

University of Windsor

Scholarship at UWindor

Electronic Theses and Dissertations

Theses, Dissertations, and Major Papers

10-5-2017

Experimental Evaluation and Comparison of Time-Multiplexed Multi-FPGA Routing Architectures

Asmeen Kashif
University of Windsor

Follow this and additional works at: <https://scholar.uwindsor.ca/etd>

Recommended Citation

Kashif, Asmeen, "Experimental Evaluation and Comparison of Time-Multiplexed Multi-FPGA Routing Architectures" (2017). *Electronic Theses and Dissertations*. 7269.
<https://scholar.uwindsor.ca/etd/7269>

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

**Experimental Evaluation and Comparison of Time-Multiplexed Multi-FPGA
Routing Architectures**

By

Asmeen Kashif

A Dissertation

Submitted to the Faculty of Graduate Studies
through the Department of **Electrical and Computer Engineering**
in Partial Fulfillment of the Requirements for
the Degree of **Doctor of Philosophy**
at the University of Windsor

Windsor, Ontario, Canada

2017

© 2017 Asmeen Kashif

Experimental Evaluation and Comparison of Time-Multiplexed Multi-FPGA Routing
Architectures

by

Asmeen Kashif

APPROVED BY:

F. Gebali, External Examiner
Department of Electrical & Computer Engineering
University of Victoria

W. Abdul-Kader
Department of Mechanical, Automotive & Materials Engineering

J. Wu
Department of Electrical & Computer Engineering

H. Wu
Department of Electrical & Computer Engineering

M. Khalid, Advisor
Department of Electrical & Computer Engineering

August 24, 2017

DECLARATION OF PREVIOUS PUBLICATIONS

This thesis includes 3 original papers that have been previously submitted for publication in peer reviewed journals, as follows:

Thesis Chapter	Publication title/full citation	Publication status*
<i>Chapter [3] [6] [7]</i>	<i>Experimental Evaluation and Comparison of Time-Multiplexed Schemes and Routing Architectures for Multi-FPGA Systems</i>	<i>submitted</i>
<i>Chapter [4] [5] [7]</i>	<i>Latency-Optimized 3D Multi-FPGA System with Serial Optical Interface</i>	<i>submitted</i>
<i>Chapter [4] [6] [7]</i>	<i>Experimental Evaluation and Comparison of Latency-Optimized Optical and Conventional Multi-FPGA Systems</i>	<i>submitted</i>

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include

such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Emulating large complex designs require multi-FPGA systems (MFS). However, inter-FPGA communication is confronted by the challenge of lack of interconnect capacity due to limited number of FPGA input/output (I/O) pins. Serializing parallel signals onto a single trace effectively addresses the limited I/O pin obstacle. Besides the multiplexing scheme and multiplexing ratio (number of inter-FPGA signals per trace), the choice of the MFS routing architecture also affect the critical path latency. The routing architecture of an MFS is the interconnection pattern of FPGAs, fixed wires and/or programmable interconnect chips. Performance of existing MFS routing architectures is also limited by off-chip interface selection.

In this dissertation we proposed novel 2D and 3D latency-optimized time-multiplexed MFS routing architectures. We used rigorous experimental approach and real sequential benchmark circuits to evaluate and compare the proposed and existing MFS routing architectures. This research provides a new insight into the encouraging effects of using off-chip optical interface and three dimensional MFS routing architectures. The vertical stacking results in shorter off-chip links improving the overall system frequency with the additional advantage of smaller footprint area. The proposed 3D architectures employed serialized interconnect between intra-plane and inter-plane FPGAs to address the pin limitation problem. Additionally, all off-chip links are replaced by optical fibers that exhibited latency improvement and resulted in faster MFS. Results indicated that exploiting third dimension provided latency and area improvements as compared to 2D MFS.

We also proposed latency-optimized planar 2D MFS architectures in which electrical interconnections are replaced by optical interface in same spatial distribution. Performance evaluation and comparison showed that the proposed architectures have reduced critical path delay and system frequency improvement as compared to conventional MFS.

We also experimentally evaluated and compared the system performance of three inter-FPGA communication schemes i.e. Logic Multiplexing, SERDES and MGT in conjunction with two routing architectures i.e. Completely Connected Graph (CCG) and TORUS. Experimental results showed that SERDES attained maximum frequency than the other two schemes. However, for very high multiplexing ratios, the performance of SERDES & MGT became comparable.

DEDICATION

Alhumdulillah (Praise to Allah) for this important achievement of my life.

I am very thankful to my husband for his continuous moral support and encouragement not only throughout this thesis work but also throughout our lives. This achievement would not have been possible without him. I am grateful to my parents for their prayers and well wishes. And last but not the least, I would like to thank my two precious and perfect kids Reyan and Manal for being incredibly understanding and patient during my work.

ACKNOWLEDGEMENTS

To my supervisor, Prof. Dr. Khalid. Thank you for the guidance, support and encouragement during my dissertation work. I have learnt a lot under your supervision. To my committee members, Dr. H. Wu, Dr. J. Wu and Dr. Abdul-Kader for agreeing to be the external reader. Thank you for all the comments and suggestions on my project. I would like to thank Prof. Dr. Rashid for allowing me to use the RCIM lab facility. I also want to say thank you to Ms. Andria Ballo and Mr. Frank Cicchello for your administrative support and helping me out whenever I needed it.

TABLE OF CONTENTS

DECLARATION OF PREVIOUS PUBLICATIONS.....	iii
ABSTRACT.....	v
DEDICATION.....	vii
ACKNOWLEDGEMENTS.....	viii
LIST OF FIGURES	xiii
LIST OF TABLES.....	xv
LIST OF ABBREVIATIONS.....	xvi

Chapter 1 Introduction

1.1. Multi-FPGA System (MFS).....	1
1.2. Multi-FPGA System Constraints	2
1.2.1. Pin Limitation Problem.....	2
1.2.2. Off-Chip Communication Strategy.....	3
1.2.3. Inter-FPGA Interface Selection	4
1.3. Thesis Goals.....	4
1.4. Thesis Contributions	5
1.5. Thesis Organization	6

Chapter 2 MFS Routing Architectures

2.1. Inter-FPGA Connections & Routing	8
2.2. Types of Inter-FPGA Connections	8
2.2.1. Hard-wired Connection	8
2.2.2. Cabling Connection	9
2.2.3. Optical Connection	11
2.3. MFS Routing Architectures	11

2.3.1.	Basic Assumptions.....	12
2.3.2.	2D and 3D Routing Architectures	13
2.4.	Previous Research on MFS Routing Architectures	14
2.4.1.	Linear Arrays	14
2.4.2.	Mesh Architectures	16
2.4.3.	Programmable Routing Architectures	19
2.4.4.	Tree Topology	21
2.4.5.	Other MFS Routing Architectures.....	22
2.5.	Summary	23

Chapter 3 Time Multiplexing in MFS

3.1	Introduction.....	25
3.2	Critical Path Delay	25
3.3	Logic Multiplexing	27
3.4	SERDES.....	29
3.4.1	LVDS Signaling	29
3.4.2	SERDES Architecture	31
3.4.3	SERDES Multiplexing	33
3.5	Multi-Gigabit Transceiver (MGT).....	35
3.5.1	CML Signaling.....	35
3.5.2	MGT Architecture	36
3.5.3	MGT Multiplexing	39
3.6	Comparison of Three Multiplexing Schemes	42
3.7	Previous Research on MFS multiplexing	43
3.8	Summary	47

Chapter 4 Optical Interface in MFS

4.1	Introduction.....	48
4.2	Short-Range Optical Interface	49
4.2.1	Optical Fibers	50
4.2.2	Optical Transceivers.....	54

4.3	MFS Serial Optical Interface	60
4.4	Proposed Latency-Optimized 2D MFS with Optical Interface.....	61
4.4.1	Proposed Architecture	61
4.4.2	Multiplexing in Proposed Architecture	65
4.4.3	Evaluation Strategy	65
4.5	Previous Research on MFS Serial Optical Interface.....	65
4.6	Summary	71

Chapter 5 Optical 3D MFS Routing Architectures

5.1	Introduction.....	72
5.2	Why 3D MFS Architecture.....	73
5.2.1	Interconnection Length Distribution	73
5.2.2	Asymptotic Behavior of Wire-length.....	74
5.2.3	Structural Distribution & Placement Optimization.....	76
5.3	Proposed 3D MFS Architecture.....	76
5.3.1	Motivation	76
5.3.2	Proposed 3D Optical MFS Routing Architectures	77
5.3.3	Multiplexing in 3D Routing Architectures	79
5.3.4	Evaluation Strategy	80
5.4	Previous Research on 3D MFS	81
5.5	Summary	83

Chapter 6 CAD Tools and Experimental Evaluation Framework

6.1	Experimental Design Mapping Flow	84
6.2	Assumptions.....	86
6.2.1	FPGA Pin Assignment	86
6.2.2	Intra-FPGA Placement and Routing	87
6.3	CAD Tools	87
6.3.1	ABC Tech-Mapper.....	87
6.3.2	Translator.....	88
6.3.3	Multi-way Partitioning	88
6.3.4	Placement	89

6.3.5	MFS Timing Analyzer.....	91
6.3.6	Time-Multiplexed Inter-FPGA Router.....	94
6.4	Evaluation Metric.....	95
6.4.1	Emulation Time and System Frequency	95
6.5	Benchmark Circuits	96
6.6	Summary	97

Chapter 7 Experimental Results and Comparison of Architectures

7.1	Introduction.....	98
7.2	Comparison of Multiplexed Routing Architectures.....	98
7.2.1	Critical Path Delay	98
7.2.2	System Frequency	102
7.3	Comparison of Proposed 2D Optical & Conventional MFS	106
7.3.1	Critical Path Delay	106
7.3.2	System Frequency	107
7.4	Comparison of Proposed 3D Optical & Conventional MFS	115
7.4.1	Critical Path Delay	115
7.4.2	System Frequency	117
7.5	Summary	125

Chapter 8 Conclusions and Future Work

8.1	Dissertation Summary.....	127
8.2	Principal Contributions	129
8.3	Future Directions	130

References	131
-------------------------	------------

Vita Auctoris.....	140
---------------------------	------------

LIST OF FIGURES

1.1	Multi-FPGA Board;DN7020K10 (DiniGroup).....	2
2.1	Hard-wired MFS (DNV7F4A).....	9
2.2	Photograph of 2 TwinStar FPGA Systems	10
2.3	Synopsys' HapsTrak 3 Connector Technology	11
2.4	Routing Architectures (a) CCG, (b) TORUS.....	14
2.5	The AnyBoard System.....	15
2.6	(a) Basic Mesh Architecture (b) 8-way Mesh (c) One-Hop Mesh.....	17
2.7	Maxwell FPGA Connectivity	19
2.8	(a) Full Crossbar (b) Partial Crossbar	20
2.9	BEE2 System Topology.....	21
2.10	HCGP Architecture	22
3.1	Critical Path Delays	27
3.2	Logic Multiplexing Scheme.....	28
3.3	LVDS Architecture	30
3.4	Generic SERDES Architecture	32
3.5	SERDES Multiplexing Scheme	33
3.6	CML Architecture	36
3.7	Multi Gigabit Transceiver Architecture.....	38
3.8	MGT Multiplexing Scheme	40
3.9	NoC Emulation Board.....	44
4.1	Optical Interface Evolution.....	49
4.2	(a) Single-Mode (b) Multi-Mode Optical Fiber Core Dimensions.....	52
4.3	Optical Transmitter Block Diagram.....	55
4.4	Optical Receiver Block Diagram	56
4.5	Optical FPGA &Transceivers	58
4.6	Simplified inter-FPGA serial optical interface structure	61
4.7	2D MFS Routing Architectures (a) CCG (b) Torus	62

4.8	Demonstrator design with the 3 optoelectronic FPGA chips and encapsulated Optical pathway	63
4.9	Samtec FireFly™ Micro Flyover System	64
4.10	High level view of the hardware setup	66
4.11	Block Diagram	69
4.12	Block diagram of remote configuration on LTDB Demonstration.....	71
5.1	Comparison of interconnection length distribution for 2D & 3D architectures	74
5.2	Possible Combination Classes in 3D (a) A-combination (b) N-combination (c) R-combination.....	75
5.3	2D MFS Architecture.....	77
5.4	3D MFS topologies with various degrees of optical interconnect.....	78
6.1	Design Flow for Time-Multiplexed MFS	85
7.1	Number of Inter-FPGA nets After Partitioning	99
7.2	System Frequency vs Multiplexing Ratio.....	102
7.3	System Frequency 2D Conventional vs Optical MFS in CCG & TORUS	108
7.4	System Frequency Gain 2D Optical vs Conventional MFS in CCG & TORUS.....	112
7.5	Average System Frequency Gain 2D Optical vs Conventional CCG & TORUS	115
7.6	System Frequency 2D vs 3D with MGT Multiplexing Scheme	118
7.7	System Frequency Gain 2D vs 3D with MGT Multiplexing Scheme	121
7.8	Average System Frequency Gain 2D vs 3D in CCG & TORUS Routing Architectures	125

LIST OF TABLES

3.1	Comparison of LVDS & CML	36
3.2	Latency Values of GTY TX & RX Blocks	41
3.3	Comparison of 3 Multiplexing Schemes.....	42
4.1	Comparison between MMF & SMF	53
6.1	Delay Values Used in Static Timing Analyzer	92
6.2(a)	Benchmark Circuits	96
6.2(b)	Benchmark Circuit	97
7.1	KaFFPaE Partitioning Results	99
7.2	Threshold Multiplexing factor ($mux_{threshold}$) for Multiplexed Routing MFS	100
7.3	Critical Path Delays (in nanoseconds) at Different levels of Circuit Implementation for 2D MFS	101
7.4	Critical Path Delays (in nanoseconds) at Different levels of Circuit Implementation for 2D MFS	106
7.5	Critical Path Delays (in nanoseconds) at Different levels of Circuit Implementation for 2D Optical MFS	107
7.6	Critical Path Delays (in nanoseconds) at Different levels of Circuit Implementation for 3D Optical MFSs.....	116

LIST OF ABBREVIATIONS

2D	Two-dimensional
3D	three-dimensional
ASIC	Application Specific Integrated Circuit
CCG	Completely Connected Graph
CLB	Configurable Logic Block
CML	Current Mode Logic
CPD	Critical Path Delay
CPU	Central Processing Unit
DSP	Digital Signal Processing
EMI	Electro-magnetic Interference
FPGA	Field Programmable Gate Array
FPID	Field Programmable Interconnect Device
Gbps	Giga bits per second
GHz	Giga Hertz
HPC	High Performance Computing
IC	Integrated Circuit
InP	Indium Phosphide
I/O	Input / Output
KaHIP	Karlsruhe High Quality Partitioning
LUT	Look-Up Table
LVDS	Low Voltage Differential Signaling
MFS	Multi-FPGA System
MHz	Mega Hertz
MGT	Multi Gigabit Transceiver
MMF	Multi Mode Fiber
PCB	Printed Circuit Board
PLL	Phase Locked Loop
PMMA	Ploy Methyl Methacrylate
POF	Plastic Optical Fiber
RX	Receiver

SERDES	SERializer / DESerializer
SFP	Small Form-Factor Pluggable
SMF	Single Mode Fiber
SoC	System on Chip
STA	Static Timing Analyzer
TDM	Time Division Multiplexing
THz	Tera Hertz
TX	Transmitter

Chapter 1 Introduction

1.1. Multi-FPGA System (MFS)

Today's general purpose microprocessors are optimized for general purpose applications. This implies that the user has to optimize his code for the processor as it is physical predefined silicon that cannot be modified to fit to user's application. Custom ICs for specific applications such as encryption use hardware that cannot be changed after fabrication. The Field Programmable Gate Array, in short "FPGA", is different. This chip allows the user to modify the silicon through software configuration to be the ideal Application Specific Circuit or short "ASIC" for user defined application, while remaining reconfigurable. Currently, a few to tens of such FPGAs are used for emulating millions of logic gates and for accelerating computationally intensive applications. Super computer level performance can be achieved at a fraction of the cost for high performance computing (HPC) applications.

Multi-FPGA systems are an important area of research. These systems connect multiple FPGAs in a fixed pattern, to implement complex logic circuits as shown in Figure 1.1. They offer the potential to deliver higher performance solutions to general computing tasks, logic emulation, rapid prototyping and reconfigurable custom computing machines [1]. Multi-FPGA boards for ASICs prototyping and High Performance Computing (HPC) enable high-speed, accurate prototyping and emulation, system and IP design. "These platforms also facilitate sub-microsecond latency market data processing and order execution and allow orders of magnitude higher performance for algorithmic trading as well as options pricing and risk management, over conventional software-based and hybrid approaches" [85]. MFS designed for individual applications can be further optimized by the type of FPGAs and interconnections employed.

In addition to FPGAs, almost all MFSs have memory chips and other devices such as CPUs, Ethernet ports, expansion slots, External Clock inputs / outputs (I/Os) and DSP blocks providing high-density supercomputing resources to a wider range of audience. For

example, the commercial platform DN7020K10 configured with 20 Intel/Altera Stratix 4SE820s can emulate up to 130 million logic gates [23].

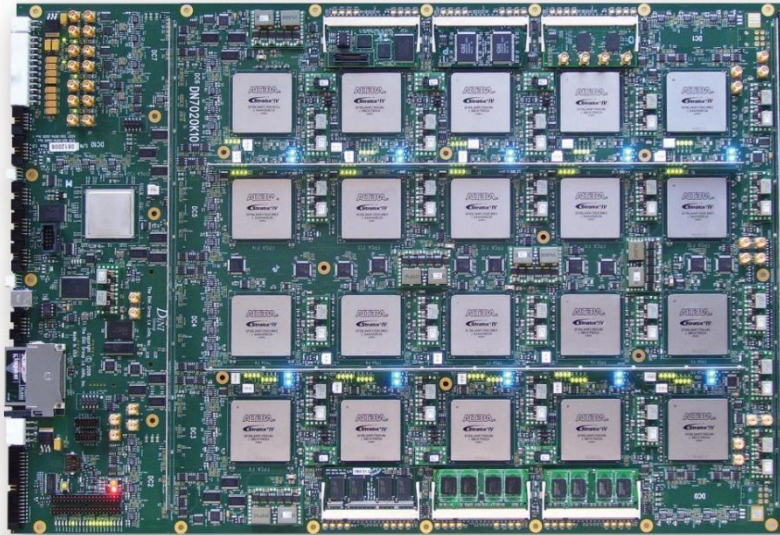


Figure 1.1: Multi-FPGA Board; DN7020K10 (*DiniGroup*) [23]

SciEngines offers RIVYERA S6-LX150; a 128 Xilinx Spartan-6 LX150 FPGA cluster with external memory and CPU cores representing high performance reconfigurable platforms [3].

However, there are multiple factors that must be taken into account in a multi-FPGA design to achieve desired performance from these systems.

1.2. Multi-FPGA System Constraints

1.2.1. Pin Limitation Problem

The first constraint of an MFS is the limited number of I/O pins. Over the past few years, the logic capacity per FPGA is increasing at a much faster pace as compared to the number of I/O pins. Large SoCs may not be routed among multiple FPGAs without overflowing the available I/O resources of a single FPGA [5]. Mapping a design to an MFS is mainly divided into two steps. In the first step, the design is partitioned into several parts. A successful partitioning approach ensures that every part fits within the logic capacity of the

single FPGA in MFS. The second step routes the inter-FPGA nets according to the available physical tracks, I/O resources of the FPGA and the routing architecture of MFS. But, out of these available pins, some need to be reserved for non-FPGA connections and in case of differential signaling, some need to be reserved to propagate the clock instead of the user data. Consequently, the number of available pins for inter-FPGA data communication is further decreased. One of the solutions to this issue can be to alter the partitioning of the design which can change the number of inter-FPGA nets [6]. However, re-partitioning does not solve the problem in every design.

1.2.2. Off-Chip Communication Strategy

Although MFSs are capable of accommodating large designs, their off-chip communication strategy imposes bandwidth constraints and limits the overall system performance [1, 4]. The selection of the MFS routing architecture exercises considerable effect on the critical path delay and system frequency of a design. The routing architecture of an MFS is the manner in which the FPGAs, fixed wires and/or programmable interconnect chips are connected together. In certain routing topologies providing full connectivity, signals can be routed from source FPGA to its destination FPGA via direct off-chip connections without any interference. However, in other routing architectures, sometimes the signals need an intermediate FPGA or a route-through to reach the destination FPGA. In such a case, the signal is sent into one pin of the route-through FPGA, through the on-chip routing and then out through the other pin, without using any of the on-chip logic. Such inter-FPGA nets inflict even larger delays than the direct connections and adversely affect the system frequency. Since the I/O pin and off-chip routing delays are much larger than the on-chip delays in an MFS, that's why the speed of the implemented design is primarily dictated by them. Moreover, the routing resources consume significant board area and scaling up an MFS only aggravates the latency, area and cost issues. Therefore, selection of the appropriate routing architecture is vital in determining the system performance.

1.2.3. Inter-FPGA Interface Selection

Over the last few decades, applications' data bandwidth requirements are constantly increasing which demand a compatible high-speed interface capable of maintaining multi-gigabits data rate. Generally, designers employ copper interconnect for chip-to-chip and chip-to-module interfaces over traces on a printed circuit board (PCB). However, copper based interconnects are incapable of scaling up with the data rate due to the frequency dependent losses. For instance, FR-4 copper trace material suffers from a loss of ~ 0.5 - 1.5 dB/inch at 5 GHz (Nyquist for 10 Gbps rate), and the loss increases to ~ 2.0 - 3.0 dB/inch at 12.5 GHz (Nyquist for 25 Gbps rate) [7]. Maximum bandwidth is also limited by return loss, insertion loss and crosstalk. In present technology, designers use copper electrical interface in MFS, however, at multi-gigabit data rates, inter-FPGA electrical interconnections are restricted in their performance due to signal integrity, latency, power and cost issue. Therefore, designers are exploring the idea of applying short-range optical fiber signaling in order to overcome these challenges.

Unlike copper interfaces, optical fiber has virtually no loss and its power consumption and penalty is relatively independent of reach length, Moreover, optical interface is immune to electromagnetic interference (EMI) and does not have amplitude crosstalk, resulting in better signal integrity resilience. Replacing PCB traces with an optical interface in MFS can provide significant power, resource, and cost reductions. Thus, the choice of off-chip interconnection type at very high data rates can determine the latency, bandwidth, area and cost constraints in an MFS.

1.3. Thesis Goals

Performance of existing MFS routing architectures is limited by many factors as discussed earlier: limited pin resources, inter-FPGA communication strategy and off-chip interface selection. This research is aimed at addressing the constraints of existing MFSs and optimizing their performance by proposing new models. We have developed CAD tools for experimentally evaluating and comparing existing and proposed time-multiplexed MFS routing architectures. The primary goals of this thesis are as follows:

- The first goal is to enable the MFS to accommodate large design which exceed the I/O pin and logic capacity of an FPGA. The proposed solution is to implement multiplexing and study the behaviour of MFS system frequency with respect to increasing multiplexing ratio in three different multiplexing schemes.
- Next goal is to investigate the effects of different routing architectures on the system frequency of an MFS.
- In the next part of this research, our goal is to improve the system frequency by decreasing the off-chip latencies in an MFS. In order to achieve this objective, we proposed latency-optimized planar 2D MFS architectures in which electrical interconnections are replaced by optical interface in same spatial distribution.
- Lastly, we aim at achieving improved MFS system frequency with smaller footprint area. For this, we proposed 3D MFS architectures with vertical stacking and optical off-chip interfaces.

1.4. Thesis Contributions

In order to resolve the problems stated above, the major contributions of this thesis include the following:

- We have proposed novel scalable 3D MFS architectures which showed improved system performance as compared to conventional 2D MFS architectures. The vertical stacking resulted in shorter off-chip links improving the overall system frequency with the additional advantage of smaller footprint area.
- The proposed 3D architectures employed serialized interconnect between intra-plane and inter-plane FPGAs to address the pin limitation problem. Additionally, all off-chip links are replaced by optical fibers that exhibited latency improvement and resulted in faster MFS. Results indicated that exploiting third dimension provided latency and area improvements as compared to 2D MFS. The experimental results have shown average 37% improvement in system frequency as compared to planar MFS with electrical interconnects.
- We also proposed latency-optimized planar 2D MFS architectures in which electrical interconnections are replaced by optical interface in same spatial distribution. Performance evaluation and comparison have shown that the proposed

architectures exhibited reduced critical path delay and system frequency improvement as compared to conventional MFS. 2D optical platforms exhibited an average frequency gain of 22% as compared to 2D MFS with electrical interconnects.

- Achieved performance of three time multiplexing schemes; Logic Multiplexing, SERDES and MGT, is compared for a given range of multiplexing ratio using different routing architectures in planar MFSs with PCB connections.

1.5. Thesis Organization

The rest of the thesis is organized as follows:

Chapter 2 studies the two multi-FPGA routing architectures i.e. Completely Connected Graph (CCG) and TORUS. It describes the two architectures' performance with electrical and optical interface in both 2D and 3D topologies. Then, the previous work done regarding MFS routing architectures is discussed in detail.

Chapter 3 focuses on the three multiplexing schemes i.e. Logic Multiplexing, SERDES and MGT, their detailed description and comparison. Then the relationship between the system frequency and the multiplexing ratio is explained and derived, and finally, the previous work done in time multiplexed MFS is presented.

Chapter 4 describes the characteristics of short-ranged optical interface and its detailed design and application in MFSs. The chapter also covers the previous research done on this topic.

Chapter 5 explains the proposed 3D MFS architectures with optical interface. The feasibility, practicality and advantages of vertically stacked MFS are discussed in detail and also the past research done on the subject is also presented.

Chapter 6 explains the framework employed for experimental evaluation of MFS routing architectures. The experimental procedure and customized set of mapping tools used for mapping circuits to architectures is described. The metric used for evaluating and

comparing multiplexed architectures are explained and the details of the benchmark circuits are also presented.

Chapter 7 compares the achieved performances for a set of designs mapped on the two multi-FPGA platforms employing three multiplexing schemes. The performance gains between these platforms are quantified. Then, the performance comparison is drawn between 2D MFS with multiplexed electrical interface and the proposed 2D MFS with optical interface. Lastly, we have drawn a comparison between 3D MFS with serialized optical interface and 2D conventional MFS.

Finally, Chapter 8 concludes the thesis and suggests directions for future work.

Chapter 2 MFS Routing Architectures

2.1. Inter-FPGA Connections & Routing

Multi-FPGA systems require chip-to-chip connections and there are several ways to organize these inter-chip connections. The *routing architecture* of an MFS is the manner in which the FPGAs, fixed wires and/or programmable interconnect chips are connected together. The routing architecture exercises a strong effect on the cost, speed and routability of the system [1]. Other than the inter-FPGA connection arrangement, the type of connectors employed is also an integral part of MFS routing architecture and impact the overall system performance [8].

2.2. Types of Inter-FPGA Connections

2.2.1. Hard-wired Connection

MFS with hard-wired connections consists of a ready-made generic multi-FPGA board, where all the inter-FPGA connections are fixed and realized using PCB traces. The connections to external interfaces are fixed as well, however these connections can be realized using PCB traces or connectors. One of the examples of such platform is the commercial DNV7F4A platform as shown in Figure 2.1, by Dini Group [2]. This platform is made up of four Virtex-7 FPGAs with all fixed FPGA to FPGA interconnects (either differential or single-ended).

Some of other major existing commercial off-the-shelf platforms with hard-wired connections are as follows:

- Cadence Protium Rapid Prototyping Platform [9],
- S2C 6th generation prototyping hardware with four Xilinx Kintex UltraScale XCKU115 FPGAs
- Quad KU115 Prodigy Logic Module [10]
- BEECubeBEE7 off-the-shelf communications platform with four Xilinx VX690T FPGAs and 400 Gbps of on-board fixed full mesh inter-FPGA connection [11]

- HyperSilicon VeriTiger-DH2000TQ prototyping board with two Xilinx Virtex-7 FPGA devices [12].

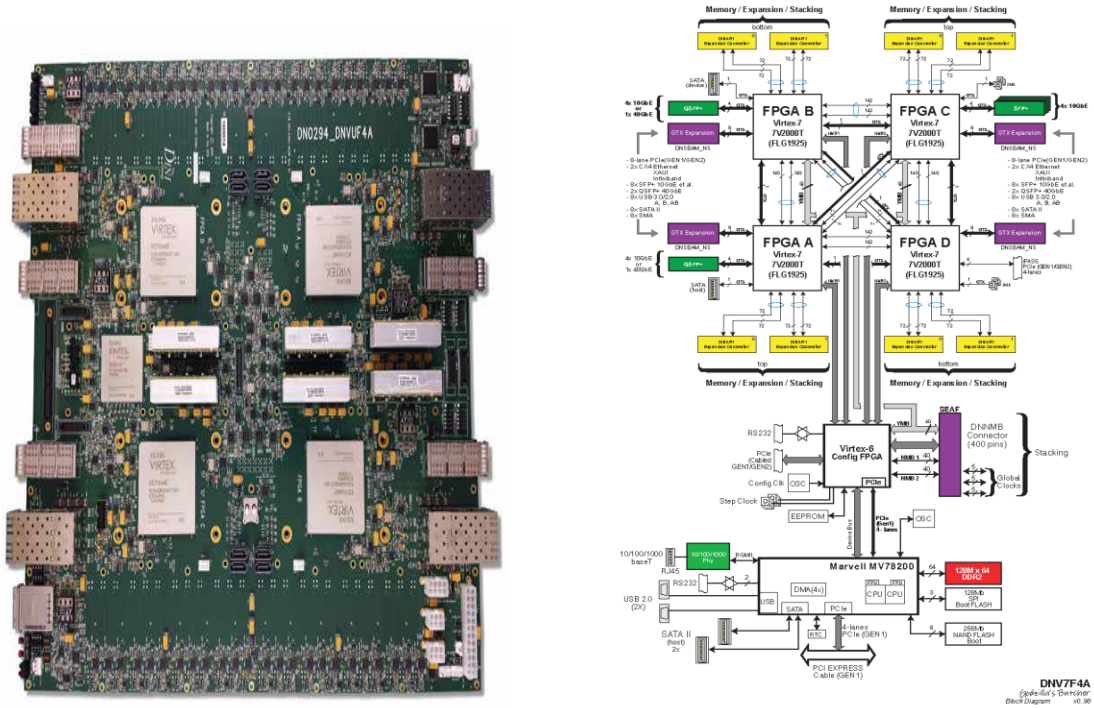


Figure 2.1: Hard-wired MFS (DNV7F4A) [2]

MFS with hard-wired connections can also be customized by tailoring the inter-FPGA connections according to the design requirements.

2.2.2. Cabling Connection

MFS with cabling connections is a relatively new technology consisting of multiple ready-made FPGA devices connected by cables and connectors. The FPGA-FPGA connections as well as the FPGA-external interfaces can be inserted or eliminated merely by connecting or disconnecting the cables with or from the connectors to meet the design requirements. MFS with cabling connections exhibits properties of both off-the-shelf and custom boards, because it employs generic ready-made devices while allowing changeable inter-FPGA connections by connecting or disconnecting the cables in order to be tailored according to the given design requirements.

One of many examples of such connections is proFPGA quad V7 multi-FPGA system which provides flexible and scalable FPGA interconnection structure with high-speed connectors and cables. These specific high speed connectors allow maximum point to point speed of up to 1.8 Gbps over the standard FPGA I/O and up to 12.5 Gbps over the high speed gigabit transceiver pins of the given FPGA. The high interconnection flexibility offers the designer a maximum speed of his/her design running in the proFPGA system. Furthermore, multiple proFPGA quad or duo systems can also be stacked or connected together resulting in unlimited scalability and no theoretical maximum in capacity [13].



Figure 2.2: Photograph of 2 TwinStar FPGA Systems [14]

Similarly, IBM's Twinstar system was configured with 24 node cards and used 45 Xilinx Virtex-5 LX330 FPGA devices, in addition to the control FPGA devices and discrete SRAM and DRAM components [14]. As shown in Figure 2.2, it was constructed with flexible cable interconnect structure facilitating multiple connection topologies. The *Active Backplane* provided flexible interconnect with high-speed LVDS-based point-to-point communication links. Synopsys' HAPS-70 FPGA-based prototyping platforms [15] is built with HapsTrak 3 interconnect cables (Figure 2.3) for high speed interconnectivity between FPGAs and systems. The off-the-shelf Hapstrak 3 connector with 50 I/Os per

connector meets the specific requirements of FPGA-based prototyping and high-speed interface via HAPS interconnect cables.



Figure 2.3: Synopsys' HapsTrak 3 Connector Technology [15]

2.2.3. Optical Connection

Reach length, power, cost, board material, and circuit board complexity are major challenges for copper based, chip-to-chip interfaces. Replacing on-board cabling or hard-wired connections by optical interface on MFS overcomes the limits of copper interconnect by integrating the latest FPGAs with state of the art photon propagation properties, providing reach-length, power, cost, density, and bandwidth advantages. Short-ranged chip-chip optical interconnection not only offers design flexibility like cabling connection, but also dramatically exceeds conventional electrical signaling and interconnects capabilities. As data rates exceed 10 Gbps and higher, optical interface technology overcomes bandwidth challenges encountered by conventional copper connections. Further details on the multi-FPGA boards with optical interface are provided later in the thesis.

2.3. MFS Routing Architectures

MFS interconnect topology influences the overall speed and performance of the system. Researchers have proposed many 2D routing architectures over the years and empirically evaluated and compared different architectures. Another distinctive aspect is whether or

not Field Programmable Interconnect devices (FPIDs) or crossbars are used for connecting the FPGAs. If FPIDs are not used, it is referred to as an FPGA-only architecture.

MFS routing architectures explored in this research are Completely Connected Graph (CCG) and TORUS. The architectural issues and assumptions that arise when mapping real circuits to these architectures are discussed in detail below.

2.3.1. Basic Assumptions

Following assumptions are made for this research:

- First assumption for this research is that the MFS architectures explored are homogeneous, in which a single type of FPGA is utilized. Heterogeneous platforms using FPGAs of different sizes are achievable however rarely used, and are restricted to application-specific (custom) MFSs. In our 2D and 3D architectural models, the chip size is considered to be a fixed parameter. Therefore, instead of adapting the chip size, the number of chips is increased or decreased according to the design requirements.
- Another important issue is the choice of FPGA. The FPGA used in this research is the Xilinx Kintex Ultrascale+ FPGA KU3P, which consists of 163,000 6-LUTs and 325,000 flip-flops. The chosen FPGA offers 16 GTY transceivers with 32.75 Gbps inter-FPGA communication data rate. GTY transceiver supports small form-factor pluggable (SFP) or SFP+ optical module required for off-chip optical interface. In terms of logic capacity and data rates, KU3P is one of the latest and fastest available FPGA in the market. Since the FPGA employed has enormous logic capacity that is why we have chosen the largest available real benchmark circuits. Large benchmark circuits not only stress the FPGA capacity but also the CAD tools developed for the purpose of experimental evaluation of 2D and 3D architectures.
- We have considered point-to-point connections in all the MFS architectures. Point-to-point connections connect two FPGAs directly to each other. After partitioning, the design is divided into several parts. All 2-point and multi-point inter-FPGA nets are routed in MFS point-to-point connections. Multi-terminal nets are split into several 2-terminal nets. This assumption is valid because we have employed multiplexing, which ensured that there are no inter-FPGA net routing failures.

- Another assumption is that the CAD tools developed are designed to handle synchronous mode where entire system uses a single global clock. There are two distinct types of time-multiplexing implementations: synchronous and asynchronous. In **synchronous** mode, the multiplexing clock and the system clock are synchronous. Whereas, in **asynchronous** mode, the multiplexing clock runs completely independent of the system clock and can supports multiple clocks.

Some commercial tools available for single-FPGA static timing analysis can handle asynchronous mode and in future, this research can be extended by developing static timing analysis tool using multiple clocks to build an asynchronous system.

