = 1/2$ or 50% and Time Period, $T = \frac{1}{f_0}$.

Similarly, the digital clock (CLK2) having same frequency (f_0) and shifted by t_θ phase is given by $x[t + t_\theta]$. The phase difference between CLK1 and CLK2 is calculated by the XOR based PD, shown in Figure 6.1. The resulting XORed waveform $y[t]$ can be expressed by the following equation:

$$\begin{aligned} y[t] &= x[t] \oplus x[t + t_\theta] \\ &= \sum_{n=-\infty}^{+\infty} \left(u \left[t - \frac{nT}{2} + \frac{D_{\theta 1}T}{4} \right] - u \left[t - \frac{nT}{2} - \frac{D_{\theta 2}T}{4} \right] \right), \\ \text{where } \begin{cases} D_{\theta 1} = 0, D_{\theta 2} = \frac{t_\theta}{T/2} & : t_\theta < \frac{T}{2} \\ D_{\theta 1} = \frac{t_\theta}{T/2}, D_{\theta 2} = 0 & : t_\theta \geq \frac{T}{2} \end{cases} \quad (6.2) \\ \text{and } D_\theta &= D_{\theta 1} + D_{\theta 2}. \end{aligned}$$

This is a pulse train of duty cycle, D_θ , composed of two independent variables $D_{\theta 1}$ (shown in Fig. 6.1 (c-d)) and $D_{\theta 2}$ (shown in Fig. 6.1 (e-f)), referred as the leading and the lagging phase duty cycles, respectively. The full duty cycle D_θ is given by the Eq. (6.2), as the ratio of the phase difference (t_θ) to the half time-period ($T/2$).

Continuous-time signals, $y[t]$ and $x[t]$, are uniformly sampled by the sampling clock of time period T_s to obtain the discrete-time signals, $y[kT_s]$ and $x[kT_s]$, respectively. The number of samples acquired for phase computation is referred as the *sample population size* (N). The ratio ($\mathcal{R}$) of the mean of N number of samples for the discretized XORed signal to the reference clock (CLK1) is given by Eq. (6.3).

$$\mathcal{R} = \frac{\sum_{k=1}^N y[kT_s]}{\sum_{k=1}^N x[kT_s]}. \quad (6.3)$$

This ratio, ($\mathcal{R}$), tends to a constant value as the sample population size (N) increases, as seen by the following equation:

$$\mathcal{R}_{N \rightarrow \infty} = \frac{D_\theta}{D} = \frac{D_{\theta 1} + D_{\theta 2}}{D}. \quad (6.4)$$

The phase difference, t_θ , can be calculated from the Eq. (6.4), by replacing $D = \frac{1}{2}$ from the Eq. (6.1) and $D_\theta = \frac{t_\theta}{T/2}$ from the Eq. (6.2). The relationship between t_θ and $\mathcal{R}$ is shown below, where $\mathcal{R}$ can take any value within the range of $[0 - 2]$:

$$t_\theta = \frac{T}{4} \times \mathcal{R}. \quad (6.5)$$

The phase shift measurement can be represented in angular form by replacing $T/4$ with 90° :

$$\tilde{\theta} = 360^\circ \times \frac{t_\theta}{T} = 90^\circ \times \mathcal{R}, \quad (6.6)$$

where $\angle \tilde{\theta}$ lies between the range of $[0^\circ - 180^\circ]$.

From the measured phase difference it cannot be distinguished whether it is the leading or the lagging phase shift. To resolve the phase polarity ambiguity, the reference clock (CLK1) is used to sample the phase shifted clock (CLK2) to determine whether the phase

magnitude ($\tilde{\theta}$) is positive (leading) or negative (lagging) (refer to Figure 6.1). Thus,

$$\theta = \begin{cases} \tilde{\theta} & : f[kT + t_\theta] = 0, \\ -\tilde{\theta} & : f[kT + t_\theta] = 1. \end{cases} \quad (6.7)$$

6.3 Sampling methodology

Commonly used stratified or random sampling [85] for signal processing holds no synchronous relationship between the sampling clock and the reference clock used to drive the logic signal. In the present work the concept of systematic sampling method [85] is used to sample the phase information signal or the XORed signal. The necessary condition for having systematic sampling is to maintain a synchronous relationship between the sampling clock (SAMPLE CLK) and the reference clock (CLK1).

Since, systematically registering only part of the possible sample set is done, hence the collected samples are referred as ‘Sub-samples’. This systematic sampling approach is illustrated in Figure 6.2. Here, the progression through the sub-samples is cyclic, which means that the sub-sample starts repeating itself after every Δ^{th} element in the selected sample frame.

The cyclic vector of sub-samples $\mathbf{S}$ of order Δ is represented as $\mathbf{S} = \{s^0, s^1, s^2, \dots, s^{\Delta-1}\}$, then the relationship between $s^\Delta = s^0$ holds. The $\mathbf{S}$ forms a subset of the selected sampling frame or the sample population vector $\mathbf{P}$ of size N . To denote this mathematically, $\mathbf{S} \subset \mathbf{P}$ where $\Delta < N$.

From the isometric property of systematic sampling on the XORed signal, the generated sample population vector ($\mathbf{P}$) constitutes multiple sets of the cyclic vector of sub-samples, $\mathbf{S}$, that repeats itself and are isomorphic to one another. The number of subsets of $\mathbf{S}$ possible from a total of $\mathbf{P}$ is expressed as the *frequency ratio* of the reference clock to the sampling clock, given by the Eq. (6.8). The *frequency ratio* is reduced to its simplest form given by $\Delta : \Psi$, such that Greatest Common Divisor between the two factors is one

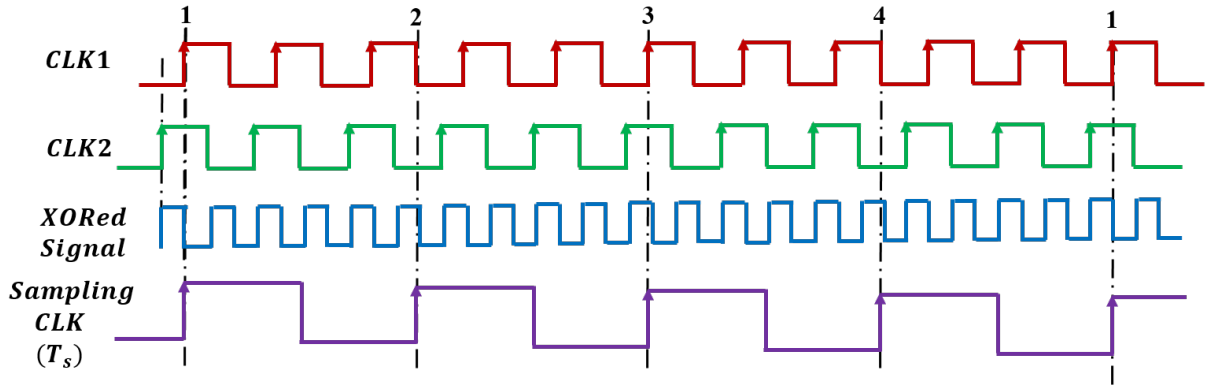


Figure 6.2: The scanning of the XORed waveform using the Sampling clock is illustrated for $F_s/F = 4/10$ or $\Delta/\Psi = 2/5$.

and both the factors belong to the set of natural numbers ($\mathbf{N}$). This is expressed in the form:

$$\frac{F_s}{F} = \frac{\Delta}{\Psi} \text{ where } \mathbf{gcd}(\Delta, \Psi) = 1 \text{ and } \{\Delta, \Psi\} \in \mathbf{N}. \quad (6.8)$$

Eq. (6.8) states that the Δ unique sub-samples of the XORed signal are registered by the sampling clock within the Ψ cycles of the measured reference clock. The ratio of Ψ/Δ is referred as the sampling interval. The sampling period (T_s) is proportional to the sampling interval.

The present phase calculator is sensitive to phase magnitude variations within the range of $[0 - T/2]$, as can be inferred from Eq. (6.5). To register the smallest phase difference of t_θ , the minimum number of elements required in the sub-sample cyclic vector ($\mathbf{S}$), is given by:

$$\Delta \geq \frac{T/2}{t_\theta}. \quad (6.9)$$

The following four constraints are constituted from the afore mentioned equations that

guide the proper choice of sampling frequency:

$$\left. \begin{aligned} \Delta &\geq \frac{1}{2Ft_\theta} \\ \text{gcd}(\Delta, \Psi) &= 1 \\ \Delta/\Psi &= \begin{cases} > 1 & : \text{ OVER-SAMPLING} \\ \leq 1 & : \text{ UNDER-SAMPLING} \end{cases} \\ F_s &= \frac{\Delta}{\Psi} \times F \end{aligned} \right\}. \quad (6.10)$$

The sampling of *CLK1* and *CLK2* with *SAMPLE CLK* involves asynchronous clock domain crossing, that can give rise to metastable output [39]. Hence, synchronization register chain or *synchronizer* are used for metastable signal to settle down, as shown in Figure 6.4.

6.4 Generating the sub-samples

During systematic sampling, there are two main methods through which one can produce more sub-samples to increase the resolution of the phase measurement. These two methods are discussed in the following subsections.

6.4.1 Method of Cascaded PLL

When the multiplication factor exceeds the VCO maximum output frequency capability in PLL circuitry, then the PLL frequency synthesizer fails. In that scenerio PLL-to-PLL cascading is needed. Here the multiplication factor (Δ) is splitted between the two PLLs. The cascaded PLL having same bandwidth settings has Jitter amplification effect [86]. To minimize jitter effect, Altera (now part of Intel) [87] recommends the use of a low-bandwidth setting for the source (upstream) PLL, and a high-bandwidth setting for the destination (downstream) PLL. The first (upstream) PLL acts as a jitter filter when

configured as low bandwidth and transfers very little jitter to the downstream PLL. A high bandwidth setting on the downstream PLL allows to track jitter from the first PLL.

To illustrate the implementation of this method, let us take for example 120 MHz clock whose phase shift of $1/100^{th}$ of the total clock period is needed to be registered. Then according to Eq. (6.10), the number of sub-samples required is 50 ($\Delta = 50$). However, the PLL multiplication factor of 50 requires VCO to operate at 6000 MHz frequency, that exceeds the maximum VCO output frequency limit of 1300 MHz in PLL circuitry for Intel Cyclone FPGAs. Hence, the multiplication factor after splitting between the two PLLs are $\Delta/\Psi|_{PLL1} = 5/7$ and $\Delta/\Psi|_{PLL2} = 10/17$ respectively.

6.4.2 Sampling Frequency for Sub-samples generation

The phase measurement design works for both over and under sampling clock frequency to generate a particular *number of sub-samples* (Δ), by choosing lower or higher PLL division factor (Ψ). The preference is given to the under-sampling frequency, because it improves the phase noise performance of the *Sampling CLK* significantly. According to the popular PLL manufacturers like Texas Instruments [88] and Silicon labs [89], one of the cardinal principle is that multiplication of frequency by N causes degradation in phase noise performance by $20 \log(N)$ with reference to the input clock, while dividing it by same improves by the same factor [88]. In a classical PLL [90], due to the existence of the divide-by M in the feedback path and the PD/CP, the divider noise amplification factor (in power) is M^2 . However, the biasness towards under-sampling can be avoided if sub-sampling technology based PLL is used as proposed by Gao in 2009 [91].

6.5 Computation of Phase Magnitude

The computation of phase magnitude in time or in angular domain needs to be evaluated using Eqs. (6.5) or (6.6), respectively. The generalized form of the equation is shown as:

$$\mathcal{P} = K \times \mathcal{R}, \quad (6.11)$$

where $\mathcal{P}$ is the phase value and K is the conversion factor. Depending on whether phase calculation is expressed in degrees or in time domain the value of the conversion factor (K) is chosen as 90° or $\frac{T}{4}$, respectively. The evaluation of Eq. (6.11) involves two operations, multiplication and division. The division operation is used while computing the ratio $\mathcal{R}$ (refer to Eq. (6.3)) and multiplication operation is needed between K and R as given in Eq. (6.11) .

For ease of formulation, lets represent the integrated values needed for division operation by $\alpha = \sum_{k=1}^N y[kT_s]$ and $\beta = \sum_{k=1}^N f[kT_s]$. The present value of dividend for each iteration step is denoted by (α_i) and the present value of divisor for each iteration is denoted by (β_i) . The dividend is initialized by the value of α and divisor by value of β . A shifted divisor binary-division technique [92] is employed for calculating the division result of α/β . The steps of division method is kept tracked by the counter variable (i). The division is done by right shifting the register value of β with each iteration of the counter variable (i) and for each iteration the divisor is stored in variable (β_i) by overwriting the past value as given in the following equation:

$$\beta_i = \beta \times 2^{-i} \quad (6.12)$$

The decision making factor $\delta[i]$ in Eq. (6.13) stores the comparison value between divisor

(β_i) and dividend (α_i) , which determines whether to continue with division operation.

$$\delta[i] = \begin{cases} \alpha_i \geq \beta_i & : -1 \\ \alpha_i < \beta_i & : 0 \end{cases} \quad (6.13)$$

The computed divisor value β_i is subtracted from the dividend α_i , based on evaluated value of the decision making factor $(\delta[i])$ and at each iteration the evaluated remainder overwrites the dividend (α_i) in next clock cycle as denoted by α_{i+1} in Eq. (6.14).

$$\alpha_{i+1} = \alpha_i + \beta_i \delta[i], \quad \alpha_0 = \alpha \quad (6.14)$$

The second operation is to multiply the conversion factor (K) with the quotient from the division method. However, this would mean to wait for the quotient value computation to finish, before performing the multiplication. To make the operation parallelized the quotient bit generated by the division method must be available simultaneously for the multiplication method to operate using the same bit. The intermediate quotient bit can be obtained by taking the modulus of the decision making factor as given by $|\delta[i]|$. Eq. (6.15) shows how multiplier value (K) is shifted synchronously with the divisor value (β) and stored in variable ϕ_i .

$$\phi_i = K \times 2^{-i} \quad (6.15)$$

The sum of the product values between ϕ_i and quotient value $|\delta[i]|$ is continued for M number of iterations, to get the final result as shown in Eq. (6.16).

$$\tilde{\theta} \approx \sum_{i=0}^M \phi_i |\delta[i]| \quad (6.16)$$

The constant M indicates the number of iterations of counter variable (i) needed, so that

the remainder (α_i) becomes negligible or zero, or to say it in other way, it is equal to the number of steps needed to shift bits rightward till β becomes 0.

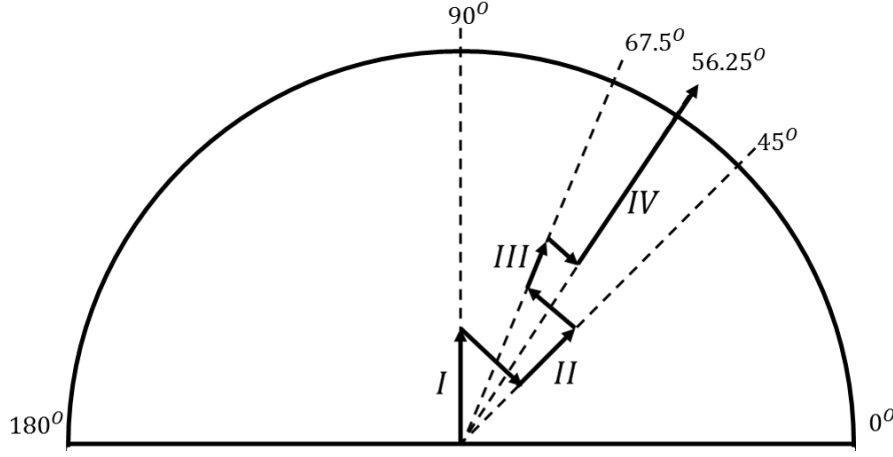


Figure 6.3: Stepwise successive approximation of phase value computation for 56.25° .

Depending on the floating point precision required, ϕ_i must be initialized with conversion factor (K) padded with zero bits. For maximum precision the number of blank bits padded must be of equivalent size as divisor β . This method of *successive approximation* of phase value computation is illustrated in Figure 6.3.

The final phase value in the study, is stored in a 64 bit register in little-endian format and represented in binary fractional form as shown in:

$$\theta = (-1)^{sign} \times \tilde{\theta} \times 2^{-32}. \quad (6.17)$$

Register size of 32 bits for β is used.

The VLSI architecture of the phase measurement system is shown in Figure 6.4.

6.6 Measurement Error

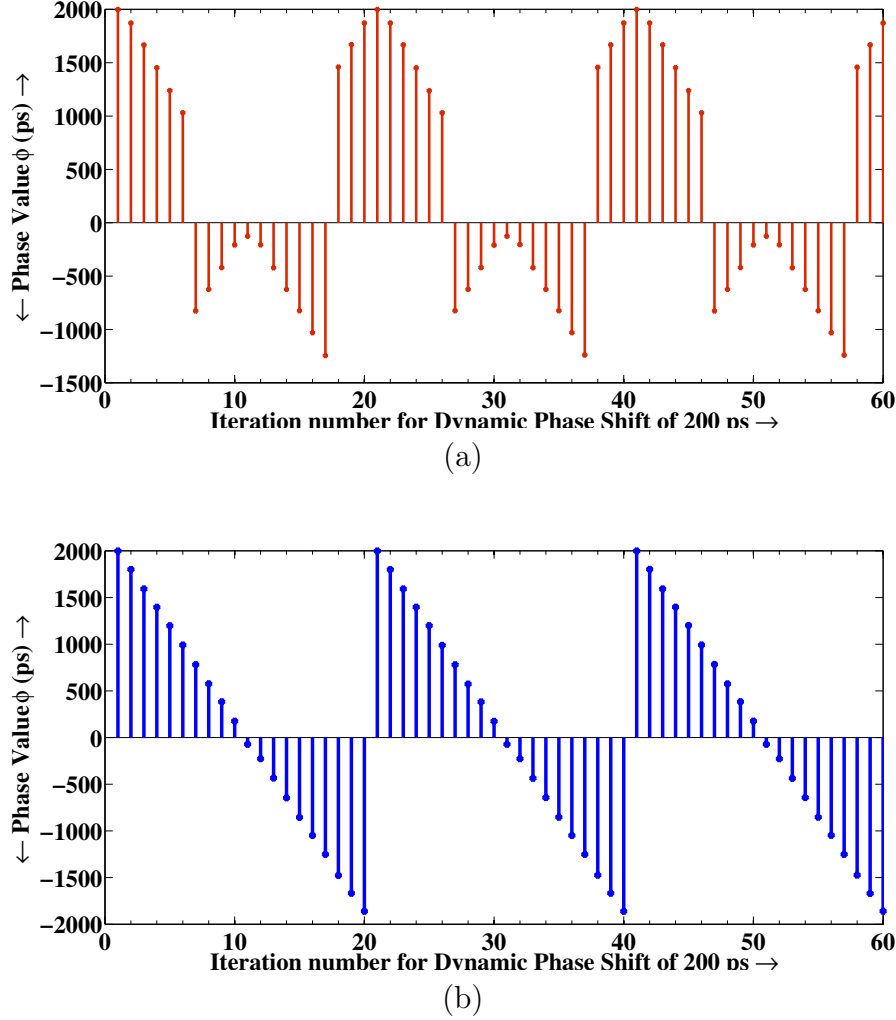


Figure 6.5: Effect of Phase polarity offset: (a) when no bank skew cancellation is in effect, the magnitude and polarity are mis-aligned leading to measurement ambiguity; (b) when bank skew cancellation is in effect the quantization steps are properly aligned.

The phase calculator, like any other instrumentation technique, is subjected to certain kinds of errors. This can be listed as follows.

Phase Magnitude offset It occurs due to clock skew ($t_{skew(m)}$) between CLK1 and CLK2. The clock skew [93] is minimized by setting proper timing constraints to instruct the fitter Place and Route (PAR) engine to do more robust timing closure for defined paths

[94], and prevents variation of clock skew by large margin with each firmware upgrade.

Phase Polarity offset It occurs due to Bank Skew ($t_{skew(p)}$) between Phase Detector block and Polarity Detector block. Bank Skew [93] is a type of skew that refers to the output skew between the signals with a single driving input terminal. Here, $t_{skew(p)} = \{t_{CLK1} - t_{CLK1*}\} - \{t_{CLK2} - t_{CLK2*}\}$. The error induced by polarity offset can be seen in Figure 6.5 (a).

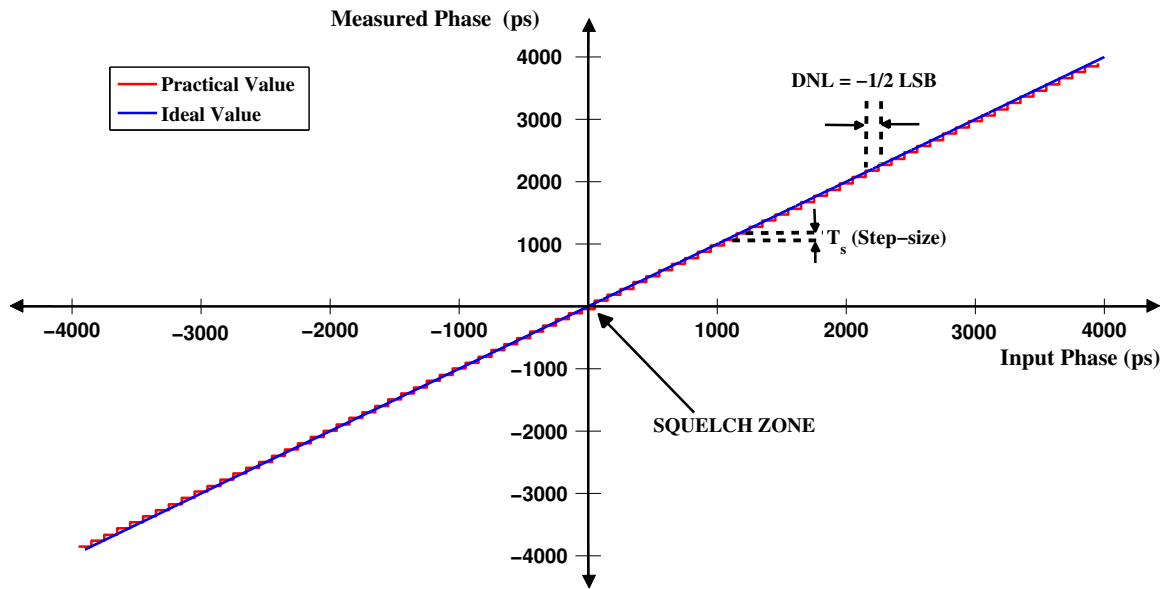


Figure 6.6: Transfer Characteristics of Phase Measurement Logic Core when operating in *Active low Sample Counting Mode*.

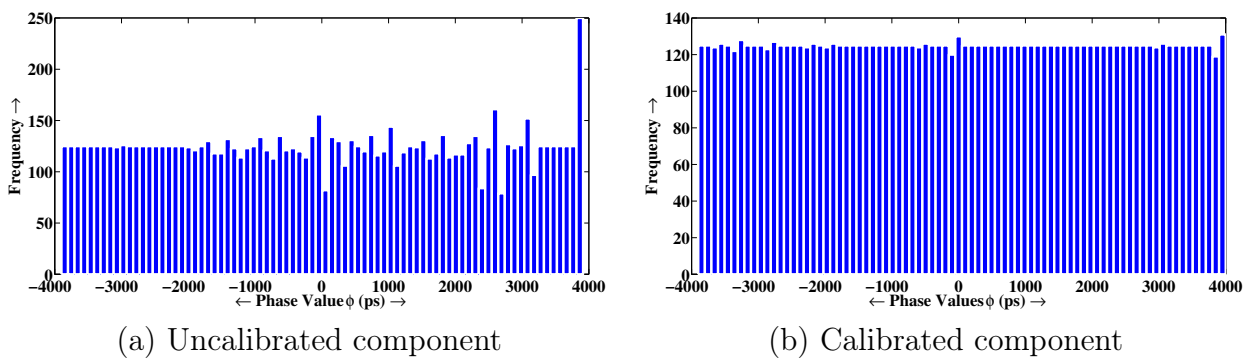


Figure 6.7: Histogram of Phase-Stepped Values.

Non-Ideal Transfer Characteristic The phase estimation logic is a typical mid-tread uniform quantizer. The quantization step is equal to the sampling interval T_s and the input-output relationship for the forward quantization step is expressed as,

$$y_k = k.T_s \text{ where } k = \left\lfloor \frac{x}{T_s} \right\rfloor + \frac{1}{2}. \quad (6.18)$$

Finite time resolution in phase measurement introduces a *quantization error* between the input phase and the reconstructed output phase value. Assuming this *quantization error* of phase value is uniformly distributed over the step width from $-1/2$ LSB to $+1/2$ LSB as shown in Figure 6.8, the mean value is equal to 0 and the variance is

$$E(\epsilon^2) = \frac{1}{T_s} \int_{-T_s/2}^{T_s/2} \epsilon^2 d\epsilon = \frac{T_s^2}{12}. \quad (6.19)$$

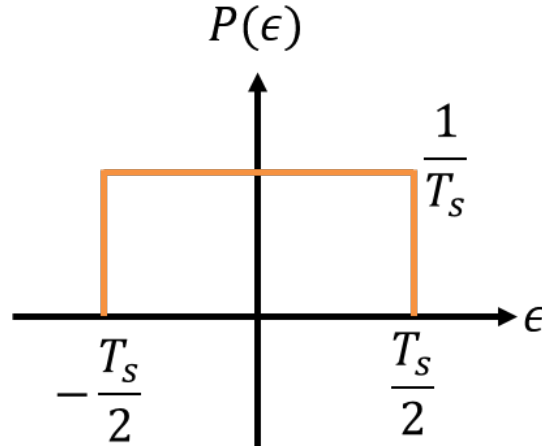


Figure 6.8: PDF of Quantization Error (ϵ).

Mid-tread quantizer shows symmetric dead-zone behavior around 0 or the end region zone, which is termed as *Squelch zone* as shown in Figure 6.6. In this zone the samples collected is too less to make decisions, whether to consider it as, very low phase difference or suppress it as noise. There also exists another kind of non-linearity, known as Differential Non-linearity (DNL) where the actual step size shows deviation from the recorded

step size during phase measurement. The non-linearities in the transfer characteristics cannot be fully eliminated, however the nature remains fixed for a particular *sample population size* (N). Using the least squares fit method to the acquired data-set, a calibration curve is fitted that exhibits a linear response characteristics with a constant slope factor of K/D , as can be derived from eqns. (6.4) and (6.11) respectively.

6.7 Calibration

The Phase measurement technique needs a calibration procedure to eliminate the offset due to random and systematic effects. A linear phase-step method is used to calibrate the component. In Intel chips Dynamic Phase Alignment (DPA) technology [95] allows phase step resolution of $1/8^{th}$ of VCO frequency. Dynamic Phase Shift method is used to detect amount of bank skew correction needed to compensate for *Phase Polarity Offset*:

$$\text{Bank Skew Cancellation delay} = -t_{skew(p)}. \quad (6.20)$$

Figure 6.5 (a) shows the effect of *Phase Polarity Offset*. This is corrected by adding bank skew cancellation as indicated in Figure 6.4 to get linear response characteristics as shown in Figure 6.5 (b). In the test case -250 ps bank skew cancellation is used.

The effect of non-linearity in the phase measurement transfer characteristic is indicated in Figure 6.6. The readings are taken at a step interval of 100 ps over 10000 repetition cycle. Figure 6.7 (a) shows without calibration how phase magnitude demonstrates non-uniform distribution, due to *squelch zone* and DNL. The quantization error is corrected using a software based Look Up Table method. Then after calibration, Figure 6.7 (b) demonstrates uniform phase magnitude distribution.