2.3.2. 2D and 3D MFS Routing Architectures

The simplest 2D mesh topology can be designed with each FPGA connected to its horizontal and vertical adjacent neighbors. Mesh architecture provides full connectivity and any combination of connections between inputs and outputs can be made. The number of traces connecting adjacent FPGAs depends upon the number of I/O pins available per FPGA. Xilinx KU3P FPGA has 208 High-Performance (HP) single-ended I/O pins. Out of these, 3 pairs are reserved for non-FPGA connections and 25 pairs are for the primary I/O signals. The connections to external interfaces can be realized using hard-wired PCB traces or connectors and cables. Therefore, 152 pins are left for inter-FPGA connections. All FPGA-FPGA interconnects can be routed as LVDS or single-ended according to the design requirements. In case of SERDES differential signaling, one pair of pins between every FPGA pair has to be reserved to propagate the clock instead of the user data and these pins should be clock capable.

Completely Connect Graph (CCG) is a topology in which all the FPGAs are connected to each other as shown in Figure 2.4 (a). Since, the MFS size is set to be 6 in this research and the available pins per FPGA are 152, this implies that there are $\lfloor 152/5 \rfloor$ tracks between any pair of FPGAs on the board.

In TORUS architecture each FPGA is connected only to its horizontal and vertical adjacent neighbors. Moreover, the peripheral FPGAs are wrapped around in horizontal and vertical directions and are connected to the FPGAs on the opposite side of the array as shown in

Figure 2.4 (b). For an MFS size of 6, each FPGA is connected to maximum 3 neighbors in TORUS and each edge in Figure 2.4 (b) represents $\lceil 152/3 \rceil$ tracks between any pair of FPGAs.

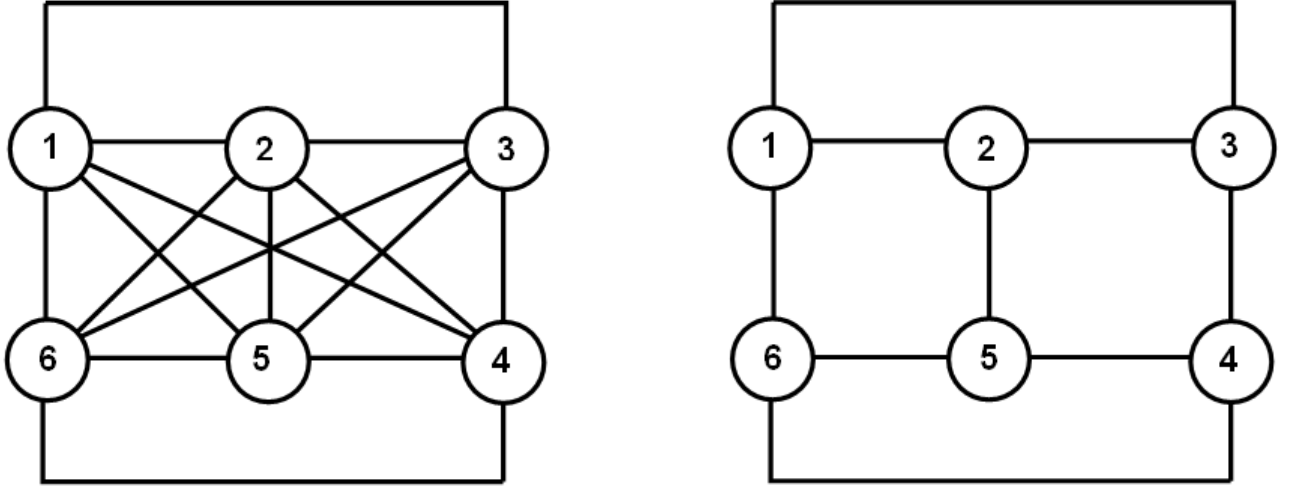


Figure 2.4: Routing Architectures (a) CCG, (b) TORUS

In case of vertically stacked MFS, the FPGA interconnection topologies of CCG and TORUS remain the same as that in planar platforms. 3D architectures are discussed in detail later in the thesis.

2.4. Previous Research on MFS Routing Architectures

In this section we will look at the different routing architectures proposed over time in MFSs. The existing routing architectures can be categorized roughly in the following three categories: linear arrays, meshes and architectures that employ programmable interconnect chips. The first two types are the examples of FPGA-only architectures.

2.4.1. Linear Arrays

FPGAs are arranged in the form of a linear array in this type of architecture, which is appropriate for one-dimensional systolic processing applications. This architecture has very restricted routing flexibility and numerous designs may run out of routing resources and therefore cannot be implemented. While the linear array architecture may be good for

certain niche applications, its utility as a general purpose MFS is extremely limited. Two historically recognized examples of this architecture are AnyBoard [16] and Splash [17].

As shown in Figure 2.5, the AnyBoard system employs five Xilinx 3090 FPGAs and three 128K x 8 RAMs.

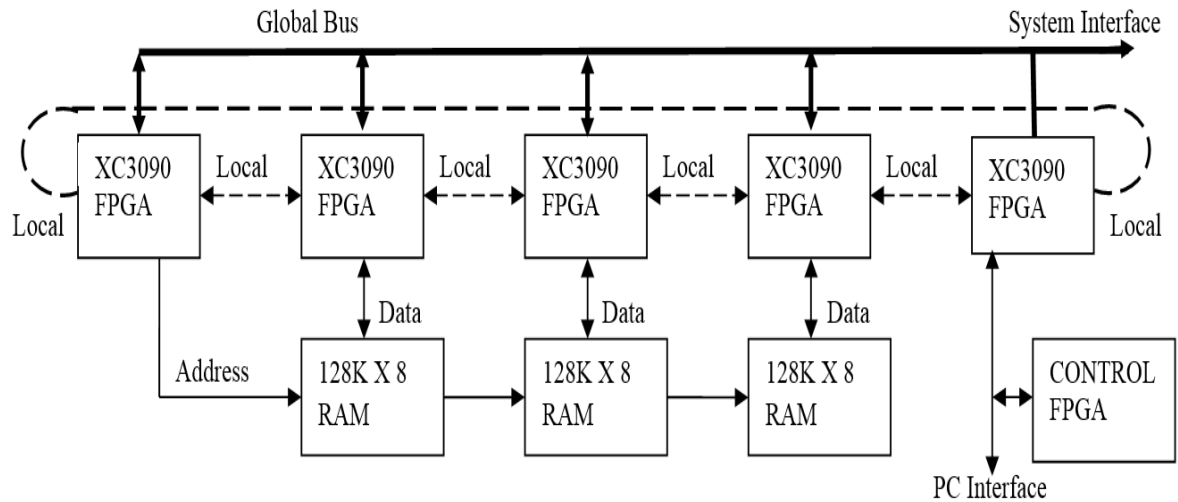


Figure 2.5: The AnyBoard System [16]

Adjacent Xilinx chips in the array are connected through local buses that offer communication between function blocks in systems. FPGAs located at the opposite ends of the array are connected to structure a ring topology and all the FPGAs are attached to a global bus. An extension of the global bus with dedicated I/O lines from each FPGA provides the system interface. This can be utilized for routing I/O signals of the circuits. The control FPGA is employed to implement circuitry for managing the PC bus interface, FPGA configuration management and hardware debugging support. The idea of using the control FPGA is to leave all the logic in other FPGAs for implementing the required design functionality. The AnyBoard system was one of the first MFSs built for accelerated prototyping of small designs. It was an economical system that demonstrated the prospective of MFSs as an attractive and low-cost means for rapid prototyping of scores of hardware designs.

The Splash logic-array board has 32 Xilinx 3090 programmable gate arrays and 32 memory chips. Two additional Xilinx chips are used for bus control. The Splash design was motivated by a systolic algorithm for DNA pattern matching [17].

Cube [25] was massively-parallel FPGA architecture with 512 FPGAs connected in systolic chain with identical interfaces between them. Each module in the Cube platform hosted 64 Xilinx FPGAs arranged in an 8 by 8 matrix. Eight FPGAs were grouped together in a row and had independent configuration inputs and power supplies. The complete system consisted of 8 connected boards in a cabinet forming an $8 \times 8 \times 8$ cluster of 512 FPGAs, and therefore named Cube.

2.4.2. Mesh Architectures

In the basic design of mesh architecture, the FPGAs are placed in the form of a two-dimensional grid with every FPGA connected only to its four nearest neighbors as shown in Figure 2.6(a). In this manner, the FPGAs are stitched together into a single, larger structure, with the Manhattan distance measure that is representative of most FPGAs carried over to the complete array structure.

In order to decrease the average number of I/O pins required to route signals and improve the routability, we can increase the number of neighbors linked to an FPGA. Rather than the simple four-way basic connection pattern of Figure 2.6(a), we can implement an 8-way topology, Figure 2.6(b). In the eight-way architecture, an FPGA is not only connected to those FPGAs horizontally and vertically adjacent, but also to those FPGAs which are diagonally adjacent. A second option is a one-hop topology, Figure 2.6(c). In this arrangement, an FPGA is linked to the two nearest FPGAs directly above, below, to the right, and to the left. Two-hop, three-hop, and longer connection patterns have also been considered [18]. In Figure 2.6(b) & (c), each line between any pair of FPGAs represent multiple number of traces and depends upon the available FPAG I/O pins.

The benefits of mesh are simplicity of local interconnections and straight forward scalability. However, using FPGAs for interconnections lessens the number of pins for logic inside each FPGA and leads to reduced logic utilization. The connection delays are

large between widely separated FPGAs (especially in bigger arrays) whereas those between neighboring FPGAs are minute. The outcome is degraded speed performance and timing problems such as setup and hold time violations because of widely variable interconnection delays. Quickturn RPM [19], DEC PeRLe-1 [20], and the MIT Virtual Wires project [21] are a few examples in this category.

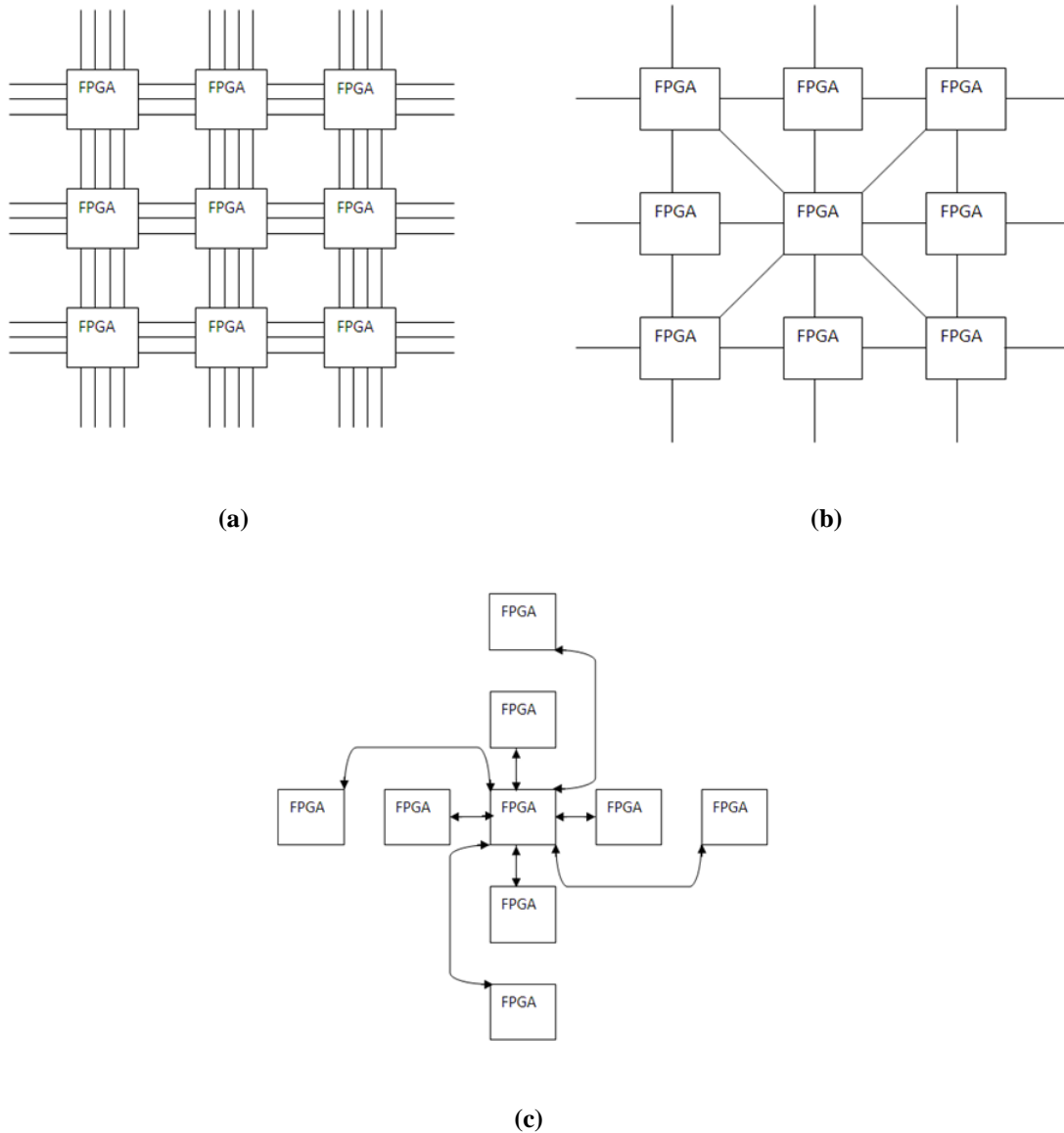


Figure 2.6: (a) Basic Mesh Architecture (b) 8-way Mesh (c) One-Hop Mesh

The Quickturn RPM Emulation System had FPGAs hardwired together on large printed-circuit boards. Each FPGA was connected to all its nearest-neighbor FPGAs in a regular

array of signal-routing channels. The routability and speed problems of the mesh architecture that arose when implementing general logic circuits, forced Quickturn to switch to a superior architecture (partial crossbar) in their next generation logic emulation systems.

Virtual wires got rid of the pin limitation problem of prior emulators by intelligently multiplexing each physical wire amongst numerous logical wires, and pipelining these connections at the highest clocking frequency of the FPGA. Consequently, the available off-chip communication bandwidth was increased by multiplexing the utilization of FPGA pin resources (physical wires) among multiple emulation signals (logical wires).

By employing virtual wires scheme on a mesh, low-cost logic emulation was achieved because in expensive low pin count FPGAs were used and the mesh architecture was reasonably simple to be manufacture. On the other hand, the drawbacks were the speed penalty and increased mapping software complexity due to pin multiplexing. Moreover, in certain cases it might not be easy to map sections of asynchronous logic that might be present in the circuit to be emulated since asynchronous signals could not be assigned to a specific time slice (phase) in the emulation clock period.

The mesh topology also performed well when implementing algorithms which matched its architecture. This was established convincingly by the DEC PeRLe-1 system which used a 4-way mesh consisting of 16 Xilinx 3090 FPGAs along with 7 control FPGAs, 4 MB of static RAM, four 64-bit global buses and FIFO devices. The said system gave superior performance and cost in contrast to every other contemporary technology of its time, including supercomputers, massively parallel machines, and conventional custom hardware for various applications, including cryptography, high energy physics, image analysis and thermodynamics.

Maxwell [24] used a 2-D TORUS routing architecture between 64 FPGAs to demonstrate its effectiveness for high-performance computing applications. The FPGAs used in Maxwell were Xilinx Virtex-4 devices in two flavors. Alpha Data cards used XC4VFX100, while Nallatech cards used XC4VLX160. Xilinx's LX Virtex range offered greatest number of logic cells, while the FX FPGAs included embedded PowerPC cores and MGTs

(“RocketIO”) for off-chip communications. These two types of Virtex-4 FPGAs were built into two flavors of plug-in PCI card: the Nallatech H101 and the Alpha DataADM-XRC-4FX. Both types of card were connected using a PCI/PCI-X bridge. The FPGA network consisted of purely point-to-point links between the MGT connectors of adjacent FPGAs and did not implement routing logic in the FPGA devices. The MGTs were connected with standard Infiband cables of 50cm and 100cm lengths, kept as short as possible.

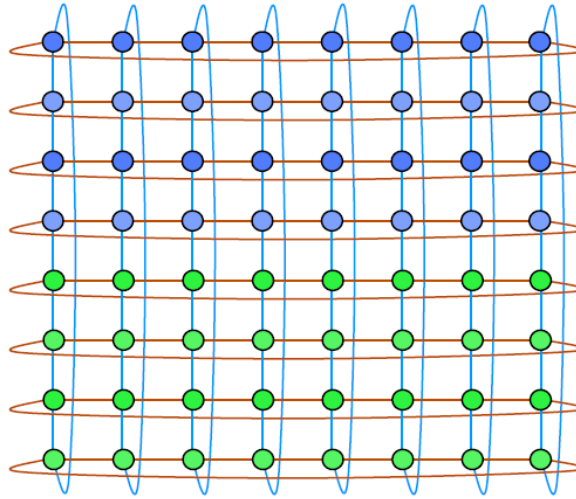


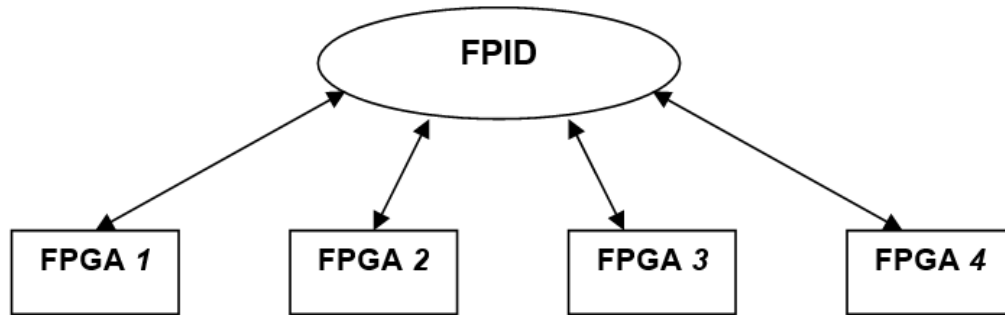
Figure 2.7: Maxwell FPGA Connectivity [24]

Catapult [26] was built on a two dimensional 6X8 network topology, which balanced routability and cabling complexity. The inter-FPGA network requirements were low latency and high bandwidth and therefore the traces were routed through mezzanine connector between daughtercard and mothercard. Note that it is a not strictly an MFS because each FPGA worked independent of the other.

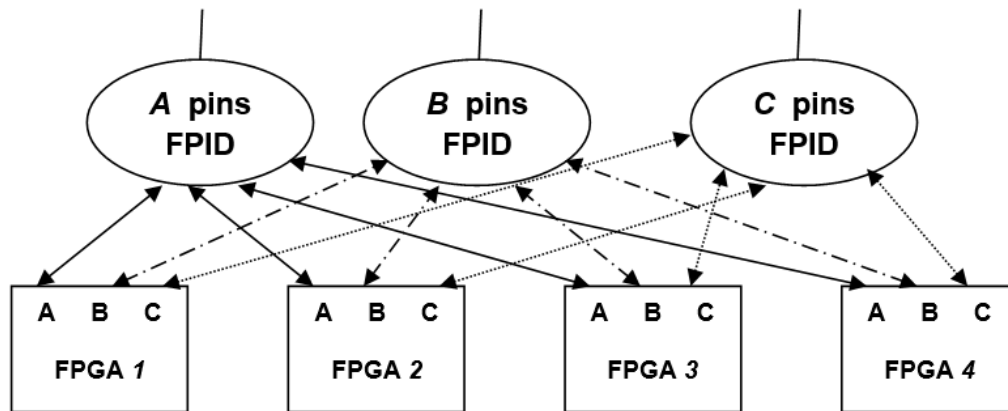
2.4.3. Programmable Routing Architectures

In this type of architecture, all the interconnections among FPGAs are routed through Field Programmable Interconnect devices (FPIDs). A superlative model of this architecture would be a full crossbar that employs a single FPID for linking all FPGAs, as shown in Figure 2.8(a). However, the complexity of a full crossbar increases as a square of its pin count and therefore it is limited to systems that have at most a few FPGAs. A brief review of FPID device architectures, their cost and commercial viability issues is discussed below.

Aptix FPIC device [22] was the first FPID brought into the market. Each FPIC had 1024 pins arranged in a 32 x 32 I/O pin matrix. Every pin was connected to two I/O tracks that orthogonally crossed the routing channels. Each routing channel consisted of sets of parallel tracks that were segmented into a variety of sizes to hold different signal paths with different lengths. Bidirectional pass transistors which were controlled by SRAM cells connected I/O tracks to routing tracks and routing tracks to other routing tracks.



(a)



(b)

Figure 2.8: (a) Full Crossbar (b) Partial Crossbar

Through selectively programming the SRAM cells, the user could connect any device pin to any number of other pins

The partial crossbar architecture shown in Figure 2.8(b) overcomes the limitations of the full crossbar by employing a set of small crossbars. This architecture is comprised of four

FPGAs and three FPIDs and the pins in each FPGA are divided into N subsets, where N is the number of FPIDs in the architecture. All the pins belonging to the same subset in different FPGAs are connected to one FPID. The number of pins per subset determines the number of FPIDs needed and the pin count of each FPID. Delay through all inter-FPGA connections is uniform and the size of the FPIDs increases linearly as a fraction of the number of FPGAs.

2.4.4. Tree Topology

A tree routing topology in a MFS resembles the structure of a directed acyclic graph, in which every node except the root node has exactly one incoming edge, and no node has more than n outgoing edges where n is the arity of the tree. Such a tree is usually referred to as an ' n -ary tree'. In the MFS implementation, every edge is considered implicitly bidirectional. For compact trees, tree depth $d = \lceil \log_n m \rceil$, where m is the number of leaf nodes. Symmetry exists at every child-bearing node in this tree topology. Obviously, increasing n would increase the symmetry of the overall system. However, when $n = m$ and $d = 1$, the tree reduces to the crossbar. The usual purpose of having a tree structure with $n < m$ is to reduce the overhead of implementing the system.

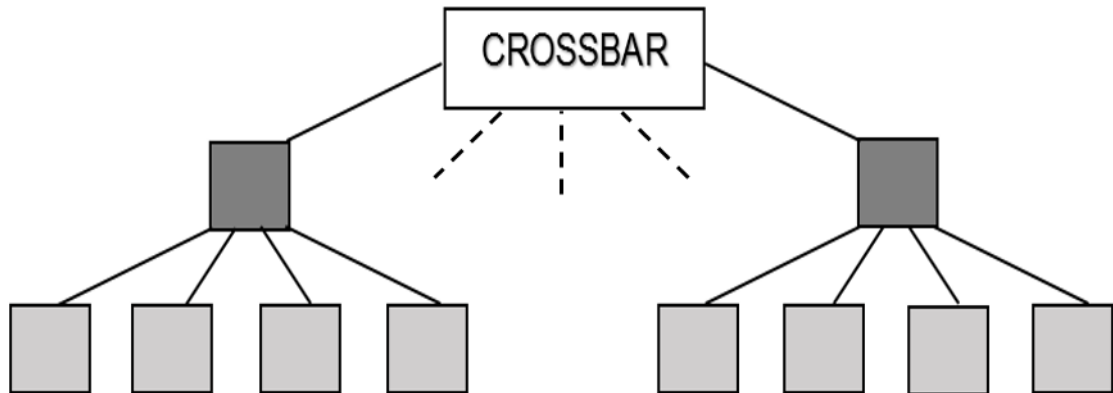


Figure 2.9: BEE2 System Topology

Berkeley emulation engine II (BEE2) [27] proposed a basic structure with a set of modules, each of which were implemented as a tree of fixed d and n . The root FPGAs of the modules were then interconnected using a full crossbar. The system was built with $n = 4$ and $d = 1$.

$n = 4$ was selected, as it appeared to be the largest number of edges that could be supported by a single FPGA. The system resulted in an $m/4$ -way crossbar that was easily implemented using off-the-shelf infiniband switches even for $m > 256$. The overhead of the BEE2 turned out to be about 1/16 that of a full crossbar.

2.4.5. Other MFS Routing Architectures

Khalid et al. [1] proposed the hybrid complete-graph and partial-crossbar (HCGP) routing architecture that used both hard-wired and programmable connections between the FPGAs. The proposed architecture was similar to partial crossbar, with the added feature that the router exploited the direct connections between FPGAs to minimize the number of FPGA and FPID pins used for routing and to minimize the net delay for critical inter-FPGA nets. The proposed architecture produced superior results as compared to partial crossbar in terms of speed and pin cost.

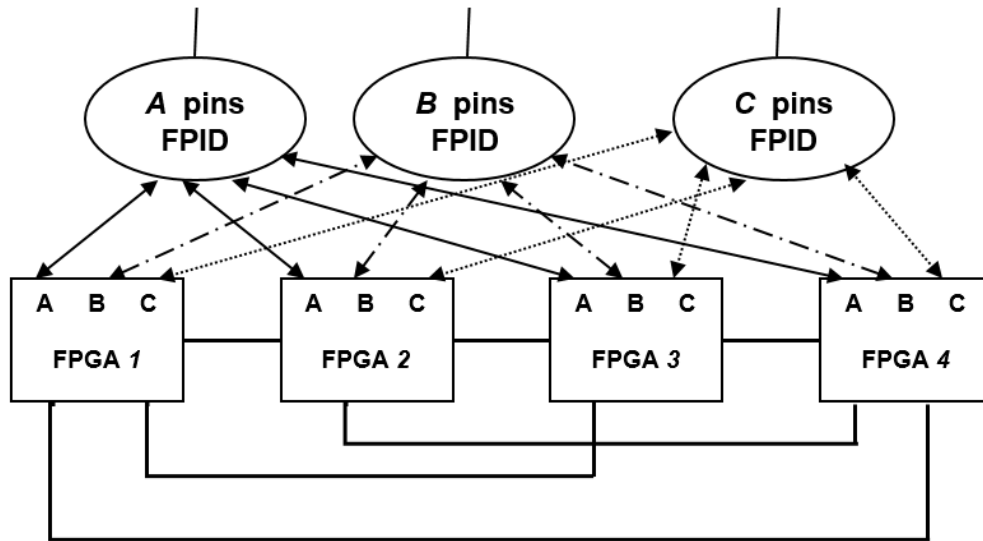


Figure 2.10: HCGP Architecture

The HCGP routing architecture for 4 FPGAs and 3 FPIDs is shown in Figure 2.10. The I/O pins in each FPGA were divided into two sets: hardwired connections and programmable connections. The pins in the first set were connected to other FPGAs and the pins in the second set were connected to FPIDs. The FPGAs were directly connected to each other using a complete graph topology, i.e. each FPGA was connected to every other FPGA. The

connections between FPGAs were evenly distributed, i.e. the number of wires between every pair of FPGAs was the same.

The FPGAs and FPIDs were connected in the same manner as that in a partial crossbar which meant that any circuit I/O had to go through FPIDs to reach FPGA pins. That's why, a certain number of pins per FPID were reserved for circuit I/Os. Using FPID for routing multi-terminal nets helped tackle that scares pin resources of an FPGA in HCGP.

In 3D domain, [4] proposed a three-dimensional concentric 4-FPGA routing architecture resulting in equal length concept between FPGA pins enabling wave-pipelined pin-multiplexing. This research concentrated on switch based routing and used pass transistor as logic element for switching technology due to its speed advantage and bidirectional functionality compared to buffer based technology. Pass transistor has a propagation delay of 0.1ns however; it has the disadvantage of degrading the slope of the signal over multiple switches. Since, the circuit behavior heavily depended on the capacity of the board traces, that's why the impact was kept minimal by using short connections. The connections between FPGAs and non-FPGA devices were kept fixed, whereas the inter-FPGA connections were mapped on the existing hardware. The switches were mounted on a specific switch-board which connected vertically with two adjacent FPGA boards. Besides the connectivity to the switch network, every FPGA pin was routed to external connectors to be accessed by non-FPGA devices. The author suggested that the main advantage of this concept was that any possible signal connectivity could be routed on the proposed structure and unused or additional pin penalty did not occur. The second advantage of this routing concept was the equal length of the connectivities which meant in this context a difference of less than 5 millimeters. Therefore, all signal connections from one FPGA to any other FPGA passed the same number of switches and had the same length. However, the author considered randomly generated designs instead of real benchmarks to evaluate the performance of the proposed architecture.

2.5. Summary

A review of different types of inter-FPGA connections in existing MFSs and the different routing architectures was presented in this chapter. Inter-FPGA connections can be

categorized as hard-wired, cable connections or optical interface. Depending upon the interconnection structure, MFSs can be grouped into three main categories; linear arrays, meshes, and architectures that use programmable interconnection chips. Relevant MFS architecture research studies were also discussed in this chapter. The chapter also presented the two routing architectures employed in this research i.e. CCG (Completely Connected Graph) and TORUS. Both routing architectures were discussed in context with 2D and 3D MFS.

Chapter 3 Time Multiplexing in MFS

3.1. Introduction

Large SoCs may not partition into multiple FPGAs without overflowing the available I/O resources. One of the possibilities to make the design routable is by changing the way that logic is partitioned into multiple FPGAs, since partitioning can alter the number of inter-FPGA nets going between partitions. However, repartitioning is not always an effective solution. Rising levels of chip functionality and data throughput requirements have persuaded the chip industry in migrating from inferior data rate parallel connections to higher speed serial connections. Employing high speed serial interface not only resolves limited pin count problem but also addresses the off-chip communication bottleneck and routing congestion issues in MFS. Multiplexing implies sending multiple signals onto the same physical trace in time shared fashion. The number of inter-FPGA nets per track is called the multiplexing ratio and greatly influences the system performance of an MFS. Exploiting the appropriate routing architecture's effects in conjunction with optimized multiplexing scheme can enhance the system clock frequency [5] [21].

Three multiplexing schemes for MFS; Logic Multiplexing, SERDES and Multi-Gigabit Transceiver (MGT) have been evaluated experimentally in this research. Each scheme has different latency and data rate and thus has distinctive influence on the system performance over a given range of multiplexing ratio.

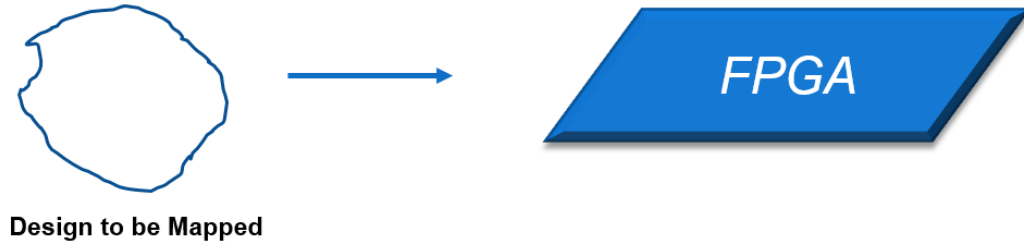
3.2. Critical Path Delay

In synchronous digital circuits, the speed of a mapped design is governed by the slowest combinational path in the circuit implementation, which is called the critical path. There are three different critical path delays: Pre-partition critical path delay (CPD), Post-partition critical path delay (CPD_PP) and Post-Routing critical path delay (CPD_PR).

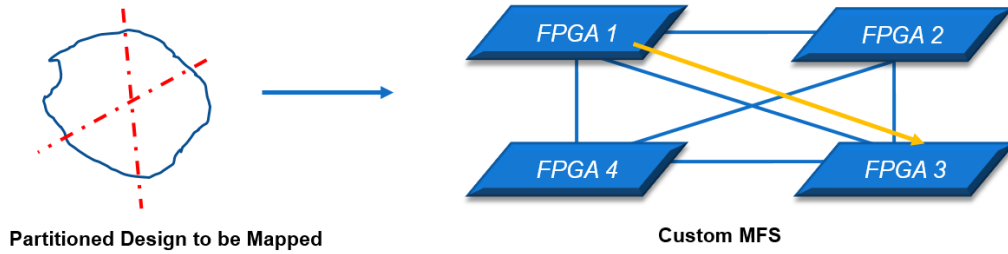
Critical path delay of the un-partitioned LUT-level netlist is called CPD. It is calculated by assuming that the complete design is mapped on a hypothetical single large FPGA and there are no off-chip delays in the critical path as shown in Figure 3.1 (a).

CPD_PP is the critical path delay obtained by analyzing the circuit netlist after it has been partitioned into multiple FPGAs. The circuit is annotated with the inter-FPGA delays. Here it is assumed that the design is mapped on a custom MFS, which has no routing limitations and it provides full connectivity as shown in Figure 3.1 (b). CPD_PP is calculated by adding all the delays encountered when connecting a CLB in one FPGA to a CLB in another FPGA. CPD_PP is the sum of the following three delay values: CLB-to-output pad routing delay, PCB or optical trace delay and input pad-to-CLB routing delay.

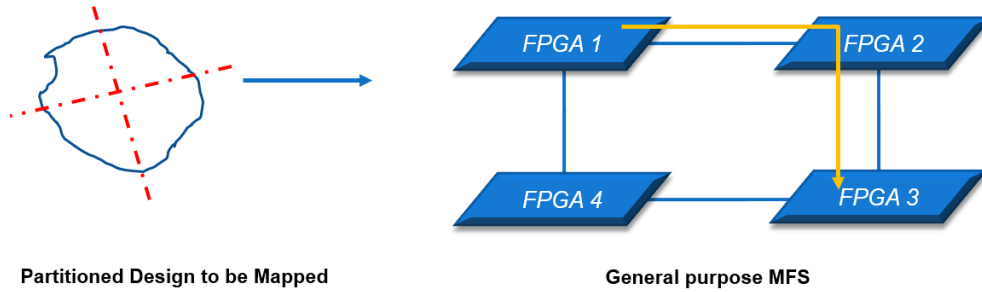
Speed of an MFS is determined primarily by the latency bound i.e. the length of the post-routing critical path (CPD_PR) obtained after a synchronous design has been placed and routed at the inter-chip level [1]. CPD_PR is governed by the internal design delay, the I/O pad delays and off-chip routing delays. As compared to the internal delay, board routing delays exercise a larger impact on the overall system performance. The routing architecture employed and the type of interconnections used mainly dictates the system routing delay. CPD_PR is the same as CPD_PP but it also takes into account any route-throughs which can occur due to MFS limited routing architectures as in TORUS. As shown in Figure 3.1 (c), in route-through scenario, the signal does not have a direct path from source to destination FPGA and therefore, it has to traverse through an intermediate FPGA. When a signal is sent from the source FPGA (FPGA 1), it enters into one pin of the intermediate FPGA (FPGA 2), travels through the on-chip routing lines and then exits through the other pin, without utilizing any of the on-chip logic of the intermediate FPGA. Then the signal reaches its destination FPGA (FPGA 3). Detailed discussion on critical path delays and their calculations is presented in Chapter 6.



(a) Pre-Partition Critical Path Delay (CPD)



(b) Post-Partition Critical Path Delay (CPD_PP)



(c) Post-Routing Critical Path Delay (CPD_PR)

Figure 3.1: Critical Path Delays

3.3. Logic Multiplexing

Logic Multiplexing requires multiple compatible inter-FPGA signals to be assembled and serialized through the same single-ended board trace and then de-multiplexed at the destination FPGA. Using I/O flip-flops makes the timing of inter-FPGA connections more predictable and generally faster in case of asynchronous multiplexing where system clock and multiplexer/de-multiplexer clock are not phase aligned [21]. In this research, we have considered the synchronous method of time multiplexing which is *system-synchronous*.

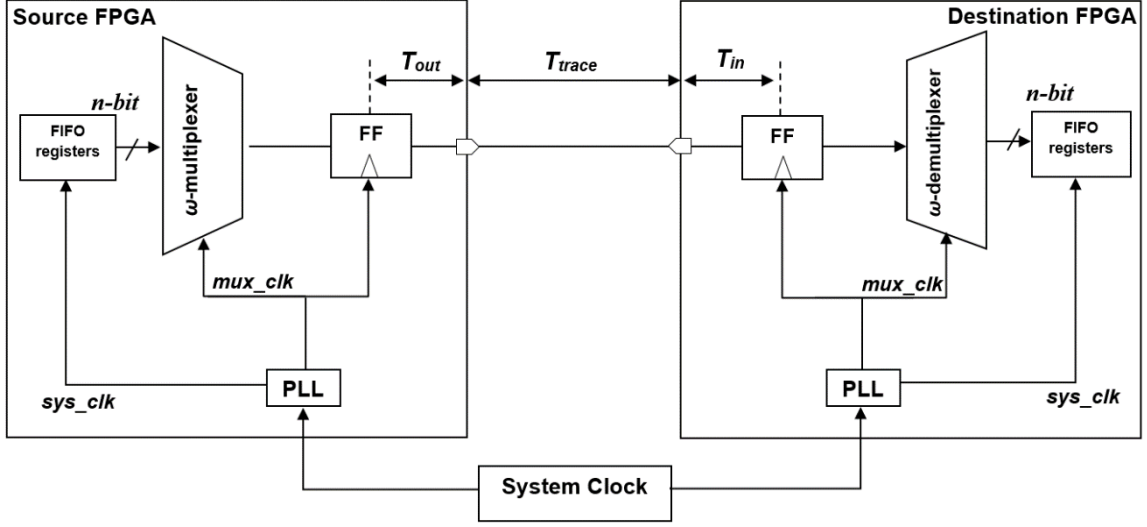


Figure 3.2: Logic Multiplexing Scheme

The multiplexer/de-multiplexer clock and the system clock for the FPGAs are mutually synchronous i.e., they are derived from one clock source, PLL (Phase Locked-Loop) and are phase aligned.

As discussed earlier, CPD_PR determines the speed of a design in an MFS. In a multi-FPGA board, CPD_PR is the sum of: source intra-FPGA routing delay, output pad delay (T_{out}), board trace delay (T_{trace}), input pad delay (T_{in}) and destination intra-FPGA routing delay as shown in Figure 3.2. In order to ensure synchronization between source and destination FPGA clocks we add a safety margin of 20% to CPD_PR. Therefore, we can obtain the delay on a multiplexed connection in Logic multiplexing scheme and mux_clk can be written as (3.1).

$$mux_clk = \frac{1}{1.20 * CPD_PR} \text{ (MHz)} \quad (3.1)$$

On the transmitter end, n -bit wide data from the internal domain is multiplexed by ω -bit wide logic multiplexer which is inserted to accommodate the signals exceeding the transmission capacity. ω is the multiplexing ratio and it represents the number of inter-FPGA nets sent onto a single board trace. When there is no multiplexing, $\omega=1$.

Therefore, the relationship between sys_clk_lm and mux_clk in logic multiplexing scheme for a given range of multiplexing ratio ω [5] can be calculated by (3.2):

$$sys_clk_lm = \frac{mux_clk}{\omega} \quad (MHz) \quad (3.2)$$

3.4. SERDES

As discussed earlier, high-speed FPGA interconnections are inevitable in present technology. The traditional method of parallel transmission is becoming inadequate and is replaced by serial data communication meeting higher bandwidth requirements. In serial high speed I/O interfaces, instead of transmitting in parallel, the stream of serial data is transmitted one bit per time on each link. All modern FPGAs are equipped with serialization and deserialization (SERDES) modules which provide serial-to-parallel conversions on incoming data and parallel-to-serial conversion on outgoing data. The common approach employed to transmit data is *single-ended signaling* where one off-chip trace is used to carry the transmitted signal as in logic multiplexing scheme. However, for data rates exceeding gigabits per second (Gbps), *differential signaling* is preferred rather than single-ended transmission. SERDES allow operation at speeds greater than 1Gbps per line, using low-voltage differential signaling (LVDS) data transmission [29].

3.4.1. LVDS Signaling

LVDS is a fast, low-power, low-voltage and low-noise general-purpose input output (I/O) interface standard which requires two pins for each serialized data stream. ANSI/TIA/EIA-644 standard and IEEE Std. 1596.3 define physical layer (PHY) of LVDS. Typical applications of LVDS include high-speed video, graphics, flat panel displays, general purpose computer buses etc.

LVDS driver has a nominal 3.5 mA current source located in it as shown in Figure 3.3. Since the input impedance of the receiver is high, the entire current flows through the 100 Ω terminating resistor resulting in a 350 mV voltage drop across the receiver inputs. LVDS receiver threshold is certain to be 100 mV or less and this sensitivity is kept constant over a wide common mode from 0V to 2.4V. This combination offers exceptional noise margins

and tolerance to common-mode shifts between the driver and the receiver. Changing the direction of current results in the same amplitude but opposite polarity at the receiver end. 350 mV typical signal swing consumes small amount of power and makes LVDS a very power efficient technology.

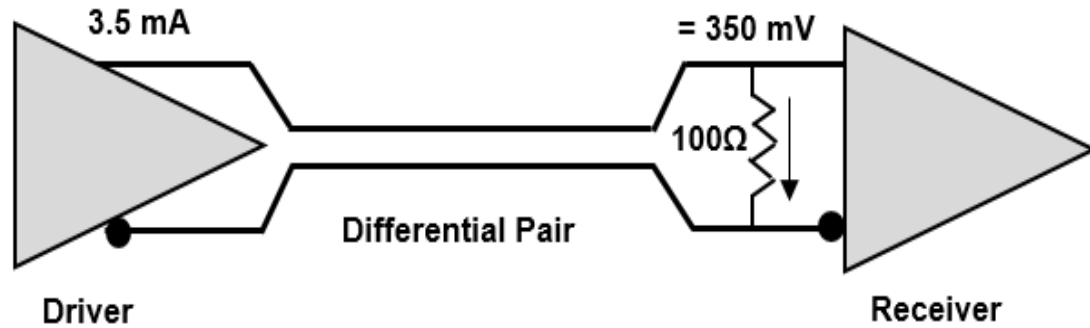


Figure 3.3: LVDS Architecture

The main advantages of LVDS signaling are as follows:

- High data rates can be attained with low power consumption.
- Better noise performance as compared to single-ended signaling.
- Low voltage swing as compared to other industry data transmission standards, consequently LVDS achieves a high aggregate bandwidth in point-to-point applications.

The main disadvantages of LVDS communication include:

- Skin effect, dielectric losses and reflections.
- Long parallel links are affected by signal integrity and skew.

In multi-FPGA setup, the OSERDES module in the transmitter FPGA translates the single input signal into a pair of output signals which are driven 180° out of phase with each other onto the PCB traces. The ISERDES module in the receiver FPGA recovers the signal as the difference in the voltages on the two lines. The voltage difference between these two signals defines the value of the resulting LVDS signal. External electromagnetic interference (EMI) tends to affect both signals equally, however, since at the receiving end only the difference between the two signals is detected that's why differential signals are

more resistant to electromagnetic noise as compared to single-ended signals. Differential signals can achieve higher speeds because they reference no other signals but themselves and the timing of signal crossover can be more tightly controlled. Since the received signal is the difference between the signals on the two traces (which are equal and opposite) the resulting signal is twice as large as compared to the ambient noise. Consequently, differential signals have higher signal/noise ratios and performance.

3.4.2. SERDES Architecture

A SERDES transmitter takes an n -bit parallel data bus, switching at a given frequency, passes it through an encoder, serializes it into a serial bit stream, and then drives the serial data onto an interconnect wire capable of handling differential signaling. Encoded data is a better fit to the physical channel and the bit detection at the receiver end becomes easier. A clock is propagated on a parallel path to the data for the purpose of synchronization and this method is called *source-synchronous*. This means SERDES requires two pins for each serialized data stream. In order to draw the comparison between the logic multiplexing and SERDES, consider a scenario, where ratio of 10:1 needs only one inter-FPGA trace to transfer ten data signals, SERDES needs two, i.e. 10:2. Therefore the SERDES reduces the interconnections only by factor 5 and not by factor of 10. However, very high multiplexing ratios give a far greater data transfer bandwidth as compared to logic multiplexing.

SERDES receiver block performs the inverse function of the serializer block. It de-serializes the incoming data onto an n -bit parallel data of similar width as that of the serializer. The de-serialization process is dependent on the clock data recovery (CDR) circuit which provides a recovered clock to aid drive the timing of the shift registers being employed to reassemble the parallel data. The de-serialized (parallel) data stream is decoded back to its original data bits format. These data bits are then forwarded to the parallel output registers and clocked out using the parallel output signal buffers. These output buffers are typically single-ended signal buffers. A recovered clock is also provided along with the parallel data. This clock is frequency-aligned to the data rate of the incoming serial data stream.

ISERDES module in the destination FPGA consists of a clock data recovery (CDR) unit which is a second order system having jitter-rejection properties and employed to extract the clock signal from the received data. It takes the received data stream and tracks its frequency and phase to recover a clock which is centered at an ideal spacing relative to the data-eye. CDR utilizes the data transitions to determine the clock speed. Since there is no separate clock signal, the transitions from 0 to 1 and from 1 to 0 in the data stream are used to infer a recovered clock. This clock is then fed to the de-serializer allowing the recovery of the data in its original format.

Phase-Locked Loop (PLL) is a closed-loop electronic control system which is employed for frequency control by generating an output clock signal with a fixed relation to the phase of the input or reference clock signal. PLL is a vital part of SERDES communication and in order to achieve maximum bandwidth, low-jitter fast-locking PLL is used to drive the parallel to serial converters on the transmitter's end. Similarly, at the receiver's end, CDR employs sophisticated PLL to recover the clock and capture and de-serialize the data back to parallel format. Generic SERDES architecture is shown in Figure 3.4.

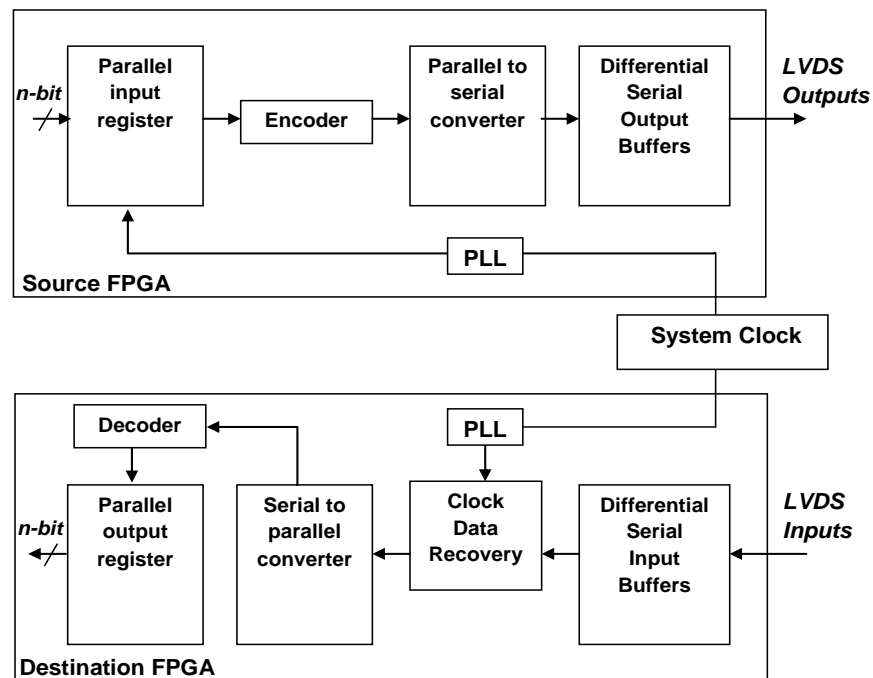


Figure 3.4: Generic SERDES Architecture

3.4.3. SERDES Multiplexing

As discussed earlier, the internal design frequency can be calculated from the post routing critical path delay of the mapped design. In case of SERDES, internal design frequency is represented by clk_core . SERDES multiplexing architecture in MFS is shown in Figure 3.5. On the transmitter end, n -bit wide data from the internal domain is multiplexed by ω -bit wide logic multiplexer which is inserted to accommodate the signals exceeding the transmission capacity. When there is no multiplexing, $\omega=1$. Multiplexed data is then sent on the clk_serdes domain combined with start pattern (for the inter-FPGA synchronization) and generated checksum data (to verify the integrity of the transmitted data), and fed to the 4-bit wide OSERDES module. clk_serdes should be a multiple of clk_core , and phase aligned with the internal design frequency. The ratio between clk_core and clk_serdes called clk_ratio , depends on the multiplexing factor and width of SERDES module.

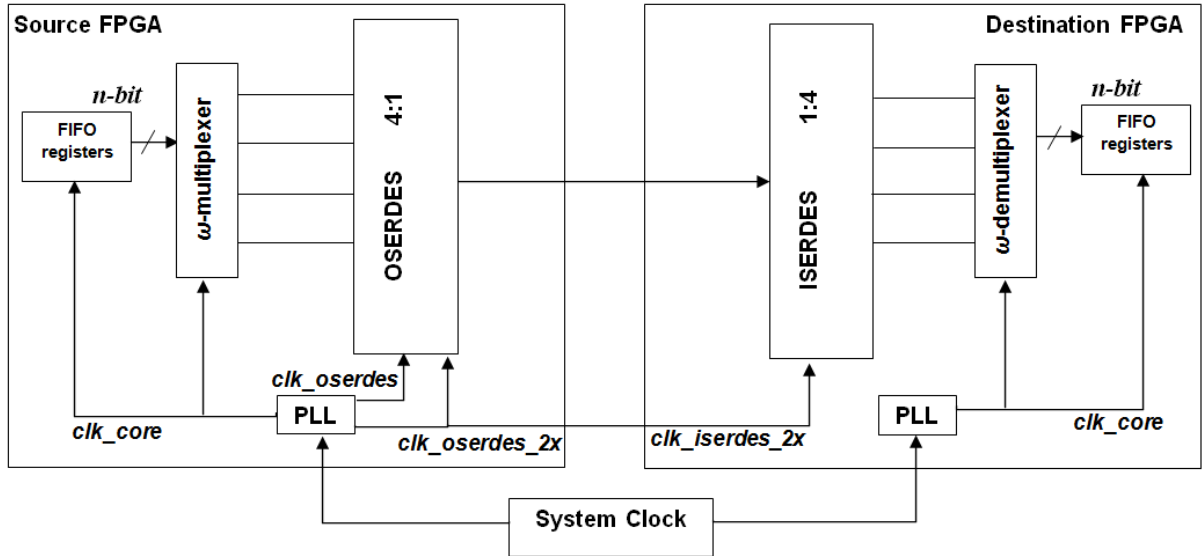


Figure 3.5: SERDES Multiplexing Scheme

All of the latency on clk_serdes domain required for physical data transfer must be accommodated by clk_core domain. Next, OSERDES module multiplexes the incoming data into serial data and is then transmitted to the receiver FPGA, across the physical

interface as source-synchronous LVDS data with the frequency clk_serdes_2x . clk_serdes_2x should be twice the clk_serdes frequency and both must be phase aligned.

On the receiver end, ISERDES module receives the source-synchronous LVDS data and clock and produces an output data of width ISERDES_WIDTH. In KU3P each I/OSERDES (ISERDES and OSERDES) is capable of performing serial-to-parallel or parallel-to-serial conversions with programmable widths of 4 or 8 bits. ISERDES_WIDTH should be same as OSERDES_WIDTH. In this research we have set SERDES width to be 4. This output data is then further de-multiplexed by ω -bit wide logic de-multiplexer into n -bit wide data. The received checksum is verified against a generated data checksum. A single clock oscillator must be used to generate all the clocks for the transmitter module i.e. clk_core , clk_serdes , and clk_serdes_2x so that phase alignment is guaranteed [61].

When inter-FPGA signals are multiplexed then $\omega = \text{ceil}\left(\frac{serdes_mux}{serdes_width}\right)$ where, $serdes_mux$ is the maximum number of signals passing through one ISERDES/OSERDES given by:

$$serdes_mux = \frac{Total_inter_FPGA_tracks * mux_ratio}{Data_Carrying_Pairs} \quad (3.3)$$

Xilinx KU3P FPGA has 208 High-Performance (HP) single-ended I/O pins. Out of these, 3 pairs are reserved for non-FPGA connections and 25 pairs are for the primary I/O signals. Therefore, 152 pins are left for inter-FPGA connections and for an MFS size of 6, there are $\lfloor 152/5 \rfloor = 30$ tracks between any pair of FPGAs in CCG and $\lfloor 152/3 \rfloor = 50$ tracks in TORUS architecture. One pair of pins is reserved to propagate the clock instead of the user data and these pins should be clock capable. This implies that $serdes_mux = 2 * mux_ratio$ for any type of routing architecture.

I/OSERDES modules each take 2 clock cycles of latency. $1 + \omega$ clock cycles are required for sending the start pattern, checksum and ω -bit wide multiplexed data. Assuming 2ns of delay across the board and 0.75ns for pad to/from SERDES and setup/hold time in I/OSERDES modules. Tolerance delay is not required in SERDES multiplexing. Therefore, the total latency comes out to be $2+2+1+\omega$ clock cycles + 2.75ns. For an I/O rate of 1.25Gbps [30], total latency turned out to be:

$$\text{ceil}\left(\frac{(5+\omega)*0.8ns+2.75ns}{0.8(ns/clock_cycle)}\right) = (9 + \omega) \text{ clock cycle} \quad (3.4)$$

Consequently, the relationship between sys_clk_sd and clk_serdes for a given range of multiplexing ratio mux_ratio can be calculated by:

$$sys_clk_sd = \frac{clk_serdes}{9 + \omega} = \frac{clk_serdes}{9 + \text{ceil}\left(\frac{mux_ratio}{2}\right)} \quad (MHz) \quad (3.5)$$

The equation presented here has been derived from [61] and its accuracy has been verified by [68] in an off-the-shelf MFS with six virtex-5 FPGAs.

3.5. Multi Gigabit Transceiver (MGT)

A Multi-Gigabit transceiver (MGT) is a power-efficient module supporting line rates up to 32.75 Gbps [31]. Similar to SERDES, the principal function of MGT is to transmit parallel data as stream of serial bits, and convert the serial bits to parallel data at the receiver's end. The key performance metric of an MGT is its line rate, which is the number of serial bits transmitted per second. It facilitates either a direct, point-to-point electrical transmission or cooperation with optoelectronic transceivers connected to optical interconnections. Xilinx introduced its first MGT under the label "RocketIO" in the Virtex-II Pro series which was capable of operating up to 3.125 Gbps [32]. Latest Xilinx UltraScale+ FPGAs offer three types of MGTs: GTR, GTH and GTY. Each MGT supports different bit rates for the given FPGA series, are highly configurable and tightly integrated with the programmable logic resources of the device. CML (Current-Mode Logic) differential signaling standard is used on all MGT line rates of 10Gbps and above, for both data and clocks.

3.5.1. CML Signaling

CML is a high-speed point-to-point interface capable of supporting data rates greater than 10 Gbps. A typical CML transmitter/receiver structure is shown in Figure 3.6. The transmitter is constructed from a common-emitter differential pair with 50Ω collector resistors for optimal signal integrity. The output voltage swing is generated by switching the tail current through the output transistors. Switching a tail current of 16 mA across a

50 Ω resistor will create a differential signal swing of 800 mV (1600 mVpp). CML typically does not require any external resistors as termination is provided internally by both the transmitter and the receiver devices. CML offers all the advantages of differential signaling as discussed earlier. Table 3.1 lists the differences between LVDS and CML signaling standards.

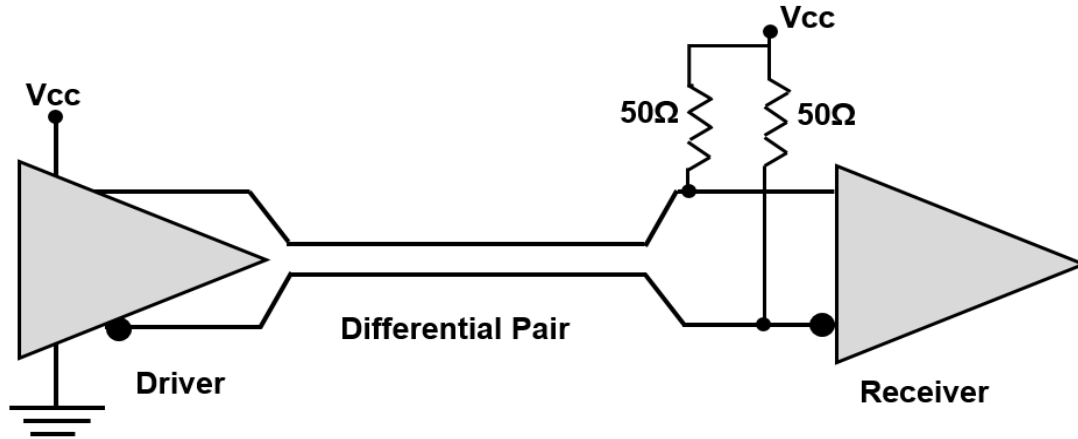


Figure 3.6: CML Architecture

Table 3.1: Comparison of LVDS & CML

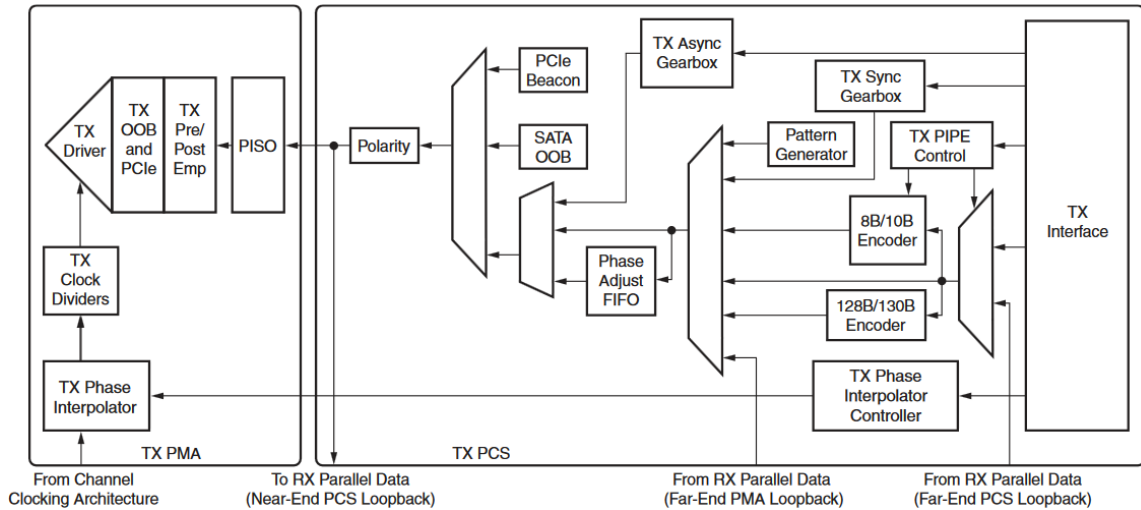
	Industry Standard	Max. Data Rate	Output Voltage Swing	Power consumption
LVDS	TIA/EIA-644	3.125 Gbps	$\pm 350\text{mV}$	Low
CML	N/A	10+ Gbps	$\pm 800\text{mV}$	Medium

3.5.2. MGT Architecture

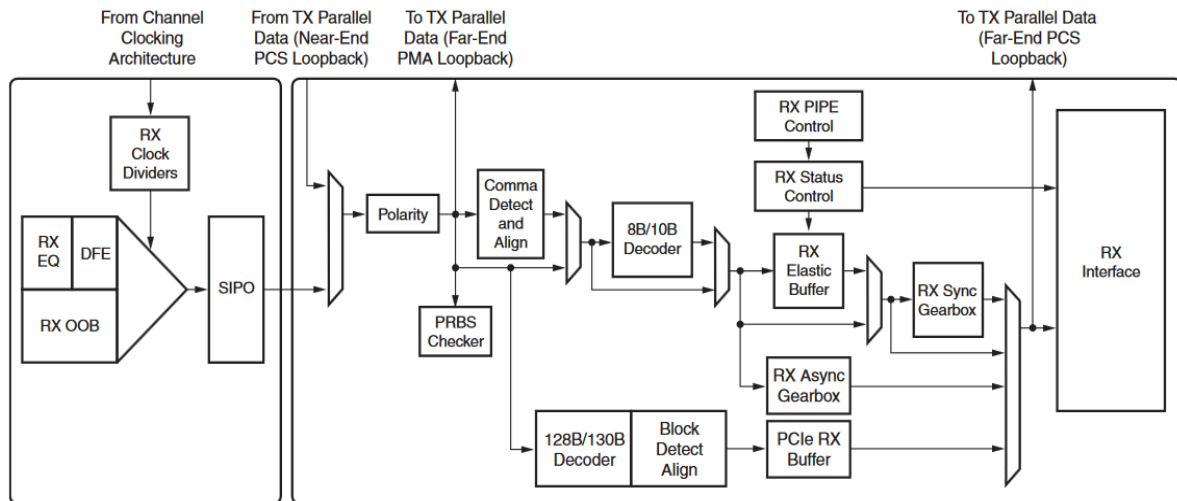
MGT employs *self-synchronous* interface where clock is embedded in the data stream [31] [33]. Figure 3.7 shows the basic architecture of MGT which consists of two sections: transmitter (TX) and receiver (RX).

Each transmitter and receiver is further sub-divided into two layers: PMA (Physical Media Attachment) and PCS (Physical Coding Sub-layer). PMA serializes the parallel data and de-serializes the serial data, while the PCS is responsible to process the data before

serialization and after de-serialization. The transmitter requires two positive-edge aligned input clocks TXUSRCLK and TXUSRCLK2. TXUSRCLK is the internal clock for the PCS logic, while TXUSRCLK2 is the primary synchronization clock for all signals into the TX side of the transceiver. Similarly, the receiver requires two positive-edge aligned input clocks RXUSRCLK and RXUSRCLK2. TXUSRCLK and TXUSRCLK2 are generated by TXOUTCLK. RXUSRCLK and RXUSRCLK2 are generated by RXOUTCLK. RXOUTCLK and TXOUTCLK are generated with PLL frequency multiplier and consequently they are synchronous. Furthermore, TXUSRCLK and TXUSRCLK2 are synchronous with RXUSRCLK and RXUSRCLK2. The PLL outputs feed the TX and RX clock divider blocks, which control the generation of serial and parallel clocks used by the PMA and PCS blocks. The transmitter consists of an Encoder, a First in first out (FIFO) and a parallel in serial out (PISO) block. The data is read from the FPGA fabric on the TXUSRCLK2 clock edges and outputted synchronously with the TXUSRCLK clock. The data from the FPGA interface is then encoded. Enabling the encoder increases latency through the TX path. The encoder can be disabled or bypassed to reduce latency, if not required.



(a) MGT Transmitter Block



(b) MGT Receiver Block

Figure 3.7: Multi Gigabit Transceiver Architecture [31]

Next, the encoded data is buffered in a FIFO which writes the data when the “write_en” pin is high and reads the data when the “read_en” pin is high. However, when the FIFO is bypassed the output of the encoder is directly fed to the PISO. The PISO block serializes the incoming parallel data and sends it out as a single-channel differential output signal. The GTY transmitter has a TX buffer and a TX phase alignment circuit to resolve any phase differences between the XCLK and TXUSRCLK domains. The TX phase alignment circuit comes into play when TX buffer is bypassed.

The incoming bit-serial differential signal is received by the Clock and Data Recovery (CDR) circuit in the RX unit, which extracts the recovered clock and uses it to sample the data. The Serial In to Parallel Output (SIPO) block de-serializes the data synchronously with XCLK. The subsequent blocks, Comma Detect and Align and Decoder in the RX data path function synchronously with PMA parallel clock domain (XCLK). Serial data should be aligned to symbol boundaries before it can be utilized as parallel data. In order to make alignment possible, TX send a recognizable sequence, called a comma. The Comma Detect and Align block searches for the comma in the received data. When it finds a comma, it moves the comma to a byte boundary so the received parallel words match the transmitted parallel words. If the received data is encoded, it must be decoded. The transceiver RX data path has two internal parallel clock domains used in the PCS: The PMA parallel clock

domain (XCLK) and the internal clock for the PCS logic (RXUSRCLK) domain. In order to receive data, the PMA parallel rate should be amply close to the RXUSRCLK rate, and any phase differences between the two clocks must be resolved. The RX elastic buffer is used to resolve differences between the XCLK and RXUSRCLK domain. Finally, the data reaches the RX interface. The RX interface includes two parallel clocks: RXUSRCLK and RXUSRCLK2. RXUSRCLK2 is the primary synchronization clock for all signals into the RX side of the transceiver. Received signals are sampled on the positive edge of RXUSRCLK2

3.5.3. MGT Multiplexing

MGT multiplexing scheme in MFS is shown in Figure 3.8. In all the 2D and 3D multi-FPGA platforms implementing MGT multiplexing, 16 2-byte wide GTY transceivers are instantiated along with ω -bit wide multiplexer/de-multiplexer. Each MGT consumes 2 MGT I/Os for transmitting and 2 MGT I/Os for receiving data. MGT transmitter and receiver are instantiated together and its duplexity is not reconfigurable. All MGT I/O pins are used for data transfer. When there is no multiplexing, ω equals 1. The maximum number of inter-FPGA signals passing through one MGT is labeled as *mux_ratio*, therefore $\omega = \text{mux_ratio} / 16$. Nevertheless, the data rate for Xilinx Kintex+ KU3P FPGA in SFVB784 package is limited to 12.5 Gbps according to [30]. In this research, the data rate of GTY transceiver is limited to 10 Gbps to facilitate the MGT reconfiguration.

According to [31], in a 2-byte multi-lane configuration, $F_{TXUSRCLK2} = F_{TXUSRCLK}$. Also, 8B/10B encoder (resp. decoder) is disabled and TX buffer is bypassed to minimize latency. This configuration is valid, because the distance travelled by the inter-FPGA nets is very small.

As discussed earlier, the internal design frequency can be calculated from post-routing critical path delay of the design. According to [30][31], T_{out} and T_{in} for MGT link can be neglected since those values are very small as compared to the board delay and TX and RX blocks latencies, therefore,

$$F_{TXUSRCLK2} = F_{TXUSRCLK} = \frac{1}{T} \quad (3.6)$$

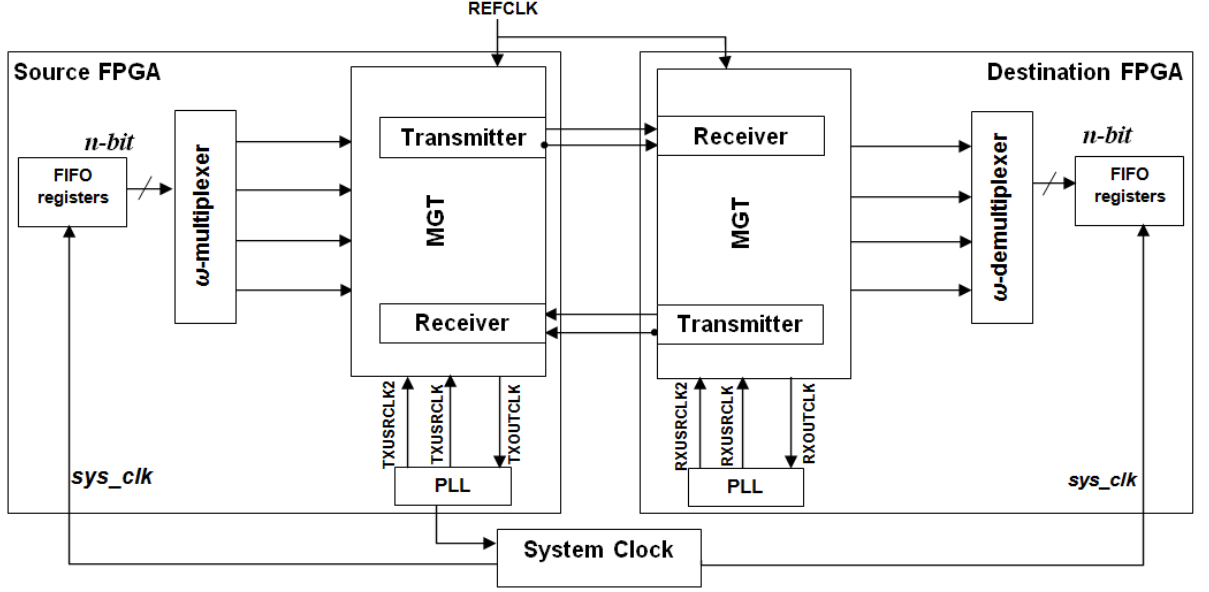


Figure 3.8: MGT Multiplexing Scheme

The latencies of the UltraScale+ GTY TX and the RX blocks have not yet been made available for public access. However, for our research, we contacted Xilinx technical support team, which graciously provided the Kintex UltraScale+ GTY TX and RX blocks latency values and are given in Table 3.2. The total latency of the TX and of the RX is respectively 75 and 93 UI (Unit Interval) for the configuration discussed earlier. UI is the minimum time interval taken to transmit one bit. Therefore, for a line rate of 10Gbps, the total latency is $168/10 = 16.8$ clock cycles. Moreover, $1 + \omega$ clock cycles are required for propagating the comma and ω -bit wide logic multiplexer/de-multiplexer. Therefore, the total latency turns out to be $((18 + \omega) \text{ clock cycles} + T_{trace})$. As the data rate is 10 Gbps, the total latency formula comes out to be:

$$\text{ceil} \left(\frac{(18+\omega)*0.1\text{ns} + T_{trace}}{0.1(\text{ns/clockcycle})} \right) \text{ clock cycles} \quad (3.7)$$

The board delays depend upon the type of interface employed in the given architecture. In this research, all 2D MFS are configured with PCB traces. Typical value of PCB board delay in an MFS is 2ns [5] [8] for a trace length of 6 inches.

Table 3.2: Latency Values of GTY TX & RX Blocks

TX	Latency (UI)	RX	Latency (UI)
TX Interface	16	PMA	36.5
8B/10B Encoder	-	PMA to PCS	8
TX FIFO	16	Comma Alignment	32
To TX PCS/PMA boundary	16	8B/10B Decoder	-
To Serializer	8		
PMA Interface	19	RX Interface	16
Total Latency	75	Total Latency	93

In the proposed 2D architectures, PCB traces are replaced by optical interfaces of the same length and all off-chip connections are realized by an optical link where an FPGA must be connected to the optical transceiver through MGT, which entails routing high speed off-chip optical traces. Board delay in such architectures includes optical transceivers delay as well as the optical trace latency. Therefore, the total board delay is the sum of 6 inches long optical trace delay i.e. 0.75ns and optical transceiver delay i.e. 500ps which turns out to be 1.25ns.

Whereas, in 3D MFS, the length of all off-chip optical connections is reduced to half of that in 2D architectures, due to vertical stacking. Therefore, the total delay is the sum of 3 inches optical trace delay i.e. 0.37ns and optical transceiver delay i.e. 500ps which turns out to be 0.87ns. Optical interface is discussed in further detail in Chapter 4 and 3D architectures are discussed comprehensively in Chapter 5.

Putting in the board delays of 2D PCB MGT, latency-optimized 2D optical MGT and 3D optical MGT architectures, the relationship between the system clock frequency and the internal design frequency is given by equations (3.8), (3.9) & (3.10) respectively.

$$sys_clk = \frac{F_{TXUSRCLK}}{38 + \omega} = \frac{F_{TXUSRCLK}}{38 + \frac{mux_ratio}{16}} \quad (MHz) \quad (3.8)$$

$$sys_clk = \frac{FTXUSRCLK}{31 + \omega} = \frac{FTXUSRCLK}{31 + \frac{mux_ratio}{16}} \quad (MHz) \quad (3.9)$$

$$sys_clk = \frac{FTXUSRCLK}{27 + \omega} = \frac{FTXUSRCLK}{27 + \frac{mux_ratio}{16}} \quad (MHz) \quad (3.10)$$

The above equations are derived from previous work [28], where authors presented the relationship between system clock and MGT clock and validated on the DNV7F2A board with a single testbench circuit.

3.6. Comparison of Three Multiplexing Schemes

After partitioning is done, the resource utilization of all FPGAs is well balanced and within the suggested range. However, there is still a chance that there are not enough FPGA pins available to satisfy design requirements. The solution is to multiplex design signals between FPGAs such that multiple compatible design signals are assembled and serialized through the same board trace and then de-multiplexed at the destination FPGA. The three multiplexing schemes are Logic Multiplexing, SERDES and MGT. We have discussed the three multiplexing schemes in great detail in precious sections. Here we present a brief comparison of the three schemes in Table 3.3. As it can be seen that Logic multiplexing provides minimum achieved data rate, whereas MGT provides maximum achieved data rate.

Table 3.3: Comparison of 3 Multiplexing Schemes

Multiplexing Scheme	Timing Model	Max. Data Rate	Single-Ended Signaling	Differential Signaling	Optical Interface
Logic Multiplexing	System-Synchronous	~100Mbps	✓		
SERDES	Source-Synchronous	>1Gbps		✓	
MGT	Self-Synchronous	> 10Gbps		✓	✓

Additionally, MGT and SERDES both operate on differential signaling. Out of the three multiplexing schemes, only MGT facilitates optical interface.

Later in the thesis, we will present the comparison of achieved system performance of the three schemes for increasing serialization factor.