Static calibration methods are used to compensate for systematic errors. Dynamic calibration method is needed to compensate for random errors, caused by the timing skew shift due to temperature and voltage drift. The procedure for dynamic calibration

are not discussed, as in this case the board is kept in controlled environment. However, the averaging process involved in computation step suppresses the errors due to random effects.

6.8 Results

6.8.1 Measurement Setup

The course phase shift of minimum granularity in the order of $70-100\text{ ps}$ is asserted by the DPA feature of the PLL, where the minimum phase shift allowed limited by the $1/8^{th}$ of the VCO frequency within FPGA. The finer phase shift in the order of $8-30\text{ ps}$ is inserted by variation of the cable length to adjust the propagation delay. For characterizing the phase measurement system in low-end FPGA Intel Cyclone IV FPGA board is used to register phase shift in 250 MHz clock with a minimum resolution of 100 ps . The phase calculator is set to collect sub-samples (Δ) of 119 within a sample population size (N) of 10000. The phase measurement system is quantified in terms of the performance parameter given by accuracy, precision and resolution.

1) *Accuracy* : It estimates the closeness of the measured value to the actual or true value. *Systematic errors* affect the accuracy involved in a system [96]. Sample population size (N) is one of the system parameter that plays a dominant role in determining the operating accuracy of the phase measurement system as shown in Figure 6.10. The total system response time increases in order of $O(N + \log_2 N)$ as N increases. The response time includes the combined effect of sample accumulator and phase computation block as shown in eqns. (6.3) and (6.16) respectively.

2) *Precision* : It measures the spread of the readings when they are repeated. *Random errors* limit the precision of a system; hence it can never be zero [96]. However, at higher sample population the precision of the system improves. The effect is visualized in a more graphical way by a bullseye representation showing only 100 readings. For lower sample

population size of $N = 10^3$ the accuracy is moderately high but precision is low as shown in the polar plot in Figure 6.13 (a), and for higher sample population size of $N = 10^4$ both the accuracy and the precision is high as plotted in Figure 6.13 (b).

The precision distribution of the phase measurement system [97] is affected by two categories of errors as illustrated in Figure 6.9. The sampling error occurs when the XORed signal undergoes systematic sampling by the *Sampling Clk* and leads to sub-sample variation. The aperture error happens when erroneous sample values accumulate over a set of sample population acquisition and lead to cumulative error in the computed average value.

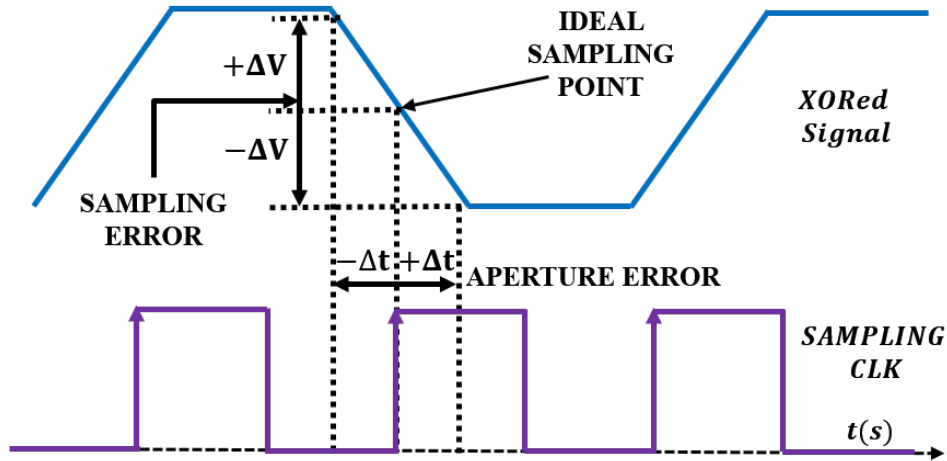


Figure 6.9: The effect of sampling on precision of the system.

According, to the Central Limit Theorem the sampling distribution of the sampling means approaches Gaussian or Normal distribution as the sample population size increases, irrespective of the probability distribution [98]. The Probability Distribution Function (PDF) of the sampling error distribution be represented as $P(x)$. The PDF of aperture error be represented as $P(y)$. For simplicity let us assume that both the PDFs are a Gaussian distribution with mean 0 and standard deviation σ , denoted by $N(0, \sigma)$. Since both the error sources are statistically independent Gaussian distribution, their joint PDF is written as $P(x, y) = P(x)P(y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$. The envelope of the sum of two quadrature Gaussian noise signals obeys a Rayleigh distribution. Solving for the

joint probability distribution by transforming it into polar coordinates (r, ϕ) , Eq. (6.21) is derived.

$$\begin{aligned}
 P(r, \phi) &= P(\phi)P(r) \\
 &= (\text{Uniform Distribution}) \times (\text{Rayleigh Distribution}) \\
 &= \frac{1}{2\pi} \times \frac{r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}} \quad : \quad r \in [0, \infty], \phi \in [-\pi, \pi].
 \end{aligned} \tag{6.21}$$

Histogram of precision distribution plotted over population size of 10000 in the range of $[-4000 \text{ ps to } 4000 \text{ ps}]$ is shown in Figure 6.11. The skewness of the precision distribution curve exhibits different behaviour for *active low sample counting mode* and for *active high sample counting mode*. For *active low sample counting mode* the skewness is towards minimum value of 0 ps and for *active high sample counting mode* the skewness is towards full-scale reading value of 4000 ps as plotted in Figure 6.11 (a) and (b). Either of the plots shows symmetric behaviour across center phase value of 0 ps because the magnitude response remains same only the polarity sign changes. To obtain the nature of the distribution a sample population size of 50×10^3 is chosen. It owes to the fact that for more number of samples averaging effect suppresses the relative uncertainty and makes it hard to decipher the distribution while for less number of samples too little statistics are gained to determine the nature of the distribution.

3) *Resolution* : It indicates the ability of the phase measurement system to detect and display smallest changes in the characteristic of the results. It is limited by the step size of the quantization step. The number of quantization steps involved in the measurement is determined by the number of sub-samples (Δ) collected by the system. For finer resolution more number of sub-samples need to be collected and quantization error of the system also reduces. To operate the phase measurement system in a particular resolution, the user can operate both in under-sampling or in over-sampling mode by properly choosing

the frequency ratio, as shown in Figure 6.12.

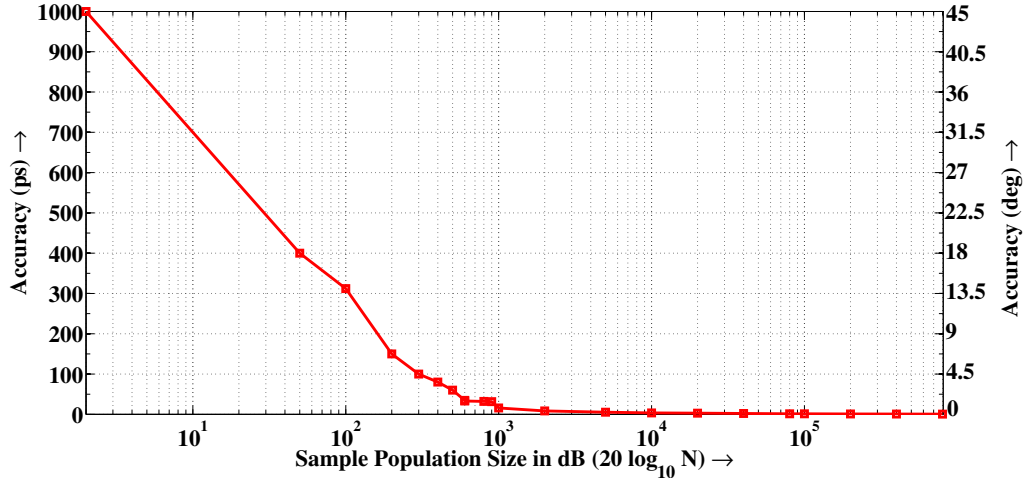


Figure 6.10: Phase measurement accuracy, presented in ps and in degrees, as a function of sample population size (N).

6.8.2 Comparison

Intel’s Cyclone IV FPGA is used to characterize the phase component in low-end FPGAs with 10,000 datasets, where measurement resolution achieved is around 34 ps. For High-End FPGAs like Stratix[®] V and Arria[®] 10 , the resolution goes to 8-10 ps depending on the input clock jitter. High end FPGA circuitry got better jitter performance leading to better performance of the phase measurement system. A detailed comparison of the proposed design competency against other recent works is shown in Table 6.1. The presented architecture outperforms other in-situ FPGA-based PD implementation in resolution and also in maximum operational frequency. The performance results are reasonably competitive with other ex-situ based implementations.

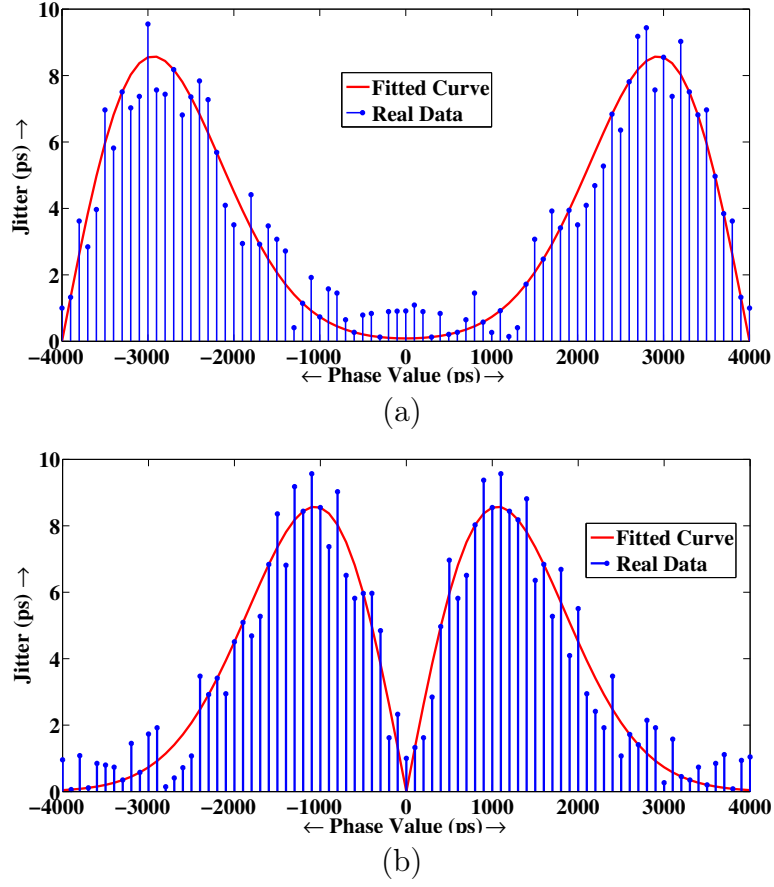


Figure 6.11: Precision Distribution for 50×10^3 sample population size operating in (a) Active High Sampling Counting Mode; (b) Active Low Sampling Counting Mode;

Table 6.1: FPGA based phase detection methods

Parameters	Proposed Method	Hidvégi <i>et. al.</i> [81]	Lu <i>et. al.</i> [78]	Trebbels <i>et. al.</i> [79]	Rong <i>et. al.</i> [80]	Guntoro <i>et. al.</i> [99]
Sampling Unit	FPGA based Internal Sampling	FPGA based Internal Sampling	ADC based Dual Channel Data Sampling	Delta Modulator (DM)	ADC based Sampling	ADC based Sampling
Phase Detection Technique	Sampling Clock Scanning	Step-wise Dynamic Phase Drift	Phase Difference between I and Q channels	DM-Integrator Triggered Feedback	I-Q channel based PD using CORDIC	Phase Difference between I and Q channels
Sampling Technique	Under-Sampling / Over-Sampling	Under-Sampling	Over-Sampling	Over-Sampling / Under-Sampling	Over-Sampling	Over-Sampling
Max. Freq. Used	250 MHz	50 MHz	200 MHz	10 MHz	1 MHz	100 MHz
Detection Range	$-180^\circ - 180^\circ$	$0 - 180^\circ$	$0 - 360^\circ$	$0 - 180^\circ$	$-180^\circ - 180^\circ$	$-180^\circ - 180^\circ$
No. of Samples Collected	33×10^4	33×10^4	0.24×10^4	100×10^4	0.013×10^4	0.1024×10^4
Max. Sampling Rate	Any (Depend on FPGA Fabric)	33.33 MHz	3 GSamp / sec	500 MHz	100 MHz	400 MHz
Accuracy	CYCLONE IV	Arria [®] 10	0.1 ⁰	-	0.05 ⁰	0.00273 ⁰
	0.27 ⁰ (3 ps)					
Precision	0.63 ⁰ (7 ps) [#]	0.18 ⁰ (2 ps) [#]	0.044 ⁰ (2.4 ps)	0.0032 ⁰ (0.044 ps)	0.0094 ⁰ (26 ps)	0.00314 ⁰ (0.872 ps)
	3.06 ⁰ (34 ps) [§]	0.72 ⁰ (8 ps) [§]	1.8 ⁰ (≥ 100 ps [*])	0.021 ⁰ (0.3 ps)	0.05 ⁰ (139 ps)	0.00195 ⁰ (0.054 ps)

[#] Depends on sample population size, [§] Depends on number of sub-samples, ^{*} Bounded by Phase Drift Step size of DPA

N.B.: Here the absolute measurement in ps should be considered, not the degrees for comparison metrics

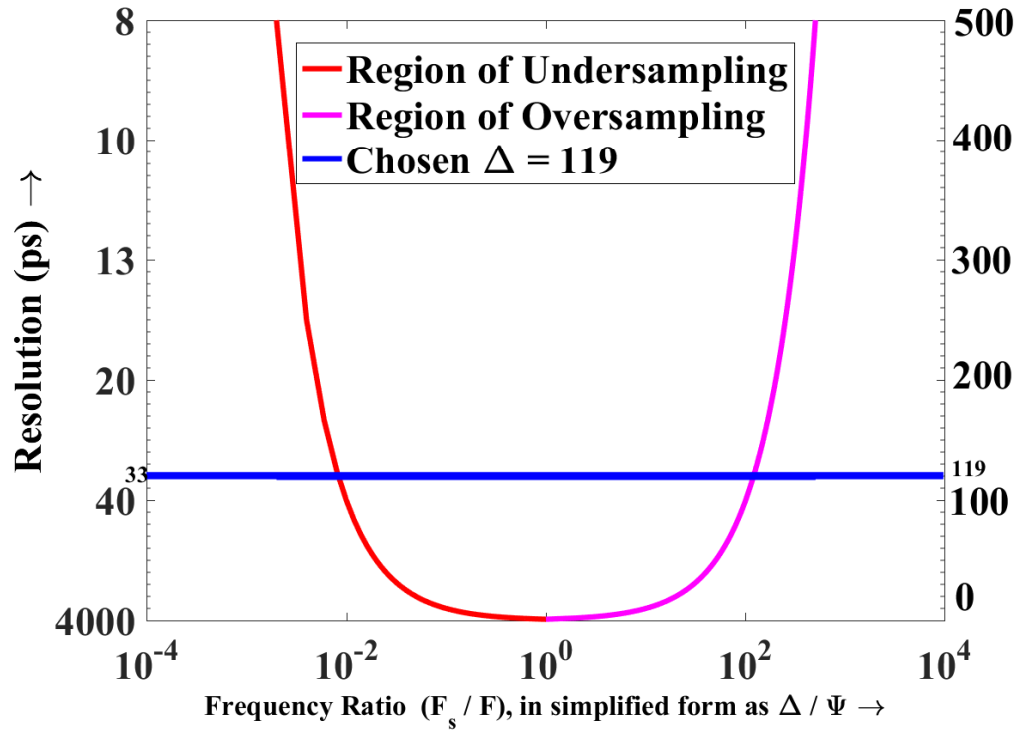


Figure 6.12: The dependence of resolution (ps) and *number of sub-samples* (Δ) with the frequency ratio (F_s/F).

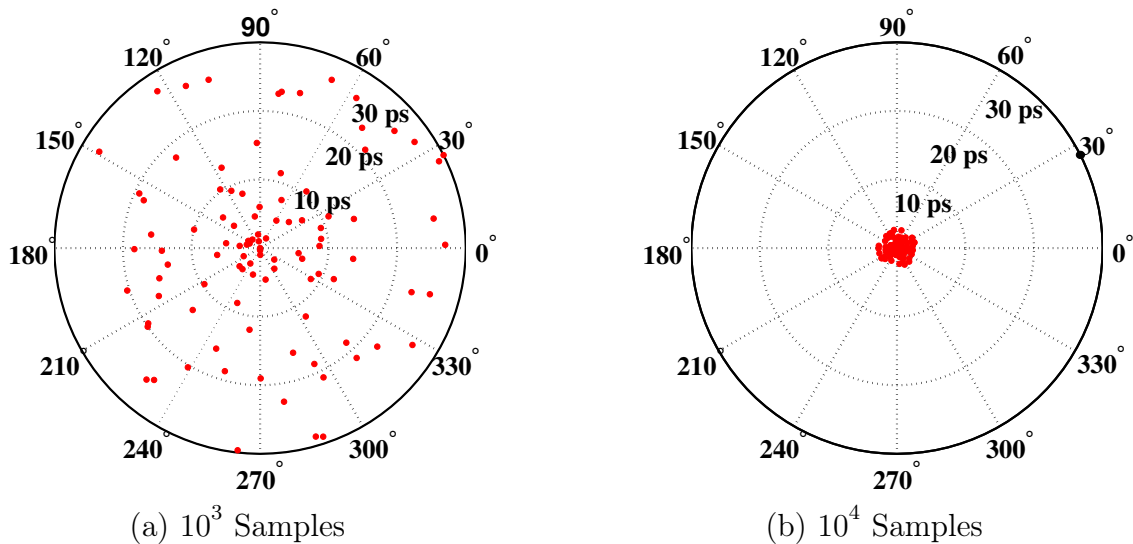


Figure 6.13: Graphical bullseye representation of accuracy and precision (a) High accuracy and low precision; (b) High accuracy and high precision.

6.9 Conclusion

In the chapter the development of a sensitive Phase Detection (PD) logic core for FPGA, having precision, accuracy and resolution in the range of few picoseconds is discussed. This is one of my key development work in the thesis. The logic core can be used within an FPGA as a monitoring device of phase relationship between digital clock pulses, without any additional circuitry. The design is modularized in a way that allows designers to modify different components for more robustness of the design, like replace XOR based PD with other phase comparator. The concept of using systematic sampling for sub-sample collection can also be extended to map other complex analog domain problem to digital domain.

Chapter 7

Use of CRC-32 Checksum in the Detector Specific Data Integrity Check

7.1 Introduction

In this age of information revolution, data integrity is everything. Protecting Detector Specific Data Packet (DSDP) within CRU is of prime importance. Even with most reliable high-speed communications the data errors do exist. So, to handle such cases protective measures are taken by applying error detection and corrective measures. In this regard, Cyclic Redundancy Check (CRC) is chosen because of its simplicity and low resource usage, as second layer of data integrity checker. In earlier days, the data rate was low, and correspondingly the algorithm was optimized at that data rate. To match the modern requirement of high-speed data transmission, the algorithm needs to be revised for minimum latency. Such low latency is possible by more pipelining and parallelism. The design is based on generalized FPGA architecture. However, the test analysis is related to Intel[®] (formerly Altera) Arria[®] 10 FPGA used in CRU PCIe40 DAQ Engine. The work presented is specific to the CRC-32 Ethernet standard. A novel idea of reconfigurable latency optimized high throughput CRC-32 architecture is presented. It has a very low

latency of one clock cycle, between the data word submitted and the result generated. The high throughput can be controlled with the usage of expandable data bus. The entire architecture supports run-time or compile-time reconfiguration. Instead of presenting the work in a generic way, the CRC-32 standard needed in CRU project is used as benchmark for elaborating the research results. This helps to understand the results much easily and then use it in a generic sense to adapt to other custom requirement.

The chapter is organized into multiple sections. Introduction section tells the motivation behind developing such architecture. It is then followed by literature survey, which explains why such redesigning is required when there already exists plenty of standards. The mathematical interpretation of the design gives an idea how to customize the CRC-32 development according to their required polynomial type and size. A special section is given just to explain the importance of latency optimized architecture and how it can be achieved. Then the functional design of CRC unit is discussed, which assumes the internal operation as black box and only highlights the bus signals functional roles. Section 7.7 is dedicated to discuss how the architecture can be re-synthesized offline or reconfigured in run-time. It is then followed by a elaborate test results, analysis and discussion of the implemented design under various test setups. The test setup is broadly grouped into two: Throughput study with various modes of interfacing (section 7.8.1); Resource utilization study with bus width variation (section 7.8.2).

7.2 Literature Survey

High throughput design is more concerned with steady-data rate than with its propagation latency. The pipelined based design requires certain handshaking signals and multiple clock cycles for doing a complete calculation. But low latency design is more concerned with minimization of intermediate delay between input and output. The hardware based CRC concurrent architecture happens to be faster than the most efficient

software implemented CRC algorithm. The concurrent VLSI implementation of byte-wise CRC generator proposed by Sait[100] requires 10 clock cycles for transmission of 8 byte message, which is just 2 more than the time required for its calculation. The most commonly used CRC is a 32 bit polynomial, which can detect all errors within the span of 32 contiguous bits and all 2 bit errors less than 2048 bits apart. For other types with varying size in error distribution is 1 in 232 bits [101]. With the steady growth of VLSI technology, many speed-up efforts have been made by the researchers by reducing the clock cycles required for CRC computation, increasing the input data vector width and increasing the clock frequency. One such prediction is given by Pei [102] in 1992 that if present development trend continues future VLSI implementation will be fast enough to meet CRC computation speedup. Various tradeoff methods are modeled for optimal design of circuit logic. CRC-32 finds common application in 10 gigabit Ethernet. There the CRC-32 generation is very time critical. In the Henriksson [103] work in 2001, it is mentioned that a speed of 8.7Gb/s is achieved with clock frequency of 1.09 GHz using 8bit parallelism. Increased clock frequency speed-up method is used in this process. But it still falls short from 10Gbps Ethernet throughput. With the advent of re-configurable boards like FPGA, it became attractive to explore design that can be configured in runtime. Toal [104] in 2009 explores programmable CRC architecture supporting 5Gb/s line rate for variable number of input octets.

The FPGA technology has evolved tremendously since then, in terms of size, power and speed. So, it requires a detailed revisiting of the algorithm to meet today's challenge, such that CRC with maximum latency optimized bus width is possible to design along with the runtime reconfigurable feature [105]. Elaborate mathematical model to produce such design is discussed that is needed for suitable implementation of the architecture along with its performance studies. This helps the user to choose the implementation options wisely.

7.3 Common CRC-32 standards in use

Varieties of cyclic redundancy checks have been incorporated in technical standards. Koopman and Chakravarty recommends that the polynomial should be chosen according to the suitability of the application requirement and the distribution of message length [106]. There is often a confusion among developers about which CRC to use from the number of CRCs that already exists in literature. Like there exist six types of CRC-32 polynomial [107]. The polynomial chosen may not be the efficient one in terms of the Hamming distance for the given message length. Sometimes the difference between two protocols can be as little as pre-inversion, post-inversion and reversed bit ordering, like CRC-32/Ethernet vs CRC32/BZIP2. Then it requires re-configuration of the entire CRC generator design. To address this issue, the design generic is made. The Table 7.1 lists only some of the popular CRC standards in use along with its parameter settings.

Table 7.1: CRC-32 TYPES

Name	Width	Polynomial	Initialize	Input Reflection	Output Reflection	Xorout
CRC-32/Ethernet	32	0x04C11DB7	0xFFFFFFFF	TRUE	TRUE	0xFFFFFFFF
CRC-32/MPEG-2	32	0x04C11DB7	0xFFFFFFFF	FALSE	FALSE	0x00000000
CRC-32/BZIP2	32	0x04C11DB7	0xFFFFFFFF	FALSE	FALSE	0xFFFFFFFF
CRC-32C	32	0x1EDC6F41	0xFFFFFFFF	TRUE	TRUE	0xFFFFFFFF
CRC-32D	32	0xA833982B	0xFFFFFFFF	TRUE	TRUE	0xFFFFFFFF
CRC-32/POSIX	32	0x04C11DB7	0x00000000	FALSE	FALSE	0xFFFFFFFF
CRC-32Q	32	0x814141AB	0x00000000	FALSE	FALSE	0x00000000
JAMCRC	32	0x04C11DB7	0xFFFFFFFF	TRUE	TRUE	0x00000000
XFER	32	0x000000AF	0x00000000	FALSE	FALSE	0x00000000

7.4 Mathematical Model

Variation in protocol includes pre-inversion, post-inversion and reversed bit ordering. Apart from those parameters, some protocols demand code word obfuscation, that involves the use of a non-trivial initial value and final XOR. For the sake of calculation consider a cyclic code operating on a data size of n bits generating a code word of k bits, with the size of the appended systematic cyclic code is r ($= n - k$) bits.

The generator polynomial for the cyclic code (n, k) is $g(x) = g_0 + g_1x + \dots + g_rx^r$ (with $r = n - k$). Given in Table 7.2 the parameter values of CRC-32 Ethernet standard as an example. For CRC-32 Ethernet standard the generator polynomial is 0x04C11DB7 (Normal polynomial representation) or 0xEDB88320 (Reversed polynomial representation).

Table 7.2: Paramater value for CRC-32 Ethernet Standard

Parameter	Value
r	32
$[g_0 \ g_1 \ \dots \ g_r]$	[1 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 1 1 0 1 1 0 1 1]
n	64
m	96
InitialValue	[11111111 11111111 11111111 11111111]
Input Reflection	true
Output Reflection	true
Xorout	[11111111 11111111 11111111 11111111]

A $n \times m$ Non-Systematic Generator matrix ($\mathbb{G}_N$) is constructed from the generator polynomial

$$g(x) = \sum_{i=0}^r g_i x^i$$

$$\mathbb{G}_N = \begin{bmatrix} g_0 & g_1 & \dots & g_r & 0 & \dots & \dots & \dots & 0 \\ 0 & g_0 & g_1 & \dots & g_r & 0 & \dots & \dots & 0 \\ \vdots & & & & & & & & \vdots \\ 0 & \dots & \dots & \dots & 0 & g_0 & g_1 & \dots & g_r \end{bmatrix}_{n \times m} \quad (7.1)$$

where $n = 2r$ and $m = 3r$

Applying Gaussian elimination technique Non-Systematic Generator matrix is converted to Systematic Generator matrix. It then takes the row echelon form.

$$\mathbb{G} = [I_n \mid \mathbb{S}] \quad (7.2)$$

where I_n is an Identity matrix of $(n \times n)$ and $\mathbb{S}$ is the syndrome matrix of $(n \times r)$.