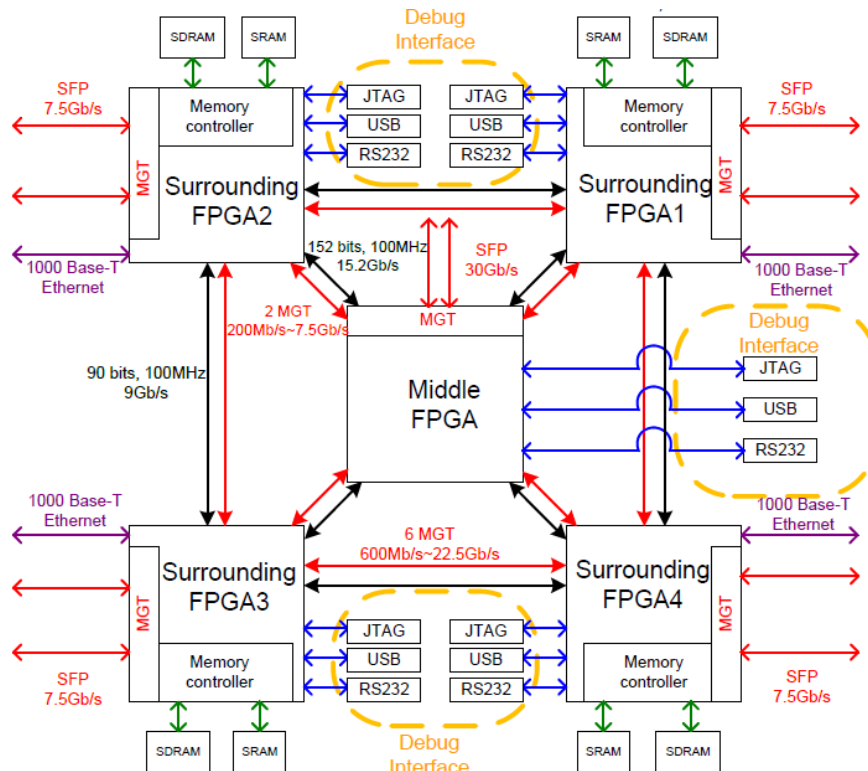
3.7. Previous Research on MFS Multiplexing

An extensive amount of research has been carried out in the field of MFS architectures and time multiplexing. Nonetheless, to our knowledge, the study done in evaluation and comparison of the system performance of the different time-multiplexed routing architectures using real benchmark circuits is extremely limited.

Babb et al. [21] presented the multiplexing concept in FPGAs, resulting in increased bandwidth and low-cost logic emulation. In virtual wires, the authors replaced the global router of traditional software with *virtual wires scheduler* and the *virtual wires synthesizer* which supported automatic pin multiplexing. Virtual Wires Scheduler determined a suitable schedule (feasible time-space route) of logical wires onto physical wires. Virtual Wires Synthesizer synthesized special multiplexers and registers to implement the chosen routing schedule. Virtual Wires emulation board contained 16 Xilinx XC4005 FPGAs interconnected in 2D nearest neighbor mesh. Designs of up to 18K gates were compiled on the demonstration system. Results including in-circuit emulation of a SPARC microprocessor indicated that Virtual Wires eliminated the need for expensive crossbar technology while increasing FPGA utilization beyond 45%.

Liu et al. [34] proposed a flexible and scalable multi-FPGA emulation platform employing high bandwidth, low latency parallel links between FPGAs to directly emulate interconnections in NoCs as shown in Figure 3.9. They presented a scalable, flexible hardware-based NoC emulation framework, through which NoCs of different types of network topologies, routing algorithms, switching protocols, and flow control schemes could be explored, compared, and validated with injected or self-generated traffic from both real and synthetic applications. The NoC emulation module board consisted of 5 Xilinx Virtex-5 FPGAs. The physical wires on the platform were organized as low-voltage CMOS (LVCMOS) parallel links and MGT links. The 4 surrounding FPGAs were

connected in a 2D mesh grid. Each link between the adjacent FPGAs on the grid provided 90 single-bit lines running at 100MHz with a total data throughput of 9Gbps. The 152-bit parallel LVCMOS interconnections were provided between the middle FPGA and the surrounding FPGAs which resulted in 15.2Gbps data bandwidth. Middle FPGA's 2 MGT transceivers were connected to each of the four surrounding FPGAs and the remaining 8 MGTs were reserved for off-board extensions.



S2C paper [35] compared the system performance of multiplexed single-ended and differential signaling in synchronous and asynchronous mode on Xilinx Ultrascale devices. Single-ended multiplexing used single-ended signals at a speedup to 290 MHz (Virtex UltraScale). This was determined by dividing the multiplexing ratio and taking into account setup, synchronization and off-chip delays. With a serialization ratio of 4:1, the system clock speed was 17.8 MHz and for ratio of 16:1, the system clock speed dropped to less than 10 MHz. However, in LVDS signaling, inter-FPGA data transmission rate of up to 1.6Gbps was achieved. The author showed that for a system with a clock speed of 11 MHz, if 12800 virtual connections were needed, single-ended multiplexing consumed 1600 physical I/O whereas, LVDS signaling consumed only 800.

In [36], fixed-latency MGT architecture was used for synchronous transfers and its latency performance was studied. The authors presented a fixed-latency synchronous architecture based on GTP transceiver of Xilinx Virtex 5 FPGA. Two different configurations were proposed for GTP. The latencies of the transmitter and the receiver in Configuration One, estimated by means of the user guide were respectively 4.5 and 9.5 clock periods. In Configuration Two, the latency of the transmitter remained the same; while latency of the receiver was 12.5 clock periods (due to the activation of the FIFO). Two off-the-shelf boards Xilinx ML-505 were deployed. The boards routed serial I/O pins of one of the GTPs on the FPGA to SubMiniature version (SMA) connectors. Transmitter and Receiver GTPs were connected with a pair of coaxial cables. One design implementing a link according to Configuration One and the other design implementing Configuration Two were presented. Due to CIMT transmission, 8b10b encoding-decoding was disabled. On the transmitter side, some logic encoded 16-bit words incoming from a payload generator into 20-bit CIMT words and transferred them to the GTP. On the receiver side, some logic received 20-bit symbols from the GTP and performed CIMT decoding and the frame alignment. The latencies of the transmitter and the receiver were measured and it was concluded that most of the latency of the transmitter was due to the fabric encoding logic, while the GTP had smaller latency. On the receiver end it was the converse, the GTP introduced more latency than the alignment and decoding logic. The latency of the transmitter was the same in both Configuration One and Configuration Two. The latency of the receiver was measured to be increased in Configuration Two, due to activation of the FIFO.

Maxwell [24] is a 32-way IBM Bladecentre containing 64 Xilinx Virtex-4s using InfiniBand cables. It targeted HPC rather than data center workloads and demonstrated much faster system performance. Physically, Maxwell comprised of two 19-inch racks and 5 IBM BladeCentre chassis. Four of the BladeCentres had 7 IBM Intel Xeon blades and the fifth had 4. Each blade was a diskless 2.8 GHz Intel Xeon with 1GB main memory and hosted two FPGAs through a PCI-X expansion module. The FPGAs were mounted on two different PCI-X card types – Nallatech HR101s and Alpha Data ADM-XRC-4FXs. The FPGAs had up to 1,024 MB external memory and 4 MGT Rocket I/O connectors. All 64 FPGAs were wired together directly in a two-dimensional TORUS. However, it did not implement the time-multiplexing concept.

In [14], the IBM's Bluegene/Q project was mapped on a Virtex-5 only platform and its performance was studied for a wide range of TDM ratio. It was configured with 24 node cards and used 45 Xilinx Virtex-5 LX330 FPGA devices, in addition to the control FPGA devices and discrete SRAM and DRAM components. It was constructed with flexible cable interconnect structure facilitating multiple connection topologies. The Active Backplane provided flexible interconnect with high-speed LVDS-based point-to-point communication links. 8:1 LVDS SERDES source-synchronous communication maximized the overall system performance and minimized link latency. The two link designs developed supporting 32:1 and 96:1 multiplexing ratio were able to achieve emulated processor clock frequency of 4MHz.

In [37], system performance was compared for a design running on HAPS-70 and HAPS-80 when pin multiplexing was employed. HAPS-80 with HAPS ProtoCompiler 2016.03 multi-FPGA pin multiplexing capability using Synopsys' proprietary HSTDM exhibited an increase in performance on an average of 15% for the same pin multiplexing ratio. However, the effect of routing architectures was not discussed.

In [8] the achieved performances for a set of designs mapped on the three different categories of multi-FPGA platforms were compared. The performance gains between these platforms are quantified with Logic and SERDES multiplexing schemes. The platform comprised of six identical Xilinx FPGAs. The author's proposed routing algorithm

increased the performance up to 25% for Logic Multiplexing and 22% for SERDES as compared to the Turki's algorithm in the off-the-shelf platform by routing multi-terminal nets in multi-point tracks.

3.8. Summary

This chapter summarized the different multiplexing schemes, their comparison and background research in this area.

The performance of MFSs is limited by the limited number of inter-FPGA traces. To accommodate the larger number of inter-FPGA signals on are fewer available inter-FPGA traces, several inter-FPGA signals need to be multiplexed and sent together onto a single board trace. There are three multiplexing techniques used for MFS prototyping: Logic Multiplexing, SERDES and MGT. The architecture and achieved system performance of these techniques was discussed comprehensively. Then the comparison among the three multiplexing scheme was presented. Also, previous research done on this subject was presented in detail. An extensive amount of research has been carried out in the field of MFS architectures and time multiplexing. Nonetheless, to our knowledge, the studies done in experimentally evaluating and comparing the three multiplexing schemes for different MFS architectures is extremely limited. Furthermore, so far no work has been done in evaluation of the system performance of the 3D MFS routing architectures with serial optical interconnections.

Chapter 4 Optical Interface in MFS

4.1. Introduction

Rising data rates in ICs and I/O density are challenging the traditional copper interconnect solutions. Over the last few decades, data bandwidth requirements in many real world applications are on the rise demanding a compatible high-speed interface capable of maintaining multi-gigabits data rate. Designers typically choose copper interconnect on a printed circuit board (PCB) for chip-to-chip and chip-to-module interfaces. However, copper based interconnects are incapable of scaling up with the data rate and exhibit lossy characteristics with increasing frequency. For instance, FR-4 copper trace material suffers from a loss of ~ 0.5-1.5 dB/inch at 5 GHz (Nyquist for 10 Gbps rate), and the loss increases to ~ 2.0-3.0 dB/inch at 12.5 GHz (Nyquist for 25 Gbps rate) [38]. Maximum bandwidth is also limited by return loss, insertion loss and crosstalk.

The performance of an MFS can be enhanced if the off-chip electrical interconnects are replaced by optical interconnects. The superiority of physical properties of photon propagation over electron propagation translates to fundamentally improved bandwidth, distance, cross-section and latency relationships of optical interconnects as compared to its electrical counterparts.

As shown in Figure 4.1, over the past few years optical interface has evolved from long range spanning over hundreds of kilometers into shorter distance links of millimeters. A couple of decades ago, fiber optic signaling was applied to distances from 10m and beyond, because of the characteristics and cost structure. However, with the advancement in technology, now short-ranged optical interconnects enable centimeters to millimeters of chip-to-chip, board-to-board, on-board and system-to-system connectivity at multi-gigabit rates overcoming the loss, signal integrity, and power challenges of copper electrical signaling.





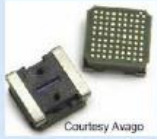


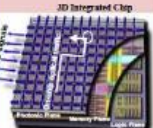
Year	1990	2004	2011	>2015?	>2020?
Application	Storage & Network I/O BOX TO IO DEVICES	Inter-Rack (node to node) BOX TO BOX	Intra-Rack & Backplane (node to node) INSIDE THE BOX	Card & Proc-memory BW INTRA CARD	optical switch (ckt) OPTICS BASED ARCH
Initial Enabling Technology	200 Mbps LEDs, MM fiber optics  1Gbps Ethernet 	2.5 Gbps vcsel arrays, Parallel fiber optics  5Gbps (2008) 	10Gbps integrated highly parallel fiber optics,  Courtesy Avago	10-25-40Gbps on module, highly parallel, dense, + WDM? Multi-core?  	High functionality Photonics switch IC's 
I/O bottleneck solved	BW at 100m	BW at >10m Off-rack	BW at >1m Off-module	BW at <1m Off-module, DIMM	Switch BW Network
System Introduction	IBM S390	IBM Federation Switch/ASCI Purple	IBM P775	?	?

Figure 4.1: Optical Interface Evolution [39]

4.2. Short-Range Optical Interface

Replacing electrical wires with on-board short-range optical fibers poses challenging requirements in terms of cost, density, power efficiency, thermal control and compactness. However, this migration is becoming more and more inevitable because at multi-gigabit data rates, electrical interconnects are restricted in their performance due to signal integrity, latency and, power issues. In contrast with copper interfaces, optical fibers render virtually no loss. Moving a serial link from the electrical domain to the optical domain has numerous advantages [39] [40]:

- Energy efficiency
 - Energy per bit at a given distance
 - Lower distortion and crosstalk
 - Immune to noise (electromagnetic interference and radio-frequency interference)
 - Signal Security (difficult to tap)

- Lower cost per Gbps
 - Lower cost-of-ownership
 - Retrofits, upgrades and infrastructure reuse
 - Integrate-ability/backwards compatibility
 - Fewer parts and interfaces
- Architectural flexibility
 - Interface simplification
 - Layout flexibility
 - Modularity
- Form factor
 - Pin-out density
 - Plug density
 - Stackability
- Nonconductive (does not radiate signals) - electrical isolation
- No common ground required
- No short circuit and sparks
- No inductive voltage drops on pins and wires
- Reduced size and weight cables (excluding connectors)
- Ability to have 2-D interconnects directly out of the area of the chip rather than from the edge
- Radiation and corrosion resistant
- Less restrictive in harsh environments

With the advancement in optical communication over the past few years, the factors of power consumption, price, protection, and maintenance, that were listed as its disadvantages have now been significantly diminished.

4.2.1. Optical Fibers

Before 1970, optical fibers were used primarily for medical imaging over short distances. Their application for communication purposes was considered unfeasible due to their high losses (~ 1000 dB/km). However, as the technology progressed, loss across an optical fiber

reduced to less than 20dB/km and then to only 0.2 dB/km near the 1.55- μ m spectral region. The availability of low-loss fibers started the era of fiber-optic communications [40].

The optical fiber is comprised of isolator materials, such as glass or plastics that operate as a waveguide to propagate light and make it immune to electromagnetic wave disturbance. The fiber is made up of a central dielectric core clad with a dielectric material having a higher index of refraction than the core to guarantee total internal reflection.

There are two types of optical cables; single-mode fiber (SMF) and multi-mode fiber (MMF). There are vital differences between multi-mode fiber and single-mode fiber. Single-mode fiber comprises of a single strand of fiber that transmits data. They have a much thinner core, typically 8 to 10 micrometers and they propagate light as an electromagnetic wave operating in a single transverse mode. Single mode fiber cables can carry signals over lengths 10 to 100 times greater than multi-mode cables and they require more expensive transceivers. SMF has a loss of ~0.4 dB/km and 0.25 dB/km at 1300-nm and 1550-nm wavelengths respectively and is more expensive due to its smaller core and has a bandwidth close 100 THz in practice. Optical transmitters for SMF are also considerably more expensive because they employ higher cost, long wavelength optical sources emitting around 1500 nm. Additionally, the higher cost of optical transmitters for SMF is because the high precision alignment of the laser to the single-mode core of the fiber is more difficult to achieve. Optical connectors for SMF are also more expensive than their multimode counterparts.

Multi-mode fiber has larger core diameter than single-mode fibers and allows multiple modes of light to propagate. Optical signals are dispersed into a number of paths or modes, as they propagate through MMF core. The laser that drives the optical signal over an MMF is usually a light emitting diode (LED) or Vertical Cavity Surface Emitting Laser (VCSEL). MMF can carry signals over distances of up to 1 km. At 10 Gbps, the reach distance for a MMF is up to 300m. Multi-mode cables have core diameters of 50 to 100 micrometers, and they propagate light using principles of geometrical optics. MMF exhibits a loss of ~3 dB/km and ~1 dB/km at 850-nm and 1300-nm wavelengths, respectively. MMF is less expensive due to its larger core and has a bandwidth ~ 2 GHz

km. Unlike their single-mode counterparts, the alignment of the lasers is more relaxed in MMF, and their transmitters and connectors are also easier to manufacture.

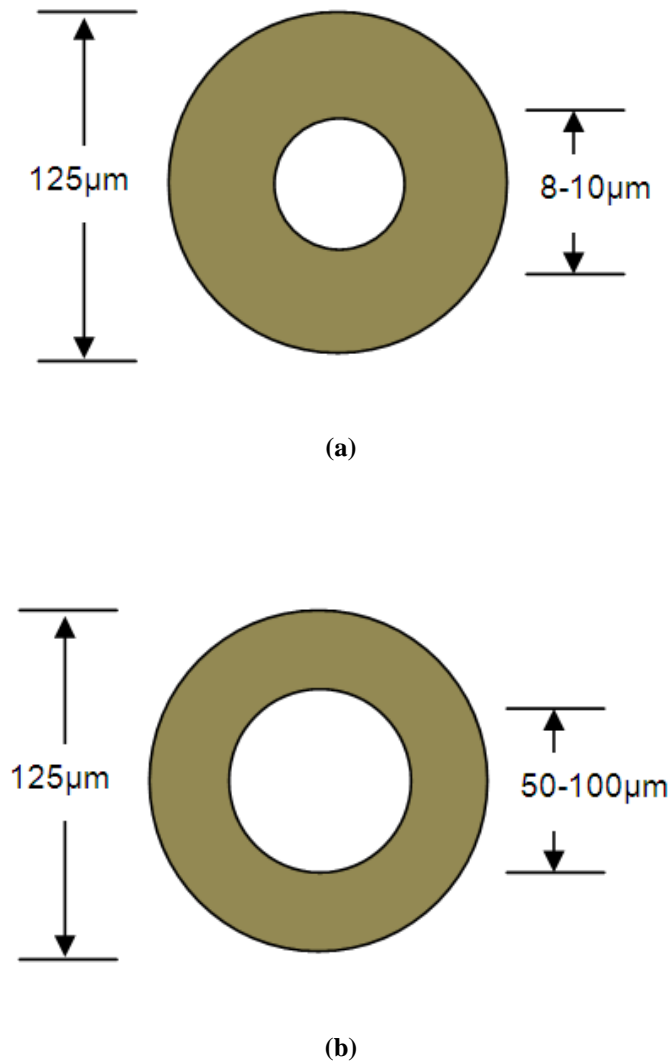


Figure 4.2: (a) Single-Mode (b) Multi-Mode Optical Fiber Core Dimensions

MMF has been the standard fiber used for 300 m or less at 10 Gbps and is the type of fiber for which short reach interconnect optical engines have been designed [51]. Table 4.1 presents the comparison between the two types of optical fibers.

In both types of fiber, an effect called dispersion can impact the fidelity of the transmitted signal. As the signal travels through the fiber, the distribution of wavelengths of light

containing signal data are interacted with in different manner, some wavelengths experiencing more delay or varying degrees of attenuation. The impact of optical dispersion on the transmitted waveform is distinctly different from the copper impacts on electrical signals.

Table 4.1: Comparison between MMF & SMF

Multi-Mode Fiber	Single-Mode Fiber
<ul style="list-style-type: none"> ➤ Low-Cost Sources <ul style="list-style-type: none"> • 850 nm & 1310 nm LEDs • 850 nm Laser at 1 &10Gbps • Low precision packaging ➤ Lower Cost Connectors ➤ Lower Installation Cost ➤ Higher Loss ➤ Lower Bandwidth ➤ Distance up to 2 km 	<ul style="list-style-type: none"> ➤ High-Cost Sources <ul style="list-style-type: none"> • 1310+ nm Laser at 1 &10Gbps • High precision packaging ➤ Higher Cost Connectors ➤ Higher Installation Cost ➤ Lower Loss ➤ Higher Bandwidth ➤ Distance up to 60 km+

Optical Fiber Latency

Plastic optical fiber (POF) is used for inter-FPGA connections in our proposed MFS architectural models. POF uses *poly methyl methacrylate* (PMMA) as the core material and is a low-cost optical fiber as compared to glass optical fiber. Moreover, unlike glass, plastic fiber is flexible and can be easily cut and bent to fit in on-board short length requirements.

In flexible plastic optical fiber, the latency of the fiber is the time taken by the light to travel a specified distance through the glass core of the fiber. Light moving through the fiber optic core depends upon the effective refractive index parameter (n_{eff}) which is described as the comparison between velocity of light in vacuum (c) and velocity of light in a medium (v).

$$n_{eff} = \frac{c}{v} \quad (4.1)$$

Typical value of n_{eff} is 1.46 for plastic PMMA optical fiber [41]. Meanwhile v is also defined in terms of propagation delay through optical fiber t_f (s) and distance travelled L (m).

$$v = \frac{L}{t_f} \quad (4.2)$$

By substituting Eq. (4.2) into (4.1) and eliminating v , the one-trip propagation delay of optical fiber can be derived [86] as:

$$t_f = \frac{n_{eff}L}{c} \quad (4.3)$$

4.2.2. Optical Transceivers

Employing optical interconnects in multi-FPGA systems requires optical transceiver (transmitter and receiver) to be inserted in the communication link. These transceivers act as translators between the on-chip electrical signaling and the optical signaling that goes over the optical fiber. Optical transceivers are multi-lane devices which can either be implemented as mono-directional transmitters / receivers or as bidirectional. The single directional approach is employed for higher lane counts typically 12 channel devices, whereas the bidirectional approach is used for lesser number of lanes like 4 bidirectional channels per device. Lately, 8 and 12 channel bidirectional devices have been sampling as pre-commercial products. Even superior devices with channel counts up to 100's are being demonstrated in research [51].

Optical Transmitter

Figure 4.3 shows the block diagram of an optical transmitter. The function of an optical transmitter is to convert an electrical input signal into the corresponding optical signal and then send it onto the optical communication channel. The major components of optical transmitters include:

- Optical Source
- Optical Modulator
- Driving Circuitry

- Channel Coupler

An optical source can be light-emitting diodes (LEDs) or semiconductor lasers. They offer several inherent advantages such as their compact size, high efficiency, and good reliability, small emissive area compatible with fiber-core dimensions, right wavelength range and possibility of direct modulation at higher frequencies.

Optical modulator converts electrical signal into its optical counterpart. The optical source is biased at a constant current to provide the continuous-wave (CW) output, and the optical modulator placed next to the optical source converts the CW light into a data-coded pulse train with the right modulation format.

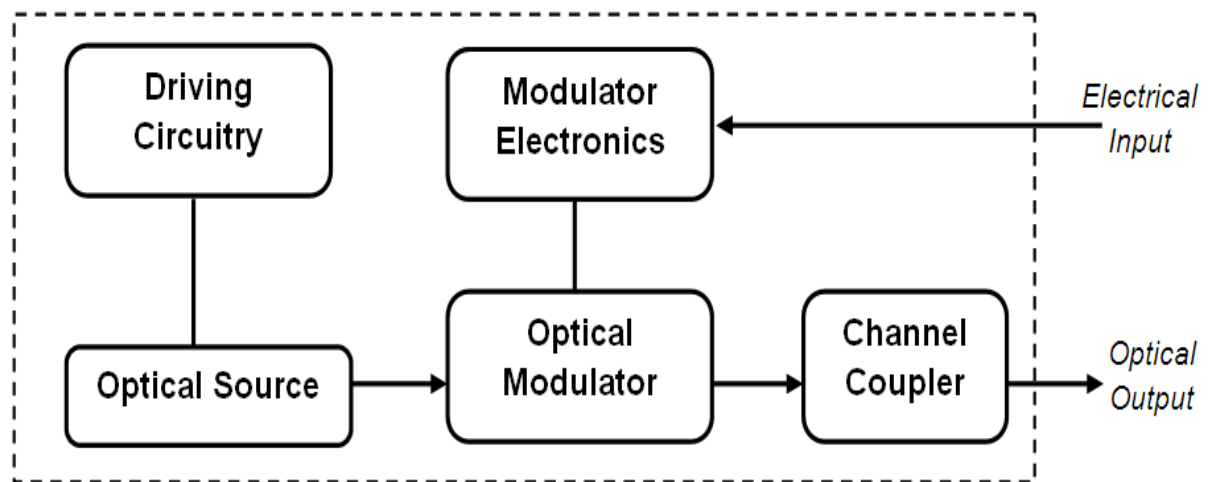


Figure 4.3: Optical Transmitter Block Diagram

The function of driving circuitry is to supply electrical power to the optical source and to modulate the optical output in accordance with the signal that is to be transmitted. Driving circuitry is relatively simple for transmitters with LED optical source as compared to high-bit-rate optical transmitters with semiconductor lasers as an optical source. Driving circuit is designed to deliver a constant bias current as well as modulated electrical signal.

The channel coupler is typically a micro-lens and its function is to focus the optical signal onto the entrance plane of the optical fiber with the maximum possible efficiency.

Furthermore, an optical transmitter also has a servo loop which is employed to maintain constant average optical power and a thermoelectric cooler to stabilize the laser

temperature. The bit-rate of optical transmitters is often limited by electronics rather than by the optical source itself.

In the transmit direction, from source FPGA to transceiver, keeping jitter low is vital. The transceiver driver must therefore provide preemptive equalization that helps to provide the lowest possible jitter to the electrical-to-optical interface: low-jitter, high-performance PLLs, and three-tap equalizers.

Optical Receiver

Figure 4.4 shows the block diagram of an optical receiver. The function of an optical receiver is to convert an input optical signal coming from the optical communication channel into the corresponding electrical signal. The design of an optical receiver depends on the modulation format employed by the optical transmitter.

The major components of optical receiver include:

- Channel Coupler
- Photo-detector
- Demodulator

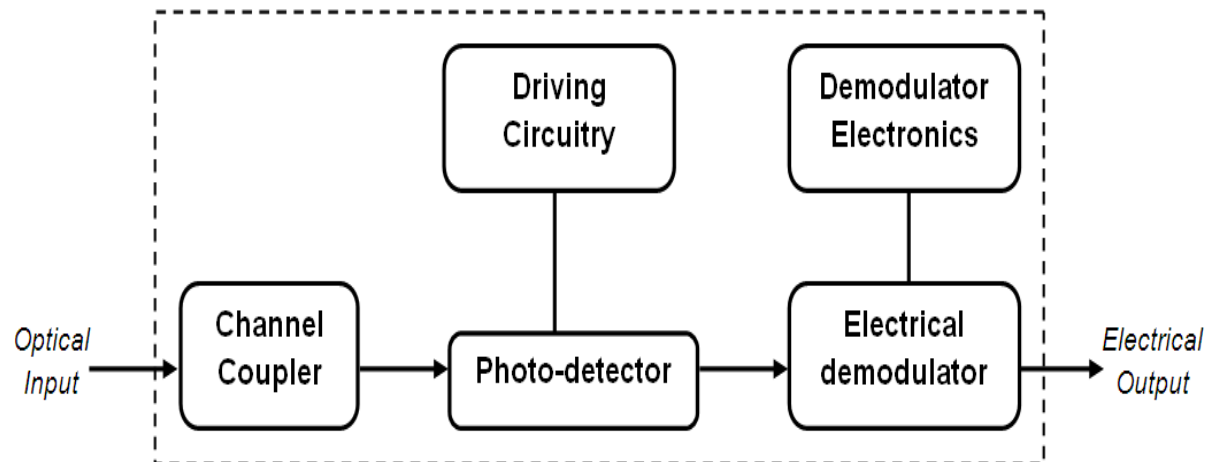


Figure 4.4: Optical Receiver Block Diagram

The function of the channel coupler is to focus the received optical signal from the optical fiber onto the photo-detector. The coupling scheme in the receiver is similar to that used in

the optical transmitter. Semiconductor photodiodes are employed as photo-detectors due to their compatibility with the whole system. They convert the incoming optical bit stream into an electrical time-varying signal. The design of the demodulator depends on the modulation scheme used by the lightwave system. Most lightwave systems use a scheme called "intensity modulation with direct detection" (IM/DD). Demodulation is done by a decision circuit which identifies bits as either 1 or 0, depending on the amplitude of the electric signal. The accuracy of the decision circuit is dependent upon the SNR of the electrical signal generated by the photo-detector. Due to the noise inherent in the optical receiver, there is always a finite probability that a bit would be identified incorrectly by the decision circuit. That's why receivers are designed to operate in such a way that the error probability is quite small i.e. typically less than 10^{-9} .

After data has been translated from the electrical domain and transmitted across the optical channel, it is translated back into the electrical domain at the receiver end. The receiver must compensate for the degradation in the electrical signal transferred from the varying amounts of dispersion and jitter introduced to the optical signal by the optical fiber. Also, for any amount of jitter that cannot be compensated for, the receiver must have a high inherent jitter tolerance to be able to receive valid and correct data. That's why, same LC tank PLLs that are used in the transmitter are also used to drive the CDR circuitry in receiver end to deliver jitter tolerance.

Optical Transceiver Latency

The latency in the optical transmitter is due to the conversion process from electronic to photonic state in the converter components. The optoelectronic module at the receiver end assigned to reconvert the photons into the electrons is a photo-detector. The total delay (D_T) on an optical link has contributions from optical transceiver (D_{opt}), optical fiber (t_f) and air propagation (D_{air}) expressed [86] as:

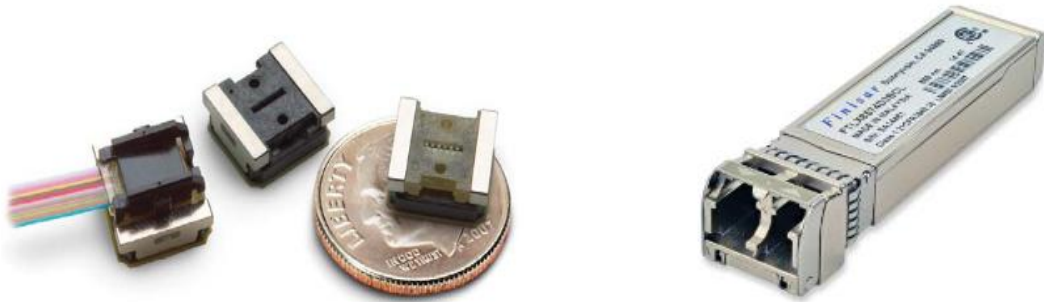
$$D_T = D_{opt} + t_f + D_{air} \quad (4.4)$$

The optical transceiver delay is based on the equipment data sheet. The air propagation has negligible contribution to the total delay and therefore can be abandoned. The roundtrip delay on an optical link will be $2*D_T$.

Optical Transceiver Types

The features that an optical transceiver needs to support must be identified based on the electrical and optical specifications of the design implemented. One common differentiator between transceivers is the length and type of optical fiber that it can drive.

Smaller form factors enable direct mounting of optical transceiver on the FPGA package requiring a footprint of mm^2 as claimed by Altera. Altera incorporated high-speed optical transceiver onto the package that held the FPGA reducing the electrical signal path from the FPGA I/O pin to the input of the optical transceiver to just a fraction of an inch. The resulting shorter path reduced signal degradation and jitter and improved the signal integrity and reduced data errors caused by parasitic elements in the signal path. Furthermore, the reduced FPGA-to-transceiver interconnect reduced the overall power consumption for both FPGA and optical module.



(a) *Avago Technologies MicroPOD Optical modules*

(b) *Finisar SFP+ Optical Transceiver*



(c) *Altera Optical FPGA*

Figure 4.5: Optical FPGA & Transceivers

As shown in Figure 4.5(c), the hybrid FPGA package Avago Technologies MicroPOD12-channel optical transceivers mounted on its two corners. One corner hosts 12 transmit channels and the other corner, supports 12 receive channels. By mounting the transceiver in the corners of the FPGA, Altera claims to have reduced the distance between the SERDES and the optical modules to less than a centimeter

Various types of optical transceivers are available depending upon their form factor, size, electrical and optical specifications, and lane width, such as SFP, SFP+, XFP, QSFP and QSFP+.

Small form-factor pluggable (SFP), is a compact, hot-pluggable transceiver used for both telecommunication and data communications applications. It has a theoretical maximum bandwidth of 5Gbps, although in practice, it is often used for 1Gbps connections. SFP can support a variety of wiring types, including Ethernet, SONET, single-mode fiber, and multi-mode fiber.

SFP+ is an enhanced version of the SFP which has a maximum transmission speed of 16Gbps, but it's generally used for data rates up to 10Gbps. SFP+ supports 8 Gbps Fiber Channel, 10 Gbps Ethernet and Optical Transport Network (OTU2). Its applications include SONET OC-192, SDH STM-64, OTN G.709, CPRI wireless, 16G Fiber Channel, and the emerging 32G Fiber Channel applications.

XFP was introduced before SFP+ and is a standardized form factor for serial 10 Gbps fiber optic transceivers. It is protocol-independent and fully compliant to the many Ethernet and Optical standards, supporting data rate from 9.95Gbps to 11.3Gbps. XFP transceivers are used in data and telecommunication optical links and offer a smaller footprint and lower power consumption as compared to its other counterparts.

QSFP stands for quad (4-channel) small form-factor pluggable. It is a compact, hot-pluggable transceiver employed for data communications applications. QSFP supports data rate up to 40Gbps on either Ethernet or Fiber, along with SONET and Infiniband. QSFP+ supports data rates higher than 40Gbps. Highest-speed format is QSFP28 that allows four simultaneous 28Gbps connections, or a total of 112Gbps.

4.3. MFS Serial Optical Interface

FPGAs with integrated high speed serial transceivers and optical interconnects offer an efficient and flexible platform. However, as discussed earlier, an FPGA must be connected to the optical transceiver, which entails routing high speed off-chip optical traces.

When implementing a short-range optical interface in MFS, several choices need to be made based on the design requirements:

- Type of fiber

The choice is between multi-mode and single-mode fibers, depending on the interface and the travel distance.

- Types of optical transceiver modules

Many form factors exist, varying in size, features, electrical and optical specifications, and lane width.

Figure 4.6 shows a simplified structure of two FPGAs connected through MGT transceivers and flexible plastic optical fiber. Optical transceivers inserted in this communication link facilitate the electrical to optical transformation required for optical interface. The data from internal FPGA fabric is transmitted to GTY transmitter in the source FPGA. From there, the electrical signals are converted to the optical signals by the optical transmitter. The data is transmitted from the source FPGA to the destination FPGA via optical fibers.

On the receiving end, the optical receiver converts the incoming optical signals back to electrical signals and sends them to the GTY receiver in the destination FPGA. And finally the data is de-serialized and sent to internal FPGA fabric for further processing.

The architectures and working of GTY transceivers and optical transceivers have already been explained in detail previously in the thesis.

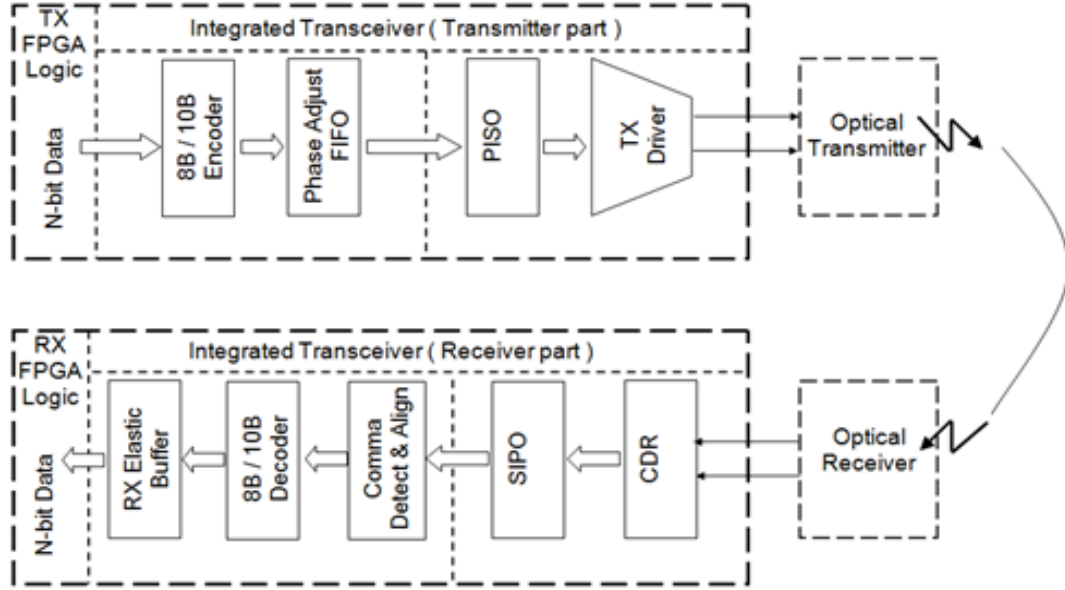


Figure 4.6: Simplified inter-FPGA serial optical interface structure

4.4. Proposed Latency-Optimized 2D MFS with Optical Interface

4.4.1. Proposed Architecture

The chosen FPGA KU3P offers 16 GTY transceivers with 32.75 Gbps inter-FPGA communication data rate and SFP+ compliance [31] [38] [44]. MGT serial communication architecture has already been discussed in detail in previous chapter. In a multi-FPGA setup with serial optical interface, each FPGA must be connected to every other FPGA via SFP+ optical transceiver, creating a bidirectional link. The distance between the GTY transceiver and optical transceiver is very short (usually fraction of an inch) and does not add to the overall latency of the link. This short distance also reduces signal degradation and jitter, consequently improving the signal integrity [7] [43].