In expanded bus mode operation, the data channel may not be fully utilized. This might generate wrong checksum, so how to tackle partially filled data bus line CRC generation. The equation consider such cases and able to handle half filled and fully filled conditions. Users can easily extend the concept for other partially filled data bus width length that might arise.

$$\mathbb{I} = \begin{cases} I_{2r} & \text{when input data is of } 2r \text{ bits} \\ \left[I_r \mid \psi_r \right] & \text{when input data is of } r \text{ bits} \end{cases} \quad (7.3)$$

where I_r is an input data row vector of $(1 \times r)$ and ψ_r is a row vector of $(1 \times r)$ zeros

Byte reflection matrix ($\mathbb{B}$) is composed of matrix elements. It contains a (8×8) square sub-matrix $\mathbf{B}$ of trailing diagonal one and (8×8) sub-square matrix Φ of null values. The dimension of byte reflection matrix is given as $(k \times k)$, when elements considered in numeric values, else it is given as $(k/8 \times k/8)$, when elements are considered in grouped (8×8) square-matrix format. For ease in representation former dimensional style is used.

$$\mathbb{B}_k = \begin{bmatrix} \mathbf{B} & \cdots & \cdots & \Phi \\ \Phi & \mathbf{B} & \cdots & \Phi \\ \vdots & & \ddots & \vdots \\ \Phi & \cdots & \cdots & \mathbf{B} \end{bmatrix}, \quad (7.4)$$

$$\text{where } \mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } \Phi = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The calculation is based on Modulo-2 Galoy Field operation, where “+” is logical XOR and “*” is logical AND. The processing of input and output vector is done for obfuscation

of the cryptographic algorithm used.

Byte Reflected Input row vector is given by

$$\mathbb{R} = \begin{cases} \mathbb{I} * \mathbb{B}_{2r} & \text{when } \mathbf{Input\ Reflection} \text{ is true} \\ \mathbb{I} & \text{otherwise} \end{cases} \quad (7.5)$$

The processed input row vector is then generated by shifting the reflected input row vector with a user defined initial value

$$\mathbb{P} = \mathbb{R} + \begin{cases} [\psi_r \mathbf{INIT}] & \text{when input data } 2r \text{ bit} \\ [\mathbf{INIT} \psi_r] & \text{when input data } r \text{ bit} \end{cases} \quad (7.6)$$

$$\mathbf{INIT} = \begin{cases} \mathbf{Initial\ Value} & \text{at } i=0 \\ \mathbb{C} & \text{otherwise} \end{cases} \quad (7.7)$$

where i indicates the step number

The Cyclic code $\mathbb{C}$ of $(1 \times r)$ is generated by operating on the processed input row vector using Syndrome matrix

$$\mathbb{C} = \mathbb{P} * \mathbb{S} \quad (7.8)$$

The byte reflection of the generated cyclic code row vector is done.

$$\mathbb{F} = \begin{cases} \mathbb{C} * \mathbb{B}_r & \text{when } \mathbf{Output\ Reflection} \text{ is true} \\ \mathbb{C} & \text{otherwise} \end{cases} \quad (7.9)$$

The processed output row vector $\mathbb{O}$ is then generated by final xoring of the byte

reflected cyclic code

$$\mathbb{O} = \mathbb{F} + \mathbf{Xorout} \quad (7.10)$$

$$(7.11)$$

Using the computational method just discussed a Syndrome Matrix/Table for CRC-32 (Ethernet Standard) with 64 bit input data bus width is constructed for illustrating the content format of the generated syndrome matrix.

$$\mathbb{S} = \begin{bmatrix} S_7 \\ S_6 \\ S_5 \\ S_4 \\ S_3 \\ S_2 \\ S_1 \\ S_0 \end{bmatrix} \quad (7.12)$$

$$S_7 = \begin{bmatrix} 79005533 \\ BEE0A442 \\ 5F705221 \\ ADD8A7CB \\ D48CDD3E \\ 6A466E9F \\ B743B994 \\ 5BA1DCCA \end{bmatrix} \quad S_6 = \begin{bmatrix} 2DD0EE65 \\ 9488F9E9 \\ C824F22F \\ E672F7CC \\ 73397BE6 \\ 399CBDF3 \\ 9EAED022 \\ 4F576811 \end{bmatrix} \quad (7.13)$$

$$S_5 = \begin{bmatrix} \text{A5CB3AD3} \\ \text{D08513B2} \\ \text{684289D9} \\ \text{B641CA37} \\ \text{D9406BC0} \\ \text{6CA035E0} \\ \text{36501AF0} \\ \text{1B280D78} \end{bmatrix} S_4 = \begin{bmatrix} \text{0D9406BC} \\ \text{06CA035E} \\ \text{036501AF} \\ \text{83D20E0C} \\ \text{41E90706} \\ \text{20F48383} \\ \text{921ACF1A} \\ \text{490D678D} \end{bmatrix} \quad (7.14)$$

$$S_3 = \begin{bmatrix} \text{A6E63D1D} \\ \text{D1139055} \\ \text{EAE946F1} \\ \text{F7142DA3} \\ \text{F9EA980A} \\ \text{7CF54C05} \\ \text{BC1A28D9} \\ \text{DC6D9AB7} \end{bmatrix} S_2 = \begin{bmatrix} \text{EC564380} \\ \text{762B21C0} \\ \text{3B1590E0} \\ \text{1D8AC870} \\ \text{0EC56438} \\ \text{0762B21C} \\ \text{03B1590E} \\ \text{01D8AC87} \end{bmatrix} \quad (7.15)$$

$$S_1 = \begin{bmatrix} 828CD898 \\ 41466C4C \\ 20A33626 \\ 10519B13 \\ 8A484352 \\ 452421A9 \\ A0F29E0F \\ D219C1DC \end{bmatrix} S_0 = \begin{bmatrix} 690CE0EE \\ 34867077 \\ 9823B6E0 \\ 4C11DB70 \\ 2608EDB8 \\ 130476DC \\ 09823B6E \\ 04C11DB7 \end{bmatrix} \quad (7.16)$$

The syndrome matrix $\mathbb{S}$ is a real $(n \times r)$ matrix, that defines a linear map from processed input row vector $(\mathbb{P}^{2r})$ to cyclic code $(\mathbb{C}^r)$. It is a linear mapping for cyclic division. The mathematical model discussed is for understanding the operational details of the latency optimized CRC-32 design architecture. It generalizes the architecture to expand data bus width in integral power of 2, and multiplexing of lesser width data lanes into larger width data bus lane. The matrix based calculation also gives an insight on how the architecture can be reconfigured or resynthesized by modifying the LUT (Look Up Table) content generated with the change in CRC generator polynomial, as shown in section 7.7.

7.5 Latency Optimization

The latency of any error detection algorithm is a combination of several delays. Gaining of high speed computation involves losing in area, bandwidth, reliability and power aspect. To increase throughput using parallelism of serialized logic involves more area, hence there is a trade-off between hardware costs versus detection latency [108]. More area utilization means more resources involved. Subsequently, more power consumption [109]. During high clock frequency operation, data reliability is a big issue as there is a chance

of jitter introduction due to improper clock routing, hence there is a tradeoff between reliability and latency [110]. If the channel is dedicated for single data line, then full bandwidth can be used. However, if it is to be shared among multiple clients, subsequently probability of data contention increases with decrease in latency. This latter requires placing of flow control; buffering and congestion control mechanisms. So, there is a tradeoff between latency and bandwidth for multiple users [111]. The goal is to optimize CRC-32 computation latency with respect to the discussed aspects.

The uniqueness of this design is its ability to calculate CRC in the same clock cycle in which the data is loaded. Only the final result is delayed by 1 clock cycle for latching the output result to prevent jitter in the CRC generated.

7.5.1 Parallelism and concurrent architecture

Understanding the data dependencies is fundamental part of parallelizing hardware implementation. Bernstein's conditions [112] states when two program segments are independent then it can be executed in parallel. Equation (7.17) shows that the mathematical steps in equations (7.5,7.6,7.9,7.10) can be parallelized, as the input data channel has one to one mapping with output response data-channel.

$$\left\{ \begin{array}{l} \mathbb{I}(x, y) \cap \mathbb{R}(x', y') = \phi \\ \mathbb{R}(x, y) \cap \mathbb{P}(x', y') = \phi \\ \mathbb{C}(x, y) \cap \mathbb{F}(x', y') = \phi \\ \mathbb{F}(x, y) \cap \mathbb{O}(x', y') = \phi \end{array} \right. \quad (7.17)$$

where (x, y) and (x', y') indicates a data index for particular row and column in the matrix and $(x, y) \neq (x', y')$

Equation 7.8 shows the cyclic code division, where the response of the computation

is based on the entire input data vector, with no dependency on the previous values. So, this part of the architecture can be designed concurrently, but it involves the longest time delay part, hence forms the most time critical portion. Equation 7.18 shows that the response lines are independent of each other.

$$\bigvee^{i,j,k,l} \left(\sum (\mathbb{B}_{i,j} \wedge \mathbb{S}_{j,i}) \right) \cap \left(\sum (\mathbb{B}_{k,l} \wedge \mathbb{S}_{l,k}) \right) = \phi \quad (7.18)$$

But, the entire response vector together forms a meaningful CRC value, hence it need to be hold for 1 clock cycle to gather the value of all the response lines. The *duration of clock cycle* has to be greater than the *maximum delay path*.

Equation 7.7 clearly indicates that in the INIT vector computation, there is a presence of data feedback part. This portion of the architecture is implemented using a positive edge sensitive shift register and multiplexer. For each consecutive data frame with 1st data set initial value is used and for rest of the data set, delayed copy of the CRC from the previous data set is used.

7.6 CRC Logic Core Functional Description

In section 7.4 working principle of CRC logic core is discussed. Same CRC block can be used as a generator or checker, and also able to handle half fill valid data lines. The Figure 7.1 shows a signal flow block diagram of CRC-32.

One of the major advantage of CRC engine that it can be shared both by the encoding block and the decoding block. The engine can start computing the checksum of a new packet while it is completing the calculation for the previous packet. Figure 7.2 shows the CRC engine to be used as CRC-32/Ethernet generator. At each clock cycle with *datavalid* signal asserted 64bit data is loaded and operated concurrently. The *byteenable* signal is set to indicate full bus data valid. The packet transmission starts on assertion of the *startofpacket* signal and stops on assertion of the *endofpacket* signal. In the

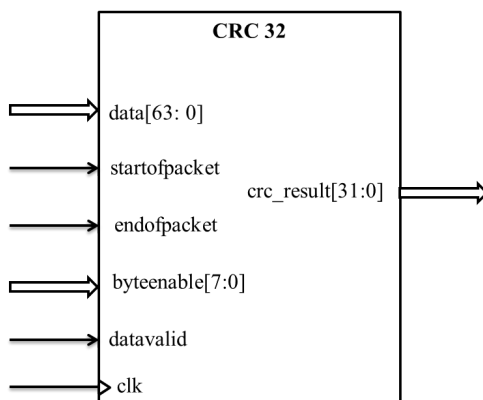


Figure 7.1: CRC-32 block diagram

immediate clock cycle following the *endofpacket* signal, the CRC checksum is latched in the *crc_result* bus signal.

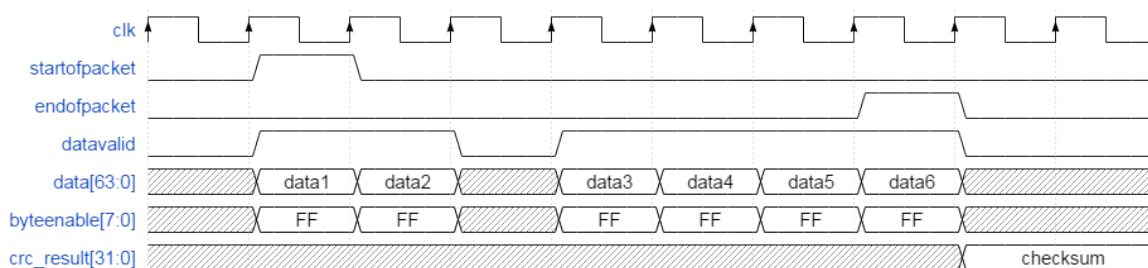


Figure 7.2: CRC-32 Generator Timing diagram

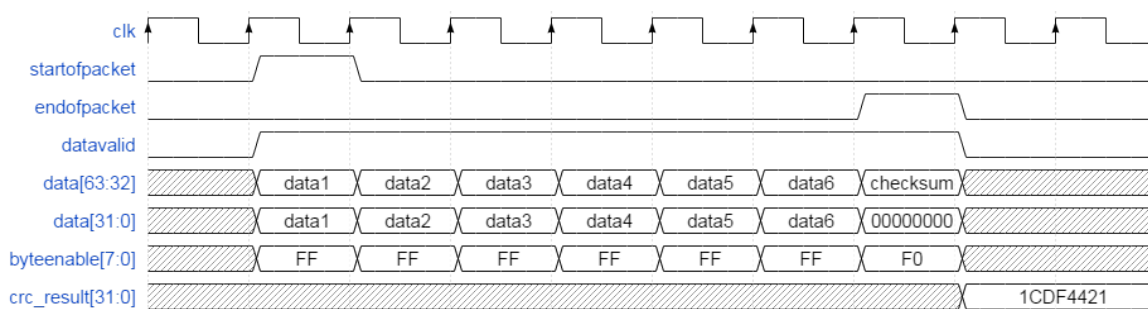


Figure 7.3: CRC-32 Checker Timing diagram with dynamic *byteenable*

The CRC checker result depends upon the design specification chosen and on the *byteenable* signal. Theoretically 32 bit cyclic redundancy check on a valid data with CRC checksum should evaluate to x“00000000”, but in the Ethernet/CRC-32 specification apart from using a 32 bit polynomial it involves x“FFFFFFFF” initialization, input reflection,

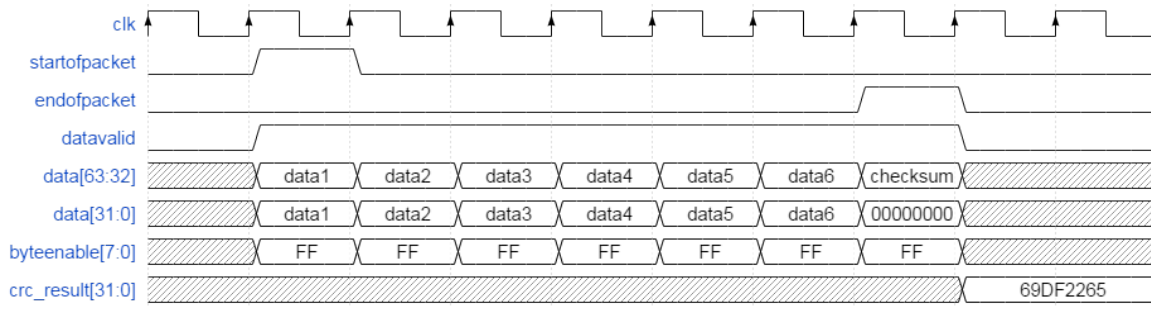


Figure 7.4: CRC-32 Checker Timing diagram with static *byteenable*

output byte reflection and final xor. This pre and post operation over the checksum gives the valid checker output. Figure 7.3 and 7.4) discuss the variation in the results of the data checker module with assertion difference in the *byteenable* lines. Both the design shows that data packet appended with the checksum is transmitted synchronously with *datavalid* signal asserted over the 64 bit data line, with begin and end marked by signals *startofpacket* and *endofpacket* respectively. Figure 7.3 shows the first checker mode, where an uncorrupted datapacket appended with CRC32 checksum evaluates to x“1CDF4421”. In this mode the *byteenable* line is asserted as x “F0” when the CRC checksum is loaded over the data bus line, to indicate the true size of the checksum which is 32 bit. Figure 7.4 shows the second checker mode, where an uncorrupted datapacket appended with CRC32 evaluates to x“69DF2265”. The *byteenable* line in this case is asserted as x“FF” when the checksum is loaded over the data bus line and it assumes the checksum size is of 64 bit that is 32bit checksum padded with 32bit zeros. The ideal values of both the checker mode are stored as constants; to be equated with the evaluated value for detecting any received data packet error.

7.7 Reconfigurable Architecture

Till now the focus is on the discussion related to the working principle and the functionality of the CRC-32 logic core. However, a particular CRC-32 polynomial is suited for a particular Hamming distance of a data traffic. For a change in data traffic the CRC-32

polynomial also need to be altered. This involves reconfiguration of the design. The developed model gives end users two types of reconfiguration choices, each with its own pros and cons. One is synthesis time reconfiguration that involves incremental or complete re-compilation of the entire design with custom polynomial and data bus width modification. But, the drawback of this type of up-gradation, requires the FPGA processing to be kept offline for a certain amount of time, and it can be an expensive solution for aggregated big data applications. The second type of reconfiguration architecture allows custom runtime modification of polynomial and parameter settings by running a simple C-code remotely in the RISC processor. The drawback of this design, it is incapable of bus width change. Had it involved bus width modification support in runtime reconfiguration design then it requires 700% more resources and that cannot be afforded in high density VLSI logic designs.

In Figures 7.5 and 7.6 the main buildings blocks are presented maintaining analogy with the mathematical representation shown in section 7.4.

7.7.1 Syndrome Table Synthesis

The computation of syndrome table from the CRC generator polynomial involves three steps, namely:

1. Generation of each row of the *Non-Systematic Generator matrix* from the shifted replica of the CRC polynomial. The row size of the matrix depends upon the data bus width (n) over which the CRC need to be calculated. From equation 7.1 it can be seen it requires n steps, for its generation.
2. Application of Gaussian elimination method to generate its Systematic form. The time complexity of this algorithm is $\frac{n(n+1)}{2}$
3. In either type of reconfiguration architecture, the objective is to load the Syndrome Mapper LUT (Look up table) with Syndrome Matrix value .

- (a) For *run-time reconfiguration* loading of the Syndrome Matrix value to the design, requires modification of the *syndrome mask*. The time complexity to reprogram the mask is $\frac{(n \times r)}{\text{write data bus width}}$.
- (b) For *compile time reconfiguration*, generation of the xor gates is done in accordance with the bit position in the Syndrome Table. The time complexity of its generation of the HDL code from Syndrome Table is $(n \times r)$.

To describe the limiting behavior of the synthesis algorithm for n tending to infinity, its big O notation takes the form $O(n^2)$. Table 7.3 draws an analogy between each VLSI logic blocks and its mathematical variable/constant as described in section 7.4.

Table 7.3: Analogy between logic blocks and mathematics

Logic Block	Linear Function / Constant
Initial Value	INIT
Input/Output Reflection	$\mathbb{B}$
Syndrome Mapper	$\mathbb{S}$
Zero Padding	ψ
XOR OUT	<i>Xorout</i>

7.7.2 DESIGN 1: Compile Time reconfiguration

The user's task is to set the CRC polynomial and its design parameters. The HDL design synthesizer automatically computes the syndrome table and generates the syndrome mapper based on the previously discussed steps. The design synthesizer then generates the pre and post operation blocks based on the given parameter settings. The advantage of this design as shown in Figure 7.5, that it does not over estimates the logic function, as it synthesizes only the blocks that are absolutely necessary based on the parameters given as input. But it carries one disadvantage that it cannot be modified during runtime, which is taken care by the design 2.

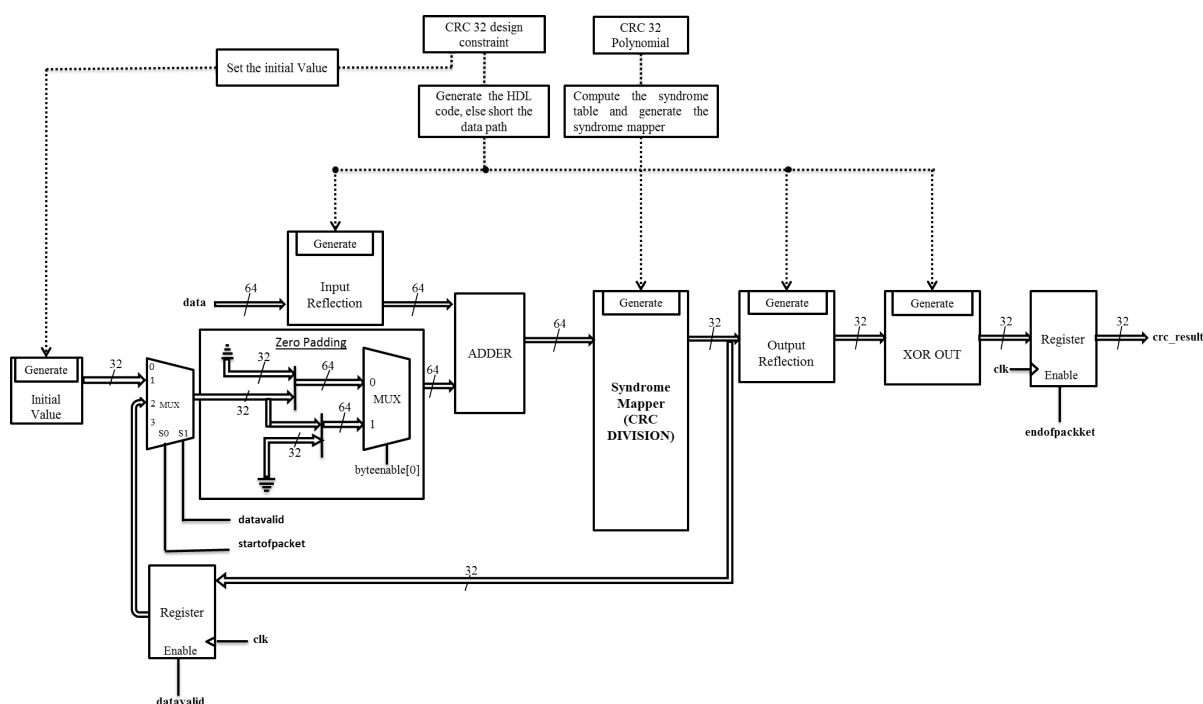


Figure 7.5: CRC-32 compile time reconfigurable architecture

7.7.3 DESIGN 2: Run-time reconfiguration

This design is the generated code of the design 1 with all blocks enabled for a specified bus width, which is 64 bit for Common Readout Unit (CRU) application. Each block is then wrapped with an interface block, such that it could able to communicate with the control unit. In the design the role of control unit is served by NIOS-II processor, which is a soft-core RISC processor provided by Altera. The initial value block, enabler block and Reconfigurable Syndrome Mapper uses Avalon Memory Mapper interface [113] for communication with the data master of the NIOS-II processor. The enabler block then supervises the activation of the Input Reflection, Output Reflection and XOROUT block.

For runtime modification of the CRC polynomial or its parameter setting, user simply needs to modify the software parameters in the application program running over the NIOS-II processor. As the application program detects any change in the design input parameters, it re-computes the entire syndrome table and accordingly it modifies the syndrome mask value. It communicates with Avalon standard bus to set the initial value

block, enabler block and loads the syndrome mask matrix. During this design modification the CRC computation is disrupted, which is only for a few millisecond. The advantage of this design is its ability to reconfigure during runtime, without reloading the entire firmware. But, it is not suitable for bus width modification.

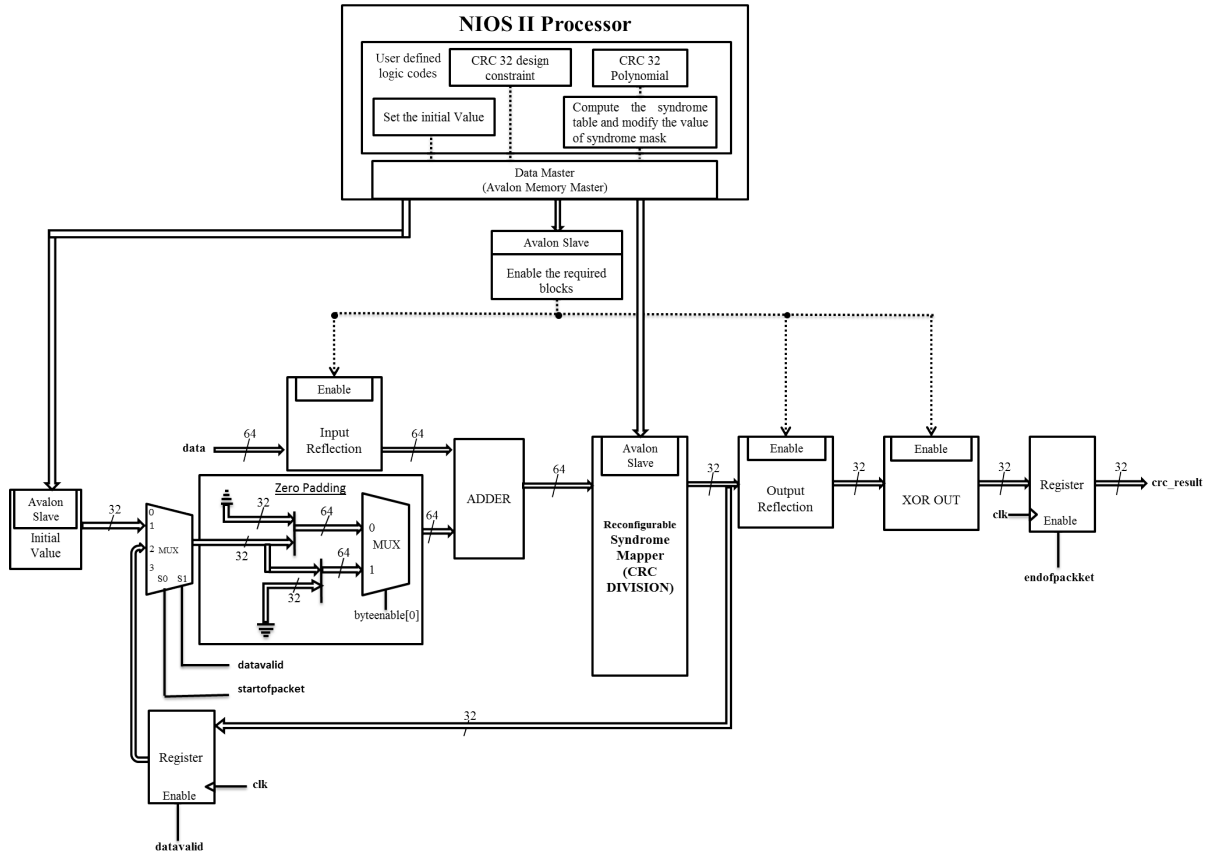


Figure 7.6: CRC-32 run-time reconfigurable architecture

7.8 Test Process

The study includes two test setups. In the first setup the performance study of the CRC core in different modes of connection is tested. In the second setup the performance study of the CRC core with variation in data bus width over different FPGA boards and its dependence on latency, power and resources is plotted. For homogeneity in test result, all through CRC32/Ethernet specification is followed.

7.8.1 Test Setup 1: Throughput study with various modes of interfacing

It is not always necessary for a user to use the CRC logic core directly. The design core can be interfaced in various connective topologies. For matching with the interface standard, user needs to develop its own wrapper. This setup illustrates and elaborates a detailed study of CRC logic core attached in different modes of connections, and it also gives a relative comparison of the speedup performance in different implementation. For uniformity in testing each component contains the same copy of the CRC logic core. Figure 7.9 shows how different CRC components are reading user data from the memory and writing back the CRC value to the memory. Each of the blocks are managed by a Nios II 32 bit soft core RISC architecture based processor provided by Altera. The Nios II/e core is used in this design for optimum use of resource utilization in FPGA. The blocks marked in red are the CRC design interfaces under test, and the blue ones are the Altera provided logic blocks. Apart from the interfacing logic, each CRC firmware wrapper also includes: a Timer Counter to track the number of clock cycles completed between *startofpacket* and *stopofpacket* signal assertion; a Data Counter to count the number of double word data(64 bit) passed for CRC computation. Both the counters are reset at the *startofpacket* signal. These signals are not shown in figures for maintaining representation clarity. The whole design is integrated using QSYS [113] system integration tool provided by Altera. When multiple masters contend for access to same slaves, QSYS automatically grants equally weighted round-robin based bus arbitration logic.