For our research, we have employed *Finisar's* (FTLX8574D3BCV) SFP+ short-range 10.3Gbps optical transceiver designed for multimode fiber. SFP+ optical transceiver is short for small form-factor pluggable which is compact, hot-pluggable transceiver having a power dissipation of less than 1W [42]. It can be operated at a commercial temperature range of -5°C to 70°C . It is also SFF-8431 optical and IEEE 802.3 Ethernet protocol compliant. Keeping the distance from the transceiver I/O pad of the FPGA chip to the input of the optical transceiver very short, reduces signal degradation and jitter, consequently

improving the signal integrity and reducing overall power consumption for both FPGA and module [43].

As discussed earlier two routing architectures are employed for this research i.e. CCG and TORUS. Figure 4.7 shows these two routing architectures for 6 FPGAs. Figure 4.7(a) shows the connectivity and routing architecture of CCG, where all FPGAs are directly connected to each other. Figure 4.7(b) is the TORUS routing architecture where FPGAs are connected only to their horizontal and vertical neighbors. All inter-FPGA links are either copper traces or optical links depending upon the MFS models.

Conventional 2D MFS employ copper traces as inter-FPGA links and their typical length is 6-7 inches [5]. Since there are 16 MGTs per FPGA, that's why 16*2 MGT I/Os per FPGA are available for data communication in each routing architecture.

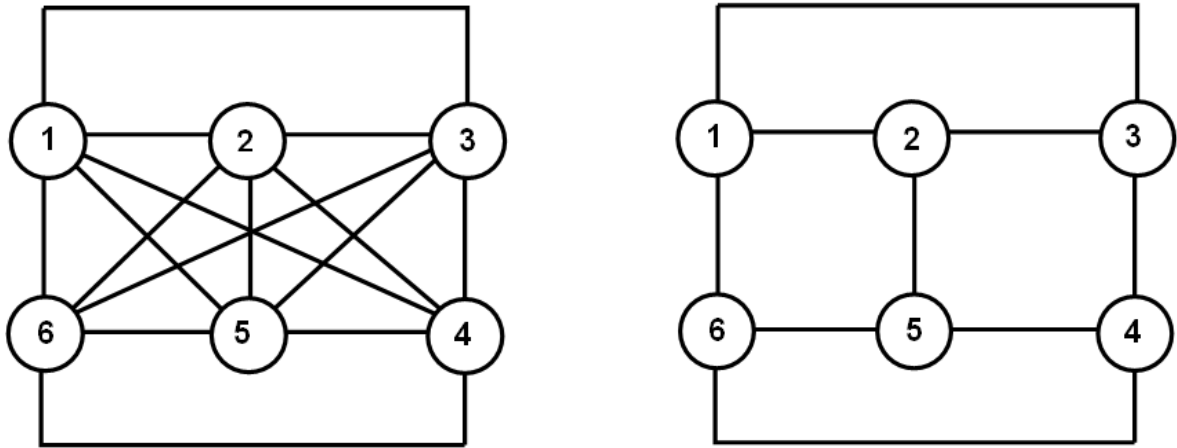


Figure 4.7: 2D MFS Routing Architectures (a) CCG (b) TORUS

In the proposed latency-optimized 2D MFS architectures, all inter-FPGA PCB connections in the two routing architectures are replaced by 6 inches long plastic optical fiber. Due to the planar nature of these MFSs, the inter-chip distances remain the same in both proposed and conventional architectures and therefore, the length of optical links does not change in the proposed MFS. As discussed previously, an off-chip PCB trace of this length has 2ns delay. Whereas, according to equation (4.3) & (4.4), the propagation delay of a 6 inches long optical fiber turns out to be 0.75ns. *Finisar's* (FTLX8574D3BCV) SFP+ optical transceiver has a delay of 500ps. This implies that total off-chip delay is 1.25ns in a 2D

MFS architecture with optical interface which is a 37.5% decrease per link as compared to conventional planar MFS.

Is it Possible?

Here, a question arises whether such short-length optical connections are even manufacturable? Is it practical to employ optical interface for such short on-board hops? The answer is, YES! [58] [59] [60] presented an optoelectronic FPGA demonstrator which used a smart-pixel like interconnect structure to create a logically 3D architecture (Figure 4.8). This architecture *conceptually* consisted of a number of FPGAs interconnected bi-directionally in a regular pattern. The optical components consisted of two 8x8 optical source arrays and two 8x8 InP detector arrays.

All 256 optical channels were designed to operate at an information rate of 80 Mbps. The optical pathways between the central chip and its two neighbors consisted of removable 8x16 POF connectors. The two outer chips were also equipped with 2x8x8 ribbons with horizontal insertion POF connectors. All optical fibers had the diameter of 125 μ m (120 μ m core) and length of 20cm.

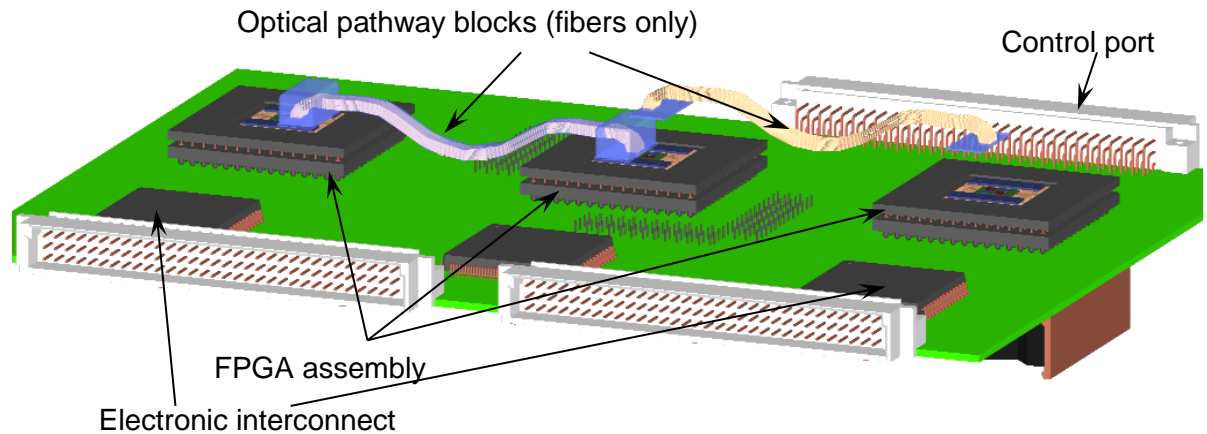


Figure 4.8: Demonstrator design with the 3 optoelectronic FPGA chips and encapsulated Optical pathway [60]

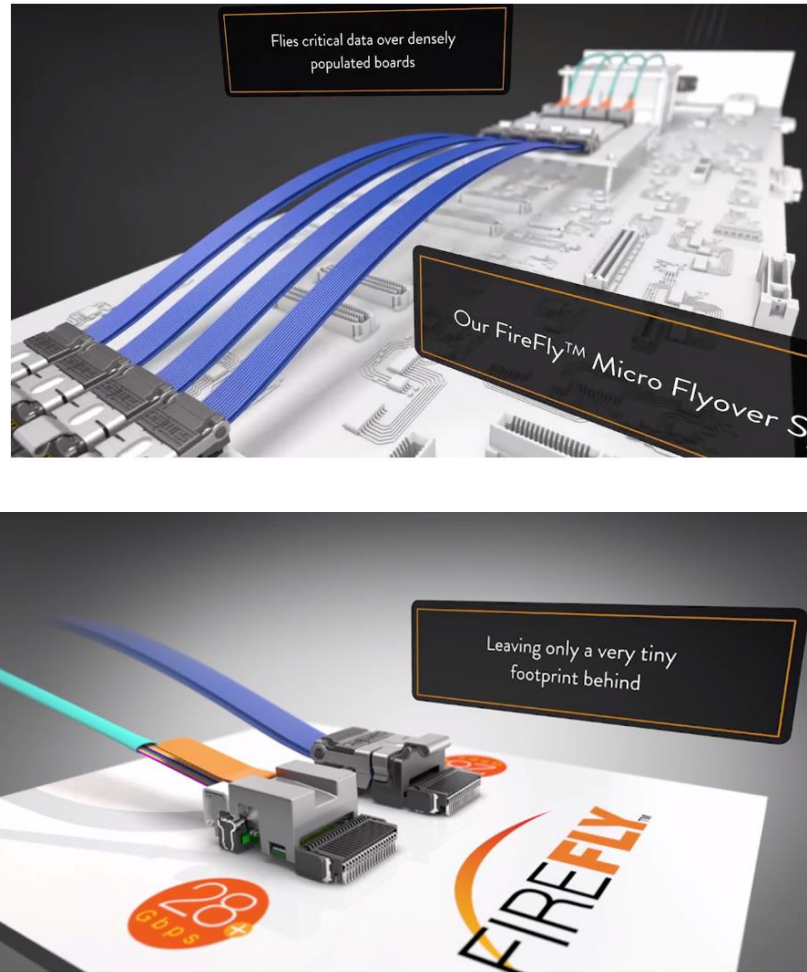


Figure 4.9: Samtec FireFly™ Micro Flyover System [62]

The Samtec FireFly™ Micro Flyover System as shown in Figure 4.9 is the first interconnect system that gives a designer the flexibility of using micro footprint optical and copper interconnects interchangeably with the same connector system. The FireFly™ system enables chip-to-chip, board-to-board, on-board and system-to-system connectivity at data rates up to 28 Gbps. The optical fibers allows the data “to fly” over the board enhancing signal integrity. The overall length of the optical fiber ranges from 8cm to 999cm and depends on the fiber type [62].

As already discussed, the electrical to optical conversion in an optical transceiver takes less than a Watt of power and its size has considerable reduced over the past few years. Adding to that, the above-mentioned instances of short-range optical fibers prove that the optical transceiver / fiber assembly is not only manufacturable and practically feasible but also

commercially available for short-range on-board chip-to-chip interconnections. Therefore, we can confidently justify their application in our research for both 2D and 3D MFS architectures with optical interface.

4.4.2. Multiplexing in Proposed Architecture

Scarce I/O resources still remains a problem even in latency-optimized structures and the obvious solution is multiplexing. Since multi-gigabit transceiver pins are the only ones capable of supporting off-chip optical interface, that's why MGT multiplexing scheme is employed in the proposed architectures.

4.4.3. Evaluation Strategy

In the later chapters, we have assessed the effect of the proposed reduced off-chip optical link latency on the overall system clock frequency. Increasing serialization factor in MGT multiplexing scheme also influences the speed of the system. In this research, we have mapped different benchmark circuits on the new latency-optimized models and conventional 2D MFS with two routing architectures and compared the system performance for a given range of serialization ratio.

In 3D MFS architectures, the inter-FPGA optical interface setup is same as that in 2D architecture and shown in Figure 4.7. However, the off-chip link length is reduced to half due to vertical stacking. Therefore, the average length of all optical fibers is set to be 3 inches. Referring to equation (4.3) & (4.4) again, we can calculate the propagation delay through a 3 inch plastic optical fiber and it turns out to be 0.37ns. Adding the optical transceiver delay, the total off-chip delay comes out to be 0.87ns for a 3D MFS architecture with optical interface. The proposed 3D architecture presents a 56.5% decrease in board latency per link as compared to conventional planar MFS. The proposed 3D MFS architectures are discussed in detail in the next chapter.

4.5. Previous Research on MFS Serial Optical Interface

Over the past few years, optical integration and interface in FPGAs have been explored and evaluated in industry and research. Benefits of optical communication have also been

studied in the integration of FPGA and non-FPGA devices. However, to our knowledge, the studies done in evaluation of the system performance of the MFS routing architectures with serial optical interconnections are very limited. Additionally, the comparison between MFS with more than 2 FPGAs with electrical and optical interface has not been considered either.

Edin [45] described the design and implementation of an optical fiber based high speed interface between two computers through Altera Stratix IV FPGAs achieving a bandwidth of 8.5 Gbps as shown in Figure 4.10. The design consisted of two computers connected to Altera's Development and Education board 4 (DE4) through a PCIeGen2 link with x4 lanes. The DE4 was equipped with a Stratix IV FPGA that temporarily stored the data being transmitted in its internal memory. A High Speed Mezzanine Connector (HSMC) connected the DE4 to a daughter card with 8 Small Form-factor Pluggable (SFP) slots. Four of them were used to connect four two-way optical fiber cables. For the optical fiber, the theoretical achievable data rate was 5Gbps per channel. Using 4 optical fiber cables brought the two links up to 20 Gbps.

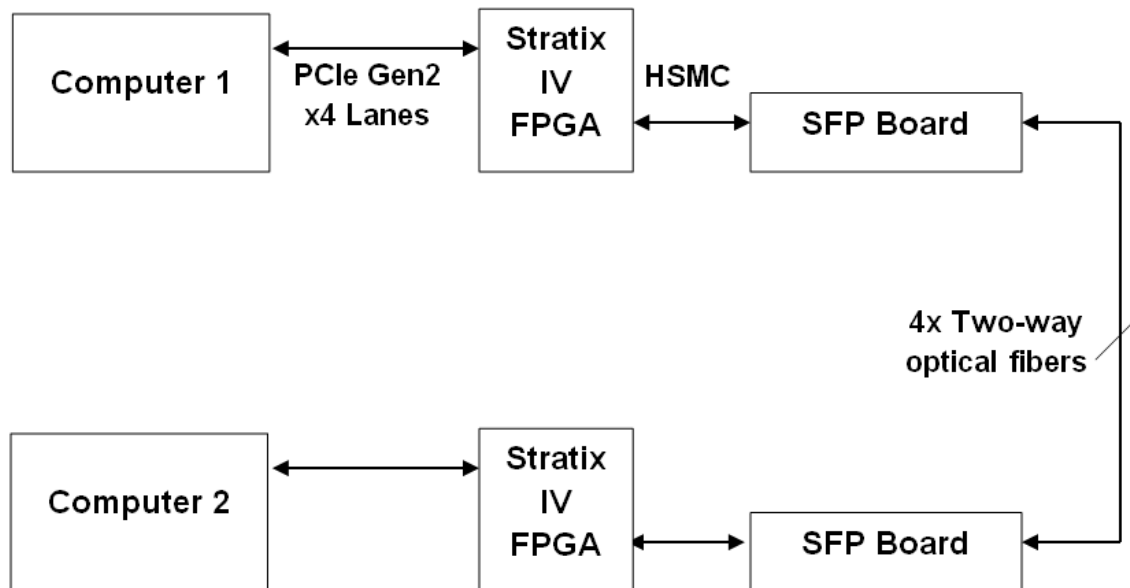


Figure 4.10: High level view of the hardware setup

The optical communication logic included 8b/10b encoding, internally controllable reset, channel bonding, channel alignment and start/end of transmission. The high-speed

interface presented in this work linked 2 computers together through 4 optical fiber channels at 25Gbps, and can provide better performance than Ethernet and InfiniBand counterparts. Figure 4.10 shows the block diagram of the configuration.

Kuzmin et al. [46] illustrated the implementation of the optical link test system demonstrating the feasibility and effectiveness of utilization of the on-chip diagnostic capabilities and soft-IP controller instantiated in FPGAs with high-speed serial transceivers. The system was based on the Altera Stratix IV GX FPGA installed on a TerasIC DE4 board. Through an adapter board with SMA connectors and a set of coaxial cables the DE4 board was connected to SFP+ evaluation boards hosting optical transceiver modules. Hot-pluggable SFP+ transceivers used in the system provided duplex LC type optical connectors for the Multi-Mode Fiber. The link data path consisted of a transmitter, an electro-optical converter (VCSEL with its driving circuits), an optical fiber, a photo detector (PIN diode and trans-impedance amplifier) and a receiver. The length of the fiber loop used in the tests ranged from 15 cm to 15 meters. The SFP+ module used in most of the experiments was the Avago AFBR-703SDDZ which was capable of data rates up to 10 Gbps. The developed hardware platform, IP blocks and embedded software formed a base for integration of twelve parallel optical links into a multi-FPGA reconfigurable computing system. It was used for the development of streaming video processing and HPC applications.

In [7], Altera discussed how optical interface technology embedded in an FPGA overcame the power, port density, cost, and circuit board complexity challenges in chip-to-chip, chip-to-module, rack-to-rack, and system-to-system interfaces providing considerable advantages over conventional electrical interconnections. Altera's transceiver technology provides electrical transmit and receive functionality with data rates up to 28 Gbps on the 28-nm process node. These transceivers also support advanced clock generation, clock recovery, and equalization capabilities. On the transmitter (TX) end, jitter generation is very low and reaches ~300 fs or lower at 28 Gbps because of the advanced LC oscillator. The receiver end compensates most of the uncorrelated jitter and noise and produces excellent locking time/range and resilience of excessive jitter on the incoming data. The transceiver also has built-in on-die instrumentation (ODI) to measure BER contour and

eye-diagram. These features simplify integrating an Altera FPGA with optics. As shown in Figure 4.2(c) FPGA is integrated with transmitter optical sub-assembly (TOSA) and receiver optical sub-assembly (ROSA), providing direct optical signal transmitting and receiving path eliminating the need for a discrete optical module. A detailed example for the use of an FPGA with optical interface in a data center (DC) is discussed presenting the intranet board-backplane-line card, board-to-board, rack-to-rack, and system-to-system interconnects using Altera's optical FPGA as LAN switch, router, SAN switch and disk array, and server array. The FPGA with optical interface enabled processing, as well as optical interconnects, for distances in the range of less than 0.3 m to greater than 100 m, and is well suited for the entire data center's interconnects. The result was significant power, density, and cost savings as compared to conventional technologies.

Similarly [43] presented the application of Altera's optical FPGA in blade server systems. For computer and storage-intensive applications, replacing pluggable optics by the new optical FPGA reduced power by 70 - 80% while increasing port density and bandwidth by orders of magnitude. Blade server systems were dense modular server systems that provided tight integration among multiple servers with storage, switching, I/Os, cooling, and power sub-systems. Most of these systems used a high-speed electrical interface between the server blades and the I/O modules through a complex electrical mid plane. This architecture presented complicated signal integrity and thermal challenges. Replacing electrical I/O channels from the blade server mezzanine by optical I/O channels, and replacing the complicated electrical mid-plane by simple optical equivalent provided high-speed connectivity between the servers and any other modules in the system, including storage, memory, and I/Os. Such replacement eliminated the complications of electrical signal integrity, EMI, crosstalk, and ESD immunity. Additionally, optical pass-through modules could interface to the mid plane to directly connect the server blades with external switches, storage, and memory.

In [47], Xilinx concentrated on the impact and compatibility of the optical interface on the error-free performance of their 7-Series FPGAs. The LC tank PLLs incorporated into the GTX, GTH, and GTZ 7 series transceivers provided a starting point for designers looking to interface to pluggable optical modules. Xilinx included programmable pre- and post-

emphasis circuits to overcome channel losses and maximize jitter performance after the data has been translated from electrical to the optical domain. The same LC tank PLLs is used to drive the CDR circuitry to deliver jitter tolerance at the receiving end. Xilinx 28 Gbps GTZ transceiver is designed to operate with the CFP2 optical modules. Furthermore, the GTX transceivers can operate at rates up to 12.5 Gbps and 13.1 Gbps in the GTH transceiver.

Ghiasi [48] presented three individual demonstrations to exhibit the applications of common electrical interfaces (CEI-28G-VSR) for two optical module form factors, namely CFP2 and QSFP28. The first one was Altera FPGA with 100 GbE MAC driving CEI-28G-VSR host card with Finisar CFP2 module plugged into it. At the receiver side Oclaro CFP2 module was plugged into CEI-28G-VSR host card with Inphi 100 GbE gearbox looping the traffic back as shown in Figure 4.11. Yamaichi CFP2 connectors were used at CFP2 module mating interface. The link operated at 25.78 GBd full duplex error-free over all 4 channels with PRBS31 as well as live 100 GbE Ethernet traffic. CFP2 100Gbase-LR4 modules support reach up to 10 km on duplex SMF. This showcased working demonstrations of the CEI-28G-VSR electrical interface implemented on a CFP2 100Gbase-LR4 optical transceiver and a QSFP28 AOC, using test scenarios similar to interoperability tests.

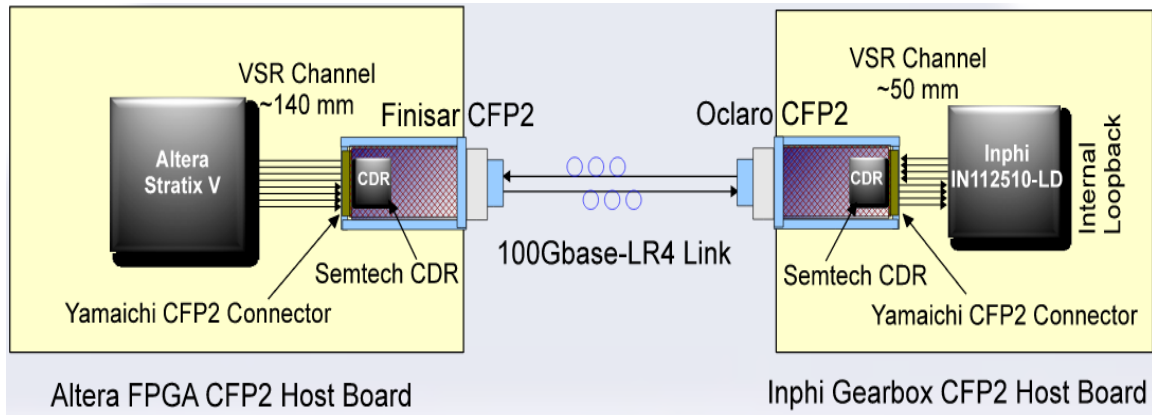


Figure 4.11: Block Diagram [48]

Deng, B. et al. [49] presented a remote FPGA-configuration method based on JTAG extension over 100 meter duplex multimode optical fibers. The remote configuration approach had three JTAG signals (TMS, TCK, and TDI) coming out of the Xilinx Kintex-

7 FPGA KC705 download cable, encoded and serialized into a high-speed serial data signal. The encoder and the serializer were integrated into a transmitter which had multiple parallel input signals and a high-speed serial data output signal. The data is converted to the optical signal in an SFP+ optical transceiver's transmitter module and transmitted across an optical fiber. Then the optical signal is converted back into the high-speed serial data electrical signal in an optical receiver module. The data is deserialized and decoded to recover the corresponding JTAG signals before they are connected to the FPGA.

As shown in Figure 4.12, the proposed remote configuration approach was implemented in the Liquid-Argon Trigger Digitization Board (LTDB) Demonstrator. Two transceivers TLK2501 were used on each LTDB Demonstrator, whereas the 8B/10B encoder/decoder and the transceiver were implemented in the Virtex6 FPGA of a Gigabit Link Interface Board (GLIB) on the other end. Two Xilinx Kintex-7 series FPGAs were configured through a single pair of optical fibers and a TLK2501 transceiver. A quad-channel optical transceiver module by Avago Technologies was used on the LTDB Demonstrator. On the other end, a Digilent USB JTAG cable was connected to the Virtex-6 FPGA through an FMC extension board. A 1:4 switch was implemented in the Virtex FPGA on the GLIB for selecting FPGA on the front end for configuration. Two SFP+ optical transceivers were used on the back end.

Minami et al. [50] developed a PCI Express (PCIe) card and front-end cards equipped with the small form-factor pluggable (SFP) transceivers for data transfer between FPGAs via optical fiber. The authors developed a PCIe to optical link interface, PEXOR (PCI-Express Optical Receiver) to connect front-end cards to standard PCs. The center of the PEXOR was a high performance SCM40 FPGA with 16 high-speed SERDES supporting data rates up to 3.8 Gbps and embedded ASIC blocks supporting PCIe, connecting two major components together; i.e. a four lane PCIe endpoint device and four high speed optical transceivers as inputs for front-end electronics. A master and slave protocol with chained connection of slave modules was designed and implemented in FPGAs. The protocol featured two data transfer modes: address mode for slow control and block mode for fast readout. The block mode data transfer per SFP port provided data rate up to 1.6 Gbps.

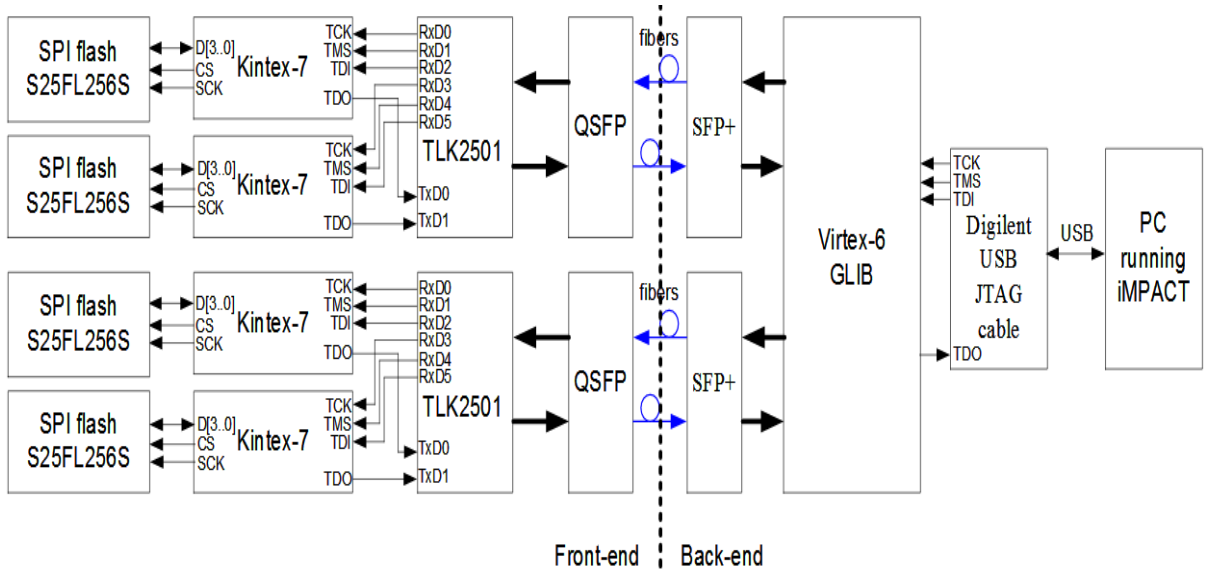


Figure 4.12: Block diagram of remote configuration on LTDB Demonstration [49]

4.6. Summary

This chapter presented the advantages and disadvantages of optical interface over its electrical counterpart. The basics and types of optical fibers and transceivers were also discussed comprehensively. In a multi-FPGA setup, incorporating an optical link has certain requirements that need to be satisfied for enhanced performance and it was described in Section 4.3. Section 4.4 presented the proposed latency optimized 2D MFS architecture with optical interface. Here we also presented and justified with real world examples the practicality and feasibility of short-length optical fibers for inter-FPGA connections in 2D and 3D MFS. Lastly, previous research done in the field of MFS and high-speed serial optical interface was presented. There is ample research available on optical integration and interface in FPGAs over the past few years. However, the evaluation of the system performance of the MFS routing architectures with serial optical interconnections has not been extensively investigated.

Chapter 5 Optical 3D MFS Routing Architectures

5.1. Introduction

While prototyping large SoC designs, the choice of prototyping machine poses significant challenges. The mapped design encounters board issues like design partitioning and placement and routing on the multi-FPGA platform. Such platform has to be faster and bigger than the design being tested and prototyped. 2D planar MFS with chips positioned next to each other, has been the most effective platform available so far for rapid prototyping and logic emulation. Although such MFSs are capable of accommodating large designs, their off-chip communication strategy imposes bandwidth constraints and limits the overall system performance. Besides introducing large delays, the routing resources consume significant board area as well. Scaling an MFS only aggravates the latency, area and cost issues.

3D MFS routing architectures become an appealing solution amid increasing cost of new technology nodes and keeping up with the Moore's law. These are flexible architectures that allow multiple FPGAs to be stacked "like Legos" to build a placement and latency optimized, almost arbitrarily large prototyping machine. Any number and type of FPGAs required to handle the implemented design can be plugged in on top of each other. The vertical orientation makes the off-chip connections much shorter, enabling the prototype to be scaled without considerable performance penalties and timing issues. 3D integration allows building topologically three-dimensional, densely interconnected architectures capable of supporting the interconnect density requirements of the application with much smaller footprint and overall reduced wire length.

Thanks in part to gigantic FPGAs available in the market today, the number of FPGAs required for the biggest prototypes has dropped significantly. Obviously, the size of the SoC designs has also increased considerably at the same time, that's why the industry is also focusing on increasing the size of every upcoming FPGA generation. With huge FPGA

devices, the designers enjoy not only the capacity and performance boost, but, combining that with the faster connections supplied by 3D orientation, the result is a much faster prototyping machine for performance-critical applications [4] [52].

5.2. Why 3D MFS Architecture?

5.2.1. Interconnection Length Distribution

Massive designs mapped onto multi-FPGA platforms require extensive communication among their different partitions with the aid of vastly complex interconnection structure. Large designs that do not fit into a single chip must be partitioned into multiple FPGAs. An empirical quantitative description of such partitioning is given by Rent's rule. It states that the relationship between the size of a sub-circuit B and the number of inter-partition connections P is given by:

$$P \sim B^r \quad (5.1)$$

Where r is the Rent's exponent ranging from 0.4 for designs with a simple interconnection structure, up to 0.75 for designs with a complex interconnection structure. Designs with a higher Rent's exponent have a proportionally larger number of long interconnections, and consequently a larger average interconnection length in two-dimensional MFS architecture because of larger spatial distance between neighboring FPGAs [53].

Figure 5.1 shows the influence of the dimension on the interconnection length distribution of an implemented design with 256K-node and $r = 0.6$ in both 2D and 3D architectures. The steeper slope in 3D architecture implies:

- There are fewer long interconnections.
- The average interconnection length is smaller because the distance between the neighboring FPGAs is reduced [57].

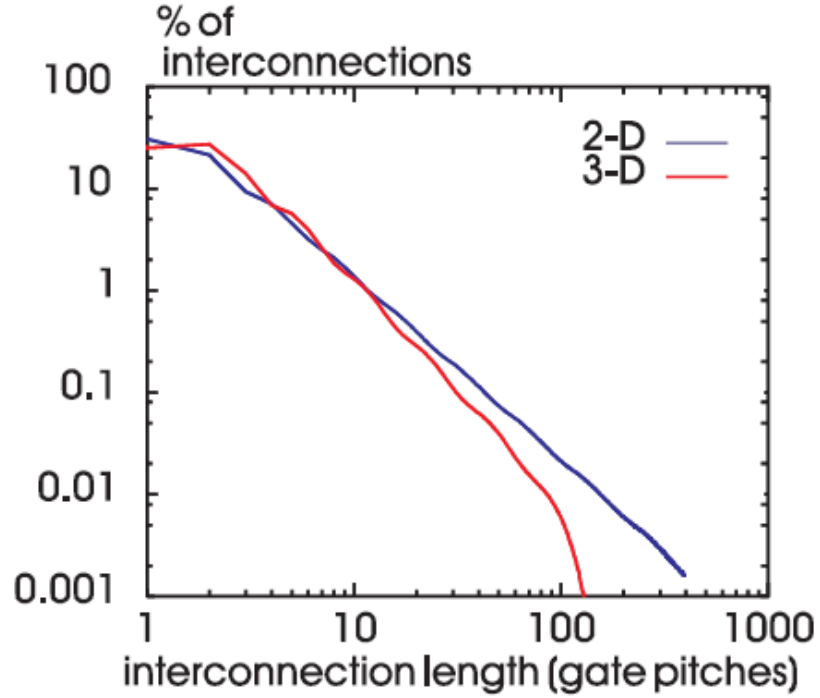


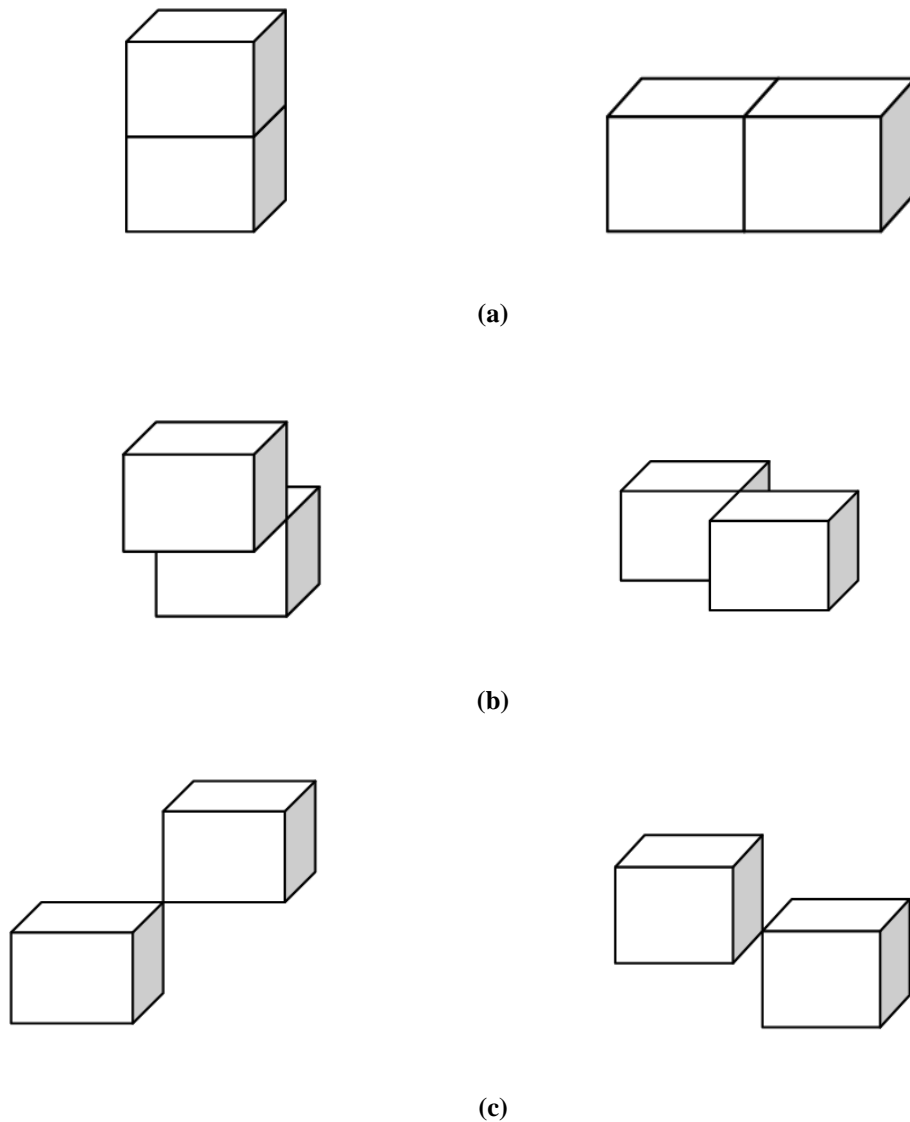
Figure 5.1: Comparison of interconnection length distribution for 2D & 3D architectures [56]

According to [63], in a 2D planar MFS platform, the nominal PCB trace length for point-to-point data signal transmission in DDR mode is 6-7 inches, resulting in a propagation delay of hundreds of picoseconds. With the increase in MFS size, the overall off-chip interconnects' length and board latency also increases, resulting in inferior system performance. In 3D topology, the FPGAs are vertically stacked on top of each other, instead of being spread out along x and y-axes. This allows reduced distance between the FPGAs resulting in shorter average wire-lengths and hence reduced inter-FPGA propagation delays.