Interface Type 1: Master-Slave Mode

Avalon[®] Memory-Mapped (Avalon-MM) [113] interfaces are used for read and write operations between master and slave components in a memory-mapped system. In this case the data master is the Nios II/e processor and slave component is the CRC-32 logic

core. Figure 7.7 shows the interfacing of the read and write Avalon-MM signals with the CRC-32 logic core. Avalon-MM is synchronized with an associated clock interface. The maximum data bus line supported by Nios II/e RISC processor is 32 bit. This requires 32 bit data bus line be multiplexed into 64 bit bus line, by using combinatorial addressing logic.

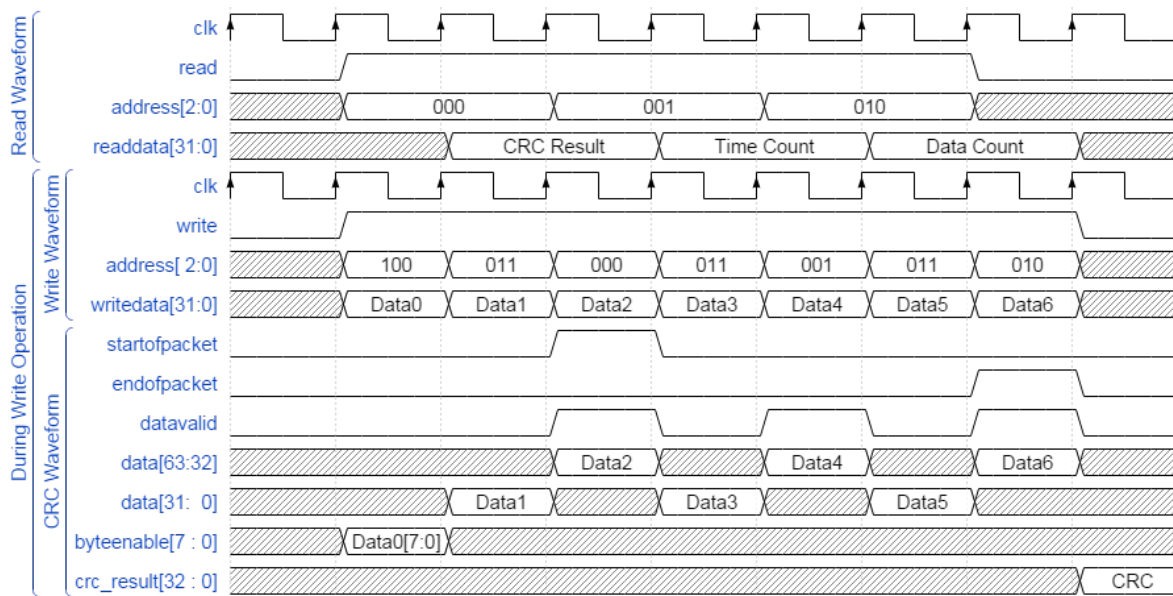


Figure 7.7: Showing the proposed CRC-32 architecture with Avalon MM connectivity

Interface Type 2: Custom Instruction Mode

Nios II custom instructions are custom defined logic blocks adjacent to the ALU (Arithmetic Logic Unit) in the NIOS II processor's datapath. It accelerates time critical algorithms by converting them to custom hardware logic blocks. It adds some user defined instruction set to already pre-existing bank of instruction set in NIOS II. For each data transfer it requires an instruction from the software interface. The advantage in this case that it is able to communicate to the processor the task to be conducted in a single instruction.

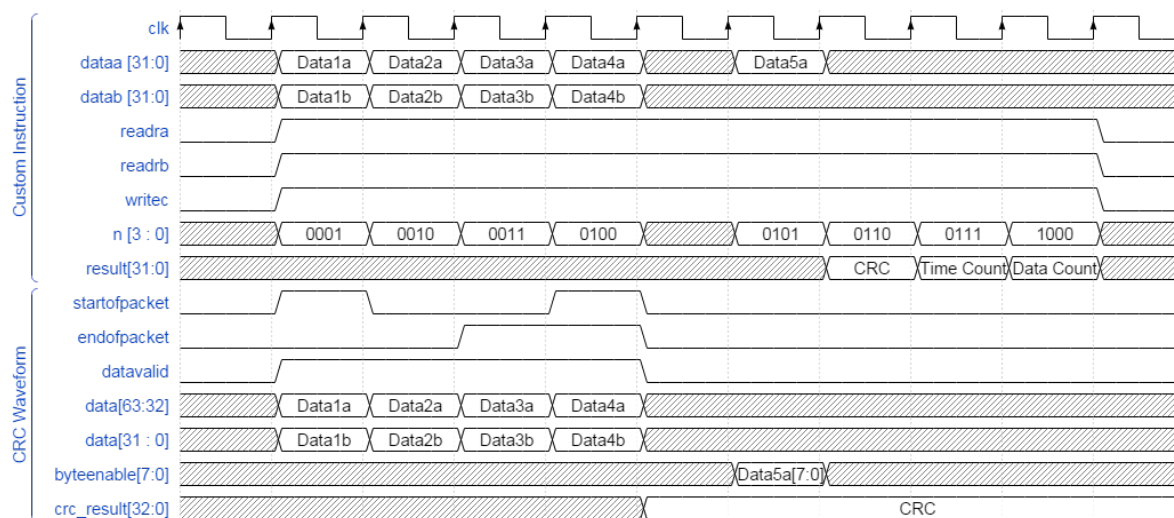


Figure 7.8: Showing the proposed CRC-32 architecture in custom instruction mode

Interface Type 3: Streaming Mode

Avalon Streaming (Avalon-ST) [113] interfaces are used for components that require high bandwidth, low latency, and unidirectional data transfer flow. This is particularly suitable for the CRC-32 component. An Avalon Streaming system is a system that has one or more Avalon-ST connections that transfer data from a source interface to a sink interface. Avalon-MM Read bus is kept to read the final outputs: CRC Result, Timer Count and Data Count values. While Avalon-MM Write is used to write data on to the *byteenable* bus of CRC logic core.

An Avalon-ST signaling standard matches closely with the CRC logic core standards. All transfers occur synchronous to the rising edge of the associated clock signal. In this design NIOS II/e the source and CRC logic core is the sink.

Scatter-Gather Direct Memory Access (SG-DMA) controller core is used along with this design for high-speed data transfer with minimum processor overload. The SG-DMA controller core achieves a significant performance enhancement over the popular DMA controller cores. The SG-DMA controller core reads a series of descriptors from the memory as described by the Nios processor for the required transactions and performs all of the transfers without further intervention from the CPU. This significantly improves

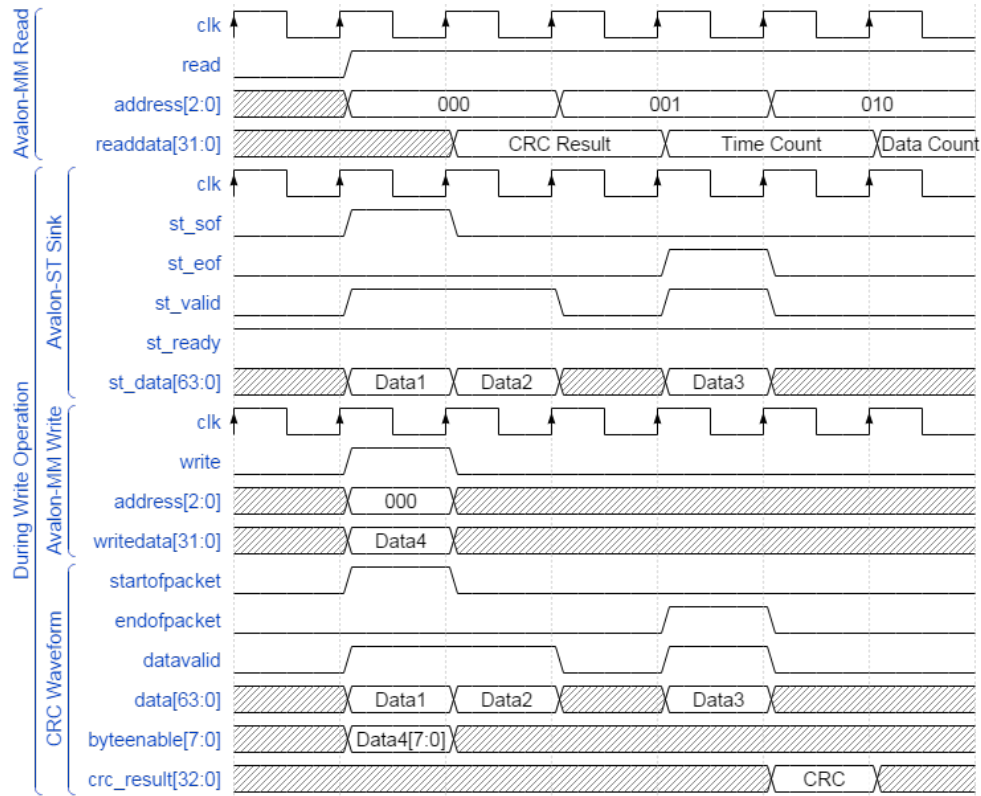


Figure 7.10: Showing the proposed CRC-32 architecture with Avalon ST Connectivity

for all possible combination of 8 bit message stored in the memory, when it receives a message it parses it byte-wise and uses the CRC value from the corresponding position at the lookup table. This algorithm during operation accesses the memory $2 * CRC_size + CRC_size/8 + 1$ times, for fetching 64 bit of the message in blocks of 32 bit, matching with 8 bit LUT values and loading the checksum value to the memory. The developed CRC algorithm is based on syndrome table. It stores the syndrome table in the memory. During computation, the particular value of the checksum bit position depends upon the corresponding row of the syndrome table which instructs the type of modulo-2 operation to be done on the 64 bit parsed message bit. The algorithm during computation accesses the memory $4 * CRC_size + 1$ size times; it fetches the 64 bit message in blocks of 32 bit from the memory, for each bit of CRC computation it fetches the particular row of the syndrome table from the memory and finally loads the value in the memory.

7.8.2 Test Setup 2: Resource utilization study with bus width variation

This setup is intended for performance study of CRC32 design arbitrarily chosen for Ethernet specification with variation in data bus width over different FPGA boards. The results tabulated in the Tables[7.7] and [7.8] , list the comparative analysis of the design for CRC32/Ethernet logic core over three kinds of Altera FPGA boards on the basis of timing, power and logic resource analysis with data bus width variation.

Same set of data vectors with same clock frequency is used to make sure a standardization in the evaluation of the performance of the algorithm in various test boards with data bus width variation. The results will vary with modification in the test vectors. Here, it is important to note that the graphs are indicative of relative qualitative measurement of the performance, for absolute measurement it is completely dependent upon the users test vectors.

7.9 Performance Analysis

7.9.1 For Test Setup 1 : Throughput study with various modes of interfacing

Table 7.4 shows the processing time of different implementation style. It can clearly be inferred from the tabulated result in Table 7.4 that as the number of hardware-software interaction decreases, throughput increases.

In the field of programming, speed up Factor (SF) is an important metric for relative performance improvement measurement when executing a task. It can be defined by throughput and latency. The latency is constant because the same CRC logic core is used, so measurement is done in terms of throughput. Table 7.5 shows a speed up ratio of all hardware implementation. The maximum achievable throughput is possible when the

Table 7.4: Performance of CRC-32 implementation in different architecture topology

Architecture Type	Processing Time	Processing Throughput
Realtime	0.0881 ms	6400.00 Mbps
SGDMA	0.4660 ms	1123.87 Mbps
Custom Instruction	126.3962 ms	4.14 Mbps
Avalon Slave	127.9012 ms	4.09 Mbps
Software Code	11500.7554 ms	0.05 Mbps

CRC design is directly connected to the immediate logical/protocol layer, like in 10 Gbps Ethernet. This type of architectural implementation is known as Run-time hardware architecture.

Maximum Achievable Processing Throughput for a 100 MHz clock freq = 6400 Mbps

Speed Up Factor (SF) is the improvement in processing throughput

$$SF = \frac{\text{Achievable Processing Throughput}}{\text{Processing Throughput}}$$

Table 7.5: Speed up ratio for all hardware implementation

	DMA	Avalon Slave	Custom Instruction	Software code
Realtime	5.69	1545.89	1564.79	128000.00
DMA		271.24	274.47	24677.25
Custom Instruction			1.01	90.98
Avalon Slave				89.91

Software version of CRC is compiler dependent and on the number of load and branch RISC instruction set. Sequential processing is not suitable for the developed algorithm that can clearly be deduced from Table 7.6. It happens so because the syndrome table based algorithm requires more number of memory accesses. It is important to note here that memory access take more time than data processing. The algorithm is specifically designed for parallelized architecture, where it assumes the loading time of syndrome table vector is either negligible or preloaded in the design.

Table 7.6: Performance of CRC-32 in software implementation

Algorithm Type	Processing Time
Lookup table based	3782.4806 ms
Division method	5640.4842 ms
Syndrome table based	11500.7554 ms

7.9.2 For Test Setup 2: Resource utilization study with bus width variation

The plotted graphs in Figures 7.11, 7.12 and 7.17 show latency, power and resource variation vs bus width modification.

Timing Analysis

Minimization of latency is done with removal of pipeline registers to operate within 1 clock cycle, but it involves increase in combinatorial delay between registers. The maximum frequency of operation is limited by the maximum delay involved in the longest concurrent path. It is very important to abide by the time critical portion, for stable output response.

Total combinatorial logic (T_{logic}) is theoretically a summation of individual logic blocks

$$T_{logic} = T_{Input\ Reflection} + T_{Adder} \quad (7.19)$$

$$+ T_{Syndrome\ Mapper} + T_{Output\ Reflection} + T_{Xorout} \quad (7.20)$$

Among all the logical blocks, Syndrome Mapper involves the lengthiest combinatorial logic. It can safely be approximated that

$$T_{logic} \approx T_{Syndrome\ Mapper} \quad (7.21)$$

The maximum allowable frequency[116] can be defined by equation 7.22.

$$F_{max} = \frac{1}{T_{clk-q} + T_{logic} + T_{timing} + T_{setup} + T_{skew}} \quad (7.22)$$

where F_{max} is maximum allowable frequency for clock; T_{clk-q} is time from clock arrival until data arrives; T_{logic} is propagation delay through logic between flip-flops; $T_{routing}$ is routing delay between flip-flops; T_{setup} is minimum time data must arrive

Table 7.7: Timing Analysis

Board No.	Fast Timing Model Delay	F_{max} = 1/(Minimum Delay)
CYCLONE III (EP3C120F780I7)	4.502 ns	181.62 MHz
CYCLONE IV E (EP4CE115F29I8L)	6.639 ns	194.29 MHz
STRATIX V (5SGSED6N1F45I2)	2.374 ns	216.78 MHz

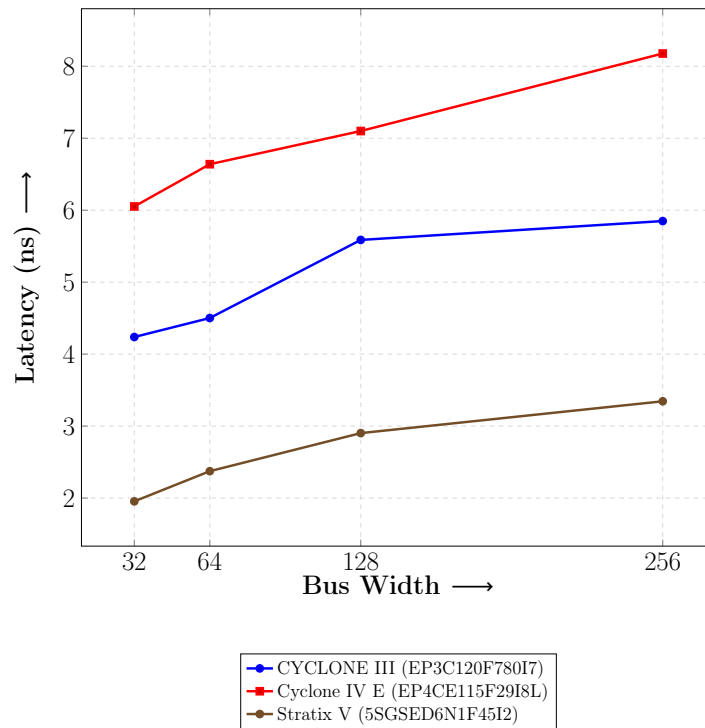


Figure 7.11: Latency vs CRC bus Variation

For guaranteed silicon performance in FPGA, different Timing Models [117] are used for analysis. FPGA able to operate at wide range of temperature variations. Commercial temperature varies from $0^{\circ}C$ to $85^{\circ}C$. Fast timing model is used for latency calculation

as it gives the minimum threshold. The timing analysis is conducted using post place and route simulation/Gate level simulation for Cyclone series FPGA and TimeQuest timing analyzer tool for Stratix series. Figure 7.11 clearly shows that the latency increases as data bus width increases.

Power Analysis

FPGA is a CMOS based technology, where dynamic power consumption is related to charging and discharging of parasitic capacitances on gates and metallic contacts. The energy dissipation is linked with current dissipation as voltage is fixed in a FPGA design [116]. So,

$$I = V \times C \times f \quad (7.23)$$

where I is total current, V is voltage, C is capacitance, and f is frequency. In it C and f can be modified. The capacitances C is directly related to the toggling rate of the gates and the distance of the routes connecting the gates. The frequency f is related with the clock frequency.

The power estimation tool provided by Altera is used for calculation of power consumption. Figure 7.12 shows at constant frequency of 100MHz, with increase in data width power consumption increases steeply. It is obvious as number of logic gate involved also increases with bus width. While Figures 7.13,7.14,7.15,7.16 show when operating frequency increases power consumption also increases, which clearly tally with the nature of equation 7.23.

Table 7.8: Power Analysis

Board	State	Block Type	Total Thermal Power	Dynamic Power	Routing Thermal Power	Block Average Toggle Rate (millions of transitions/sec)
CYCLONE III(EP3C120F780I7)	Fully ACTIVE	Combinatorial Cell	8.35 mW	3.23 mW	5.12 mW	91.720
		Clock Control Block	0.84 mW	0.00 mW	0.84 mW	10.123
		Register Cell	0.65 mW	0.39 mW	0.26 mW	5.347
CYCLONE IV E(EP4CE115F29I8L)	Fully ACTIVE	Combinatorial Cell	5.76 mW	2.21 mW	3.55 mW	95.927
		Clock Control Block	0.44 mW	0.00 mW	0.44 mW	10.123
		Register Cell	0.50 mW	0.24 mW	0.26 mW	5.347
STRATIX V(5SGSED6N1F45I2)	Fully ACTIVE	Combinatorial Cell	8.83 mW	4.06 mW	4.78 mW	122.498
		Clock Control Block	0.52 mW	0.00 mW	0.52 mW	29.491
		Register Cell	0.40 mW	0.12 mW	0.28 mW	5.601

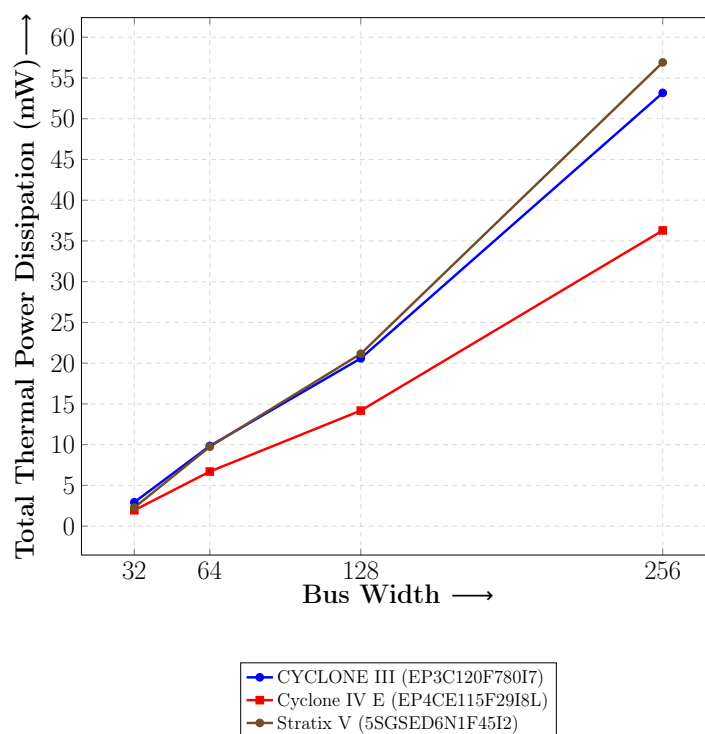


Figure 7.12: Power vs CRC bus Width Variation

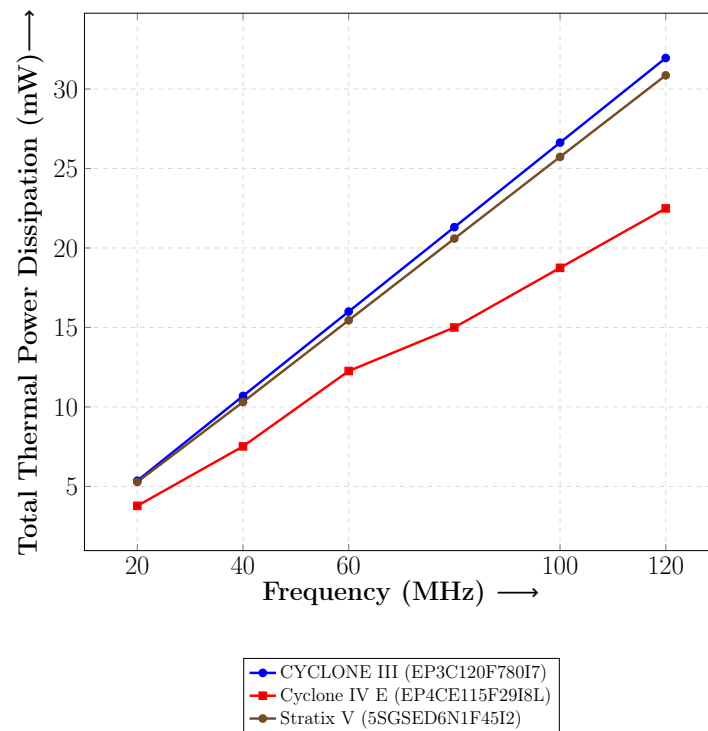


Figure 7.13: Power vs Frequency for 32 bit data bus

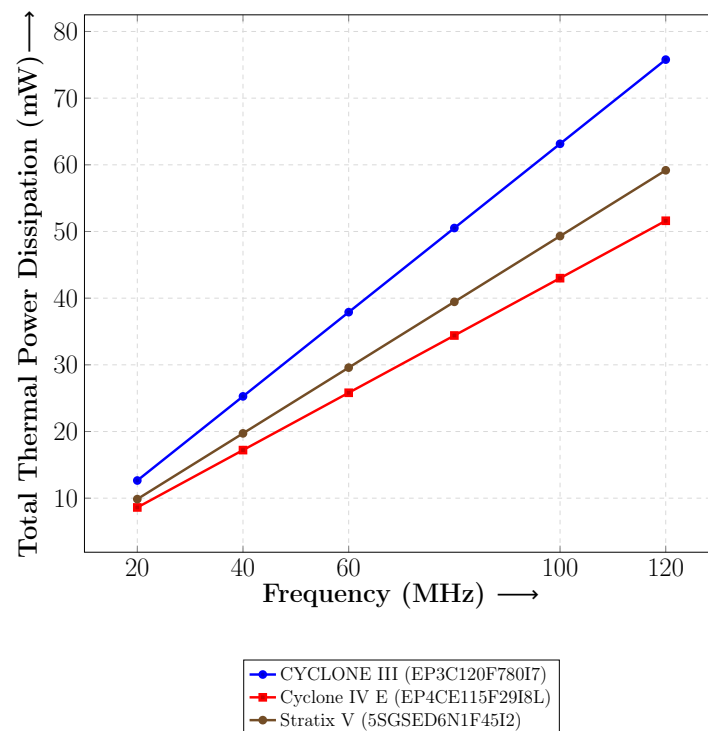


Figure 7.14: Power vs Frequency for 64 bit data bus

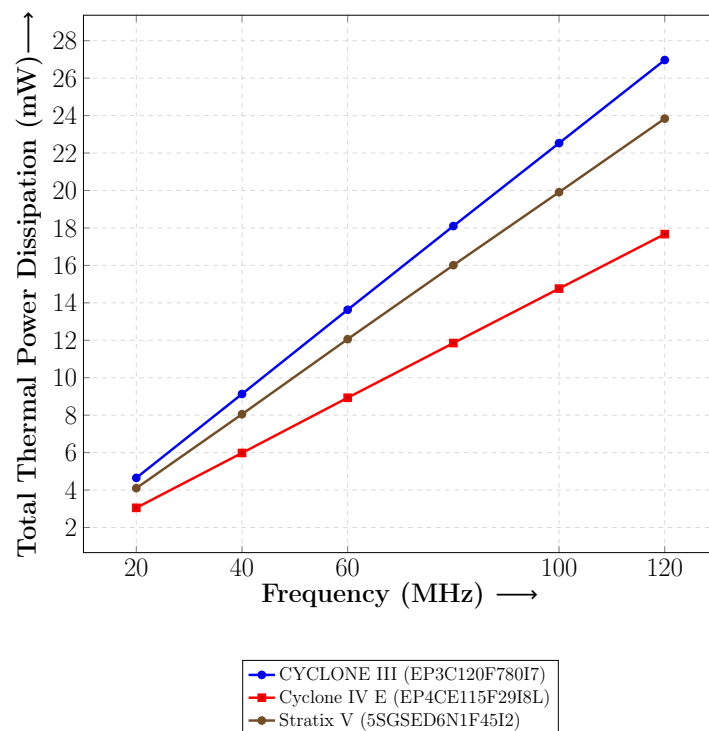


Figure 7.15: Power vs Frequency for 128 bit data bus

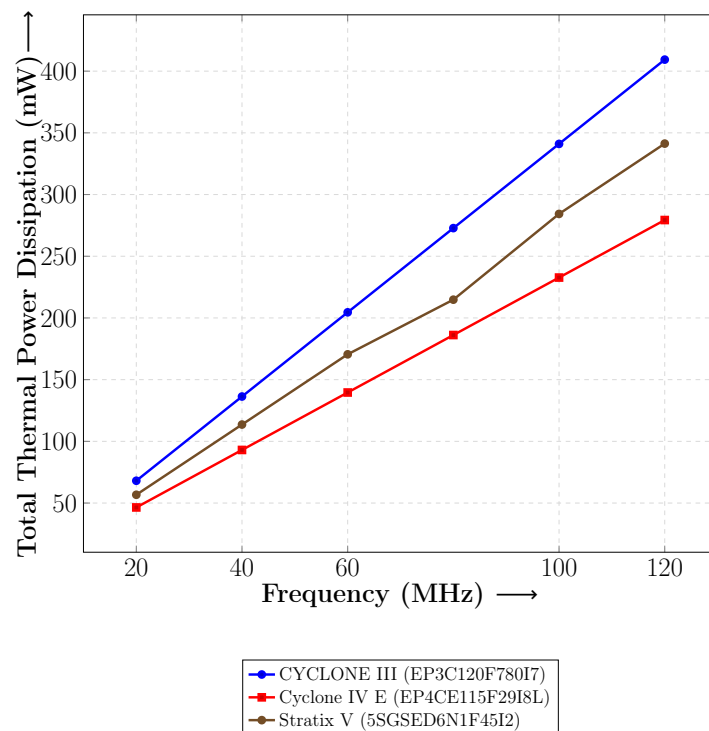


Figure 7.16: Power vs Frequency for 256 bit data bus

Logic resource utilization

Device densities and corresponding design sizes have become very large (over millions of gates), and newer methodologies have been developed to assist the placement tools. Place and routing tools helps in laying out logic elements from a sea of gates. Floorplanning is used by designers who are having difficulty in setting the critical paths. The proposed design includes many time critical path for operation at very high frequency. Proper physical constraints must annotated into the fitter tool to concentrate the logic resources and optimizing the overall routing resources and improving timing performance. Figure 7.18 shows that the logic resources is closely packed and placed near the global clock. It also shows a magnified view of the small portion of the used register bank.