5.2.2. Asymptotic Behavior of Wire-Length

The average interconnection length or wire-length shows asymptotic behavior as the size of a design increases [55]. In designs where Rent's exponent r is less than a threshold value r_t , the average interconnection length converges to a constant value. However, in designs where $r > r_t$, the average interconnection length runs off to infinity, at a rate proportional to the design size raised to the power $(r - r_t)$. The threshold value r_t can therefore be

interpreted as the capacity of an architecture to fit a certain design with a Rent's exponent r . The larger the difference $(r - r_t)$, the harder it is for the architecture to fit the design. For 2D planar architectures, this value is $r_t = 0.5$. Whereas, 3-D architectures can contain designs with Rent's exponents up to $r_t = 0.67$. Even for designs with $r_t > 0.67$, average wire-length increases with the design size at a much slower rate in 3D architectures as compared to 2D structures.



**Figure 5.2: Possible Combination Classes in 3D (a) A-combination (b) N-combination
(c) R-combination**

5.2.3. Structural Distribution & Placement Optimization

Increase in design complexity triggers MFS spatial scalability issues. Although one can continue to stitch together multiple FPGAs to expand, however, the resulting footprint area of the platform continues to grow as well in planar architectures.

In the third dimension, nodes can be positioned in three different combination classes as shown in Figure 5.2: *adjacent combination* (A-combination), *diagonally opposed combination located at a near diagonal* (N-combination), and *diagonally opposed combination located at a remote diagonal* (R-combination) [55]. It was shown that the spatial distribution of nodes in 3D architectures results in placement optimization favoring smaller distances between the nodes. The outcome was an obvious advantage of overall smaller footprint area, with a cleaner setup and manageable on-board interconnections. Furthermore, vertical structural distribution exhibits shorter average wire-length as discussed earlier resulting in speed advantages, making 3D architectures highly suitable for the implementation of complex designs with higher Rent's exponent [54] [55] [56].

5.3. Proposed 3D MFS Architectures

5.3.1. Motivation

Multi-FPGA systems are attaining increasingly critical role in many industries like aerospace defense, automotive industry, high performance computing, communication and medicine. Most applications are latency sensitive and require highly accurate, complex, sophisticated and extremely fast processing speeds for computationally intensive tasks. That's why researchers continuously aim at achieving superior performance in multi-FPGA systems with additional advantage of smaller footprint area.

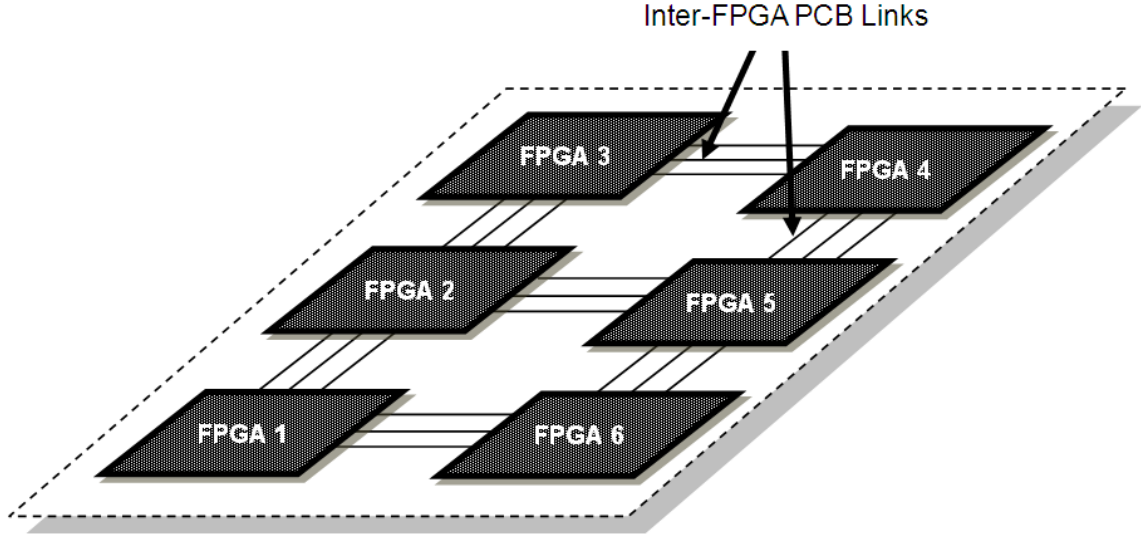


Figure 5.3: 2D MFS Architecture

Conventional 2D planar MFS with electrical interconnections as shown in Figure 5.3 has broader spatial distribution resulting in larger footprint area. Additionally, 2D platforms have dominant off-chip delays as compared to on-chip latencies that dictate the overall system frequency. Off-chip copper interconnections in 2D MFS are typically 6 to 7 inches [63], which results in a delay of 2ns [5] [61].

For this reason, the primary motivation behind this research is to propose latency-optimized MFS with smaller footprint area.

5.3.2. Proposed 3D Optical MFS Routing Architectures

We have proposed three-dimensional, vertically stacked densely interconnected architectures presenting several advantages including significantly shorter trace lengths, cleaner setup, inter-layer equal-length connectivity and smaller footprint.

Additionally, we have proposed using short-range optical interconnects instead of copper connections for improving the system performance significantly. The superiority of physical properties of photon propagation over electron propagation translates to fundamentally improved bandwidth, distance, cross-section and latency of optical interconnects as compared to its electrical counterparts

As discussed in Chapter 4, the propagation delay through an optical link depends on the length of the link, and 3D topology allows interconnect length per link to be reduced by nearly one half. Therefore, the nominal length of a POF interconnection is set to 3 inches in the proposed 3D MFS which leads to the propagation delay of 0.37ns (Refer to equation 4.3 & 4.4, Chapter 4).

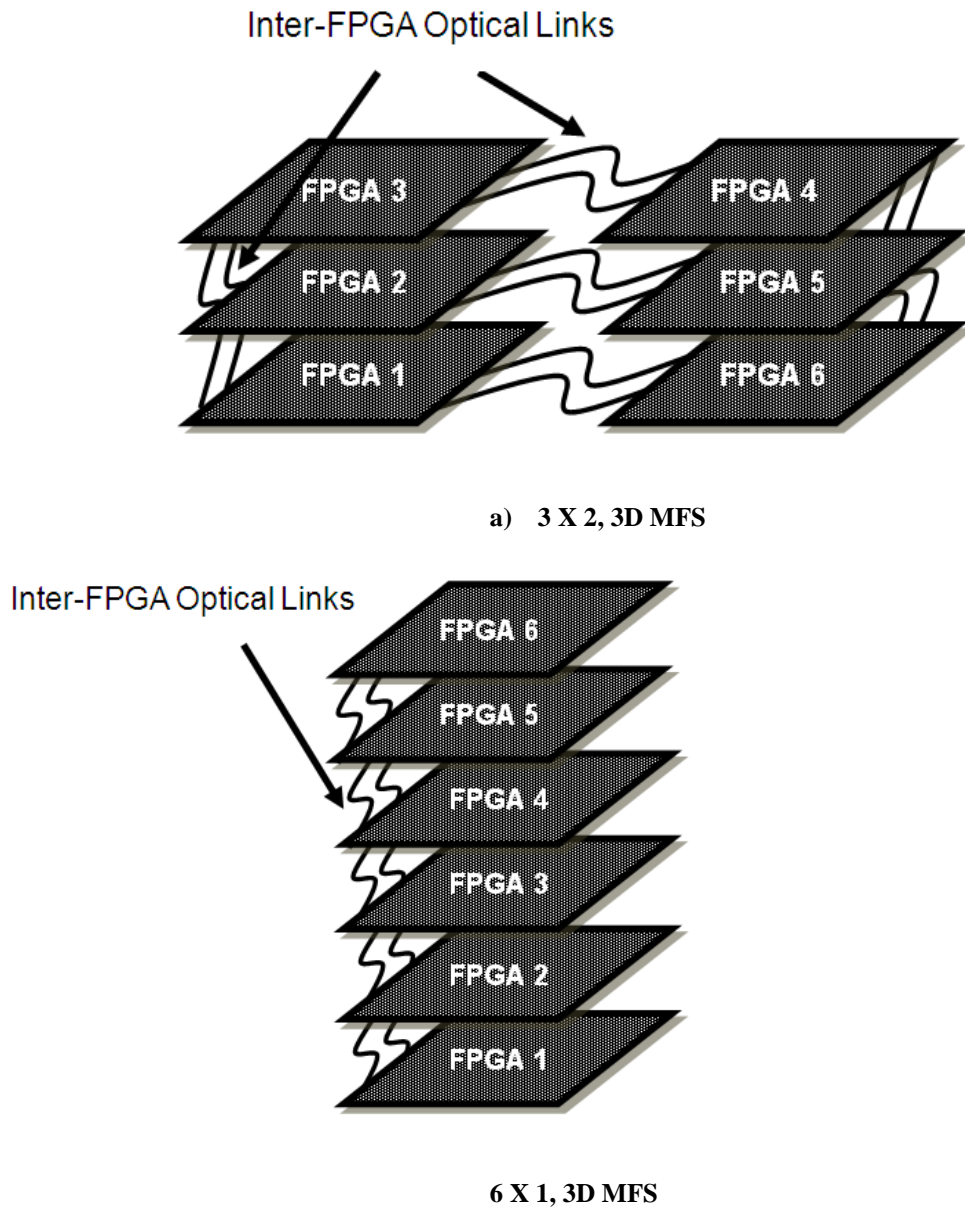


Figure 5.4: 3D MFS topologies with various degrees of optical interconnect

Finisar's (FTLX8574D3BCV) SFP+ optical transceiver has a delay of 500ps. This implies that total off-chip delay is 0.87ns in the proposed 3D MFS architecture with optical interface which is a 56.5% decrease per link as compared to conventional planar MFS.

As shown in Figure 5.4, we have proposed two 3D architectural models i.e. 3X2 and 6X1. Both models are built with optical interconnects routed through optical transceiver connected with every FPGA (not shown in Figure 5.4). In our architectural models we have considered the FPGA size to be a fixed parameter, and thus we increase or decrease the number of FPGAs according to the design requirements. In these architectures FPGAs are arranged in multiple *planes*, and each plane contains equal number of chips. 3X2 platform consists of 3 planes having 2 FPGAs per plane. Whereas, 6X1 topology consists of 6 planes having 1 FPGA per plane. 3 X 2 platform is built in two versions; 3X2 CCG and 3X2 TORUS. The interconnection distribution in 3D CCG and TORUS routing architectures is same as that in 2D architectures except that here all the intra-plane and inter-plane interconnections are short-length optical fibers instead of PCB. In CCG, every FPGA on every plane is directly connected to every other FPGA via horizontal, vertical and diagonal optical link. In TORUS architecture each FPGA is connected only to its horizontal and vertically adjacent neighbors. Moreover, the peripheral FPGAs are wrapped around in horizontal and vertical directions and are connected to the FPGAs on the opposite side of the plane through optical links.

6X1 3D architecture is built by stacking 6 FPGAs on top of each other with vertical interconnects among all the planes.

The proposed architectures are scalable and the number of FPGAs per plane can be increased as per the design requirements. We have chosen 6 FPGAs in each model, because they are sufficient to accommodate the benchmark circuits. Also, all planar MFS that we evaluated have 6 FPGAs.

5.3.3. Multiplexing in 3D Routing Architectures

Although 3D architectures are better capable of accommodating designs with higher connectivity count, there is still a fair chance that they can suffer from a lack of interconnect

capacity due to limited number of input/output (I/O) pins per chip. Pin limitation issue can arise in 3D MFS architectures as well, since the basic building block of the 2D and 3D topology is the same. Limited pin count problem can be very effectively addressed using pin multiplexing. Employing high speed serial interface not only addresses the off-chip communication bottleneck, but also reduces routing congestion not only in planar MFS but also in 3D MFS architectures.

As discussed earlier in the thesis, only the multi-gigabit transceivers in KU3P FPGA support optical interface. That's why we have employed MGT multiplexing scheme in the proposed 3D architectures to optimize the system performance.

5.3.4. Evaluation Strategy

These 3D MFS platforms with high-speed optical serial interface are scalable and highly flexible providing a wide variety of inter-FPGA routing choices. The number of FPGAs per plane and the number of planes can be increased according to the design requirements and numerous architectural variants of the proposed 3D MFS can be built. However, it would be unrealistic to assume that any number of chips can be stacked on top of each other connected via low-latency optical interface in 3D MFS and every time better performance is achieved as compared to its 2D counterpart.

In this research, we have investigated the influence of increased number of planes and evaluated the system clock frequency of the proposed architectures and compared it with that of 2D architectures. The proposed models are validated with experiments on real sequential benchmark circuits.

Besides the selection of routing architecture and topology, the multiplexing ratio also exercised considerable effect on the system clock frequency. That is why in our work, the system performance of 3D architectures has been evaluated experimentally by increasing the serialization factor.

5.4. Previous Research on 3D MFS

The idea of investigating the third dimension on-chip is not new. Several studies (discussed below) have presented latency improvement due to reduction in the communication channel lengths in 3D chips and have proved their superior performance efficiency as compared to their 2D counterparts.

In [64], various topologies for 3D NoC were presented. The authors also described analytic models for the zero-load latency and the power consumption with delay constraints of these networks that captured the effects of the topology on the performance of 3D NoC. It was proved that optimum topologies exist that minimized the zero-load latency and power consumption of a network and they depended upon numerous parameters characterizing both the router and the communication channel, like the number of ports of the router, the length of the communication channel, and the impedance characteristics of the interconnect. The authors presented a 3D topology where the interconnect network was contained within one physical plane, whereas each PE was integrated in multiple planes. A hybrid 3D NoC was also presented where both the interconnect network and the PEs spanned more than one physical plane of the stack.

Lin et al. [65] presented 3D FPGA consisting of multiple active layers, each performing a different FPGA function. The proposed 3D FPGA required monolithic stacking, which enabled much higher vertical interconnect density than chip/wafer stacking. The study quantified the potential improvements in logic density, delay and power of monolithically stacked 3D FPGA over 2D FPGA. The architectural baseline was Virtex-II-style 2D FPGA. It was assumed that only the switch transistors and configuration memory cells were moved to the top layers. A technology-independent FPGA area model was developed and used to compare the logic density of a stacked FPGA to the 2D FPGA as a function of configuration memory element size. RC circuit models for interconnect segments were also developed and used to calculate the improvements in interconnect delay in the 3D FPGA relative to 2D FPGA. The interconnect delay results were then used to estimate the relative improvements in the geometric average net delays and critical path delays achieved by 3D FPGA for 20 MCNC benchmark circuits that were placed and routed using VPR.

In 2011, Xilinx announced its first heterogeneous 3D Virtex-7 2000T FPGA containing 6.8 billion transistors, providing designers with access to 2 million logic cells. The capacity was made possible by Xilinx's Stacked Silicon Interconnect (SSI) technology which involved a special layer of silicon called a "silicon interposer" combined with through-silicon vias (TSVs) [66].

In [67] a 3D physical design and validation methodology for Tree-based FPGA architecture was studied and implemented. Horizontal and vertical design partitioning methods were also presented to support 3D design and implementation. A CAD tool set for 3D physical design and verification based on Global Foundries 130 nm technology node modified to use Tezzaron's TSV technology was developed.

An extensive amount of research has been carried out in the field of 2D MFS architectures and time multiplexing. Nonetheless, to our knowledge, the studies done in exploring and evaluating the third dimension for MFS architectures are extremely limited. Furthermore, so far no work has been done in evaluation of the system performance of the 3D MFS routing architectures with serial optical interconnections.

Strauch [4] proposed a three-dimensional concentric multi-FPGA architecture resulting in equal length concept between FPGA pins enabling wave-pipelined pin-multiplexing. The proposed structure placed four FPGAs on top of each other and introduced a multiplexed horizontal and vertical routing concept. The author claimed that the advantage of this routing concept was that any possible signal connectivity could be routed on the proposed 3D structure. Also, the proposed vertical system prototyping resulted in very low physical distance between FPGAs and switches. The small and equal delay values between FPGA pairs enabled high speed data transfer, because it allowed wave-pipeline based pin multiplexing. The author mapped 1000 randomly generated design scenarios automatically on the structure and compared the achievable system prototyping speed with a group of alternative concepts.

Dambre [57] addressed the performance advantage of using optoelectronic area-I/O to realize 3D MFS architectures. The authors used minimum achievable system clock period as an evaluation metric by implementing limited set of synchronous designs. The clock

period was determined by the largest combinational path delay between memory cells. The benchmark circuits were partitioned and then placed in FPGA chips with an array of 20 X 20 CLBs. Every benchmark circuit was implemented in 1, 2 and 4-plane proposed architectures with different optical link latency values. The resulting clock periods were then compared to the clock period of an implementation in a purely electrical single-plane architecture. The authors indicated that three-dimensional optoelectronic multi-FPGA architectures exhibited higher performance than traditional two-dimensional electronic FPGAs, provided the optical link latency is sufficiently low. Performance gain also strongly depended on the number of optically interconnected planes.

Li [7] compared the differences in building a 32 FPGA system using a 2D approach versus a 3D build using a chassis, and demonstrated that the latter required fewer cables. A 4 by 8 MESH system was mapped to 32 FPGAs on 8 Quad FPGA modules. FPGAs were grouped in a specific 3D pattern in order to minimize the cable connections, cable lengths, and to keep the cables from crossing from one side to the other.

5.5. Summary

MFS can experience large off-chip delays owing to long wire-length and inferior performance of electrical interconnections. In this chapter we have proposed three different 3D optical MFS routing architectures built by stacking multiple FPGAs on top of each other. The advantages of exploring third dimension in MFS were described in detail. Additionally, we have replaced electrical interconnections by optical interface to further reduce the off-chip delays. Since, GTY transceivers support optical interface, that's why MGT multiplexing is employed to address limited pin count issue in the proposed architectures.

Chapter 6 CAD Tools and Experimental Evaluation Framework

In order to evaluate and compare the performance of 3D and 2D time multiplexed MFS routing architectures, an experimental platform was developed that facilitated optimized mapping of real sequential benchmark circuits to MFS routing architecture. This chapter provides an outline of the experimental procedure used for mapping a benchmark circuit to the target architectures, layout synthesis and timing analysis tools developed, benchmark circuits and evaluation metric employed.

6.1. Experimental Design Mapping Flow

The experimental procedure used for mapping a circuit to a given architecture is illustrated in Figure 6.1. We start with a gate-level netlist of a circuit. The first step is to perform technology-mapping to obtain the LUT-level netlist that can be mapped directly onto an FPGA. In the next step, the LUT-level netlist is translated into a graph format, which is then fed to the partitioning tool. The circuit is partitioned into sub-circuits using KaHIP partitioning tool which is discussed in detail in a later section. The partitioning for both 2D and 3D platforms can be carried out in similar manner using KaHIP specifying the number of FPGAs per platform.

At the end of 6-way partitioning, the nodes are assigned to 6 different partitions, such that the number of edges between partitions is minimized.

The next step is the placement of each sub-circuit on the given FPGA in the MFS. Given the sub-circuits and the inter-FPGA netlist, each sub-circuit is placed in a specific FPGA in the MFS. The goal is to position highly connected sub-circuits into adjacent FPGAs so that the routing resources required for inter-FPGA connections can be minimized.

For the given MFS architecture, placement and inter-FPGA netlist, the next step is to route all inter-FPGA nets using the most suitable routing path.

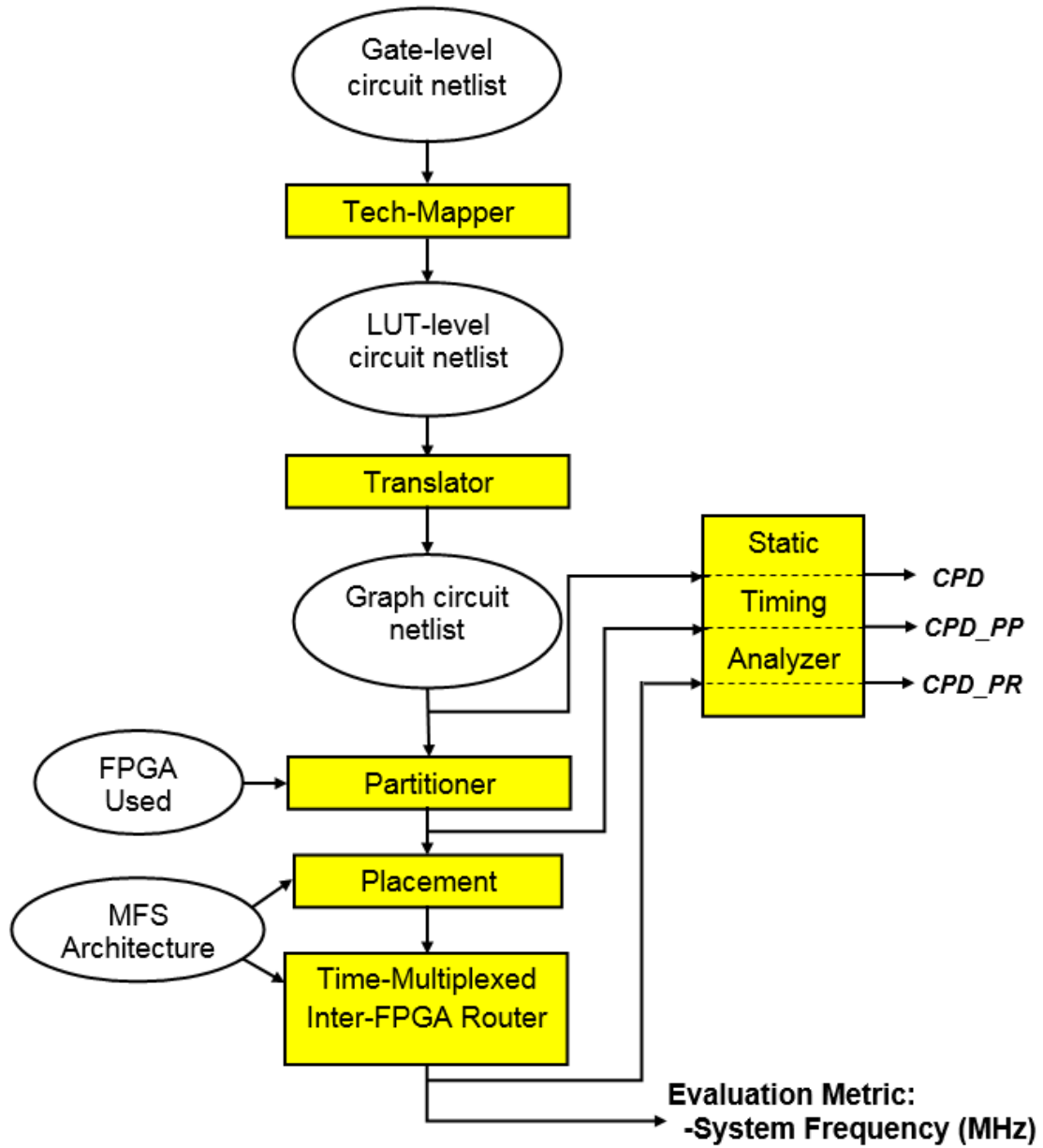


Figure 6.1: Design Flow for Time-Multiplexed MFS

In the context of MFSs, this implies that the routing path selected should utilize minimum routing resources and thus minimize the routing delay for the inter-FPGA nets. However, the available pins per FPGA for data transfer are always less than the total inter-FPGA nets. Therefore, in order to ensure that the routing succeeds, appropriate multiplexing scheme is incorporated in the given architecture. Lastly, the post-routing critical path delay and system frequency for a range of serialization ratios is calculated.

A static timing analyzer (STA) is developed to calculate the critical path delay (CPD) for a given circuit. The STA developed is described in Section 6.3.5. The pre-partitioned LUT-level netlist is fed to the STA to obtain pre-partitioning CPD. After partitioning, the circuit netlist with the updated net delays is fed to the STA to obtain post-partitioning CPD_PP. And finally, the timing analysis is performed again on the same circuit netlist after 100% routing is accomplished with the updated routing delays for each net.

As discussed earlier, the FPGA used in this research is the Xilinx Kintex Ultrascale+ FPGA KU3P [30], which consists of 163,000 6-LUTs and 325,000 flip-flops. The chosen FPGA offers 16 GTY transceivers with 32.75 Gbps inter-FPGA communication data rate. The CAD tools were developed in C++ and executed on Centos 6.9 Linux environment on a 16 core Intel(R) Xeon CPU E5-2620 v4 with 125.8GB of memory, clocked at 2.10 GHz.

6.2. Assumptions

In a real MFS, once the inter-FPGA nets are routed, the next step is the pin assignment, placement and routing within individual FPGAs. By doing so, accurate routing delays within each FPGA can be obtained; however this requires an excessive amount of time and effort. Therefore, a more practical alternative is to perform static timing analysis after inter-FPGA routing, assuming that all the routing delays within an FPGA are constant. This assumption provides reasonably accurate estimate of the MFS speed because in CPD_PR, off-chip delays determine the critical path. It is assumed that after inter-FPGA routing, the intra-FPGA pin assignment, placement, and routing step will succeed for the reasons outlined below.

6.2.1. FPGA Pin Assignment

The pin assignment step selects specific wires and pins for every connection given by the inter-FPGA router. If the FPGA pin assignment is carried out randomly, it might lock pins in places making intra-FPGA placement and routing more complex. This can lead to routability and speed issues for the FPGA. Khalid and Rose [69] carried out an experimental study to explore the effects of pin locking on the routability and speed of FPGAs. They placed and routed sixteen benchmark circuits on Xilinx (XC4000) and Altera (FLEX 8000) FPGAs with and without a variety of pin constraints. The experimental

results showed that there was an average increase of 5% in the critical path delay due to random pin assignment for Xilinx FPGAs and only 3.6% for the Altera FPGAs(as compared to the no pin constraints case). No routing failures were recorded for the XC4000FPGAs; however there were three routing failures (out of 14 circuits) for the FLEX 8000FPGAs. Since we use one of the largest available FPGAs in our experiments, the results support our assumption that pin locking will not significantly impact placement and routing results for each FPGA in the MFS.

6.2.2. Intra-FPGA Placement and Routing

Once the inter-FPGA routing and FPGA pin assignment is done, then it is safe to assume that each sub-circuit can be successfully placed and routed within an FPGA. This assumption is based on previous work [69] [70] which showed that the placement and routing of a circuit within an FPGA generally succeeds provided the FPGA logic utilization is restricted to less than 70%. For that reason, during multi-FPGA partitioning in this study, the size of each sub-circuit is restricted to at most 70% of the FPGA logic capacity. This ensures that the placement and routing of each sub-circuit within an FPGA does not fail.

6.3. CAD Tools

This section describes in detail the CAD tools developed for mapping benchmark circuits to 2D and 3D time-multiplexed MFS routing architectures. The key goals in creating these CAD tools were:

1. To create a tool set that was flexible enough to map benchmark circuits and employ multiplexing for the given MFS routing architectures.
2. To employ the most suitable algorithm for each task to obtain results that were comparable with the results reported elsewhere or at least not significantly worse.
3. To focus on the architectural exploration of 2D and 3D MFS routing architectures. Excessive amount of time and effort was not spent on CAD tool development.

6.3.1. ABC Tech-Mapper

Technology mapping (tech-mapping) transforms a technology-independent logic network into a functionally equivalent netlist of primitive elements available for implementation on

a target device. For an FPGA, the logic primitive is a look-up table (LUT). Each LUT is restricted to a maximum of K inputs (a K -LUT), where K is defined by the FPGA's architecture. The size of a LUT is the number of inputs actually used. First of all, the tech-mapper read *.blif* gate-level input file. The input netlist is a structure of nets, logic nodes, latches, and PI/PO (primary input / primary output) terminals. Nodes can have any number of fanouts and fanins. An edge is a connection between any two objects like between two nodes, or between a node and a latch. The tech-mapping is accomplished by using ABC command *if -K 6* [71] which performed FPGA mapping into 6-LUTs using exhaustive cut enumeration. This algorithm is based on the traditional area flow and exact area recovery. Lastly, the LUT-based output netlist is written in *.bench* format.

6.3.2. Translator

A translator was developed to convert the *.bench* output of the ABC tech-mapping step into graph format. It read the LUT-level netlist and translated it into a graph netlist. A graph $G = (V, E)$ with n vertices (nodes) and m edges is specified using $(n + 1)$ lines in a text file. The first line contains two integers representing number of vertices n and the number of undirected edges m in the graph G . After the first line, the remaining n lines contain vertices information. For example, the i th line has information of i th vertex. i th line lists the vertices connected to the i th vertex. It is important to note here, that we have considered an un-weighted graph i.e. the nodes/ vertices and the edges are considered not to have any weight associated to them.

6.3.3. Multi-way Partitioning

The graph G is then partitioned using KaHIP [72]. KaHIP - Karlsruhe High Quality Partitioning - is a family of graph partitioning programs including KaFFPa (Karlsruhe Fast Flow Partitioner), KaFFPaE (KaFFPaEvolutionary) which is a parallel evolutionary algorithm that uses KaFFPa to provide combine and mutation operations, and KaBaPE which extends the evolutionary algorithm.

For this research, KaFFPaE partitioning is carried out with the following parameters. We set *--k* to 6 indicating the number of partitions. *--preconfiguration* is set to 'strong' to ensure paramount partitioning quality. *--imbalance* is set to 1% to ensure high quality of balance

in nodes between partitions. *--balance_edges* is also enabled to balance the edges among the partitions as well as the nodes.

The output partitioned file of the graph with n vertices, consists of n lines with a single number per line. The i^{th} line of the file contains the partition number that the i^{th} vertex belongs to. Partition numbers always start from 0.

Resulting partitions are considered acceptable only if the partition size is at least 30% less than the available number of 6-LUTs in Xilinx Kintex UltraScale+ KU3P FPGA. The partitioning for both 2D and 3D platforms can be carried out in similar manner using KaFFPaE specifying the number of FPGAs per platform.

6.3.4. Placement

Current placement algorithms can be identified based on the algorithm employed. Based on this identification, there are four types of placement algorithms: partition-based placement or min-cut placement, force-directed algorithm, quadratic placement algorithm and the simulated annealing [73].

Two objectives i.e. net-cut and wirelength are normally used when solving a placement problem. Optimizing net-cut means reducing the number of inter-partition connections, whereas, wirelength optimization aims to decrease the global interconnection length that improves the routability [74].

Partition based algorithms (e.g. KaHIP) minimize the number of interconnects being cut at the partitioning stage and serve as efficient min-cut placement tools. They are generally fast and create placement with reasonable quality. However, they don't address the wirelength optimization objective.

Net-cut for a given placement can be formulated as:

$$C_{ut} = |N| . (\alpha_1 + \alpha_2 + \alpha_3 + \dots)$$

Where, $|N|$ represent the total number of nets in the circuit and α is defined as the percentage of nets which has a normalized wirelength. Therefore, the percentage of un-cut nets will be α_0 and the percentage of nets having a normalized wirelength of 1 is α_1 and so on.

For a given placement, the normalized wirelength is:

$$WL_n = |N|. (\alpha_1 + 2\alpha_2 + 3\alpha_3 + \dots)$$

Net-cut optimized placement α 's are symbolized by α_{ci} and wirelength optimized placement α 's are symbolized by α_{wi} . The optimal normalized wirelength WL_n obeys the following relation:

$$|N|. \sum_{i>1} \alpha_{ci} \leq WL_n \leq |N|. \sum_{i>1} i. \alpha_{ci}$$

This equation shows that the optimal wirelength is bounded by the α 's from the optimal net-cut placement [74].

In case of 2D completely connected graph, any arbitrary placement is acceptable because in this architecture every pair of FPGAs is uniformly connected providing direct connections. However, for 2D TORUS mesh architectures, a placement algorithm is required to position highly connected sub-circuits into adjacent FPGAs, to reduce the inter-FPGA routing resources needed.

As discussed earlier, any partitioning tool that optimizes net cut can be used as a placement tool in MFS. Therefore, KaHIP partitioner also serves as a tool to get net-cut optimized placement and satisfies the requirements for this work.

However, a separate 3D placement tool is developed for 6X1 3D topology; because it's very limited routing resources introduce large number of route-throughs. This placement tool aims at minimizing the routing cost for the inter-FPGA nets and post-routing critical path delay by assigning highly-connected sub-designs to adjacent FPGAs. This placement tool is an extended version of the force-directed placement algorithm for 2D MFS [1]. The process starts by randomly placing the sub-circuits on an MFS and then iteratively changing their placement until minimum possible routing cost for each inter-FPGA net is achieved. Better and faster results can be achieved by placing highly-connected sub-designs in the adjacent FPGAs during the initial random placement phase.

6.3.5. MFS Timing Analyzer

In synchronous digital circuits, the maximum achievable speed is governed by the slowest combinational path in the circuit implementation, which is called the critical path. Timing analysis can be employed in timing-driven layout tools, to calculate the slack of each connection in the circuit. The slack of a connection is defined as the delay that can be added to the connection without increasing the critical path delay. Connections with low slack values are routed using fast paths to avoid slowing down the circuit [75]. For a given benchmark circuit, the critical path delay determines the speed of an MFS after a circuit has been placed and routed at the inter-FPGA level.

MFS static timing analyzer based on block-oriented technique was developed to calculate the post-routing critical path delay for any given circuit and MFS routing architecture whether 2D or 3D. The need to develop a custom static timing analyzer for multi-FPGA platform arose because existing commercial timing analysis tools target single-FPGA architectures; for instance Vivado by Xilinx [76], TimeQuest by Altera [77], Synopsys' PrimeTime [78]. Wasga [79] by Flexras Technologies targeted multi-FPGA platform for partitioning, routing and timing analysis, however, it is no longer commercially available and also it did not meet our requirements for static timing analysis of 3D MFS routing architectures. It is not possible to execute multi-chip timing analysis and obtain post-routing path completely or automatically with the existing tools since they are designed for single FPGA alone and cannot handle off-chip optical interface.

Static Timing Analysis Technique

The algorithm starts from the sources (primary inputs and flip-flop outputs) and records the delay of each vertex. All the primary inputs have only the routing delay and not the logic delay. However, the rest of the vertices of the circuit have both the routing and the logic delay. Since the primary inputs do not have logic delay that is why the arrival time $T_{arrival}$ of all the sources is the same as their respective delays. In the next step, the algorithm calculates the arrival time $T_{arrival}$ for all the nets of the circuit using breadth-first search. The arrival time is given the following equation:

$$T_{arrival}(i) = D(i) + \text{Max}_{j \in fanin(i)} \{T_{arrival}(j, i)\}$$

Where, $D(i)$ is the routing delay of the said net.

In the next step, the algorithm sets the required time (T_{reqd}) at all sinks (primary outputs and flip-flop inputs) to be $T_{arrival_max}$. Required arrival time is then propagated backwards starting from the sinks with the following equation:

$$T_{reqd}(i) = \text{Min}_{j \in \text{fanout}(i)} \{T_{reqd}(j) - D(i, j)\}$$

And finally, the *slack* of each net is calculated by the following equation:

$$\text{Slack} = T_{reqd} - T_{arrival}$$

The algorithm continues until every single net in the graph has been labeled with its *slack*. Connections with zero slack are on the *critical path*. The different delay values used by the analyzer are given in Table 6.1. These values are obtained from the Xilinx Kintex UltraScale+ data sheet [80].

Table 6.1: Delay Values Used in Static Timing Analyzer

Item	Delay (nsec)
Intra-FPGA CLB-to-CLB routing delay	0.625
FPGA input pad delay	0.34
FPGA output pad delay	0.41
FPGA Route through delay	10

The propagation delay through a LUT is independent of the function implemented and is set to be 0.05ns. CLB-to-CLB delay is approximated as a constant, because individual FPGA placement and routing is not performed in this research. The value of 0.625 for CLB-to-CLB routing delay is $1/16^{\text{th}}$ of the delay on a long line for KU3P FPGA, which is a pessimistic estimate. Since the post-routing critical path delay of an MFS is dominated by the off-chip delay values, therefore the internal delay values can be safely approximated to a reasonable constant value.