It is interesting to notice here that the slope of the graph is always positive. This means that the performance of the module is consistently compromised with increase in data bus width. Now it is upto user's discretion to choose the particular bus width after optimizing the design against latency, power and resource variation. For example, there are certain design requirements which requires extra wide data bus, but it will operate at low clock frequency. Now this trade-off is clearly reflected in equations (7.22) and (7.23). But as the bus width increases the number of resource utilization also increases as shown in Figure 7.17. Correspondingly requiring more power as plotted in figure 7.12. But power consumption might also be decreased with lowering in operating clock frequency as shown in Figures 7.13,7.14,7.15 and 7.16.

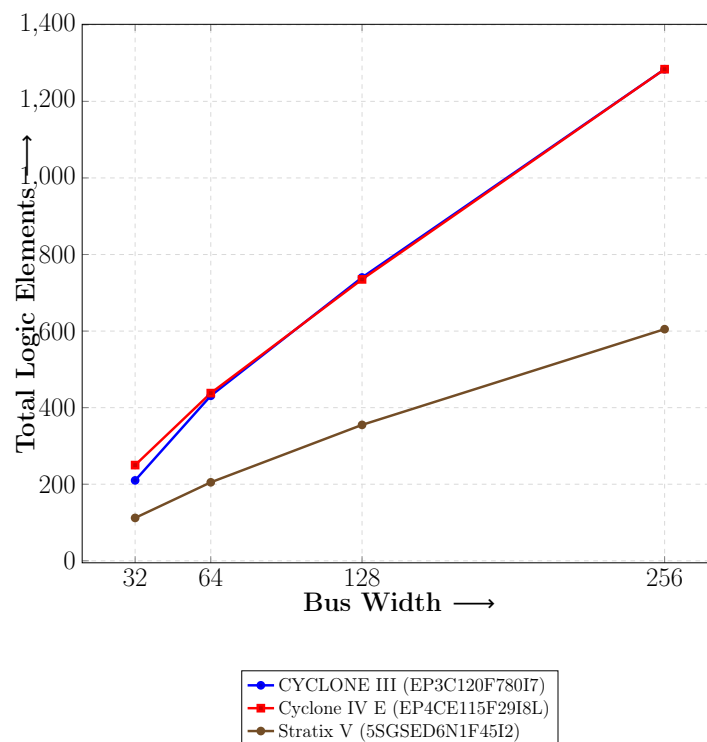


Figure 7.17: Resources vs CRC bus Variation

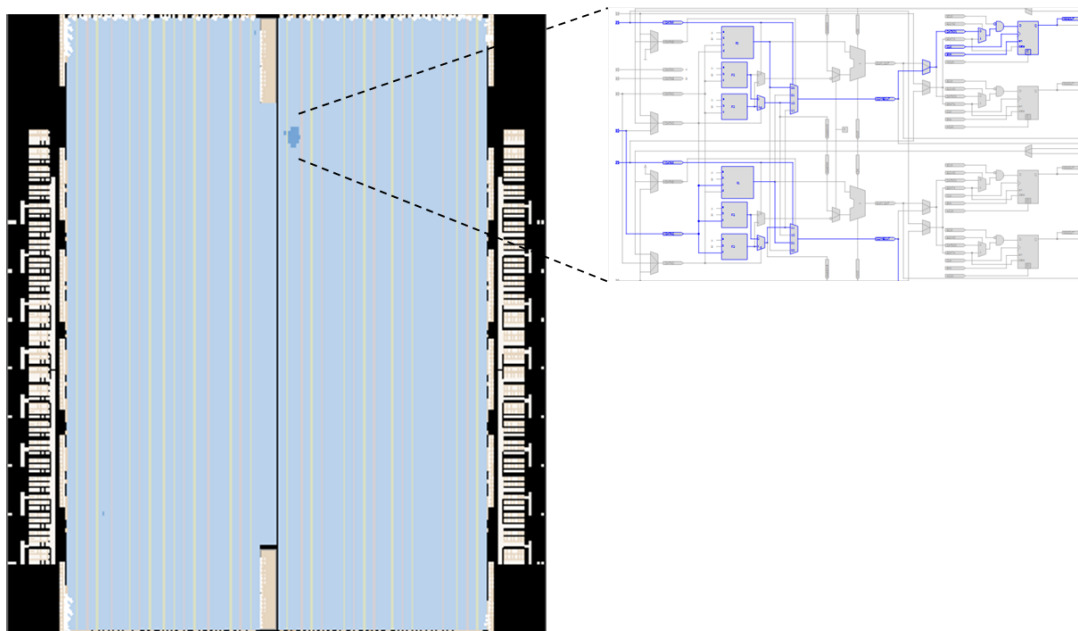


Figure 7.18: Showing chip planner view of used resources ($< 1\%$) in Stratix V FPGA

7.10 Design Comparison

The design approach presented in this chapter is a special case of parallel CRC or fast CRC implementation. Pipelined latency is used to increase throughput [118]. However, the design avoids pipelined architecture to maintain the latency of 1 clock cycle, as needed in low latency protocol design. So, in this case, maximum throughput is limited by critical path delay as detailed in subsection 7.9.2. In Table 7.9 the performance of the design is compared against other relevant CRC research work.

Table 7.9: Comparison of High-Speed CRC-32 Logic Implementation

References	Technology Used	Solution Type	No. of Input Parallel Bits	Reconfigurable Type	Clock Rate	Throughput
Proposed Method	28nm FPGA Technology	Syndrome Table	256	Both Compile-Time and Run-Time CRC type modification; Compile Time i/p length change	150 MHz	38.4 Gbps
Lin et. al. [119]	0.18 μ m CMOS process	State-Space	32	Not Supported	333 MHz	10.66 Gbps
Ming et. al.[120]	0.6 μ m Silicon process	Lookahead Technique	8	Lookahead Function Order Selection	25 MHz	200 Mbps
Ji et. al. [121]	Configurable Processor	Galois Field multiplication and accumulation	128	Design Time i/p length change	200 MHz	8.5 Gbps (3 Clk cycle)
Walma [118]	65 nm FPGA Technology	Pipelined Algorithm	256	Scalable Architecture	345 MHz	88 Gbps (8 Clk Cycle)

7.11 Future Scope

Test setup 1 gives the possibility of different implementation style open for the user. It will be interesting to do further research on analyzing the presented algorithm performance on other advance parallelized processor architecture like GPUs. From Test setup 2 it can clearly be inferred that the CRC design of particular bus width depends on the parameters preferred by the user's choice of area, power and latency. It is a multi-optimization problem to satisfy all the design constraints set by the user. Future research can be done in this direction to develop an optimization algorithm to choose ideal design parameters, and automatically suggesting the user with the best choice.

7.12 Conclusion

The purpose of this chapter is to discuss the development of a CRC design architecture that is fully reconfigurable, got very high throughput and has a single clock latency with data pipelining support. The work presents widely popular CRC algorithm in a VLSI design friendly manner. Throughout the chapter CRC-32 Ethernet standard is used as a reference to present the concepts. It can easily be generalized to suit the designer's requirement. The test cases show the performances of different interfacing scenarios and shows how to maximize the performance with proper interface solution.

Chapter 8

A Secure Error Resilient High-speed Link Standard

8.1 Introduction

In recent years critical infrastructures faced various security exploitations. Interfaces that are local to ALICE detectors like GBT and TTC-PON communicates data packets without encryption, making the readout data stream entirely vulnerable to security exploitations. Although the DAQ system is protected by an active firewall, however, if there is a security breach the underlying electronic sub-systems can easily be compromised or manipulated. Hence, raises the interest for designing the interface links with security and error resilience. Typically there always exists a trade-off between error detection capability and channel bandwidth. A design is proposed that is capable of triple bit error correction to protect against radiation-induced soft-errors and prevention of eavesdropping of classified information by encrypting the data packet using Advanced Encryption Standard (AES) while transmitting over the optical medium.

The design architecture is elaborated in Section 8.2, with proper transceiver transmitter data-path, receiver data-path, and clocking connection scheme. Sections 8.3-8.9

discuss in details the individual composite blocks of the architecture, along with well illustrated figures. The novelty claimed for this design is then compared with the CERN standard protocol in section 8.10, to justify the added benefit of this design. Finally, the different metrics used for performance estimation is reported in section 8.11.

8.2 System Architecture

The architecture as shown in fig. 8.1 can be broadly grouped into four main categories namely: forward error correction (FEC) Coding, Data Encryption Coding, Deterministic Clock Domain Crossing and Frame Synchronization Block.

The FEC block consists of 3 sub-blocks: Self-Synchronizing Scrambler, Golay FEC Coder, and Helical Interleaver. Due to the perfect nature of Golay code it could able to correct any 3-bit errors and detect any combination of 7-bit errors or odd bit errors [122]. For spreading of burst errors over correctable error frame, Helical Interleaver block is employed after it [123]. It adds the ability of 15 bit consecutive error correction capability to the FEC coder. The Scrambler block is used to randomize the data pattern for proper distribution of ‘1’ and ‘0’ in the data stream. It is used for DC-balancing of the transmitted data to facilitate clock recovery. But, it has a drawback of error multiplication if it used just before data serialization reducing the FEC performance. So, this block precedes all the coder block for optimal operation of the error correcting block [124].

Data encryption block applies a powerful AES encryption algorithm with 256 bit cipher [125] for encoding messages or information in such a way that only authorized parties can access it. Though, in principle it does not prevent interception, but it is strong enough to deny the message content to the interceptor.

The design is addressed to carry multiple varieties of data, which includes timing and trigger data signals, detector data frames and control signals. Of the three data types, timing and trigger signal data is latency sensitive, so latency optimized gearbox is used

for clock domain crossing.

Finally, frame synchronization block deals with header - trailer addition/deletion, and proper alignment of frame based on header bits.

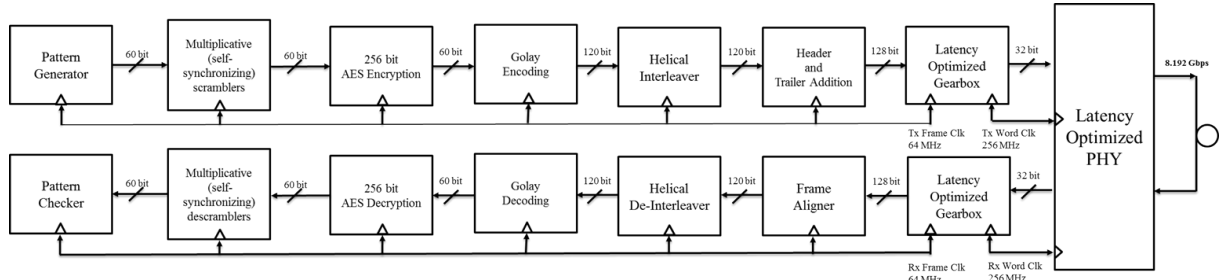


Figure 8.1: System Architecture Block Diagram

Apart from the physical coding blocks, Native PHY forms a very important part of the sub-system [37]. It controls how the serialization of the data needs to be done and also data rate control (single or double). The Advanced Transmit PLL (ATX PLL) is used to drive the transceiver. It is very important that the reference clock fed to the transceiver is jitter free. For this purpose CDCE62005 [126] jitter cleaner PLL is used to fed 256 MHz frequency.

Since detector data are not readily available, so it is emulated using pattern generator and pattern checker for testing of the firmware. Pseudo random binary sequence is used for testing the efficiency of the firmware. The developed design is entirely based on latest state of the art 20nm FPGA chip technology Arria[®] 10 available from Altera [37].

8.3 Scrambler

Scramblers are a class of substitution ciphers [127]. In the design multiplicative self-synchronizing scrambler is applied to avoid long strings of 0's and 1's, which are responsible for DC wander and synchronization problems. Although in self-synchronization data delay or loss may be problematic. But its ability to fast resynchronize without any seed value, far outweighs its drawback. The polynomial used for its computation is shown in

eq. 8.1. It does both temporal and spatial scrambling of data bits to remove long run of 1's and 0's in both domain.

$$y(t)[i] = x(t-1)[i] + x(t-1)[i+2] + x(t)[i] \quad (8.1)$$

where i shows the bit position of the i^{th} bit in a vector size of 60 bit, while $x(t-1)$ represents the content of 1 clock cycle delayed buffered copy of data vector

8.4 Advanced Encryption Standard (AES)

In 2001 U.S. National Institute of Standards and Technology (NIST) announced Advanced Encryption Standard AES [125] as a specification for the encryption of electronic data. In recent dates AES is the most widely used symmetric block cipher in the crypto world which finds applications in banking sector, defense, Internet security protocols (Secure Socket Layer, Transport layer Security) and in other various domains. The variable key size, different modes, and efficient security related performances issues are significant reasons of its widely acceptance.

8.4.1 Algorithm Description

The algorithm has three parts: Cipher, Inverse Cipher and Key Expansion. Cipher converts plain text to cipher text. Key Expansion generates a key schedule, which is used in the Cipher and the Inverse Cipher procedure. Cipher and Inverse Cipher consists of specific number of rounds. For the AES algorithm, the number of rounds to be performed during the execution of the algorithm is dependent on the key length [125] which is 14 for 256 bit key size. AES operates on a 4x4 array of bytes (referred to as “state”). The algorithm consists of four different simple operations. These operations are:

- Sub Bytes

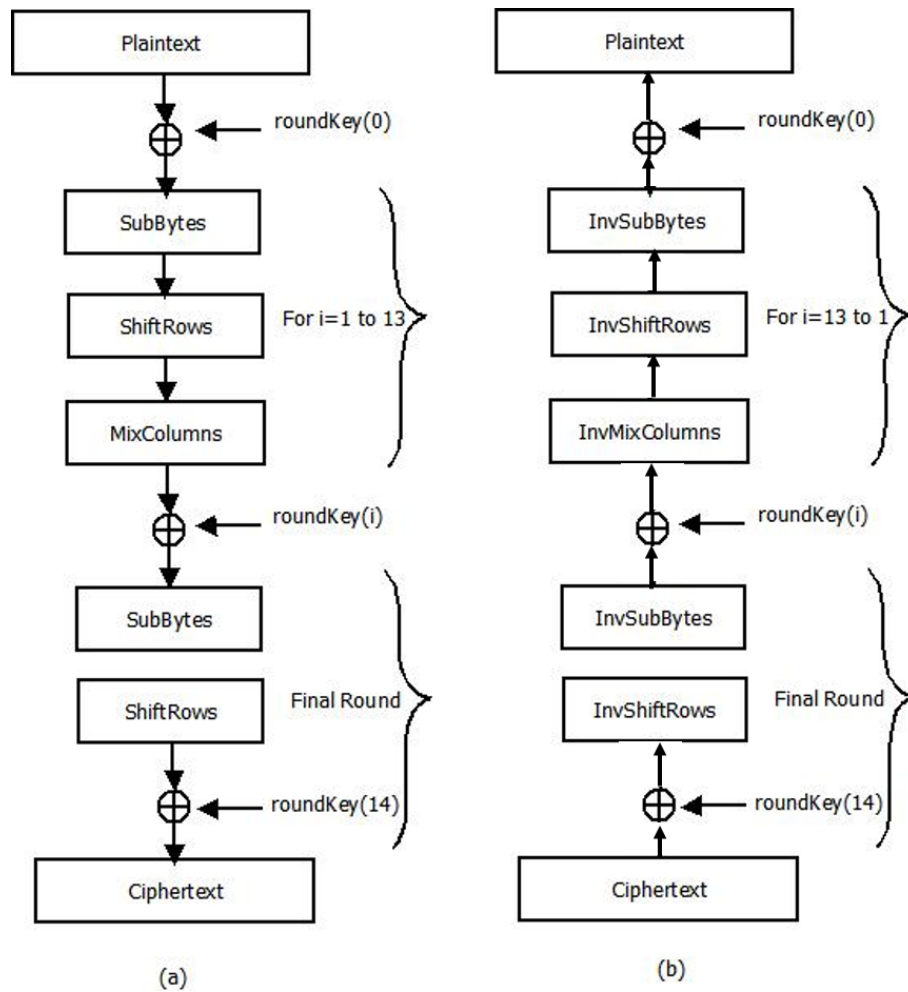


Figure 8.2: Showing AES Algorithm Flow

- Shift Rows
- Mix Columns
- Add Round Key

Encryption Process

The Encryption and decryption process consist of a number of different transformations applied consecutively over the data, in a fixed number of rounds. The number of rounds depends on the length of the key. For a key length of 256 bits, the number of iterations required are 10 ($Nr = 14$). As shown in fig. 8.2(a), each of the first $Nr - 1$ rounds consists

of 4 Transformations: SubBytes(), ShiftRows(), MixColumns() and AddRoundKey().

Sub Bytes Transformation It is a non-linear substitution of bytes where each byte $a_{i,j}$ in the state matrix is replaced with a SubByte $S(a_{i,j})$ using an 8-bit substitution box (Sbox). This invertible S-box is constructed by first taking the multiplicative inverse in the finite field $\text{GF}(2^8)$ with irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$. The element 00 is mapped to itself. Then affine transformation is applied (over $\text{GF}(2)$).

Shift Rows Transformation It shifts the rows of the state over the different offsets in a cyclic fashion. The operation is almost selfsame in the decryption process except for the fact that the shifting offsets have different values.

Mix Columns Transformation The MixColumns process takes four bytes as input and outputs four bytes, where each input byte affects all four output bytes. The transformation operates on the state column-by-column, treating each column as a four-term polynomial. The columns are considered as polynomials over $\text{GF}(2^8)$ and multiplied by modulo $x^4 + 1$ with a fixed polynomial $a(x) = 03x^3 + 01x^2 + 01x + 02$.

Add Round Key Transformation A simple bitwise XOR operation is used in Add Round Key Transformation. Each Round Key consists of Nb words from the key expansion. Those Nb words are each added into the columns of the state. Key Addition is identical for the decryption process.

Key Expansion As a product of the previous round key a 4-word (128-bit) array is generated in each round key. A series of S-Box lookups for each 32-bit word of the key. The Key schedule Expansion generates a total of $Nb * (Nr + 1)$ words.

Decryption Process

The decryption process is identical with encryption process. It occurs simply in reverse order, taking the 256-bit block of cipher text as input and converts it to plaintext by the application of the inverse of the four operations. AddRoundKey is the same for both encryption and decryption. However, the three other functions have inverses used in the decryption process: Inverse SubBytes, Inverse ShiftRows, and Inverse MixColumns. All the transformations applied in the encryption process are inversely applied to this process. Hence, the last round values of both the data and key are first round inputs for the decryption process and follows in decreasing order as shown fig. 8.2(b).

8.5 Golay Forward Error Correction Coder

The Golay code is a perfect or densely packed code and has attracted the interests of mathematicians for centuries. In the design extended Golay code (24,12,8) with extra parity check bit is used. It has a hamming distance of 8 between two code words, and having error correction capability of 3 bits. It can detect any combination of 7 bit errors or odd bit errors. The Golay FEC consists of two parts; encoder on the transmitter side and decoder on the receiver side. Encoding of golay is based on Look-UpTable (LUT) based scanning [128]. LUT is constructed based on the generator matrix as shown in fig.

8.2.

$$\begin{aligned}
 P = & \begin{bmatrix} 0x7FF \\ 0xEE2 \\ 0xDC5 \\ 0xB8B \\ 0xF16 \\ 0xE2D \\ 0xC5B \\ 0x8B7 \\ 0x96E \\ 0xADC \\ 0xDB8 \\ 0xB71 \end{bmatrix} & G = [I_{12 \times 12} | P] & H = [P | I_{12 \times 12}] & (8.2)
 \end{aligned}$$

where P is parity matrix, G is generator matrix and H is Parity Check Matrix.

The arithmetic decoding of Golay is based on works of Wicker (1995) [129]. The Syndrome of the received data vector is computed, by $S = rH^T$. Then Hamming Weight of S is measured and compared for further iteration in error vector detection. Elaborate details of the algorithm is available in the book by Morelos-Zaragoza (2006) [128].

In the design *five* blocks of Golay coder for independent data encoding/decoding is used and then multiplexing it before forwarding it to the next block.

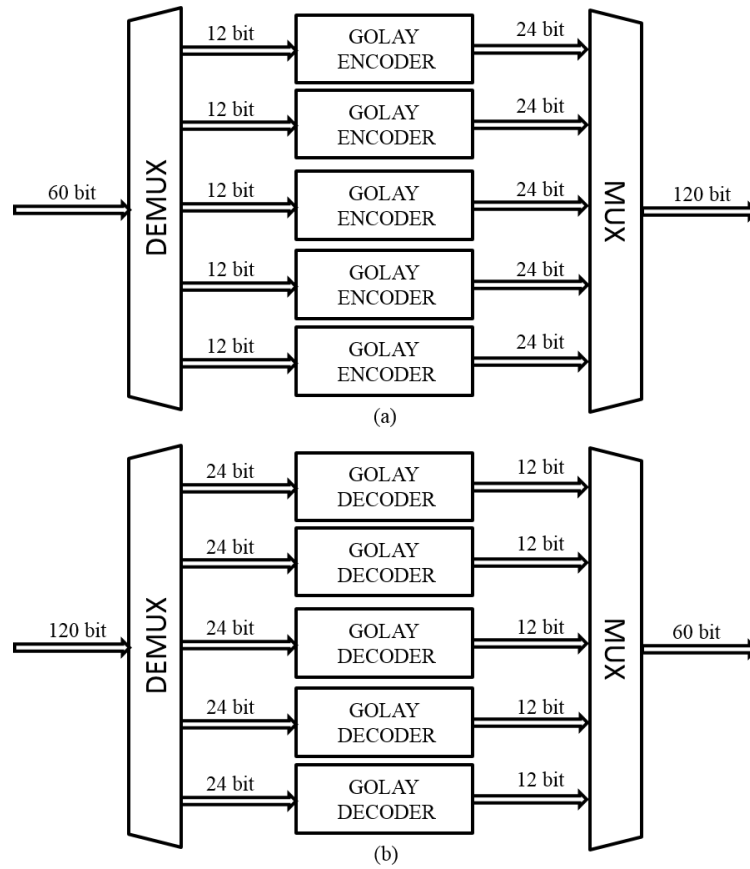


Figure 8.3: Showing multiple Golay encoder/decoder operating in-situ

8.6 Helical Interleaver

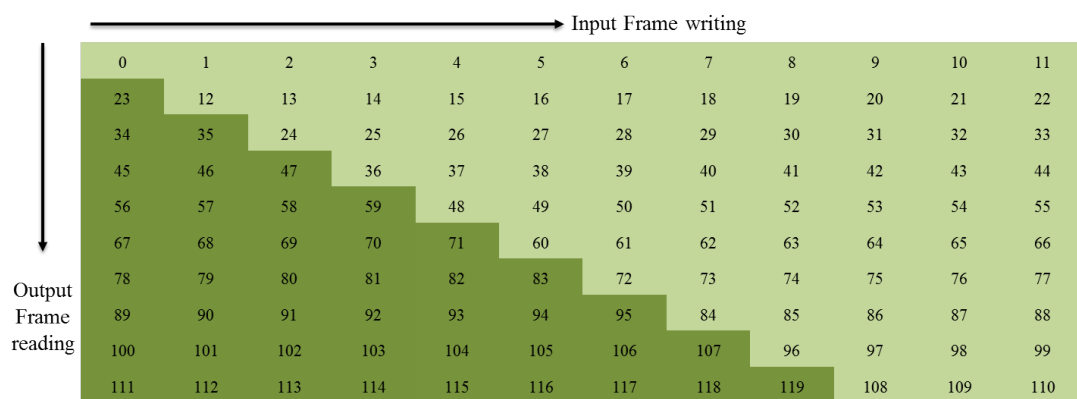


Figure 8.4: Showing Helical Interleaving of 120 bit

The Helical Interleaver block change the order of arrangement of the input data, by placing them in an array in a helical fashion and then send rows of the array at the

output port. If C is the Number of columns in the helical array input, then the array has C columns and may have unlimited rows. If N is the Group size, then the block can accept an input of length $C \times N$ data at each instant and partitions the input into consecutive groups of N symbols. The block places the k th group in the array along column $(k \bmod C)$. Here the placement is helical as the reduction is modulo of C . The first symbol in the k th group is in row $1 + (k-1)xs$, where s is the Helical array step size.

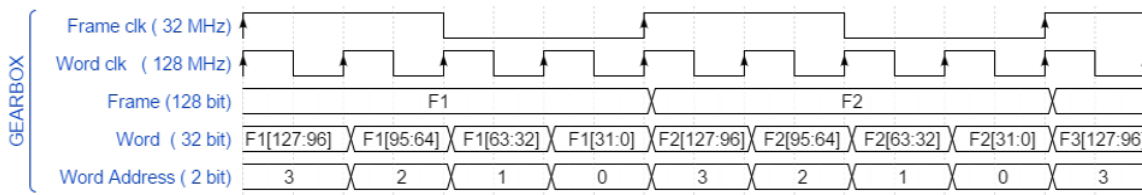


Figure 8.5: Latency optimized gearbox

8.7 Frame Synchronization Block

8.7.1 Header and Trailer Addition Block

HEADER	DATA	TRAILER
1010	120 bit	4 bit

Figure 8.6: Data Frame format

A 4-bit header field is appended at the beginning of each frame. This field is required to synchronize the data stream at the frame level. To enable reliable header recognition it is chosen to be DC-balanced. In this case bit pattern ‘1010’ is used. Smaller header pattern is chosen as large words are more prone to errors.

The 4-bit trailer field uses Walsh code [130] sequence to represent the data type signature. Nearest Hamming distance search is used to correct errors in the trailer. Four types of data transmission are marked in the trailer, as shown in Table 8.1

Table 8.1: Trailer pattern meaning

4 bit Code	Data Type
1001	Valid detector Data
1100	Control signal
1010	Trigger data
1111	Bad detector data

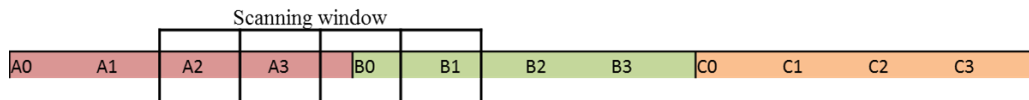


Figure 8.7: Scanning window searching for FAW

8.7.2 Rx Frame Aligner Block

The frame boundaries are marked by inserting Frame Alignment Word (FAW) or header. Proper frame alignment is necessary before the system is ready to receive data [131]. Loss of frame alignment could result in catastrophic loss of message, control and trigger information. So, a robust frame alignment algorithm is designed to have correct alignment detection, misalignment detection and fast searching of FAW.

The FAW may be mutilated by transmission errors due to its proximity to radiation zone, in such cases a tolerance band is necessary before switching the state from frame lock to frame search mode.

Table 8.2: Frame Aligner State Machine

FAW	Number of Frame checks before changing states			
	Lock to Search		Search to Lock	
1010	24	70% tolerance	12	90% tolerance

8.8 Latency Optimized Gearbox

Common approaches for Clock Domain Crossing (CDC) requires the use of asynchronous FIFO or Dual Port RAM. But this comes up with a penalty of clock cycle loss, which cannot be afford during timing information transmission. So, for deterministic clock

domain crossing a register based approach is used for guaranteed latency. These type of gearbox is termed as latency optimized gearbox [33]. Clock calibration of *frame clk* and *word clk* is must for proper sampling of the data as shown in Figure 8.5. The Latency optimized Gearbox must be able to meet the condition as stated in eq. 8.3, Figure 8.4 for its implementation using register bank.

$$Frame\ Clock \times Frame\ Size = Word\ clock \times Word\ size \quad (8.3)$$

$$Word\ clock = N \times Frame\ Clock \quad (8.4)$$

where N must be an integer value. Arria 10 Transceiver Native PHY IP core is used in this firmware development. It is made to operate in an enhanced PCS mode, with most features disabled, as the logic used already implements some of it. The width of PMA used is 16 bit while serializer operates at double data rate, this makes PCS data line of 32 bit.

8.9 Transceiver Native PHY

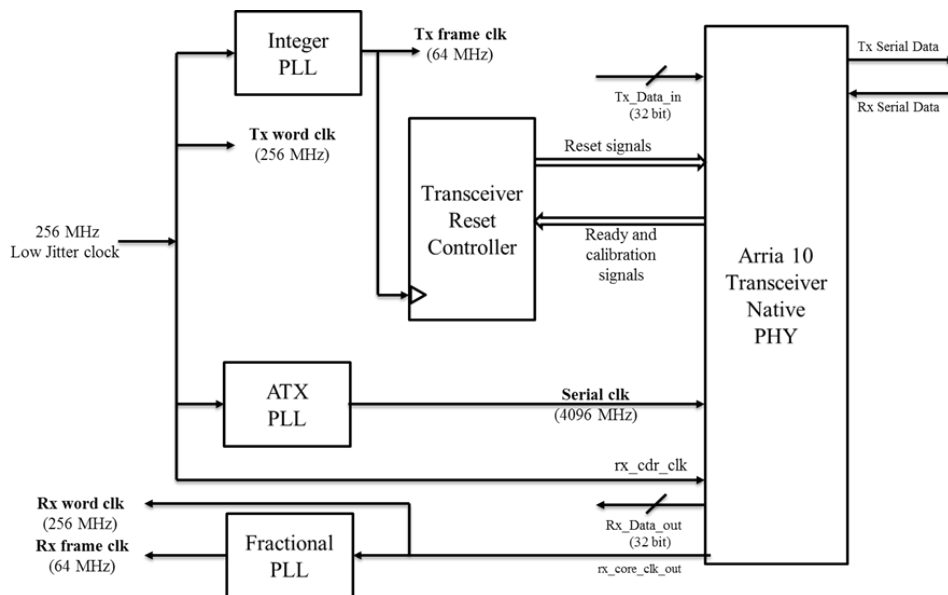


Figure 8.8: Simplified diagram of transceiver data-path and clock connectivity

Advanced Transmit PLL (ATX PLL) is used to feed the serial clock of 4096 MHz to the transceiver. It is an LC tank based Voltage Controlled Oscillator (VCO), having the best jitter performance. The ATX PLL and the clock data recovery (CDR) block need an input reference clock source to generate the clocks required for transceiver operation. The reference clock must be very stable, low jitter and free running at device power-up for proper PLL calibrations. CDCE62005 Jitter cleaner PLL is used before feeding the detector synchronizing clock. Reset Controller IP is used for asserting in correct sequence the reset signals to ensure transceiver channel for proper initialization and functioning. The Frame clock for transmission side is generated by integer PLL from reference clock, while for receiver side frame clock is generated using fractional PLL from recovered clock.

8.10 Comparison

The implementation method is compared against the most widely used CERN GBT data transmission protocol [20, 19]. Table 8.3 lists out some main parameters of the two protocols which have major differences. From the table it is clearly visible that there are some pros and cons of the developed design. The advantages of the design can be accredited to its high data rate, application of AES encryption scheme, more error detection capability. Apparently it might seem why to forego 50% bandwidth for the sake of error correction. Since, this link is operating at roughly the double speed of GBT in the radiation zone, it is very important that the encoding scheme chosen have a wide hamming distance to have very high error detection capability. A FEC with more hamming distance has less ambiguity in error correction in case of multi-bit error and more capable in error detection. Golay Encoder has 100% capability in detecting any odd bit errors, the maximum it can go up to 115 bits. This makes the data transmission scheme more reliable, as it helps to detect and mark corrupt data with a very high confidence.

Table 8.3: Comparison between two data transmission protocol

Parameters	Proposed Method	GBT Data Transmission Protocol [20, 19, 33, 26]
Security	AES-256 Encryption	Not Applicable
Channel Data Throughput	8.192 Gbps	4.8 Gbps
Raw data Throughput	3.84 Gbps	3.2 Gbps
Bandwidth Utilization Efficiency	46.88% (60 / 128)	66.67% (80 / 120)
Forward Error Correction	Golay Encoding (24,12,8)	RS Encoding (15,11) with symbol size of 4 bits
Number of FEC block used	5	2
Resource consumption of FEC	Less	High
Error Detection	35 bits Or any odd bits error (max 115 bits if odd)	32 bits
Error Correction	15 bits	16 bits
Burst Error Correction	15 bits	16 bits
Latency optimized data transmission	Supported	Supported
Interleaver	Helical	Block
Supported Data Type	4	2

8.11 Results

8.11.1 Register resource utilization

Table 8.4: Register Resource Utilization

Name of the Block	AES	GOLAY	Fame synchronization Block	Latency Optimized Gearbox	Scrambler Block	Native PHY with simplified pcs	Helical Interleaver Block
#Registers	10319	752	481	424	180	48	2

Table 8.5: Latency Measured

Parameter	Latency (Clock Cycle)
Scrambler	1
Descrambler	1
AES Encryption	15
AES Decryption	14
Golay Encoder	1
Golay Decoder	4
Helical Interleaver	Concurrent
Helical Deinterleaver	Concurrent
Header and Trailer Addition	Concurrent
Frame Aligner	2
Tx Gearbox	1
Rx Gearbox	1
Native PHY	5

where 1 clock cycle is $1/64$ MHz or 15.625 ns

Table 8.4 shows the register utilization of the individual logic blocks. It is important to note here that the Golay encoder block takes very little resources to fit. While AES-256 encryption block uses most of the design resources. In Figure 8.5 the latency measurement of each of the sub-blocks are shown.

8.11.2 Data Throughput

The developed design does not contain any elastic buffer to modify data throughput based on channel congestion. This means the arrival rate of the packet (λ) and the departure rate of the packet (μ), is equal ($\mu = \lambda$)

This makes the design suitable for trigger transmission with a guaranteed latency. Data throughput of achieved for the system is 3.84 Gbps, having its operating baud rate at 8.192 Gbps.

8.11.3 Optical MiniPOD Performance

Singal integrity analysis of the transmitted optical signal is analyzed for the link rate of 8.192 Gbps. Data from the FPGA transceivers are transmitted to the on board MiniPOD optical modules. It is coupled with flat ribbon cable using a Prizm connector and terminated on the other side with industry standard MTP connector. Lecroy Serial Data Analyser (SDA) oscilloscope is used for analyzing MiniPOD signal quality.

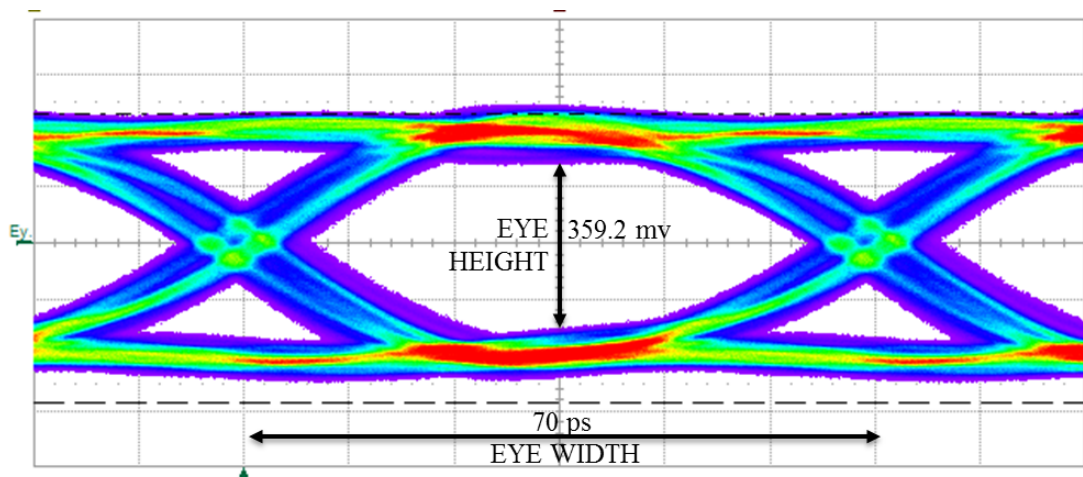


Figure 8.9: Showing the Eye Diagram of 8.192 Gbps optical link

Fig. 8.9 shows wide eye opening indicating good quality of the signal.

$$\text{Eye Width} = 70 \text{ ps}$$

$$\text{Eye Height} = 359.2 \text{ mV}$$

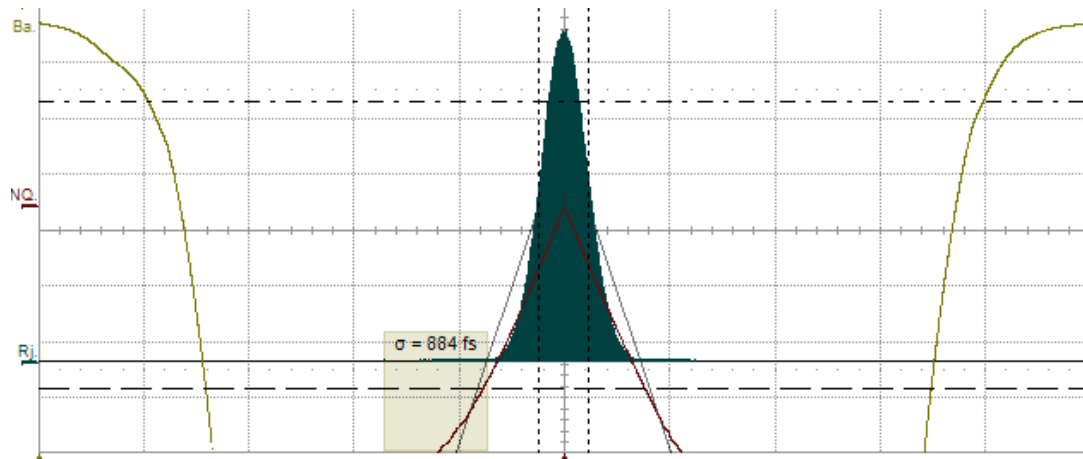


Figure 8.10: Showing Jitter measurement of 8.192Gbps optical link

Jitter spectrum is the measurement of timing variation of a signal edge from its ideal values. Contributing factors include power load variation, thermal noise and interference coupled from nearby devices. It is very important for correct data sampling; the Jitter to be within tolerable hold time requirement of the processing block.

Deterministic Jitter (Dj)	=	2.786 ps
Periodic Jitter (Pj)	=	3.33 ps
Data Dependant Jitter (DDj)	=	15.672 ps
or Inter Symbol Interference (ISI)		
Duty Cycle Jitter (DCD)	=	277 fs
Random Jitter(Rj)	=	962 fs
Total Jitter (Tj)	=	29.315 ps

The random component of the Jitter is specified using the following statistical terms.

$$\text{Standard Deviation } (\sigma) = 884 \text{ fs}$$

8.12 Conclusion

The design is highly capable of detector data transmission in High Energy Physics experiment. It provides soft error correction to handle SEU and data encryption method for security purpose. The design is more concerned with data integrity, data security and error detection reliability than that of maximum bandwidth utilization. The design has currently been implemented on processing FPGA board sitting at low-radiation zone. Future efforts will be made to use other combination of channel coding, error correction coding and cryptography coding for efficient use of channel bandwidth.

Chapter 9

A Generalized Data Aggregation Unit (DAU) for HEP Experiments

9.1 Introduction

High Energy Physicists in their journey to unravel the mysteries of the microscopic world, have constructed sophisticated and large particle accelerators to probe into the whimsical world of sub-atoms. In High Energy Physics (HEP) experiments millions of particle collisions take place per second, producing the raw data at the rate of millions of gigabytes per second. Software converts that raw data to readable data objects for later event analysis. Current experiments produce more than a Peta-Bytes of new data annually, to help scientists understand the fitful nature of the sub-atomic world with more accuracy and insight. Big Science experiments like LHC, which has the world's largest and most powerful particle collider, generates data more than 50 PB/year. In most cases the next generation accelerator operates at a higher energy frontier or intensity than the previous one. The volume of data passing through data centres is ballooning with each subsequent year as shown in Figure 9.1.

A methodology is needed to draw accurate and meaningful conclusions from such a

large amount of data sets, for proper analysis, search, integration, reporting, and system maintenance with keeping in pace to the exponential growth of data. The tools to deal with large data comes under the purview of the subject “BIG DATA”. The fact that data is doubling every two years mimics one of electronics’ most famous laws: Moore’s law [132]. The Moore’s law for transistor have slowed down already while for data generation it still remains valid for the coming decade. The current data scientists are lucky that the growth of electronics is optimum enough to sustain the current growth of data generation.

The BIG DATA applications have accelerated the use of novel computing architectures. These range from computationally intense multi-threaded processor that can hide memory latency, data parallel streaming processors such as GPUs or reconfigurable parallelized architectures like FPGAs. Computational techniques that deals with BIG DATA assimilation are termed as Data Intensive Computing (DIC) [133]. One of the fundamental element in DIC is aggregation of raw data. This chapter discusses the need for a dedicated Electronic System called **Data Aggregation Unit** (DAU), to aggregate the incoming raw data from several channels on to a few manageable ultra-high bandwidth channels, before forwarding it to the Computing Unit for processing. The other utilities which DAU can serve is fast rearrangement of tier architecture, trigger granularity, provide topological cut for efficient data flow management, load sharing to adapt for unprecedented system stress and real-time processing for data reduction [134].

The chapter is divided into multiple sections for presenting the ideas with more clarity without intermixing the goal of each section. The functional idea for DAU is not new, many of the HEP experiments have already realized the necessity for separate architecture, however the unit is not recognized as a separate modular entity before. How the FPGA based DAU acts as a middle-ware in this hybrid Hardware Ecosystem of HEP readout electronics, is discussed in section 9.2. In Section 9.3 defined the most important design constraints required for the experimentalist to decide before development. After making the proper choice how to choose the correct interface standards and FPGA hardware

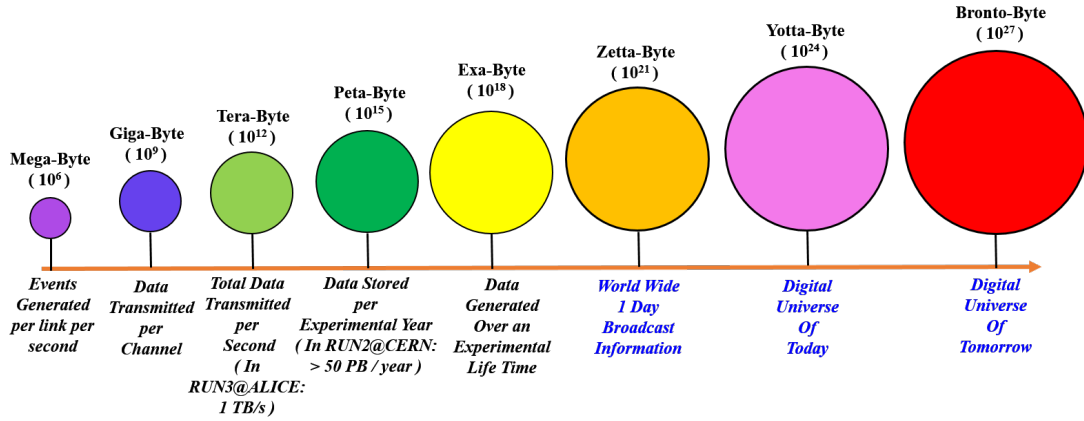


Figure 9.1: Data Volume expansion in High Energy Physics Experiments in comparison to other digital data.

Table 9.1: New Readout Architecture in upgraded or future HEP experiments having the concept of Data Aggregation Unit

Experiment	Unit	Technology
ALICE at CERN	CRU (Common Readout Unit) [15]	Arria [®] 10 FPGA
ATLAS at CERN	FELIX (Front End Link eXchange) [135]	FPGA-based PCIe card
CBM at FAIR	DPB (Data Processing Boards) [136]	Kintex 7 FPGA
LHCb at CERN	SOL40, ODIN40, TRIG40, TELL40 [137]	Arria [®] 10 FPGA
ITER	CODAC (Control, Data Access and Communication) [138]	Distributed Network (Can be seen as IOT)

according to the requirement, is in listed section 9.4. Then in section 9.5, a generalized architecture of DAU is developes using latest popular HEP standards, and also discuss its ability to work in different mode based on current system traffic load. A queue model is used to show how it can predict a sudden upsurge in traffic load and switch to appropriate mode to save system from overflowing (section 9.6). DAU connected in mesh mode could also add to design elasticity and help in smooth transition section 9.7.

9.2 Comparison with other existing architectures

The real time data analysis has always been an integral part of the HEP electronic systems. Currently, it is used for real time trigger generation to select roughly between 0.001% to 0.01% of the most interesting data collected by the detectors for further analysis and discard the rest. However, this trend is going to change. Take for example the LHC four large experiments: ALICE, LHCb, ATLAS and CMS. The ALICE input data rate will rise to 1 TBs^{-1} in 2019-2020, while the LHCb input data rate will rise to 5 TBs^{-1} at the same time. ATLAS and CMS will increase their input rates later, around 2024-2027, reaching around 150 TBs^{-1} [134]. The trigger decision complexity will increase many-fold and also the detector pileup, so to remove the design complexity the hardware will run in full triggerless mode to capture all the events. A software trigger running on each core of the processing farm analyses the events and overall selects subset of them to be saved to disk [139]. An important and integral part of such DIC application is *aggregation* of data to *gather* from multiple detectors and *dispatch* to computing nodes with proper load balancing. Table 9.1 shows that in many experiments the concept is already in use but under different nomenclature.

9.3 Design Constraints of the DAU

Every experiments have different experimental goals, and hence each time there is a need for new architecture development. It is very important to properly identify the constraints originate from a variety of sources. To say in generalized terms this might mean balancing demands for Performance, Scalability, and Reliability with Cost Constraints. In DAU important constraints have been identified that designers required to pay attention, before proceeding further into the problem.

9.3.1 Types of Information and Constraints involved

It is absolutely necessary to identify the information variety in the system, then search for the boundary conditions for stable communication. The design is broadly categorized based on the information content into 3 groups. Related constraints for each group are listed as follows :

- Trigger and Timing Information :
 - Deterministic Latency
 - Jitter Clean Recovered Clock
 - Intra-channel skew cancellation
 - Path with minimum latency
- Detector Data information :
 - Error resilient communication
 - Radiation tolerant electronic system and standards
 - Consensus of different detector data links
 - Push data with minimum back-pressure to prevent packet drop

- Monitoring, Control and Configuration Information :
 - Priority based packet transaction
 - Request-Response-Acknowledgement Cycle
 - Minimum System Response Time
 - Redundant readings for critical system parameters
 - Minimum Packet Loss

9.3.2 Degrees of Synchrony

For latency critical communications, deterministic behavior of the system is expected. Using asynchronous link to behave as synchronous communication is a challenge. According to Veríssimo (1997) [50], a system can behave as synchronous system if it is compliant with five Degrees of Synchrony:

1. Bounded and known processing speed
2. Bounded and known message delivery delay
3. Bounded and known local clock rate drift
4. Bounded and known load pattern
5. Bounded and known difference between local clocks

Depending on the probabilistic or partial satisfaction of synchrony assumption, different intermediary model might exist. Example of such intermediate models are the timed asynchronous system model [140] and the quasi-synchronous model [141]. Consensus in asynchronous systems can be achieved at different levels which relaxes the design constraints. Three kinds of asynchrony in the system can be identified, namely process asynchrony, communication asynchrony and message order asynchrony [142]. Fischer

Table 9.2: Comparison of SerDes Speed among popular FPGA Vendors

	ALTERA		XILINX		ACHRONIX		
SILICON	DEVICE	SERDES	DEVICE	SERDES	DEVICE	SERDES	
TECHNOLOGY	FAMILY	MAX. SPEED	FAMILY	MAX. SPEED	FAMILY	MAX. SPEED	
60 nm	Cyclone IV <i>(GX Variant)</i>	3.125 Gbps					
45 nm			Spartan 6 LXT	3.2 Gbps			
28 nm	Cyclone V <i>(GX Variant)</i>	3.125 Gbps	Artex 7	6.6 Gbps			
	<i>(GT Variant)</i>	6.144 Gbps					
	Arria V <i>(GX / GT /GZ Variant)</i>	6.5536 Gbps	Kintex 7	12.5 Gbps			
	<i>(GT Variant)</i>	10.3125 Gbps					
<i>(GZ Variant)</i>	12.5 Gbps						
	Stratix V <i>(GS / GX Variant)</i>	14.1 Gbps	Virtex 7 <i>(GTX Transceiver)</i>	12.5 Gbps			
	<i>(GT Variant)</i>	12.5 Gbps			13.1 Gbps		
	<i>(GT Variant)</i>	28.05 Gbps				28.05 Gbps	
22 nm							Speedster 22i HD (HD680, HD1000, HD1500)
					(HD1500)		28 Gbps
					Speedster 22i HP (HP360, HP560)	12.75 Gbps	
					(HP560)	28 Gbps	
20 nm	Arria 10 <i>(GX / GT Variant)</i>	17.4 Gbps 25.78 Gbps	Kintex Ultrascale <i>(GTX / GTY Transceiver)</i>	16.3 Gbps			
			Virtex Ultrascale <i>(GTH Transceiver)</i> <i>(GTY Transceiver)</i>	16.3 Gbps 30.5 Gbps			
16 nm			Kintex Ultrascale + <i>(GTH Transceiver)</i>	16.3 Gbps			
			<i>(GTY Transceiver)</i>	32.75 Gbps			
			Virtex Ultrascale +	32.75 Gbps			
14 nm	Stratix 10 <i>(GX Variant)</i>	17.4 Gbps					
	<i>(GXT Variant)</i>	30 Gbps					

et. al. [143] coined the phrase “Bounded Asynchrony”. It means partial satisfaction of synchrony assumption is possible if the system boundary condition meets system time tracking granularity requirement at each levels.

9.4 Choice of DAU Hardware and Interfaces

Once the design the constraints have been identified, the stage is set to choose correct hardware and interface standards. Three comparison tables are provided for quick reference to the recent standards in use, namely Table 9.2 for selecting the right kind of FPGA with the needed transceiver switching speed, Table 9.3 for choosing the required commercial interfaces from the current/upcoming ones and in Table 9.4 gives a comparison between currently popular radiation zone suitable for HEP interface standards.

Some of the high energy physics big data experiment are moving towards PCIe form factor based DAQ boards, instead of using old VME or more popular ATCA form-factor. Three of the LHC High energy physics experiment namely; ALICE: Readout based on CRU [15], LHCb: Readout based on TELL40, and ATLAS: Readout based on FELIX are all opting for PCIe form factor board. The design is inspired from the work in CRU project at ALICE collaboration. The hardware used for generating the test result is PCIe40.

9.5 Implementation

A new design configuration for Data Aggregation that uses latest electronic hardware and protocol standards currently most popular in HEP experiments is proposed. The Data from High Energy Physics involves mainly three types of information content, namely: Trigger and Timing, Detector Readings and Control Information. The design treats these three data traffic separately to meet different delay constraints. Treating the data traffic

CHAPTER 9. A GENERALIZED DATA AGGREGATION UNIT (DAU) FOR HEP EXPERIMENTS

separately and then integrating it over same channel improves Reliability, Availability and Serviceability (RAS). To show the hidden complexity behind the design of data traffic manager a mathematical queuing model is used to explain it more elaborately. The DAU firmware uses recent state of the art Arria[®] 10 FPGA of 20nm chip technology available from Altera to house the design.

Table 9.3: Comparison of popular Communication standards speed along with evolution timeline

TIMELINE	Communication Standards for High Performance Computing (HPC)				
	ETHERNET	PCIe x8	INFINIBAND x4	RAPIDIO x4	INTEL's OMNI-PATH
1983	10 Mb/s				
1995	100 Mb/s				
1998	1 Gb/s				
2002	10 Gb/s	4 GB/s (GEN 1.0)	20 Gb/s (DDR)	12.5 GBd (v1.x)	
2003					
2004					
2005					
2006					
2007					
2008					
2009					
2010	40 Gb/s	16 GB/s (GEN 3.0)	40 Gb/s (QDR)	25 GBd (v2.x)	
2011					
2012	100 Gb/s	32 GB/s (GEN 4.0)	56 Gb/s (FDR)	40 GBd (v3.x)	
2013					
2014					
2015					
2016					
2017	400 Gb/s		200 Gb/s (HDR)	100 GBd (v4.x)	100 Gb/s

The trigger and timing information from the Front End Electronics to DAU are received over TTC-PON (Timing and Tigger Control Passive Optical Network) [144], which is then used for time stamp generation and multi-link event building. Data from different detectors sitting in radiation zone are transmitted to DAU situated in controlled-radiation zone over GBT protocol[17]. GBT is a radiation tolerant error resilient data communication standard with fixed latency support. A single GBT channel link can handle detector data, timing, trigger and control information traffic. Upto 36 GBT links can come to DAU from Front end Detectors. Each link speed is about 4.8 Gbps for full channel occupancy. Both GBT and TTC-PON are custom protocol standards developed by CERN team to

Table 9.4: Comparison of three high speed interface protocol standards used in the design

Parameters	GBT	TTC-PON	10G Ethernet
Technology Specification	Custom	XGPON1 with modifications	802.3ae Specification Standard
Designer Group	CERN	ITU-T with CERN modifications	IEEE
Line Rate	4.8 Gbps	Downstream: 9.6 Gbps Upstream: 2.4 Gbps	10.3125 Gbps per lane
Payload Rate	3.2 Gbps	Downstream: ~7.68 Gbps Upstream: 640 Mbps	10 Gbps per lane
Payload Size	120 bits@40 MHz	Downstream: 192 bits@40 MHz Upstream: ~16 bits@40 MHz	64 bits@156.25 MHz for 8 lanes
Wavelength	850 nm (Multi-mode Tx) 1310 nm (Single-mode Tx)	Downstream: 1577 nm Upstream: 1270 nm	850 nm (10 GBASE-SR)
Network Topology	Point-to-Point	Point-to-Multipoint	Point-to-Point
Encoding	RS Encoding with Block Interleaver	8b/10b	64b/66b
Synchronous Trigger Support	Yes	Yes	No
Trigger Latency	Optical loop-back Round-trip: 150 ns	Downstream: ~100 ns Upstream: 1.6 μ s	Not Applicable

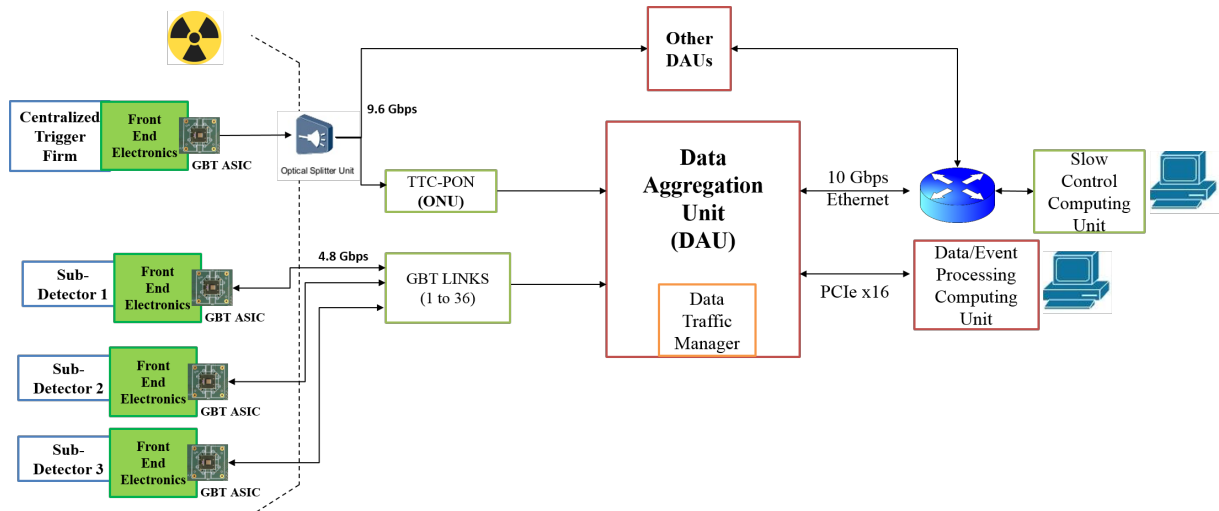


Figure 9.2: Architecture of DAU Implementation

support LHC extreme experiment conditions. The data aggregated from multiple GBT links are then forwarded onto commercially available high bandwidth channels. Two types of link standards are chosen to address two different cases. PCIe Gen 3 x16 to push data on to the next level of DAQ for event building; and UDP/SCTP over 6 links of 10 Gbps Gigabit Ethernet for slow control of detector, data and experimental control.