The static timing analyzer first calculates the critical path delay of the un-partitioned LUT-level netlist. In this step it is assumed that the complete design is mapped on a hypothetical single large FPGA. The logic block and the interconnect delays (shown in Table 6.1) are assumed to be the same as the FPGA used in the MFS. The critical path delay of the un-partitioned design is denoted by CPD.

In the next step, the analyzer calculates the post-partition critical path delay (CPD_PP). This is the critical path delay obtained by analyzing the circuit netlist after it has been partitioned into 6 FPGAs. A fair assumption made here is that the FPGAs are interconnected on a custom PCB and the circuit is annotated with the inter-FPGA delays from which CPD_PP is calculated. The inter-FPGA delay for connecting a CLB in one FPGA to a CLB in another FPGA is the sum of the following three delay values (given in Table 6.1): CLB-to-output pad routing delay, PCB or optical trace delay and input pad-to-CLB routing delay. The signaling standards employed are single-ended HSTL_I_12 and differential DIFF_SSTL15 standard [80]. Typical value of PCB board delay in an MFS is 2ns [5] [61]. In case of optical link, the board delay is the sum of propagation delay of optical interface and the delay through optical transceiver. As discussed in Chapter 4, the propagation delay through an optical link depends on the length of the link. For a 6 inch long optical link, the delay is 0.75ns and for a 3 inch long optical interface, the propagation delay turns out to be 0.37ns (Refer to equation 3 & 4, Chapter 4). *Finisar's* (FTLX8574D3BCV) SFP+ short-range 10.3Gbps optical transceiver produces a delay of 500ps [42].

CPD_PP provided a lower bound on the post-routing critical path delay (CPD_PR) that is calculated for general purpose MFS. Since board-level programmable routing for any circuit in general purpose MFSs introduces a significant delay, that's why CPD_PR can be no better than CPD_PP.

Lastly, the static timing analyzer read the given 2D or 3D MFS architecture and the routing path for every inter-FPGA net provided by the inter-FPGA router. Using this information, the benchmark circuit is annotated with the inter-FPGA routing delays for the given MFS and the post-routing critical path delay (CPD_PR) is calculated. CPD_PR is dominated by route-throughs in TORUS. In route-through scenario, the signal sent from the source

FPGA, enters into one pin of the intermediate FPGA, travels through the on-chip routing lines and then exits through the other pin, without utilizing any of the on-chip logic of the intermediate FPGA. These types of nets impose even larger latencies as compared to direct nets and are responsible for reducing system clock frequency. The route-through delay value is set to be 10ns based on the following assumptions. The spatial distance between the source and destination FPGAs is very large and they are not adjacent to each other. Secondly, the input buffer of the intermediate FPGA is on one side and the output buffer is on the opposite side of the chip and the signal has to “route through” the chip from one end to the other [81] [82].

It is important to note here that the characterization of the critical path employed here has two limitations: First of all, some designs might implement multi-cycle operations and the critical path in each cycle might diverge from the definition of critical path. Secondly, there can be a possibility that the critical path calculated is a false path. Despite these constraints, block-oriented technique provides realistically acceptable results and can be used for accurate pre and post routing static timing analysis estimates based on back annotation. Furthermore, the developed STA is suitable only for synchronous mode of communication.

6.3.6. Time-Multiplexed Inter-FPGA Router

An architecture-specific time-multiplexed scalable router was developed to obtain the routing results for both 2D and 3D architectures. The number of FPGAs can be increased or decreased according to the MFS size.

The router developed for this research not only exerts to find the shortest path for each inter-FPGA net but also addresses issues like routing congestion by employing multiplexing. The routing problem in the TORUS is slightly more complicated because of its lesser connectivity as compared to CCG. FPGAs in TORUS are used for both logic and routing. After a circuit is mapped to an architecture, each FPGA will have a number of I/O pins specified for primary inputs and outputs. The rest of the pins are used for inter-FPGA data communication and for route-throughs.

After efficient partitioning and appropriate resource utilization of all FPGAs using accurate placement tools, larger circuits still have the tendency to run out of the I/Os, resulting in routing failures. To address this issue, the router used multiplexing which assembled multiple compatible design signals (nets) and serialized them through the same FPGA pin and board trace and then de-multiplexed them at the receiving FPGA. Multiplexing ratio is the ratio of the number of inter-FPGA nets to the number of pins. Multiplexing ratio of 1 means no multiplexing is done. As discussed earlier, 152 pins per FPGA are available for data transfer in both CCG and TORUS routing architectures. In differential signaling, one pair of pins is specified for clock transfer and therefore 150 pins are left for data transfer in both CCG and TORUS routing architectures. When FPGAs are connected via GTY transceivers, only 16×2 pairs of MGT I/Os per FPGA are available for data transfer. As it can be seen that the available pins in each scenario are not sufficient to ensure 100% routing of inter-FPGA nets. Therefore, the time-multiplexed inter-FPGA router iteratively increases the multiplexing ratio, until all inter-FPGA nets are successfully routed with least number of route-throughs. The multiplexing ratio value that ensures 100% routing of inter-FPGA nets is called threshold multiplexing ratio ($mux_{threshold}$).

The time-multiplexed routing for both 2D and 3D platforms can be carried out in similar manner by specifying the number of FPGAs per platform and the interconnection grid. The justification for this assumption lies in the fact that using flexible plastic optical fiber in 3D platforms allows to unfold the system back into a planar one, thus making it conceptually similar to its 2D counterpart.

6.4. Evaluation Metric

To compare time-multiplexing schemes on different 2D and 3D routing architectures, benchmark circuits are implemented on each and compared for the emulation time and system frequency, as described below.

6.4.1. Emulation Time & System Frequency

The speed of an MFS, for a given circuit, is determined predominantly by the latency bound i.e. the length of the post-routing critical path obtained after a circuit has been placed and routed at the inter-chip level. MFS static timing analysis tool developed in this study

calculated the post routing critical path delay for a given circuit and MFS architecture. Post-routing critical path delay is governed by the internal design delay and system routing delay. As compared to the internal delay, board routing delay has a larger impact on the overall system performance. The routing architecture employed mainly dictates the system routing delay. Multiplexing only scales up the external routing delay and in turn reduces the emulation frequency of the MFS.

System frequency is the reciprocal of the emulation time period obtained from the post-routing critical-path delay. The relationship between the system clock frequency and different multiplexing schemes has been discussed in detail and derived in Chapter 3.

6.5. Benchmark Circuits

Six largest open-source real sequential benchmark circuits are used for this experimental work. All of these benchmark circuits are FPGA proven, clock synchronous and utilize heterogeneous on-chip resources. Table 6.2(a) and (b) provide the circuit name and size of each design. These digital sequential benchmark circuits are obtained from Gaisler [83] which are accessible as a gate-level netlist in *.blif* format.

Table 6.2 (a): Benchmark Circuits

design	sequential	inverter	buffer	Logic	tristate	total
vga_lcd	17,079	21,397	2,542	83,013	-	124,031
leon2	149,381	104,393	14,964	511,665	53	780,456
netcard	97,831	61,712	11,946	552,506	48	724,043
leon3mp	108,839	87,122	3,303	346,539	33	545,836
leon3- avnet- 3s1500	185,025	169,668	4,333	540,522	84	899,632

Table 6.2 (b): Benchmark Circuit

design	BLIF Blocks	ALUTs	REGs	DSP
mcml	49,210	11,748	36,771	72

Leon2 processor was developed for and by the European Space Agency (ESA) whereas, Leon3 processor is a 32-bit processor based on the SPARC-V8 architecture with support for multiprocessing configurations.

One benchmark circuit (*vga_lcd*) is taken from OpenCores [83], and one (*mcml*) from VTR 7.0 benchmark suite [84]. *mcml* is an application that uses Monte Carlo simulation of photons. Each circuit is tech-mapped and converted to LUT-level netlist for further processing as mentioned earlier.

6.6. Summary

The experimental platform and the CAD tools developed for mapping the benchmark circuits to different MFS routing architectures were described in this chapter. The architecture evaluation metric and the benchmark circuits employed are also presented. In this research, particular attention is paid to the development of static timing analyzer for evaluating the speed of different MFS routing architectures. To our knowledge, none of the existing timing analysis tool encompasses multi-chip platforms and definitely do not cover 3D architectures or off-chip optical interface. The architectures are then evaluated and compared on the basis of critical path delay and system frequency. The evaluation and comparison results are presented in the next chapter for varying multiplexing ratio and for different existing and proposed MFS routing architectures.

Chapter 7 Experimental Results and Comparison of Architectures

7.1. Introduction

In this chapter, we present the experimental results obtained after mapping the benchmark circuits using the CAD tools developed to different MFS routing architectures. Three multiplexing schemes with two routing architectures are evaluated and compared for their performance. We also compared the system frequency behavior of the existing MFS and the proposed 2D and 3D MFS architectures with optical interface. The key evaluation parameters are system frequency behavior (primary) and critical path delay (secondary) with the increasing serialization factor.

7.2. Comparison of Multiplexed Routing Architectures

7.2.1. Critical Path Delay

In this section, we present the effect of different routing architectures on the critical path delay in three multiplexing schemes. Six benchmark circuits are mapped on CCG and TORUS routing architectures, using the CAD flow described in previous chapter. KaFFPaE partitioner partitioned the design into six sub-circuits. Table 7.1 shows the partitioning results. Number of nets before partitioning are represented by m . Number of inter-FPGA nets after partitioning are represented by n . Time taken by the partitioner to partition each design is also provided in seconds. Figure 7.1 shows the number of inter-FPGA cut nets after partitioning.

Table 7.1: KaFFPaE Partitioning Results

	vga_lcd	leon2	netcard	leon3mp	leon3- avnet- 3s1500	mcml
<i>m</i>	154,989	1,609,675	963,625	1,048,573	1,573,018	465,945
<i>n</i>	35,118	353,606	101,004	91,794	130,320	155,057
Time taken for partitioning (sec)	566.8	51,486.7	9,725.24	10,984.6	26,750.6	5,462.3

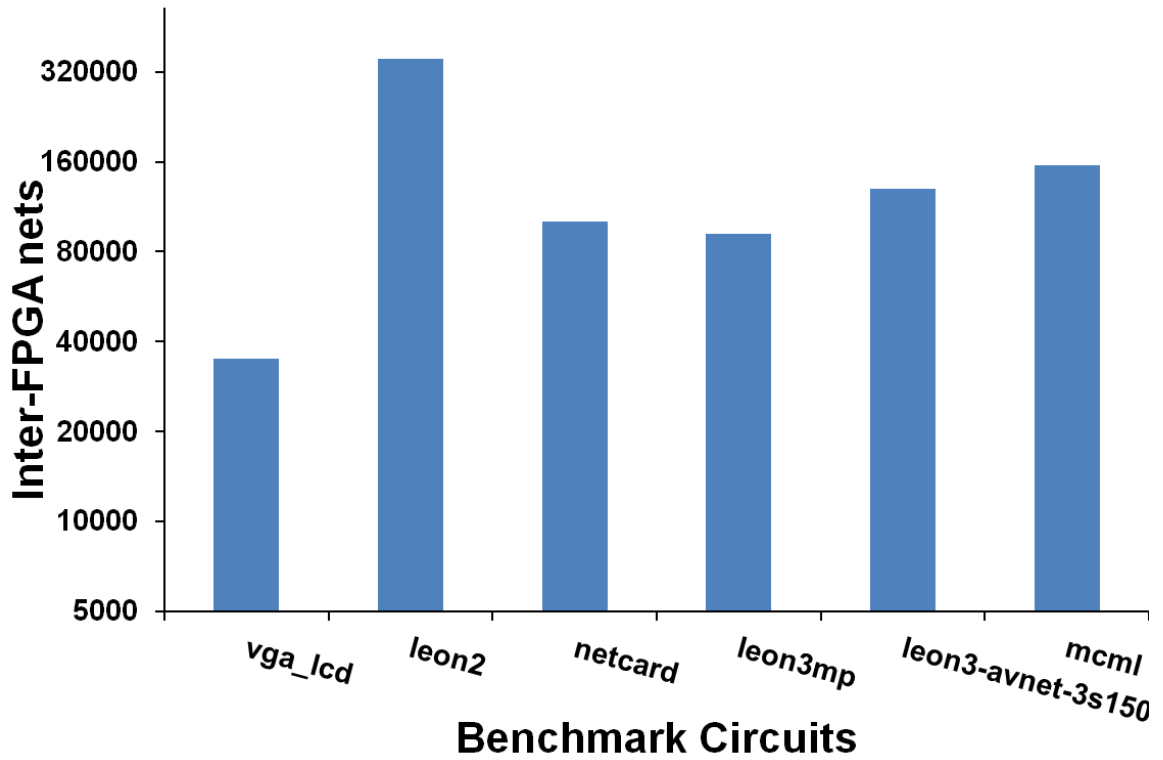


Figure 7.1: Number of Inter-FPGA nets After Partitioning

The results indicate that KaFFPaE took lesser time to partition medium-sized designs (*vga_lcd* & *mcml*) as compared to other large designs. Secondly, in all benchmark circuits, the total number of inter-FPGA nets after partitioning is clearly very large than the

available I/O capacity of KU3P FPGA, which justifies the need to employ multiplexing. As discussed earlier, logic multiplexing scheme allowed 152 pins per FPGA for data transfer. SERDES allowed 150 pins per FPGA for data transfer and in MGT there were 16*4 pins per FPGA for data transfer. Each benchmark circuit has a threshold multiplexing ratio ($mux_{threshold}$), below which the inter-FPGA net routing fails. Routing architecture dictates $mux_{threshold}$. Table 7.2 presents $mux_{threshold}$ for CCG and TORUS multiplexed architectures.

Results showed that limited routing resources in TORUS resulted in higher $mux_{threshold}$ as compared to CCG in all three multiplexing schemes. Moreover, MGT exhibited highest $mux_{threshold}$ because there are only 16 MGTs per FPGA for data transfer.

Table 7.2: Threshold Multiplexing Factor ($mux_{threshold}$) for Multiplexed Routing MFS Architectures

	LM ($mux_{threshold}$)		SERDES ($mux_{threshold}$)		MGT ($mux_{threshold}$)	
	CCG	TORUS	CCG	TORUS	CCG	TORUS
vga_lcd	154	162	156	164	365	385
leon2	713	742	722	752	1694	1764
netcard	360	382	364	386	854	907
leon3mp	201	207	203	209	478	493
leon3-avnet-3s1500	428	457	434	463	1018	1087
mcml	312	326	317	331	742	776

After the time-multiplexed router completed the routing, static timing analyzer determined the critical path delay of each design. STA also calculated the pre-partitioned and post-partitioned critical path delays. Table 7.3 presents CPD, CPD_PP and CPD_PR values for CCG and TORUS multiplexed MFS routing architectures. All these platforms are 2D

planar MFS built with electrical interconnections. The table also indicates the number of route-throughs (RT) encountered in the post-routing CPD in TORUS architectures.

Table 7.3: Critical Path Delays (in nanoseconds) at Different Levels of Circuit Implementation for 2D MFS

	vga_lcd	leon2	netcard	leon3mp	leon3- avnet- 3s1500	mcml
CPD	5.4	8.775	16.875	26.325	23.675	22.275
CPD_PP	13.075	29.225	23.3	39.125	34.3	47.6
CPD_PR LM_CCG	15.69	35.07	27.96	46.95	41.16	57.12
CPD_PR LM_TORUS	30.99	65.67	43.26	108.15	71.76	87.72
CPD_PR SERDES_CCG	13.075	29.225	23.3	39.125	34.3	47.6
CPD_PR SERDES_TORUS	25.825	54.725	36.05	90.125	59.8	73.1
CPD_PR MGT_CCG	11.575	26.225	21.05	34.625	30.55	42.35
CPD_PR MGT_TORUS	23.575	50.225	33.05	82.625	54.55	66.35
RT in CPD_PR (TORUS)	1	2	1	4	2	2

As the results indicate pre-partitioned CPD is the least, because it is assumed that the design is mapped on a hypothetically large FPGA without any off-chip delays. The post-partitioned critical path delay is higher than CPD because it incorporated I/O pad and off-chip electrical interconnections delay. However, it is assumed that CPD_PP is calculated for platforms that have full connectivity and no route-throughs are involved. That is why

post-routing delay in CCG is the same as post-partitioning critical path delay. Routing penalties across all CCG platforms are lesser as compared to their TORUS counter-parts, because of the direct connections among all FPGAs on the board. However, in TORUS, the post-routing delay is significantly affected by the number of route-throughs encountered in critical path.

7.2.2. System Frequency

In this section, we present the effects of increasing multiplexing ratio in three different multiplexing schemes (LM, SERDES & MGT) in CCG and TORUS routing architectures. All platforms are 2D planar and are built with electrical interconnections. The relationship between the system frequency and the multiplexing ratio was derived in Chapter 3 for the three multiplexing schemes. Each benchmark circuit had a threshold value of multiplexing, below which the routing failed. The multiplexing ratio is increased from that threshold value onwards to study its impact on system frequency.

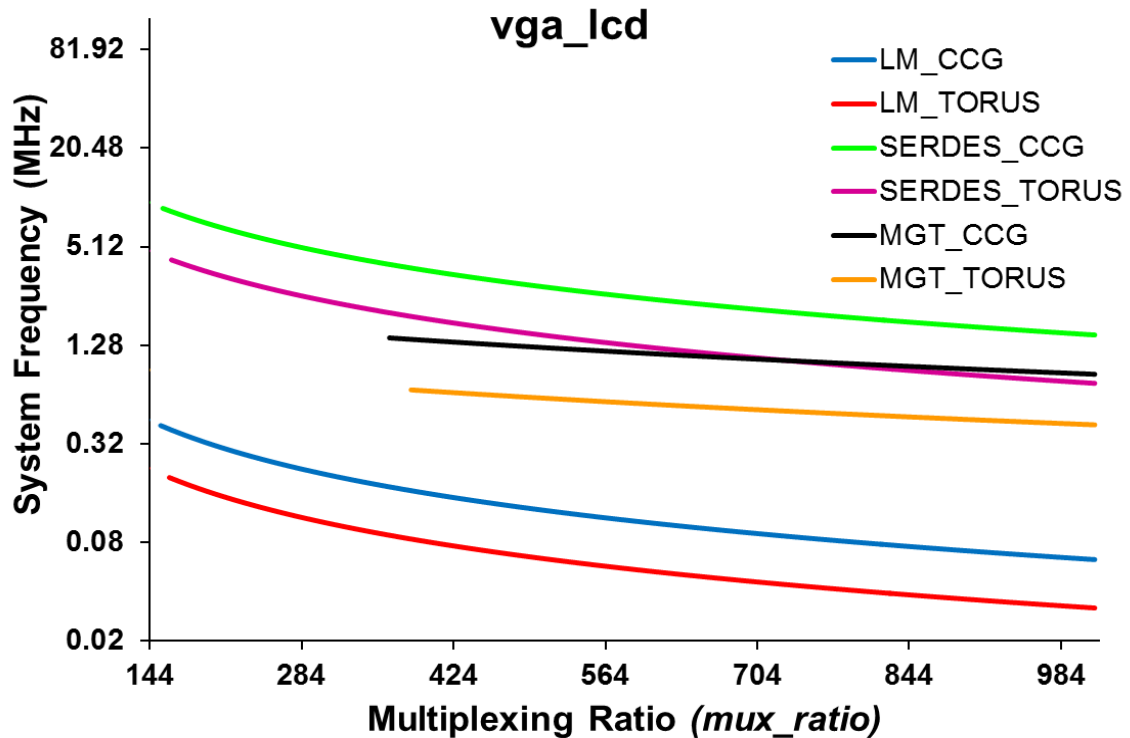


Figure 7.2(a): System Frequency vs Multiplexing Ratio

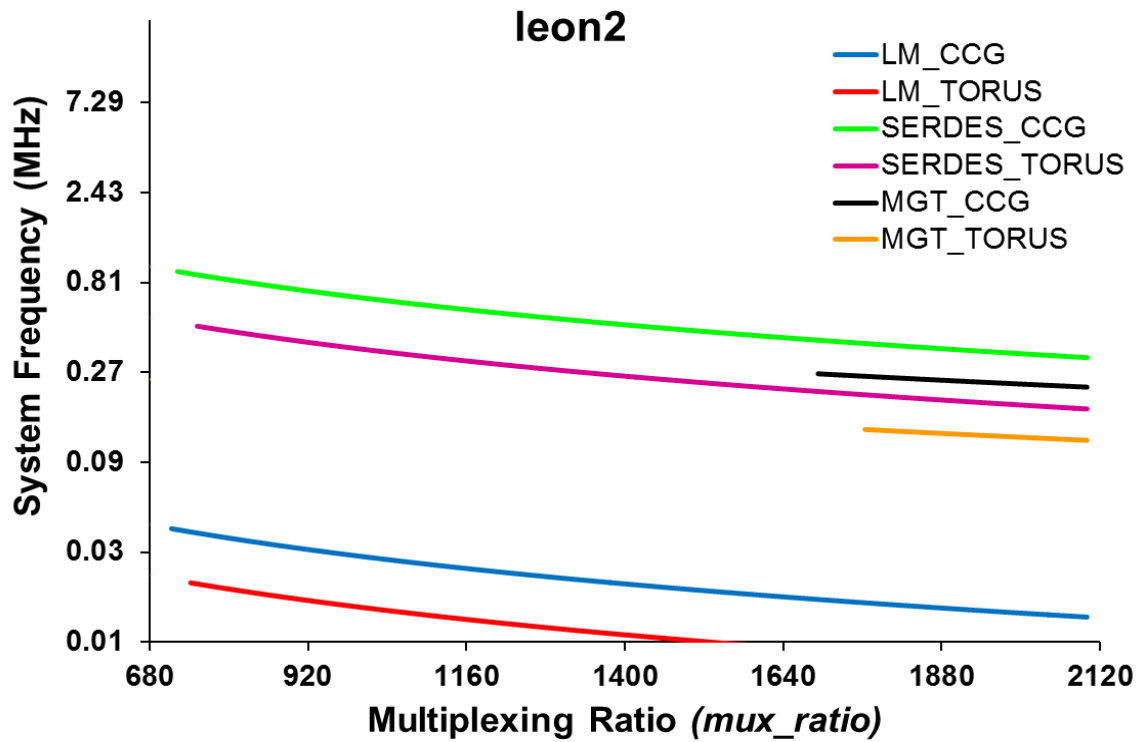


Figure 7.2(b): System Frequency vs Multiplexing Ratio

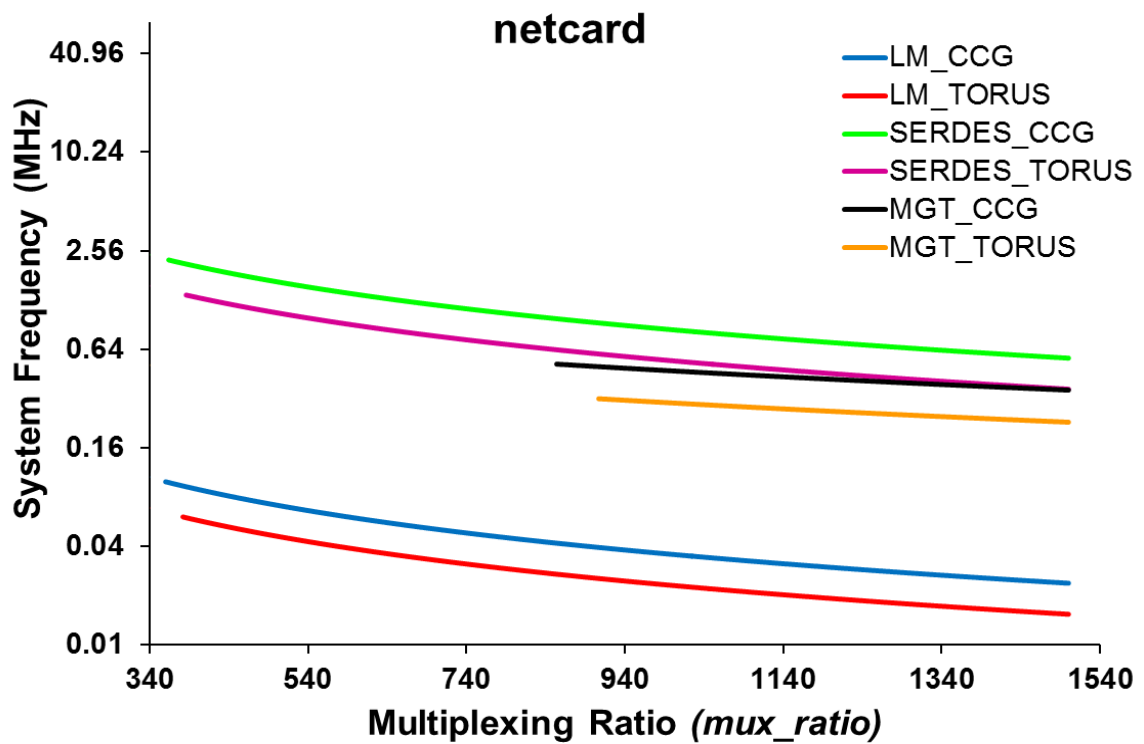


Figure 7.2(c): System Frequency vs Multiplexing Ratio

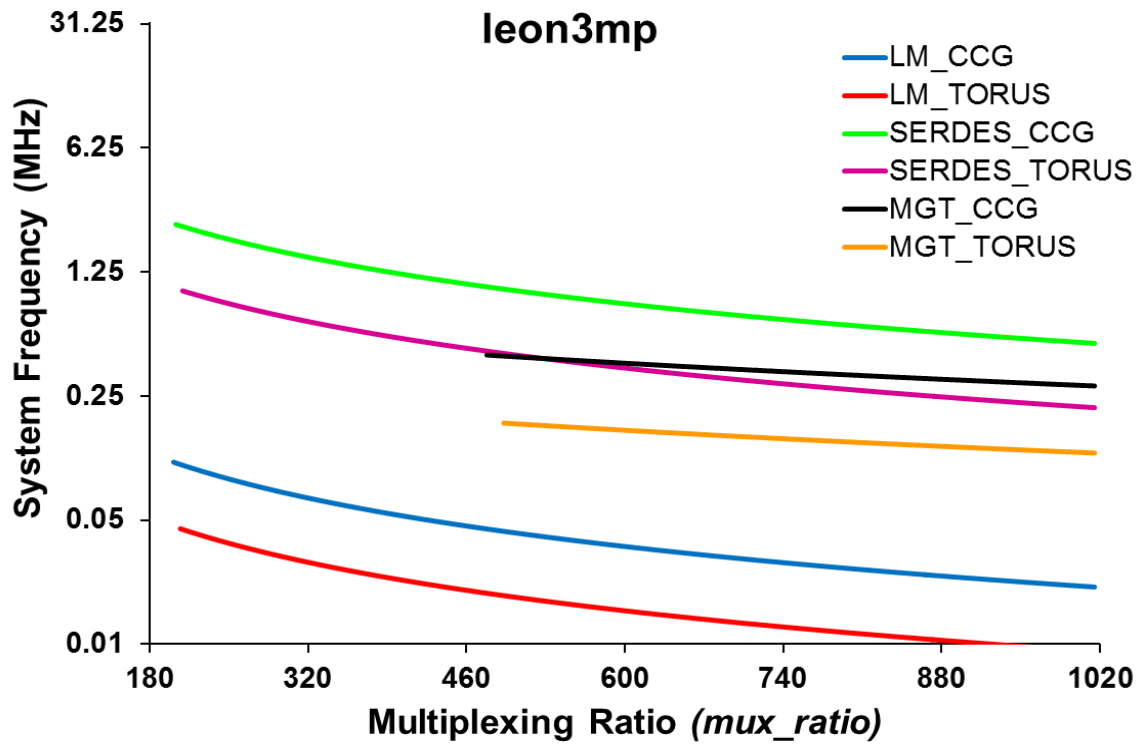


Figure 7.2(d): System Frequency vs Multiplexing Ratio

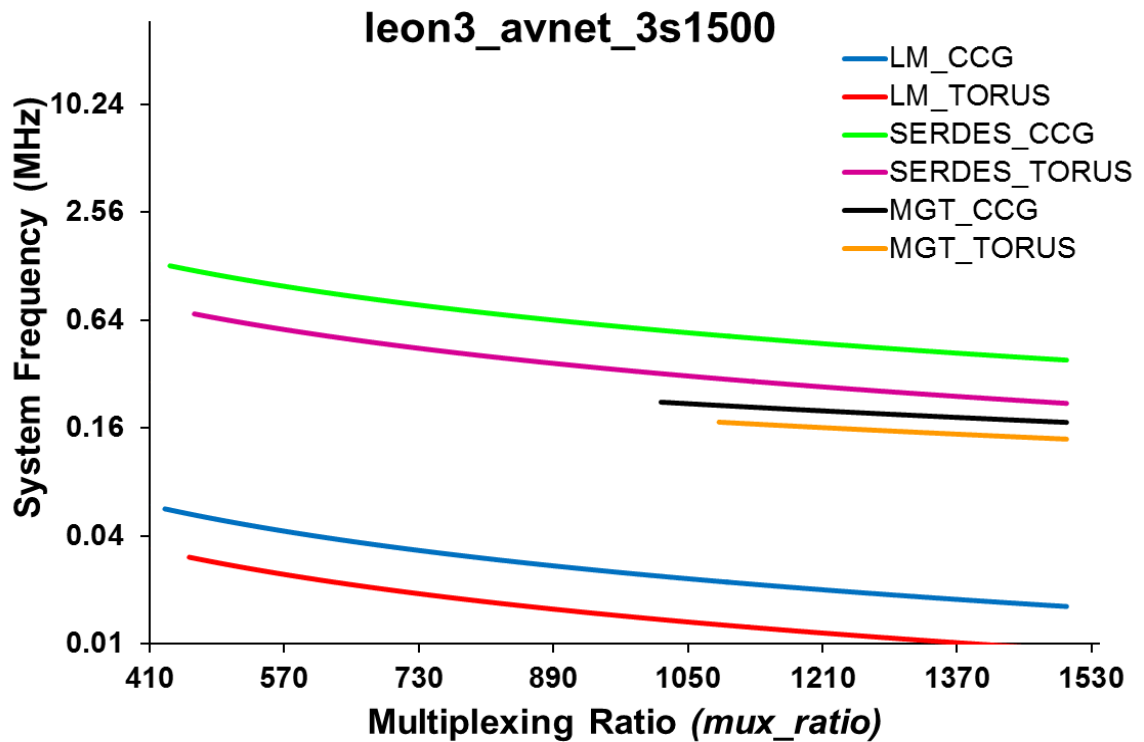


Figure 7.2(e): System Frequency vs Multiplexing Ratio

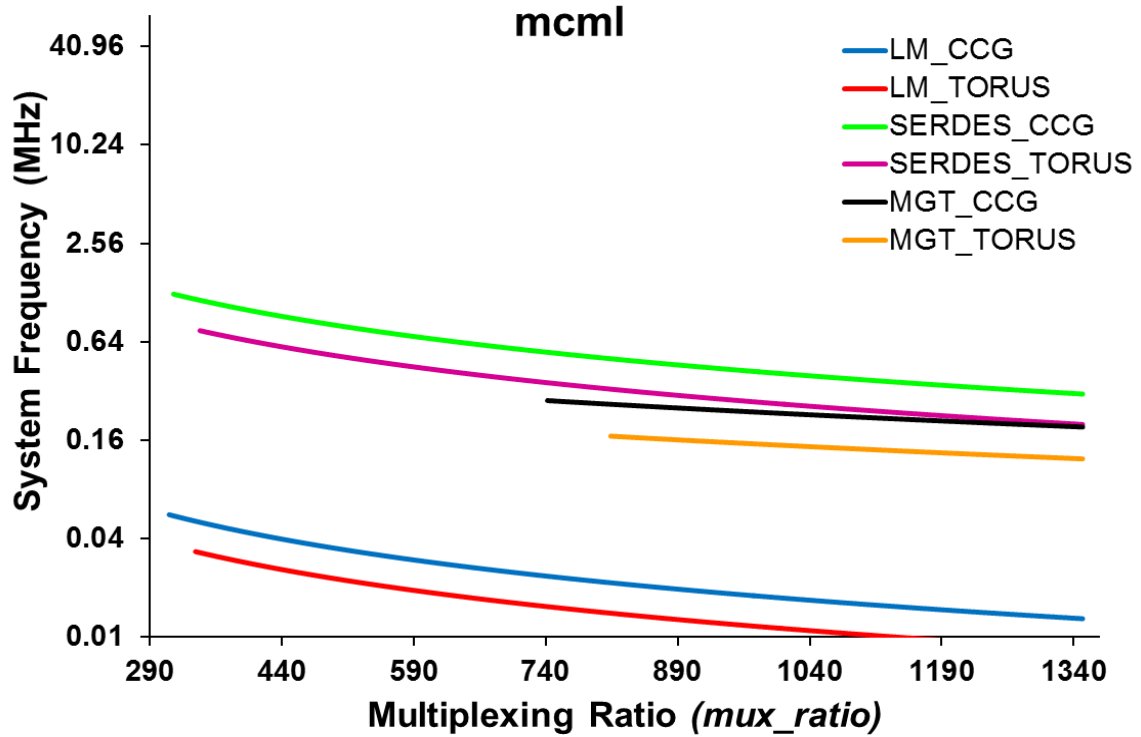


Figure 7.2(f): System Frequency vs Multiplexing Ratio

Figure 7.2(a)-(f) presents the system frequency performance of six benchmark circuits with increasing multiplexing ratio. The results indicate that with increasing serialization ratio, the achieved performance in both CCG and TORUS routing architectures is reduced. But, CCG always has notably higher performance than TORUS for every multiplexing scheme, because of full connectivity. Out of the three schemes, logic multiplexing exhibited least gain, whereas differential signaling based SERDES attained maximum frequency for the given range of multiplexing ratio. As discussed in Chapter 3, SERDES always performs better than single-ended scheme even with twice pin requirements and this concept has been experimentally validated here as well.

MGT's performance also decreased with the serialization ratio; however, the decline is not as prominent as the other two schemes. Moreover, the difference between SERDES and MGT performance in CCG decreased rapidly with increasing multiplexing factor. MGT performance is always lower than SERDES, because the number of GTY transceivers is very small i.e. 16. If the number of transceivers increases in future generations of FPGA, the performance of MGT can improve significantly.

In TORUS architecture, the routing penalties dominate the critical path delay and decrease the system performance. In benchmark circuits with more than two route-throughs, TORUS showed a system clock frequency decrease of up to 66% as compared to its CCG counterpart in all multiplexing scheme. Whereas, the benchmark circuits with up to two route-throughs, TORUS exhibited a percentage decrease of up to 19% as compared to CCG in all the multiplexing schemes.

7.3. Comparison of Proposed 2D Optical & Conventional MFS

7.3.1. Critical Path Delay

Since only GTY transceivers on KU3P enable an off-chip optical interface, that's why the proposed 2D latency-optimized MFS with optical interface only has MGT multiplexing. In this section we compare the critical path delay improvements in the proposed architecture as compared to that in conventional MFS. As discussed earlier, CPD_PR is calculated from the threshold multiplexing ratio ($mux_{threshold}$) onwards for each benchmark circuit. Both conventional and proposed architectures are built with two routing architectures CCG and TORUS.

Table 7.4: Critical Path Delays (in nanoseconds) at Different Levels of Circuit Implementation for 2D MFS

Circuit	CPD	CPD_PP	CPD_PR		RT in CPD_PR (TORUS)
			CCG	TORUS	
vga_lcd	5.4	11.575	11.575	23.575	1
leon2	8.775	26.225	26.225	50.225	2
netcard	16.875	21.05	21.05	33.05	1
leon3mp	26.325	34.625	34.625	82.625	4
leon3-avnet-3s1500	23.675	30.55	30.55	54.55	2
Mcml	22.275	42.35	42.35	66.35	2

Table 7.5: Critical Path Delays (in nanoseconds) at Different Levels of Circuit Implementation for 2D Optical MFS

Circuit	CPD	CPD_PP	CPD_PR		RT in CPD_PR (TORUS)
			CCG	TORUS	
vga_lcd	5.4	10.075	10.075	21.325	1
leon2	8.775	23.225	23.225	45.725	2
netcard	16.875	18.8	18.8	30.05	1
leon3mp	26.325	30.125	30.125	75.125	4
leon3-avnet-3s1500	23.675	26.8	26.8	49.3	2
mcml	22.275	37.1	37.1	59.6	2

Critical path delays only for conventional 2D MFS with MGT are presented again in Table 7.4. Whereas, Table 7.5 presents the critical path delays for the proposed 2D MFS with optical links.