9.5.1 Modes of Operation

PCIe Gen 3 and 10 Gigabit Ethernet is used to push the data to the computing section. However, 10 Gigabit Ethernet can also be used to pull the data. Accordingly, at any instance of time the DAU can operate in 4 different modes based on requirement.

- Nominal Mode : In this mode all incoming channels and outgoing channels are operating in $<100\%$ occupancy.
- Extended Mode : In this mode all incoming channels operate at full occupancy and there is a probability of high back-pressure (throttle) in a particular DAU. The excess data traffic load is shared by adjacent DAU which receives the multi-cast unprocessed data over 10 Gigabit Ethernet.
- Twin Mode : In this mode a duplicate copy of the incoming data is multi-casted to other level of processors to have redundancy in data processing.
- Throttling Mode: In this mode if there high back-pressure (throttle) in a particular DAU, it will take decision on real time analysis of data or use *trigger signal* to keep important detector data and permanently reject the rest.

9.6 Queue model of DAU

For prediction of data traffic load with the use of Queuing analysis a data driven mathematical model for DAU is proposed. An efficient load prediction model helps to plan

for architecture or data distribution scheme in advance, and also help to decide when to switch architecture topology dynamically when there is sudden upsurge of data.

Delay-history-based predictors based on flow monitor are used in the calculations. It helps make delay prediction for uncertain and time-varying demand system, for improved Quality of Service (QoS). Let, us define a time window of ΔT as population size for averaged value calculation for the following defined parameters.

$$\text{Average Queue Length} = \mathbb{L}_q \quad (9.1)$$

$$\text{Average Change in Queue Length} = \frac{\Delta \mathbb{L}_q}{\Delta T} \quad (9.2)$$

$$\text{Average Service rate of each channel} = \mathbb{S}_{Ch} \quad (9.3)$$

$$\begin{aligned} \text{Average Service rate of each DAUs} &= \mathbb{S}_{DAU} \\ &= \text{Avg}(\mathbb{S}_{Ch}) : \forall \text{ Channels in the DAU} \end{aligned} \quad (9.4)$$

$$\begin{aligned} \text{Average Service rate of the System} &= \mathbb{S} \\ &= \text{Avg}(\mathbb{S}_{DAU}) : \forall \text{ DAUs in the System} \end{aligned} \quad (9.5)$$

$$\text{Average Packet arrival rate per channel} = \mathbb{A}_{Ch} \quad (9.6)$$

$$\begin{aligned} \text{Average Packet arrival rate of each DAUs} &= \mathbb{A}_{DAU} \\ &= \text{Avg}(\mathbb{A}_{Ch}) : \forall \text{ Channels in the DAU} \end{aligned} \quad (9.7)$$

$$\begin{aligned} \text{Average Packet arrival rate of the System} &= \mathbb{A} \\ &= \text{Avg}(\mathbb{A}_{DAU}) : \forall \text{ DAUs in the System} \end{aligned} \quad (9.8)$$

During Data spike the buffered memory or Maximum Allowed Queue length must not overflow due to Service Delay, this prevents drop of data due to channel congestion. So,

selection of Queue length is based on the stability criteria as shown in Eq. (9.9).

$$\mathbb{L}_q = \frac{[\mathbb{A}_{DAU} - \mathbb{S}_{DAU}]}{No. \ of \ Channels} \Big|_{Max. \ Data \ Spike \ Test} \quad (9.9)$$

Stress testing can provide a means to measure graceful degradation and the ability of a system to maintain limited functionality even when a large part of it has been compromised. A parameter *System Stress Factor* for DAU in Eq. 9.10 is defined to make decisions on mode selection as discussed in section 9.5.1.

$$System \ Stress \ Factor \ (\gamma) = \frac{\mathbb{S}_i^{DAU}}{\mathbb{A}_i^{DAU}} \quad (9.10)$$

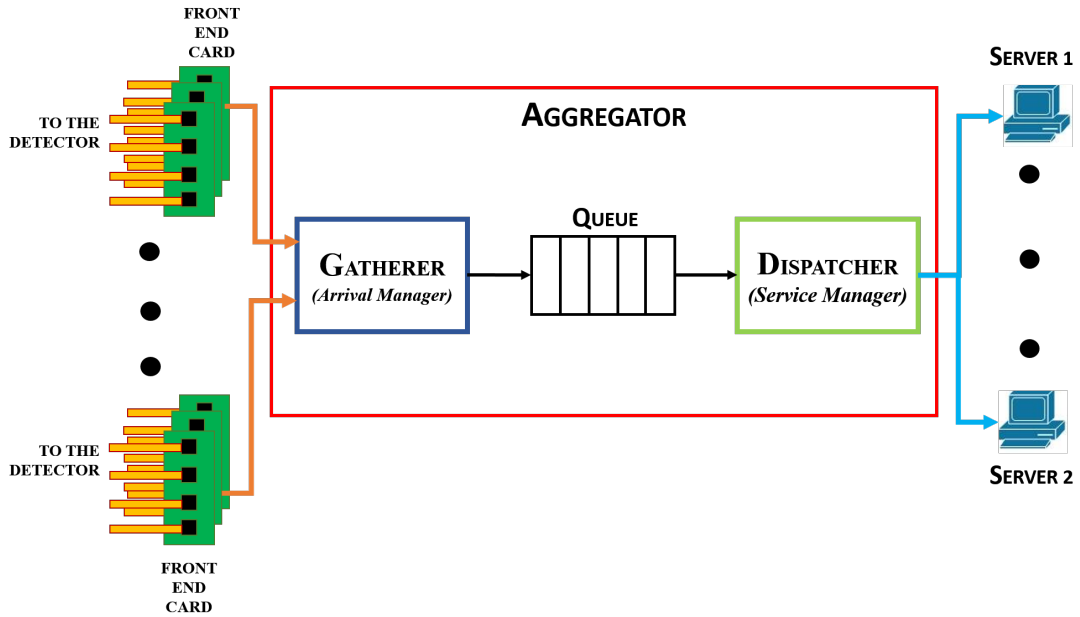


Figure 9.3: Queue Model for the Data Aggregation Unit

The Optimum System Stress Factor is defined as γ_{opt} . The current DAU *System Health*

Table 9.5: Queueing Model Performance comparison of some of the widely used models

Performance Parameters	MODEL A	MODEL B	MODEL C	Proposed Model
Number of Channels	Single	Multi-Channel	Single	Multi-Channel
Number of Phases	Single	Single	Single	Multi-Phases
Arrival Rate Pattern	Poisson	Poisson	Poisson	Any
Serrvice Time Pattern	Exponential	Exponential	Constant	Any
Population Size	Unlimited	Unlimited	Unlimited	Limited
Queue Discipline	FIFO	FIFO	FIFO	FIFO
Average number of datagram buffered in the system (L_s)	$\frac{\lambda}{\mu-\lambda}$	$\frac{\lambda\mu(\lambda/\mu)^M}{(M-1)!(M\mu-\lambda)^2}P_0 + \frac{\lambda}{\mu}$	$L_q + \frac{\lambda}{\mu}$	$\gamma (\leq \mathbb{L}_q)$
Average time a datagram spends inside the DAU (W_s)	$\frac{1}{\mu-\lambda}$	$\frac{L_s}{\lambda}$	$W_q + \frac{1}{\mu}$	$\frac{L_s}{\mathbb{S}_{DAU}}$
Average number of datagram waiting to be served by a unit dispatcher (L_q)	$\frac{\lambda^2}{\mu(\mu-\lambda)}$	$L_s - \frac{\lambda}{\mu}$	$\frac{\lambda^2}{2\mu(\mu-\lambda)}$	$L_s - \frac{\mathbb{S}_{Ch}}{\mathbb{A}_{Ch}}$
Average time a datagram spends waiting to be served by a unit dispatcher (W_q)	$\frac{\lambda}{\mu(\mu-\lambda)}$	$W_s - \frac{1}{\mu} = \frac{L_q}{\lambda}$	$\frac{\lambda}{2\mu(\mu-\lambda)}$	$\frac{L_q}{\mathbb{S}_{Ch}}$

where

λ = Average arrival rate

μ = Average service rate at each channel

P_0 = Probability that there are zero datagram in the system

M = Number of service channels open

is specified as

$$System\ Health = \begin{cases} \text{Nominal Condition} & : \gamma \leq \gamma_{opt} \\ \text{Under Stress} & : \gamma_{opt} < \gamma < \mathcal{T} \\ \text{Excess Data Forwarded} & : \mathcal{T} \leq \gamma < 1 \\ \text{Packet Drop Started} & : 1 \leq \gamma \end{cases} \quad (9.11)$$

When “*System Health is Under Stress*”, then system is about to get congested, prepare for mode switch for data diversion through other DAU. When “*System Health is Excess Data Forwarded*”, DAU operating in *Extended Mode*. And when “*System Health is*

Packet Drop Started”, DAU operating in *Rejection Mode*.

9.7 Readout Architecture Transition from Hierarchy to IOT

The Guiding Principles for data producers and publishers, for scientific data management and stewardship are four foundational principles: Findability, Accessibility, Interoperability, and Reusability (FAIR) [145]. The phrase “Big science” is coined by Galison and Hevly (1992) [146] to define the growth of HEP large-scale research. Huge data volumes of HEP experiment require parallel end-user analysis on clusters of hundreds of machines, the solution is to create a more distributed mesh like architecture for parallel data processing of different working models like the GRID computing evolution [147]. “Cloud + Exascale Ecosystem” cyberinfrastructure is the next big computing thing for scientific workflows [148], it includes faster on-demand service with any-time any-where network access and provides an efficient pool of resource management leading to economies of scale in performance and electrical power (Green IT). The Data Acquisition (DAQ) system is also not lagging in up-gradation to match with computing data distribution topology restructuring. Internet of Things (IoT) has already started to appear within DAQ infrastructures, more fine-grained and intelligent way to interact on specialized high-density DAQ system, providing opportunity for low-power, local intelligence with standardized access to future readout-systems and associated sensors [149]. These Large-scale, data-intensive, distributed computing also offers virtual organization of data nodes, thus optimizing the design for different analysis condition [150].

It is an established fact to enable long-lasting data usability of a research enterprise and to facilitate discovery or support of new knowledge, maximizing more community engagement is a necessity. Recently, a consortium including CERN, DESY and CNRS in Europe submitted a proposal for International Long-term Data and Analysis Preservation (ILDAP) to develop and prototype a first set of data reuse archival and prototyping services in the context of LHC. The LHC program when it comes to an end, probably the

last data in this energy frontier for many years and the world cannot afford to lose such valuable information. LHC plan for moving towards open data future, such that knowledge and research results are accessible to the people outside the HEP community [151]. The Data Preservation in Long Term Analysis in High Energy Physics (DPHEP) study group is making effort to push data from mere storage to a system of open sharing [152]. Data and Software Preservation for Open Science (DASPOS) which has a similar goal as DHEP with an added “Curation Challenge”, where, for example, an ATLAS physicist might perform an analysis on curated and archived CMS data [153].

Challenges of BIG DATA Electronics in HEP experiment is to deal with data having high Volume, Veracity, Variety and Velocity. Moreover, these 4Vs are increasing every year. It is getting more complex with the interconnection of physical detectors and sensors, together forming Internet of Things (IoT). Till now distribution was at the computing level, now it will be on the readout level also. DAU perfectly fits as the most adaptable and necessary unit in the picture of this transition of readout architecture, as shown in Figure 9.4.

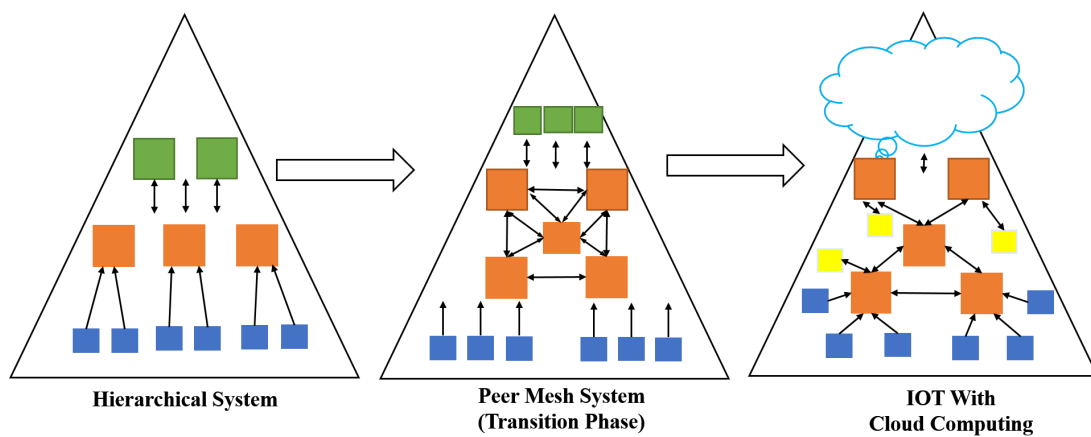


Figure 9.4: Transition of Data readout Architecture in High Energy Physics Experiments

9.8 Conclusion

The work highlights the need for a dedicated Data Aggregation Unit (DAU) to adapt to the upcoming HEP readout architecture requirement efficiently. DAU can be beneficial for easy readout architecture transition from current hierarchical architecture to future mesh-based architecture. The goal of the chapter is to give an overview of the existing systems in order to find the best architectures, an emulation of envisaged architectures to identify possible bottlenecks and building of small-scale prototype to find limitations of possible hardware implementations. In the future study, if the results are encouraging in terms of performance and cost, a study of the scaling by a factor ten can be engaged.

Chapter 10

Summary and Future Scope

The thesis is composed of ten chapters. Each chapter consists of an introduction and a concluding section. Individually, the chapters deal with the different aspect of a specific research sub-topic. A subtle transition of one research sub-topic to the other is maintained to assist in easy readability.

Chapter 1 is introductory and describes in detail about the world's largest collider LHC and how ALICE is planning to align its upgrade with the LHC high luminosity run period for more exciting physics research. To cope up with LHC RUN3 luminosity, ALICE has opted for a complete detector and technical upgrade. It is expected that high amount of data volume (>1 TB/s) and events will be generated from the detectors due to large interaction rate. Hence a separate and dedicated block is phased out from the electronic system to deal with load balancing and trigger distribution. The block is termed as CRU. CRU is the name of the firmware functionality programmed on PCIe40 DAQ Engine, developed in collaboration with LHCb. The central processing hardware that is used by CRU is reconfigurable unit based on FPGA technology. This is not the first time an FPGA based readout board found application in ALICE experiment. Previously FPGA based readout board is used in RUN1 as D-RORC, in RUN2 as C-RORC and finally as CRU in RUN3. However what is exceptional from previous runtime experience is that the

same board will be used by the trigger group, the DAQ group and the detector front-end electronics group for custom functionality. Since PCIe40 would also be used by LHCb detector group, hence substantial firmware development and R&D that are of common need for both ALICE and LHCb collaboration is developed in a joint effort.

Chapter 2 gives an overview of CRU architecture in ALICE, discusses the different interfaces needed for the firmware design and the need for resource estimation before initiating implementation of CRU firmware on the PCIe40 board. CRU act as a common interface between the sub-detector electronic systems, the computing systems and the trigger processors. The interface links include GBT, TTC-PON and PCIe. Giga-Bit Transceiver optical link (GBT) is used for detector data payload transmission and constant latency path for trigger distribution between CRU and detector readout electronics. Timing, Trigger and Control system based on Passive Optical Networks (TTC-PON) is employed for time multiplex trigger distribution between CRU and Central Trigger Processor (CTP). Peripheral Component Interconnect Express (PCIe) is the high-speed serial computer expansion bus standard for bulk data transport between CRU boards and processors. CRU is used in ALICE readout electronic system for data aggregation, trigger and timing distribution and control moderation.

Chapter 3 examines GBT interface link standard and describes in detail the prototyping & evaluation involved for CRU Development. Discussion includes on GBT FPGA Core firmware implementation strategy over latest Intel[®] (formerly Altera) FPGAs, along with inter-FPGA firmware migration, clock calibration strategy and comparative performance study. Two custom DAQ boards AMC40 and PCIe40 based on latest Intel[®] (formerly Altera) Stratix[®] V and Arria[®] 10 FPGA were used for the detailed comparative study of GBT FPGA Core firmware implementation.

Chapter 4 touches the subtle intricacies of trigger and timing communication using asynchronous link communication chain of TTC-PON GBT Bridge in CRU. The trigger distribution system using CRU is an amalgamation and integration of multi-technology

or protocol standards like GBT and TTC-PON. For entire system to operate correctly and efficiently, the individual design needs to be optimized and be able to work in coherence to another block. A detailed measurement and analysis results are discussed. Four performance metrics are used to characterize the communication bridge: (a) the latency added by the firmware logic, (b) the jitter cleaning effect of the PLL on the timing signal, (c) BER analysis for quantitative measurement of signal quality, and (d) the effect of optical transceivers parameter settings on the signal strength. Reliability study of the bridge connection in maintaining the phase consistency of timing signals is conducted by performing multiple iterations of power on/off cycle, firmware upgrade and reset assertion/de-assertion cycle (PFR cycle).

Chapter 5 discusses the qualification tests done to characterize the SI534X PLL component. In the Large Hadron Collider (LHC) Bunch Clock (BC) is one of the most critical accelerator signals delivered to all the participating experiments. The clock is directly derived from the Radio Frequency (RF) driving the beams in the accelerator. The whole range of electronic systems from ADC to high-speed transmission link, work synchronously with the Bunch Clock and are sensitive to jitter. Throughout the clock distribution chain, high-frequency components increase the jitter. Multiple Phase-Locked Loops (PLLs) are used in the entire chain to keep the jitter noise low. Si5344 PLL by Silicon Labs is chosen as one of the candidate PLLs for jitter cleaning of the embedded clock in LHC gigabit serial link transmission. The testing system emulates the LHC runtime condition that involves variation in temperature or repetitive reset assertion of the PLL system.

Chapter 6 describes a novel design for a phase measurement logic core having resolution and precision in the range of a few picoseconds. The design is developed out of the necessity to measure small phase shift changes inside an FPGA. Phase measurement is required in CRU applications where a synchronous relationship between the signals needs to be preserved. Traditional electronic systems used for time measurement are designed

using a classical mixed-signal approach. With the advent of reconfigurable hardware like FPGAs, it is more advantageous for designers to opt for all-digital architecture. Most high-speed serial transceivers of the FPGA circuitry do not ensure the same chip latency after each power cycle, reset cycle or firmware upgrade. These cause uncertainty of phase relationship between the recovered signals. Hence, an inbuilt phase monitoring system is designed to detect and mitigate any phase-related errors.

Chapter 7 discusses how the integrity of data payload can effectively be protected in high-speed communication with little overhead using CRC32 standard. To match the modern requirement of high-speed data transmission, the algorithm needs to be revised for minimum latency. Such low latency is possible by more pipelining and parallelism. This chapter revisits the CRC algorithm to redesign its architecture to suit modern day requirement for high throughput low latency computation. The entire architecture supports run-time or compile-time reconfiguration of CRC-32 standards.

Chapter 8 introduces a new error resilient High-speed Link Standard developed for HEP environment. It operates at a link speed of 8.192 Gbps. A triple bit data error correction scheme is used to protect against radiation-induced soft-errors and also prevents eavesdropping of classified information by encrypting the data packet using Advanced Encryption Standard (AES) while transmitting over the optical medium.

Finally, in **Chapter 9** the future scope of electronic readout cards like CRU are highlighted. A design concept is proposed for the development of a generalized Data Aggregation Unit (DAU) for High Energy Physics (HEP) Experiments. To draw analogy let us consider a router, it is used in network distribution system as a dedicated functional module to deal with multiple network protocols. Similarly, the idea of DAU as a modularized functional module that can be utilized in the future experiment is presented as a dedicated one-stop solution for multiple HEP communication protocols. DAU in HEP can serve as fast rearrangement of tier architecture, trigger granularity, provide topological cut for efficient data flow management, load sharing to adapt to unprecedented system stress

and real-time processing for data reduction. Also, for application of artificial intelligence and big data processing, DAU concept can find a promising application.

The dissertation discusses the design and development work involved in the building of firmware for the Common Readout Unit (CRU). CRU has been planned by the ALICE collaboration, to prepare for high luminosity run of LHC during RUN3 starting from 2021. The entire thesis revolves around the core idea of CRU design conception to its design implementation, where significant emphasis is given to the trigger and timing signal communication link chain development to ensure phase consistency and signal integrity.

REFERENCES

- [1] Walt Kester, “[Converting Oscillator Phase Noise to Time Jitter](#),” *Analog Devices MT-008 Tutorial*, pp. 1–10, 2009. [xxxvi](#), [90](#), [91](#)
- [2] R. Schwemmer, J. Cachemiche, N. Neufeld, C. Soos, J. Troska, and K. Wyllie, “[Evaluation of 400 m, 4.8 Gbit/s Versatile Link lengths over OM3 and OM4 fibres for the LHCb upgrade](#),” *Journal of Instrumentation*, vol. 9, no. 03, p. C03030, 2014. [xxxvii](#), [104](#)
- [3] A. Breskin and R. Voss, *The CERN large hadron collider: accelerator and experiments*. Cern Geneva, 2009. [4](#)
- [4] The LHC Study Group, “[The Large Hadron Collider, conceptual design](#),” CERN/AC/95-05 (LHC) Geneva, Tech. Rep., 1995. [6](#), [65](#)
- [5] ALICE-collaboration *et al.*, “[CERN faq LHC the guide, CERNBrochure-2008-001-Eng, 2008](#),” *CERN-Brochure-2017-002-Eng*, 2017. [6](#)
- [6] G. L. Kane and A. Pierce, *Perspectives on LHC physics*. World Scientific, 2008. [11](#)
- [7] K. Aamodt, A. A. Quintana, R. Achenbach, S. Acounis, D. Adamová, C. Adler, M. Aggarwal, F. Agnese, G. A. Rinella, Z. Ahammed *et al.*, “[The ALICE experiment at the CERN LHC](#),” *Journal of Instrumentation*, vol. 3, no. 08, p. S08002, 2008. [11](#)

- [8] B. A. et al and the ALICE Collaboration, “[Upgrade of the ALICE Experiment: Letter Of Intent](#),” *Journal of Physics G: Nuclear and Particle Physics*, vol. 41, no. 8, p. 087001, 2014. [12](#), [23](#), [68](#)
- [9] F. Carena, W. Carena, V. C. Barroso, F. Costa, S. Chapeland, C. Delort, E. Dénes, R. Divià, U. Fuchs, A. Grigore *et al.*, “[DDL, the ALICE data transmission protocol and its evolution from 2 to 6 Gb/s](#),” *Journal of Instrumentation*, vol. 10, no. 04, p. C04008, 2015. [14](#), [15](#)
- [10] F. Carena, W. Carena, S. Chapeland, V. C. Barroso, F. Costa, E. Dénes, R. Divià, U. Fuchs, A. Grigore, T. Kiss *et al.*, “[The ALICE data acquisition system](#),” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 741, pp. 130–162, 2014. [14](#)
- [11] W. Carena, J. Marin, J. Sulyán, C. Soós, P. Van de Vyvre, P. Csató, R. Divià, A. Vascotto, F. Carena, K. Schossmaier *et al.*, “[PCI-based readout receiver card in the ALICE DAQ system](#),” in *8th Workshop on Electronics for LHC Experiments, Colmar, France*, 2002, pp. 9–13. [14](#)
- [12] ALICE collaboration, “[Technical design report of the trigger, data acquisition, high level trigger and control system](#),” *CERN*, *January*, 2004. [14](#)
- [13] H. Engel and U. Keschull, “[Common read-out receiver card for ALICE Run2](#),” *Journal of Instrumentation*, vol. 8, no. 12, p. C12016, 2013. [15](#)
- [14] J. Alme, T. Alt, L. Bratrud, P. Christiansen *et al.*, “[RCU2-The ALICE TPC readout electronics consolidation for Run2](#),” *Journal of Instrumentation*, vol. 8, no. 12, p. C12032, 2013. [15](#)
- [15] J. Mitra, S. Khan, S. Mukherjee, R. Paul, and for the ALICE collaboration, “[Common Readout Unit \(CRU\)-A new readout architecture for the ALICE experiment](#),” *Journal of Instrumentation*, vol. 11, no. 03, p. C03021, 2016. [15](#), [80](#), [129](#), [209](#), [214](#)

- [16] M. Bellato, G. Collazuol, I. D’Antone, P. Durante, D. Galli, B. Jost, I. Lax, G. Liu, U. Marconi, N. Neufeld *et al.*, “[A PCIe Gen3 based readout for the LHCb upgrade](#),” in *Journal of Physics: Conference Series*, vol. 513. IOP Publishing, 2014, p. 012023. [15](#), [39](#), [48](#), [92](#)
- [17] J. Mitra, S. A. Khan, M. B. Marin, J. P. Cachemiche, E. David, F. Hachon, F. Rethore, T. Kiss, S. Baron, A. Kluge *et al.*, “[GBT link testing and performance measurement on PCIe40 and AMC40 custom design FPGA boards](#),” *Journal of Instrumentation*, vol. 11, no. 03, p. C03039, 2016. [26](#), [77](#), [82](#), [117](#), [129](#), [215](#)
- [18] D.-M. Kolotouros, S. Baron, C. Soos, and F. Vasey, “[A TTC upgrade proposal using bidirectional 10G-PON FTTH technology](#),” *Journal of Instrumentation*, vol. 10, no. 04, p. C04001, 2015. [26](#), [31](#), [32](#), [75](#), [79](#)
- [19] S. Baron, J. Cachemiche, F. Marin, P. Moreira, and C. Soos, “[Implementing the GBT data transmission protocol in FPGAs](#),” in *Topical Workshop on Electronics for Particle Physics (TWEPP)*, 2009. [26](#), [32](#), [48](#), [75](#), [76](#), [201](#), [202](#)
- [20] P. Moreira, R. Ballabriga, S. Baron, S. Bonacini, O. Cobanoglu, F. Faccio, T. Fedorov, R. Francisco, P. Gui, P. Hartin *et al.*, “[The GBT project](#),” in *Proceedings of the Topical Workshop on Electronics for Particle Physics*, 2009, pp. 342–346. [26](#), [32](#), [47](#), [48](#), [62](#), [201](#), [202](#)
- [21] ALICE Collaboration , “[Technical Design Report for the Upgrade of the ALICE Read-out & Trigger System](#),” *CERN-LHCC-2013-019 / LHCC-TDR-015*, 2014. [28](#), [33](#)
- [22] L. Amaral, S. Dris, A. Gerardin, T. Huffman, C. Issever, A. J. Pacheco, M. Jones, S. Kwan, S.-C. Lee, Z. Liang *et al.*, “[The versatile link, a common project for super-LHC](#),” *Journal of Instrumentation*, vol. 4, no. 12, p. P12003, 2009. [32](#), [75](#)

- [23] PCI-SIG[®] Group, “[PCI Express[®] Base Specification Revision 3.0](#),” *PCI-SIG*, 2010. 32
- [24] J. Christiansen, A. Marchioro, P. Moreira, and T. Toiff, “[TTCrx reference manual](#),” *A Timing, Trigger and Control Receiver ASIC for LHC Detectors, CERN EP/MIC*, vol. 3.9, 2004. 34
- [25] ALICE Collaboration, “[Technical Design Report for the Upgrade of the Online-Offline Computing System](#),” *CERN-LHCC-2015-006*, 2015. 35
- [26] A. Caratelli, S. Bonacini, K. Kloukinas, A. Marchioro, P. Moreira, R. De Oliveira, and C. Paillard, “[The GBT-SCA, a radiation tolerant ASIC for detector control and monitoring applications in HEP experiments](#),” *Journal of Instrumentation*, vol. 10, no. 03, p. C03034, 2015. 36, 48, 202
- [27] D. Vaughan, R. Hannah, and M. Fields, “[Applications for Embedded Optic Modules in Data Communications](#),” *Avago Technologies White Paper*, 2011. 41, 55
- [28] Altera Corporation, “[V-Series Avalon-MM DMA Interface for PCIe Solutions](#),” *UG-01154*, 2014. 41
- [29] Altera Corporation, “[Intel[®] Arria[®] 10 Device Overview](#),” *a10_overview*, 2015. 43
- [30] O. Boyle, R. McLaren, and E. van der Bij, “[The S-LINK interface specification](#),” *ECP division CERN*, 1997. 47
- [31] B. Taylor, “[Timing Distribution at the LHC](#),” *CERN EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH-REPORTS-CERN*, vol. 3, pp. 63–74, 2002. 47
- [32] F. Vasey, D. Hall, T. Huffman, S. Kwan, A. Prosser, C. Soos, J. Troska, T. Weidberg, A. Xiang, and J. Ye, “[The Versatile Link common project: feasibility report](#),” *Journal of Instrumentation*, vol. 7, no. 01, p. C01075, 2012. 47

- [33] M. B. Marin, S. Baron, S. Feger, P. Leitao, E. Lupu, C. Soos, P. Vichoudis, and K. Wyllie, “[The GBT-FPGA core: features and challenges](#),” *Journal of Instrumentation*, vol. 10, no. 03, p. C03021, 2015. [48](#), [74](#), [200](#), [202](#)
- [34] J. Cachemiche, P. Duval, F. Hachon, R. Le Gac, and F. Réthoré, “[Recent developments for the upgrade of the LHCb readout system](#),” *Journal of Instrumentation*, vol. 8, no. 02, p. C02014, 2013. [48](#)
- [35] F. Alessio, J. Cachemiche, R. Jacobsson, F. Hachon, M. Barros Marin, S. Baron, and K. Wyllie, “[Clock and timing distribution in the LHCb upgraded detector and readout system](#),” in *Conference poster*, no. Poster-2014-461, 2014. [48](#)
- [36] Altera Corporation, “[Transceiver Clocking in Stratix-V Devices](#),” *Stratix V Device Handbook*, vol. 3, 2012. [48](#), [49](#), [58](#), [61](#)
- [37] Intel Corporation, “[Intel[®] Arria[®] 10 Transceiver PHY User Guide](#),” *Arria 10 Device Handbook*, p. 615, 2015. [48](#), [49](#), [57](#), [58](#), [61](#), [82](#), [88](#), [107](#), [191](#)
- [38] R. Giordano and A. Aloisio, “[Fixed-latency, multi-gigabit serial links with Xilinx FPGAs](#),” *IEEE Transactions on Nuclear Science*, vol. 58, no. 1, pp. 194–201, 2011. [58](#)
- [39] J. Stephenson, D. Chen, R. Fung, and J. Chromczak, “[Understanding metastability in FPGAs](#),” *Altera Corporation white paper*, 2009. [59](#), [61](#), [82](#), [136](#)
- [40] J. Cachemiche, P. Duval, F. Hachon, R. Le Gac, and F. Réthoré, “[The PCIe-based readout system for the LHCb experiment](#),” *Journal of Instrumentation*, vol. 11, no. 02, p. P02013, 2016. [64](#)
- [41] M. Ferro-Luzzi and S. Baron, “[TYPICAL RF CLOCK FREQUENCY CHANGES: An Exhaustive lists](#),” *CERN Internal Slide*, 2010. [65](#)

- [42] S. Baron, T. Mastoridis, J. Troska, and P. Baudrenghien, “[Jitter impact on clock distribution in LHC experiments](#),” *Journal of Instrumentation*, vol. 7, no. 12, p. C12023, 2012. 68, 117, 118, 121
- [43] The ALICE Collaboration, “[Performance of the ALICE experiment at the CERN LHC](#),” *International Journal of Modern Physics A*, vol. 29, no. 24, p. 1430044, 2014. 68
- [44] ALICE Collaboration, “[ALICE trigger data-acquisition high-level trigger and control system](#),” *Technical Design Report ALICE*, 2004. 68
- [45] P. Antonioli, A. Kluge, W. Riegler, and for the ALICE Collaboration, “[Upgrade of the ALICE Readout & Trigger System](#),” *CERN Technical Design Report*, vol. CERN-LHCC-2013-019, ALICE-TDR-015, 2013. 68, 72
- [46] F. Costa, A. Kluge, P. V. Vyvre, and A. Collaboration, “[The detector read-out in ALICE during Run 3 and 4](#),” *Journal of Physics: Conference Series*, vol. 898, no. 3, 2017. 69
- [47] P. Buncic, M. Krzewicki, P. Vande Vyvre, and for the ALICE Collaboration, “[Upgrade of the Online-Offline Computing System](#),” *CERN Technical Design Report*, vol. CERN-LHCC-2015-006 ; ALICE-TDR-019, 2015. 69
- [48] A. Kluge, “[ALICE upgrade in LS2](#),” *ACECS 2016-Fifth Common ATLAS CMS Electronics Workshop for LHC Upgrades*, 2016. 69
- [49] X. C. C. Y. Kyung Suk (Dan) Oh, *High-Speed Signaling: Jitter Modeling, Analysis, and Budgeting*, 1st ed. Prentice Hall Modern Semiconductor Design Series, 2011. 73
-

- [50] P. Veríssimo, “[On the role of time in distributed systems](#),” in *Distributed Computing Systems, Proceedings of the Sixth IEEE Computer Society Workshop on Future Trends of*. IEEE, 1997, pp. 316–321. [73](#), [212](#)
- [51] P. Moreira, R. Ballabriga, S. Baron, S. Bonacini, O. Cobanoglu, F. Faccio, T. Fedorov, R. Francisco, P. Gui, P. Hartin *et al.*, “[The GBT project](#),” in *Proceedings of the Topical Workshop on Electronics for Particle Physics*, 2009, pp. 342–346. [75](#)
- [52] S. Baron, J. Cachemiche, F. Marin, P. Moreira, and C. Soos, “[Implementing the GBT data transmission protocol in FPGAs](#),” in *TWEPP-09 Topical Workshop on Electronics for Particle Physics*, 2009, pp. 631–635. [77](#)
- [53] S. Papadopoulos, I. Darwazeh, I. Papakonstantinou, J. Troska, and J. Mitchell, “[Passive Optical Networks for Particle Physics Applications](#),” *London Communications Symposium*, 2013. [78](#)
- [54] I. Papakonstantinou, C. Soos, S. Papadopoulos, S. Détraz, C. Sigaud, P. Stejskal, S. Storey, J. Troska, F. Vasey, and I. Darwazeh, “[A fully bidirectional optical network with latency monitoring capability for the distribution of timing-trigger and control signals in high-energy physics experiments](#),” *IEEE Transactions on Nuclear Science*, vol. 58, no. 4, pp. 1628–1640, 2011. [79](#)
- [55] E. Mendes, S. Baron, D. Kolotouros, C. Soos, and F. Vasey, “[The 10G TTC-PON: challenges, solutions and performance](#),” *Journal of Instrumentation*, vol. 12, no. 02, p. C02041, 2017. [80](#)
- [56] C. SOOS, D. KOLOTOUROS, V. VUJICIC, and S. BARON, “[TTC-PON History& Status](#),” *CERN Indico Event TTC-PON_ALICE_and_LHCb_meeting*, 11-06-2015. [80](#)

- [57] J. Mitra and T. K. Nayak, “[An FPGA-Based Phase Measurement System](#),” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 1, pp. 133–142, Jan 2018. 84
- [58] P. J. Davis and P. Rabinowitz, [Methods of numerical integration](#). Dover Publications, 2007. 90
- [59] R. Neil, “[Understanding Jitter and Wander Measurement and Standards](#),” *Agilent Technologies*, Feb, 2003. 91
- [60] Intel Corporation, “[Arria 10 Device](#),” *A10-Datasheet*, p. 28, 2016. 92
- [61] Silicon Laboratories, “[Si5345/44/42 Rev D Data Sheet](#),” *Datasheet*, 2017. 92
- [62] M. Schnecker, “[Jitter Transfer Measurement in Clock Circuits](#),” *LeCroy Corporation, DesignCon*, 2009. 96
- [63] J. Mitra and S. A. Khan, “[GBT BER measurement & Transceiver Optimization - For Arria10 FPGA Development board](#),” *CRU Internal Technical Note*, 2017. 98
- [64] J. Rudd, “[Statistical confidence levels for estimating error probability](#),” *Maxim Engineering Journal*, vol. 37, pp. 12–15, 2000. 99
- [65] S. Detraz, C. Sigaud, S. Seif El Nasr, I. Papakonstantinou, S. Papadopoulos, H. Versmissen, P. Moreira, C. Soos, P. Stejskal, S. Silva *et al.*, “[FPGA-based bit-error-rate tester for SEU-hardened optical links](#),” *JINST*, 2009. 100
- [66] Csaba SOOS, “[GBT protocol implementation on GBT protocol implementation on Xilinx FPGAs](#),” *LHCB meeting*, 2008. 101
- [67] Altera. Corporation, “[Extending Transceiver Leadership at 28 nm](#),” *White Paper*, 2012. 106

- [68] Altera Corporation, “[High-Speed link tuning using signal conditioning circuitry in Stratix-V transceivers](#),” *White Paper*, 2015. 106
- [69] Bob Blake, Altera Corporation, “[Ensuring Serial Protocol Signal Integrity with FPGAs and Embedded Transceivers](#),” *Web Article: Design & Reuse*, 2002. 106
- [70] Altera. Corporation, “[Understanding the Pre-Emphasis and Linear Equalization Features in Stratix-IV GX Devices](#),” *Application Note*, 2010. 106
- [71] CRU Team Members. (2018) [ALICE CRU Hardware, Firmware, and Software Development–ALICE Web TWiki](#). 116
- [72] E. Stott and P. Y. K. Cheung, “[Improving FPGA reliability with wear-levelling](#),” *Proceedings - 21st International Conference on Field Programmable Logic and Applications, FPL 2011*, pp. 323–328, 2011. 116
- [73] M. Wang, A. Ailamaki, and C. Faloutsos, “[Capturing the spatio-temporal behavior of real traffic data](#),” *Performance Evaluation*, vol. 49, no. 1-4, pp. 147–163, 2002. 116
- [74] E. Mendes *et al.*, “[The 10G TTC-PON: challenges, solutions and performance](#),” *Journal of Instrumentation*, vol. 12, no. 02, p. C02041, 2017. 117
- [75] Silicon Labs, “[10-Channel, Any-Frequency, Any-Output Jitter Attenuator/ Clock Multiplier](#),” *Technical Note*, 2016. 118
- [76] K. Mustafa, “[Defining skew, propagation-delay, phase offset \(phase error\)](#),” *Texas Instruments Incorporated, High Performance Analog/CDC*, pp. 1–10, 2001. 121
- [77] X. Liu *et al.*, “[Design and FPGA implementation of high-speed, fixed-latency serial transceivers](#),” *IEEE Transactions on Nuclear Science*, vol. 61, no. 1, pp. 561–567, 2014. 129

- [78] S. J. Lu *et al.*, “Real-time differential signal phase estimation for space-based systems using FPGAs,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 2, pp. 1192–1209, 2013. 130, 150
- [79] D. Trebbels *et al.*, “High precision Phase Measurement technique for Cell Impedance Spectroscopy,” in *Journal of Physics: Conference Series*, vol. 224, no. 1. IOP Publishing, 2010, p. 012159. 130, 150
- [80] L. Rong *et al.*, “FPGA-based amplitude and phase detection in DLLRF,” *Chinese physics C*, vol. 33, no. 7, p. 594, 2009. 130, 150
- [81] A. Hidvégi *et al.*, “FPGA based phase detector for high-speed clocks with picoseconds resolution,” in *Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*. IEEE, 2013, pp. 1–4. 130, 150
- [82] D. W. Kuddes, “Method and system for aligning the phase of high speed clocks in telecommunications systems,” 1997, US Patent 5,638,410. 130
- [83] H. Choi, A. V. Gomes, and A. Chatterjee, “Signal acquisition of high-speed periodic signals using incoherent sub-sampling and back-end signal reconstruction algorithms,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 7, pp. 1125–1135, 2011. 130
- [84] A. V. Oppenheim, *Discrete-time signal processing*. Pearson Education India, 1999. 132
- [85] S. K. Thompson, *Sampling*. Wiley, 2012. 134
- [86] D. Chow, V. Tsui, and S. Wong, “A Jitter Estimation Method for Cascaded, Programmable Phase Locked Loops,” *DesignCon, CP-01036-1.0*, 2008. 136
- [87] Altera Corporation, UG-01087, “Altera Phase-Locked Loop (Altera PLL) IP Core,” *User Guide*, 2017. 136

- [88] C. Barrett, “[Fractional/Integer-N PLL Basics](#),” *Texas Instrument Technical Brief, Wireless Communication Business Unit, Technical Brief SWRA029*, 1999. 137
- [89] Silicon Labs, “[Clock Division with Jitter and Phase Noise Measurements](#),” *White Paper*, vol. Rev-1.0, 2015. 137
- [90] Analog Devices, “[Fundamentals of Phase Locked Loops \(PLLs\)](#) ,” *MT-086 Tutorial*, 2009. 137
- [91] X. Gao, E. A. Klumperink, M. Bohsali, and B. Nauta, “[A low noise sub-sampling PLL in which divider noise is eliminated and PD/CP noise is not multiplied by \$N^2\$](#) ,” *IEEE journal of solid-state circuits*, vol. 44, no. 12, pp. 3253–3263, 2009. 137
- [92] K. R. Wadleigh and I. L. Crawford, *Software optimization for high-performance computing*. Prentice Hall Professional, 2000. 138
- [93] JEDEC Solid State Technology Association, JEDEC Standard JESD65B, “[Definition of Skew Specifications for Standard Logic Devices](#),” 2003. 142, 143
- [94] Altera Corporations, “[Using Timing Analysis in the Quartus II Software](#),” *Application Note*, vol. A-AN-123-02, 2001. 143
- [95] Altera Corporation, “[The Need for Dynamic Phase Alignment in High-Speed FPGAs](#) ,” *White Paper*, February 2004. 145
- [96] J. Taylor, *Introduction to error analysis, the study of uncertainties in physical measurements*. University Science Books, 1997. 146
- [97] R. S. Turgel and D. F. Vecchia, “[Precision calibration of phase meters](#),” *IEEE Transactions on Instrumentation and Measurement*, vol. 1001, no. 4, pp. 918–922, 1987. 147

- [98] K. R. Vijay and A. K. M. Saleh, Ehsanes, *An Introduction to Probability and Statistics*. John Wiley & Sons, 2015. 147
- [99] A. Guntoro *et al.*, “Implementation of Realtime and Highspeed phase detector on FPGA,” in *International Workshop on Applied Reconfigurable Computing*. Springer, 2006, pp. 1–11. 150
- [100] S. M. Sait and W. Hasan, “Hardware design and VLSI implementation of a byte-wise CRC generator chip,” *IEEE Transactions on Consumer Electronics*, vol. 41, no. 1, pp. 195–200, 1995. 155
- [101] J. Stone, M. Greenwald, C. Partridge, and J. Hughes, “Performance of checksums and CRCs over real data,” *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 529–543, 1998. 155
- [102] T.-B. Pei and C. Zukowski, “High-speed parallel CRC circuits in VLSI,” *IEEE Transactions on Communications*, vol. 40, no. 4, pp. 653–657, 1992. 155
- [103] T. Henriksson, H. Eriksson, U. Nordqvist, P. Larsson-Edefors, and D. Liu, “VLSI implementation of CRC-32 for 10 Gigabit Ethernet,” in *The 8th IEEE International Conference on Electronics, Circuits and Systems(ICECS)*, vol. 3. IEEE, 2001, pp. 1215–1218. 155
- [104] C. Toal, K. McLaughlin, S. Sezer, and X. Yang, “Design and implementation of a field programmable CRC circuit architecture,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 8, pp. 1142–1147, 2009. 155
- [105] J. Mitra and T. K. Nayak, “Reconfigurable Concurrent VLSI (FPGA) Design Architecture of CRC-32 for High-Speed Data Communication,” in *2015 IEEE International Symposium on Nanoelectronic and Information Systems*. IEEE, 2015, pp. 112–117. 155

- [106] P. Koopman and T. Chakravarty, “[Cyclic redundancy code \(CRC\) polynomial selection for embedded networks](#),” in *International Conference on Dependable Systems and Networks*. IEEE, 2004, pp. 145–154. 156
- [107] G. Cook, “[Catalogue of parametrised CRC algorithms](#),” 2010. 156
- [108] S. G. Pestana, E. Rijpkema, A. Radulescu, K. Goossens, and O. P. Gangwal, “[Cost-performance trade-offs in networks on chip: A simulation-based approach](#),” in *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, vol. 2. IEEE, 2004, pp. 764–769. 162
- [109] I. Kuon and J. Rose, “[Area and delay trade-offs in the circuit and architecture design of FPGAs](#),” in *Proceedings of the 16th international ACM/SIGDA symposium on Field programmable gate arrays*. ACM, 2008, pp. 149–158. 162
- [110] A. Singh, A. Mukherjee, L. Macchiarulo, and M. Marek-Sadowska, “[PITIA: an FPGA for throughput-intensive applications](#),” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 3, pp. 354–363, 2003. 163
- [111] L. Kleinrock, “[The latency/bandwidth tradeoff in gigabit networks](#),” *IEEE Communications Magazine*, vol. 30, no. 4, pp. 36–40, 1992. 163
- [112] A. J. Bernstein, “[Analysis of programs for parallel processing](#),” *IEEE Transactions on Electronic Computers*, no. 5, pp. 757–763, 1966. 163
- [113] Altera Corporation, “[Quartus II Handbook Version 13.1, QII5V1-13.1.0](#),” *Volume 1: Design and Synthesis*, 2013. 169, 171, 173
- [114] D. E. Knuth, [The Art of Computer Programming](#). Addison-Wesley Professional, 3rd Revised ed., 2011. 174

REFERENCES

- [115] M. E. Kounavis and F. L. Berry, “[Novel table lookup-based algorithms for high-performance CRC generation](#),” *IEEE Transactions on Computers*, vol. 57, no. 11, pp. 1550–1560, 2008. 174
- [116] S. Kilts, *Advanced FPGA design: architecture, implementation, and optimization*. John Wiley & Sons, 2007. 179, 180
- [117] M. Mac and C. Wysocki, “[Guaranteeing Silicon Performance with FPGA Timing Models](#),” *White Paper*, no. WP-01139-1.0, 2010. 179
- [118] M. Walma, “[Pipelined cyclic redundancy check \(CRC\) calculation](#),” in *Proceedings of 16th International Conference on Computer Communications and Networks, ICCN*. IEEE, 2007, pp. 365–370. 186
- [119] J.-S. Lin, C.-K. Lee, M.-D. Shieh, and J.-H. Chen, “[High-speed CRC design for 10 Gbps applications](#),” in *2006 IEEE International Symposium on Circuits and Systems*. IEEE, 2006, pp. 4–pp. 186
- [120] M.-H. S. Ming-Der Shieh, C.-h. Chen, and H.-F. Lo, “[A systematic approach for parallel CRC computations](#),” *Journal of information science and engineering*, vol. 17, pp. 445–461, 2001. 186
- [121] H. M. Ji and E. Killian, “[Fast parallel CRC algorithm and implementation on a configurable processor](#),” in *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 3. IEEE, 2002, pp. 1813–1817. 186
- [122] V. Pless, “[On the uniqueness of the Golay codes](#),” *Journal of Combinatorial theory*, vol. 5, no. 3, pp. 215–228, 1968. 190
- [123] M. Kimmitt, “[Interleaver for burst error correction](#),” Dec. 2003, US Patent 6,662,332. 190

- [124] D. Amrany, “[Self-synchronizing scrambler/descrambler without error multiplication](#),” jun 1996, uS Patent 5,530,959. 190
- [125] N. F. Pub, “[Advanced encryption standard \(AES\)](#),” *Federal Information Processing Standards Publication*, vol. 197, pp. 441–0311, 2001. 190, 192
- [126] Texas Instruments, “[CDCE62005 3:5 Clock Generator, Jitter Cleaner with Integrated Dual VCOs](#),” *Datasheet*, 2015. 191
- [127] Z.-l. YANG and Y.-j. LIU, “[Algorithm research of self-synchronizing scrambler sequence](#),” *Information Technology*, vol. 2, pp. 30–32, 2005. 191
- [128] R. H. Morelos-Zaragoza, *[The art of error correcting coding](#)*. John Wiley & Sons, 2006. 195, 196
- [129] S. B. Wicker, *[Error control systems for digital communication and storage](#)*. Prentice hall Englewood Cliffs, 1995, vol. 1. 196
- [130] Y. Kim and J.-L. Pan, “[OVSF code system and methods](#),” Apr. 2003, uS Patent 6,552,996. 198
- [131] F. Mazda, *[Telecommunications engineer’s reference book](#)*. Butterworth-Heinemann, 2014. 199
- [132] K. G. Coffman and A. M. Odlyzko, “[Internet growth: Is there a “Moore’s Law” for data traffic?](#)” in *Handbook of massive data sets*. Springer, 2002, pp. 47–93. 208
- [133] R. Moore, C. Baru, R. Marciano, A. Rajasekar, and M. Wan, “[Data-intensive computing](#),” *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, pp. 105–129, 1999. 208
- [134] V. Gligorov, “[Real-time data analysis at the LHC: present and future](#),” in *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, 2015, pp. 1–18. 208, 210

- [135] J. Anderson *et al.*, “[FELIX: a High-Throughput Network Approach for Interfacing to Front End Electronics for ATLAS Upgrades](#),” in *Journal of Physics: Conference Series*, vol. 664, no. 8. IOP Publishing, 2015, p. 082050. 209
- [136] W. Zabolotny *et al.*, “[Versatile prototyping platform for Data Processing Boards for CBM experiment](#),” *Journal of Instrumentation*, vol. 11, no. 02, p. C02031, 2016. 209
- [137] J. Cachemiche, P. Duval, F. Hachon, R. L. Gac, and F. Réthoré, “[The PCIe-based readout system for the LHCb experiment](#),” p. P02013, 2016. 209
- [138] J. Lister, J. Farthing, M. Greenwald, and I. Yonekawa, “[The ITER CODAC conceptual design](#),” *Fusion Engineering and Design*, vol. 82, no. 5, pp. 1167–1173, 2007. 209
- [139] G. Vouters and On behalf of the LHCb Collaboration, “[Technological Aspect of the Trigger-Less Readout Architecture for the LHCb Upgrade at CERN](#),” in *IEEE Nuclear Science Symposium and Medical Imaging Conference*, 2013. 210
- [140] F. Cristian and C. Fetzer, “[The Timed Asynchronous Distributed System Model](#),” *IEEE Transactions on Parallel and Distributed systems*, vol. 10, no. 6, pp. 642–657, 1999. 212
- [141] C. Almeida and P. Verissimo, “[Timing failure detection and real-time group communication in quasi-synchronous systems](#),” in *Real-Time Systems*. IEEE, 1996, pp. 230–235. 212
- [142] M. J. Fischer, N. A. Lynch, and M. S. Paterson, “[Impossibility of distributed consensus with one faulty process](#),” *Journal of the ACM (JACM)*, vol. 32, no. 2, pp. 374–382, 1985. 212

- [143] J. Fisher, T. A. Henzinger, M. Mateescu, and N. Piterman, “[Bounded asynchrony: Concurrency for modeling cell-cell interactions](#),” in *Formal Methods in Systems Biology*. Springer, 2008, pp. 17–32. 214
- [144] D. Kolotouros, S. Baron, M. B. Marin, C. Fountas, C. Soos, J. Troska, F. Vasey, and P. Vichoudis, “[Metrics and methods for TTC-PON system characterization](#),” *Journal of Instrumentation*, vol. 9, no. 01, p. C01015, 2014. 215
- [145] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne *et al.*, “[The FAIR Guiding Principles for scientific data management and stewardship](#),” *Scientific data*, vol. 3, 2016. 221
- [146] P. Galison and B. W. Hevly, *Big science: The growth of large-scale research*. Stanford University Press, 1992. 221
- [147] J. Mościcki, “[Distributed analysis environment for HEP and interdisciplinary applications](#),” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 502, no. 2, pp. 426–429, 2003. 221
- [148] C. Hoffa, G. Mehta, T. Freeman, E. Deelman, K. Keahey, B. Berriman, and J. Good, “[On the use of cloud computing for scientific workflows](#),” in *IEEE Fourth International Conference on eScience, eScience’08*. IEEE, 2008, pp. 640–645. 221
- [149] N. Neufeld, “[Future of DAQ Frameworks and Approaches, and Their Evolution towards the “Internet of Things”](#),” in *Journal of Physics: Conference Series*, vol. 664, no. 8. IOP Publishing, 2015, p. 082038. 221
- [150] I. Bird, B. Jones, and K. F. Kee, “[The Organization and Management of Grid Infrastructures](#),” *IEEE Computer Society*, 2009. 221

REFERENCES

- [151] E. Gibney *et al.*, “[LHC plans for open data future](#),” *Nature*, vol. 503, no. 7477, pp. 447–447, 2013. 222
- [152] R. Kogler, D. M. South, M. Steder, I. D. S. Group *et al.*, “[Data preservation in High Energy Physics](#),” in *Journal of Physics: Conference Series*, vol. 368, no. 1. IOP Publishing, 2012, p. 012026. 222
- [153] M. D. Hildreth, “[Data and Software Preservation for Open Science \(DASPOS\)](#),” in *IEEE 30th International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 2014, pp. 142–146. 222