As the numbers indicate, CCG's full connectivity resulted in smaller routing penalties as compared to TORUS where route-through latencies increased CPD_PR significantly. However, in proposed 2D MFS with optical interface, reduced per link latency has translated to faster speeds as compared to electrical interconnections in conventional MFS. Latency-optimized 2D MFS showed approximately 12% average decrease in critical path delay values in CCG, whereas, in TORUS the average decrease is nearly 9%. This is obvious, because in TORUS the route-through delays diminish the optical link latency improvement.

7.3.2. System Frequency

Figure 7.3(a)-(f) shows the system frequency of the six benchmarks after they have been routed across the four architectural models. The proposed architectures have off-chip optical links instead of electrical interconnections. As discussed in Chapter 4, optical

links in the proposed architectures have the same length as that of electrical traces in conventional 2D MFS. However, they tend to exhibit lesser latency as compared to their electrical counterparts improving the system frequency significantly. The results clearly validate the concept and show the encouraging impact of exploiting high speed optical interface.

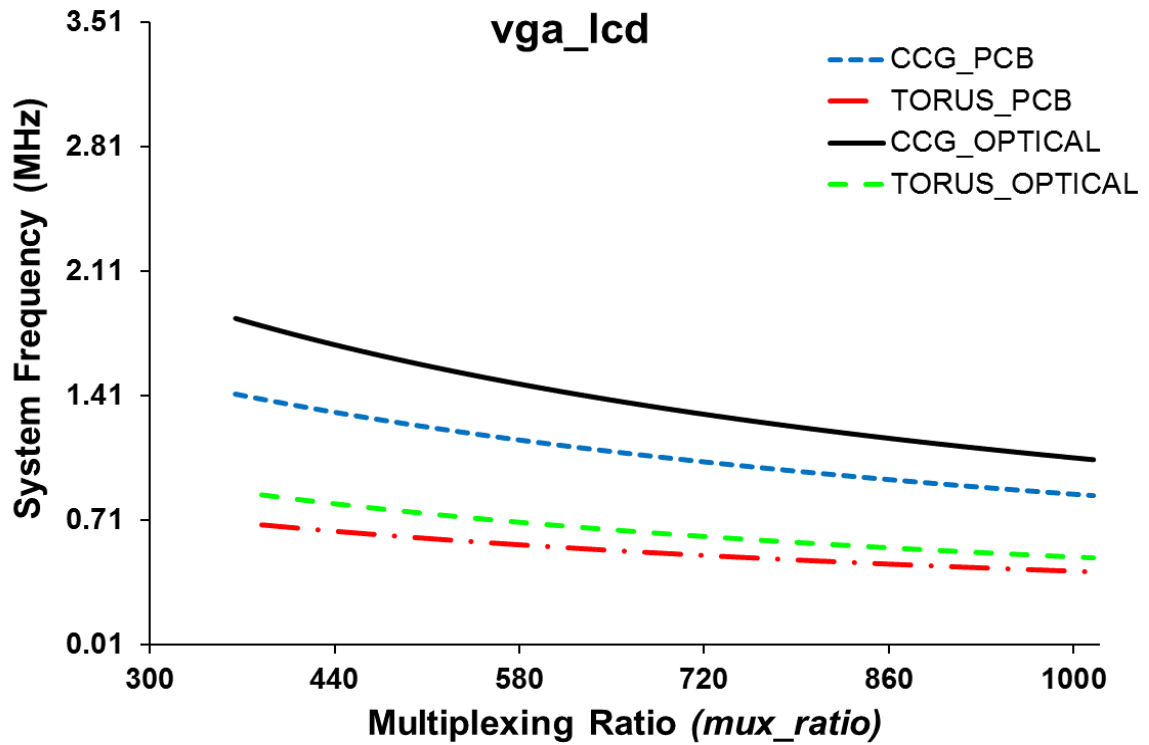


Figure 7.3(a): System Frequency 2D Conventional vs Optical MFS in CCG & TORUS

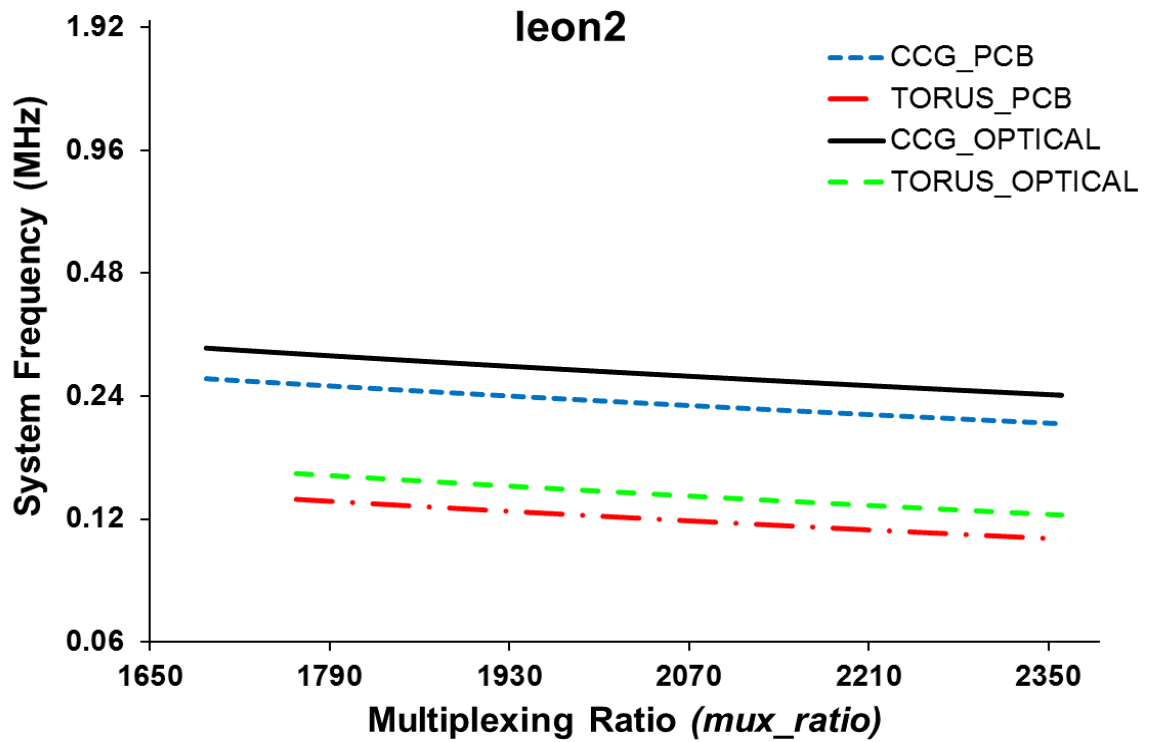


Figure 7.3(b): System Frequency 2D Conventional vs Optical MFS in CCG & TORUS

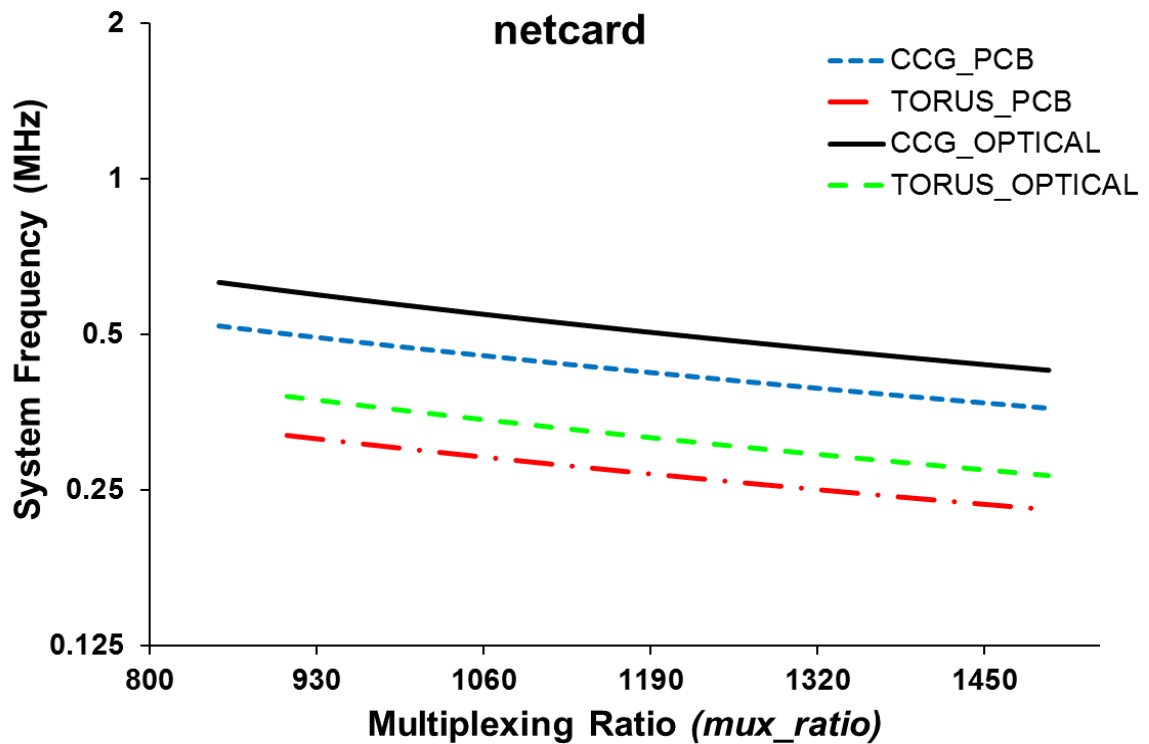


Figure 7.3(c): System Frequency 2D Conventional vs Optical MFS in CCG & TORUS

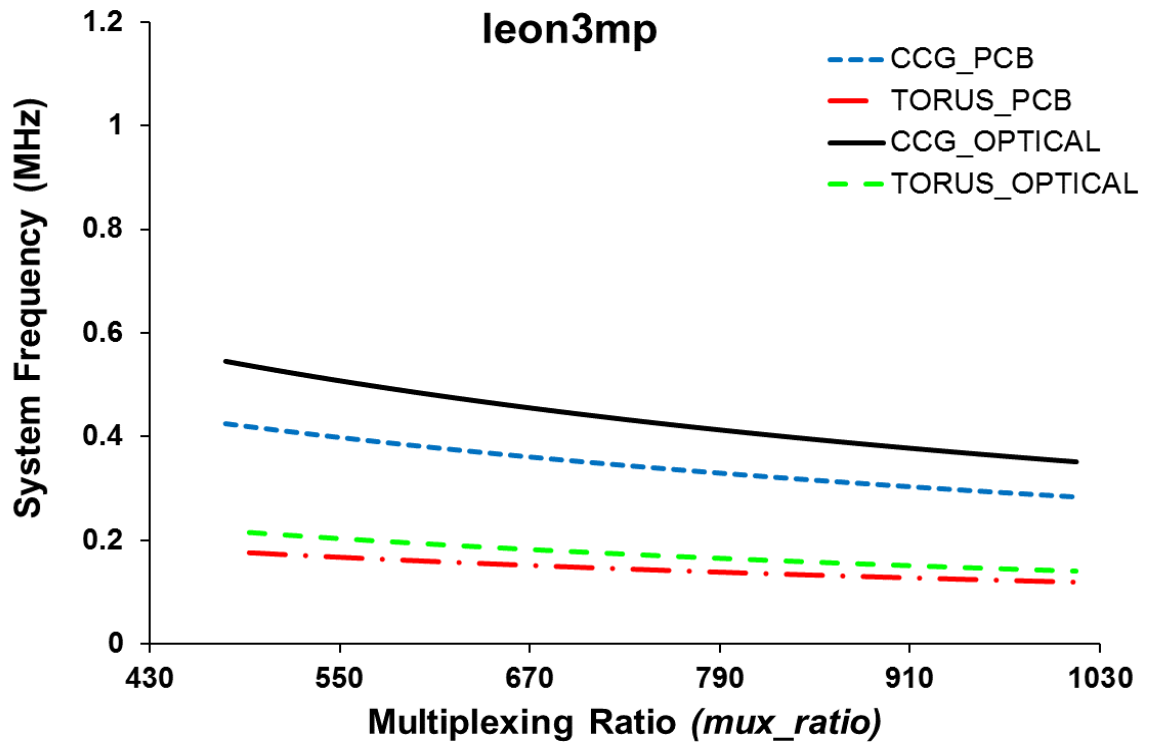


Figure 7.3(d): System Frequency 2D Conventional vs Optical MFS in CCG & TORUS

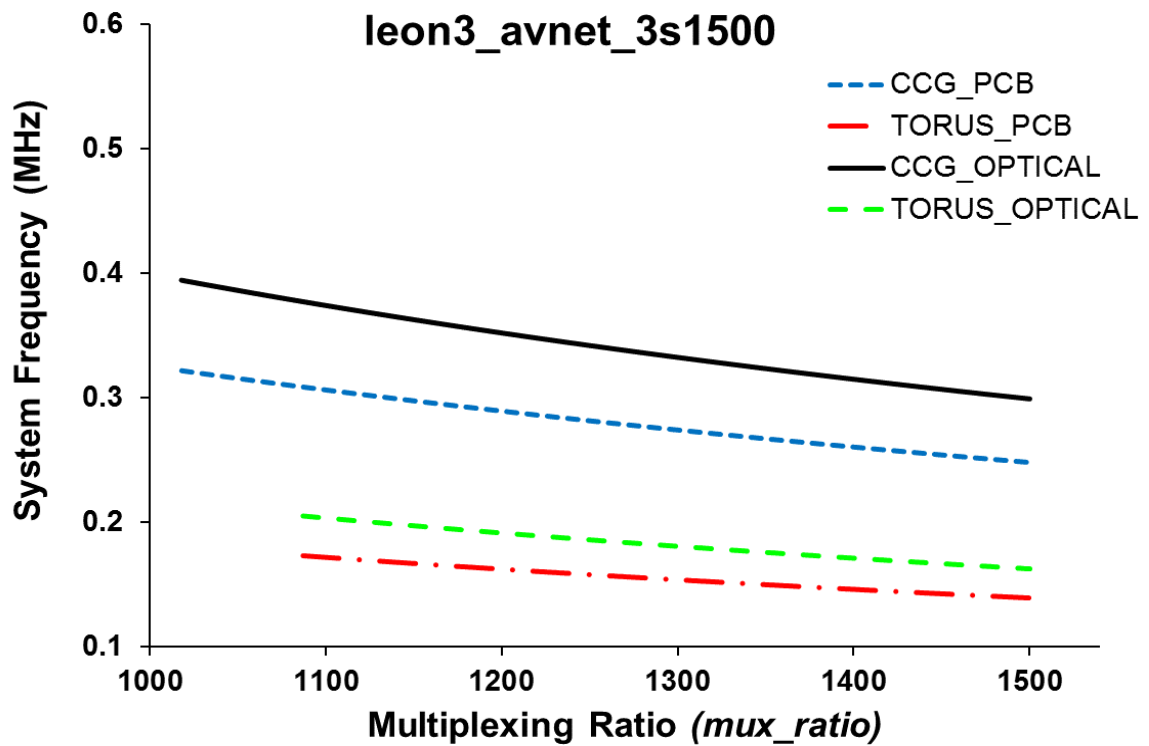


Figure 7.3(e): System Frequency 2D Conventional vs Optical MFS in CCG & TORUS

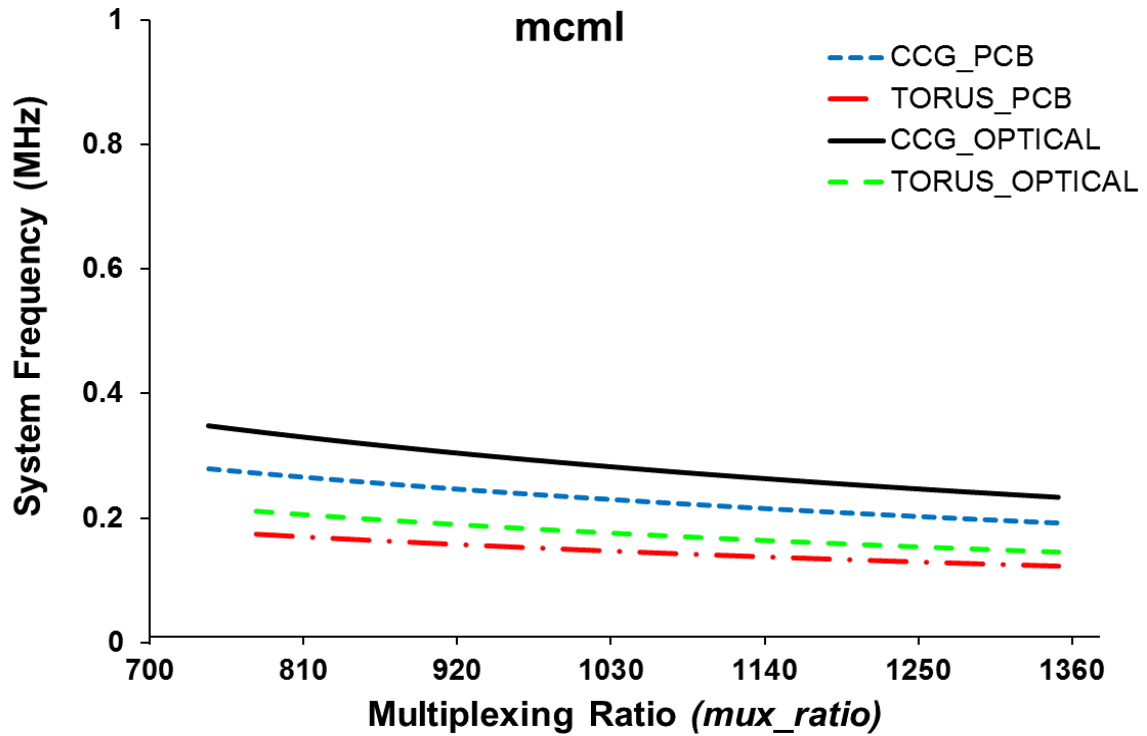


Figure 7.3(f): System Frequency 2D Conventional vs Optical MFS in CCG & TORUS

Experimental results indicate that as the multiplexing ratio increased, system frequency decreased in all the architectural models. However, CCG showed better performance than TORUS in every case owing to the lower off-chip latencies as expected. Additionally, all 2D optical CCG MFS performed better than any other structure, because the latencies across the board were greatly reduced thanks to full connectivity of CCG and better speed of optical links.

Similarly, TORUS with optical interface performed better than TORUS with electrical interconnects, owing to optical links. However, TORUS could not produce better results than CCG, because the route-throughs weakened the optical interface effects.

Figure 7.4(a)-(f) presents the frequency gain in the proposed 2D platforms when compared to their counterparts with electrical interconnections for multiplexing range above threshold value for all six benchmark circuits. The results indicate that all optical 2D MFSs clearly have performance advantage over their electrical counterparts.

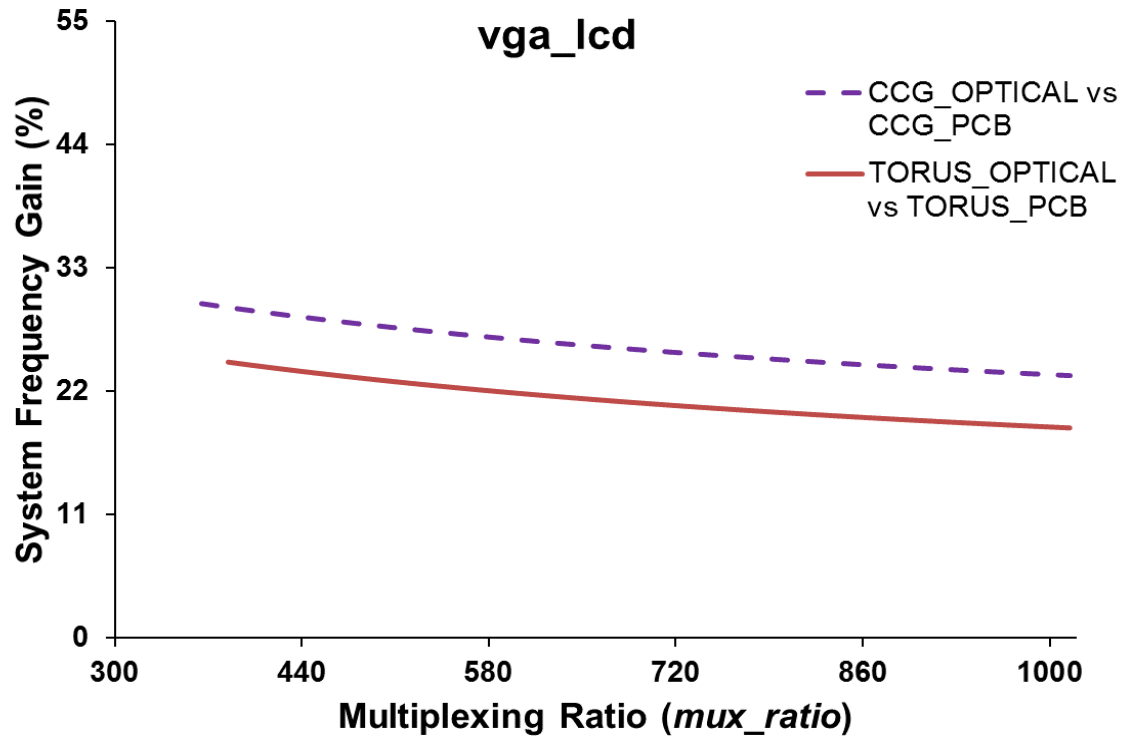


Figure 7.4(a): System Frequency Gain 2D Optical vs Conventional MFS in CCG & TORUS

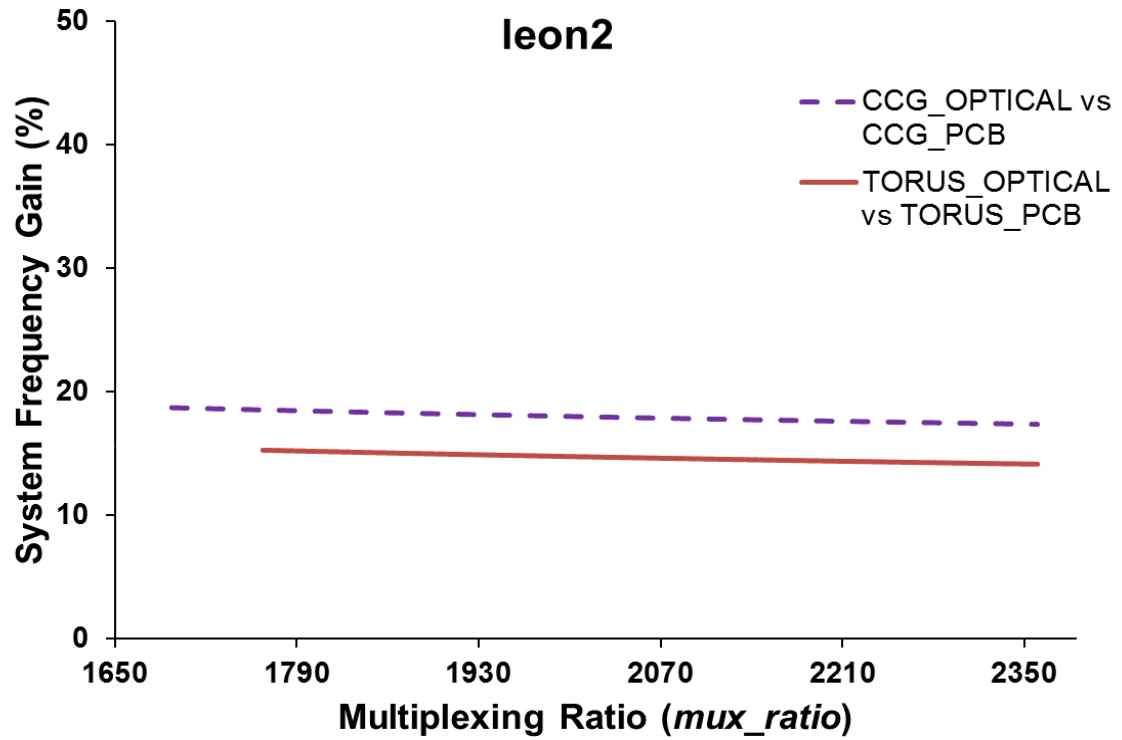


Figure 7.4(b): System Frequency Gain 2D Optical vs Conventional MFS in CCG & TORUS

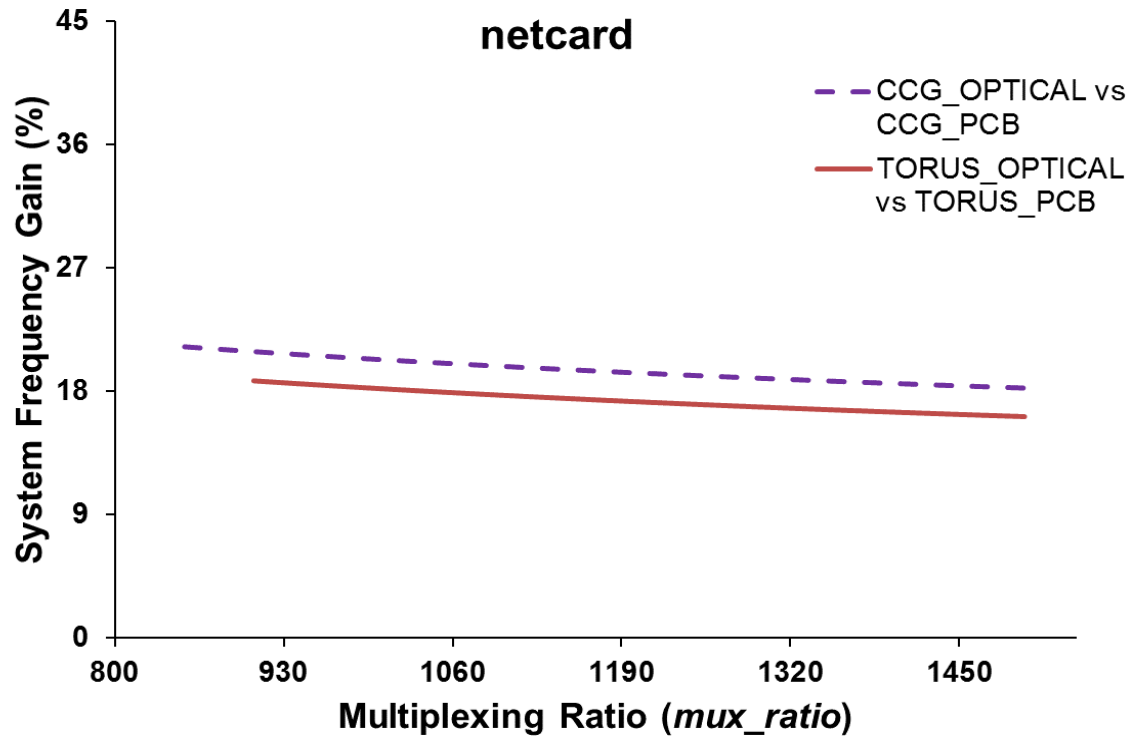


Figure 7.4(c): System Frequency Gain 2D Optical vs Conventional MFS in CCG & TORUS

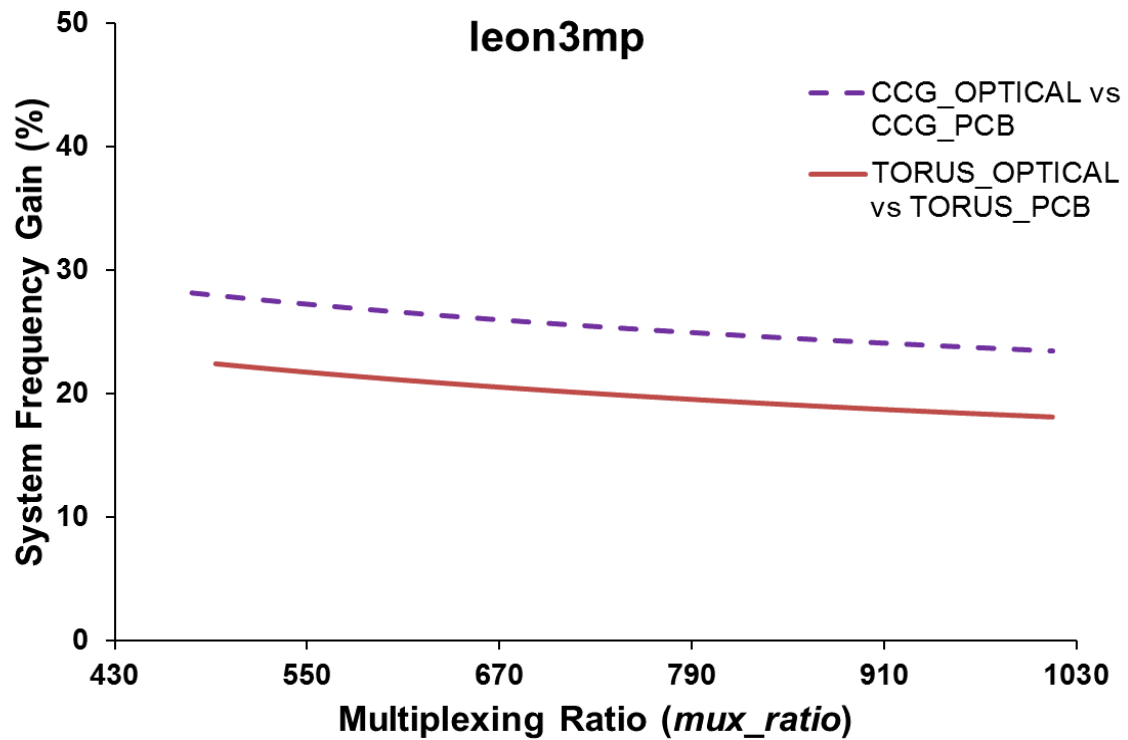


Figure 7.4(d): System Frequency Gain 2D Optical vs Conventional MFS in CCG & TORUS

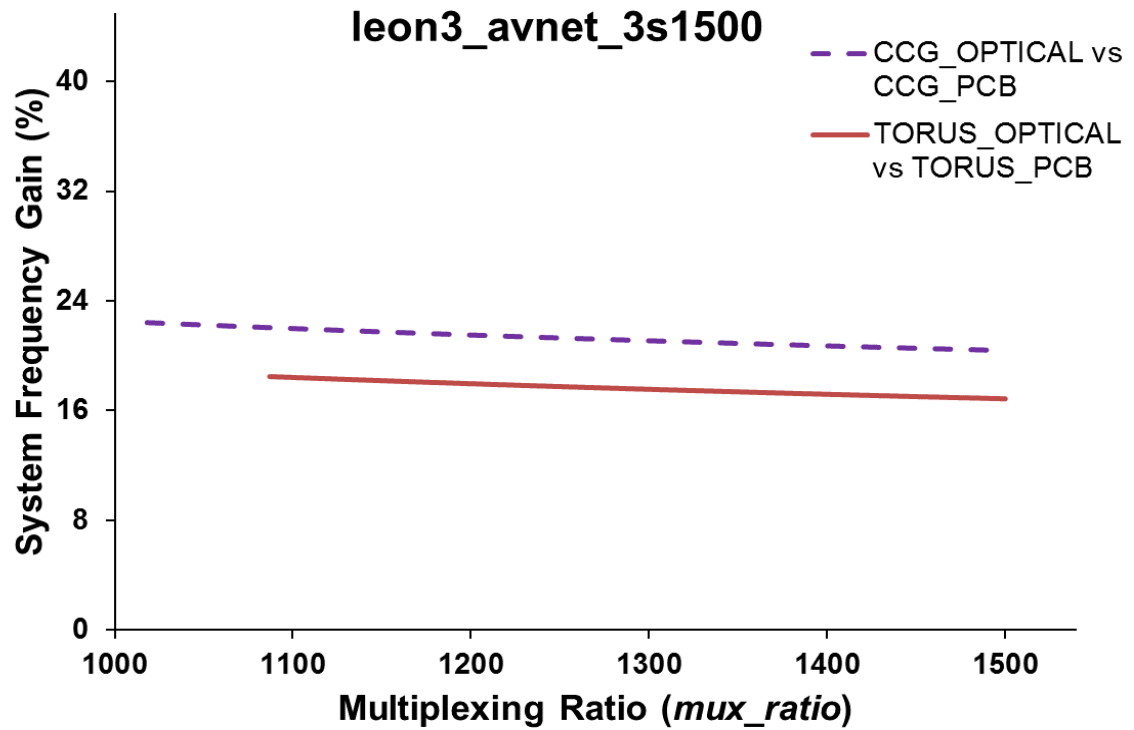


Figure 7.4(e): System Frequency Gain 2D Optical vs Conventional MFS in CCG & TORUS

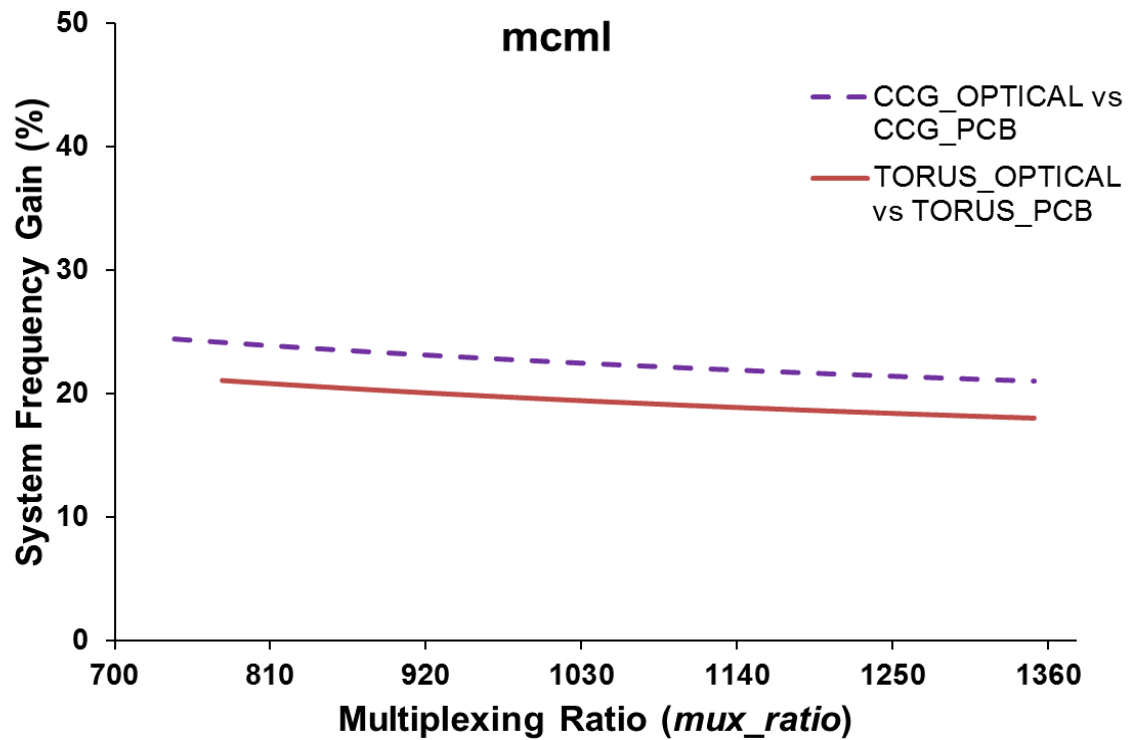


Figure 7.4(f): System Frequency Gain 2D Optical vs Conventional MFS in CCG & TORUS

Figure 7.5 shows the average performance gain in the proposed architectures and conventional MFS with MGT multiplexing. All 2D optical CCG platforms exhibited an average frequency gain of 22% as compared to 2D CCG architectures with electrical interconnects. In best case scenario i.e. *vga_lcd*, the performance gain was up to 26%. 2D TORUS optical MFS showed an improvement of 18% on average as compared to 2D TORUS with electrical interconnects. In best case scenario i.e. *vga_lcd*, 2D TORUS optical MFS system frequency gain was up to 21%.

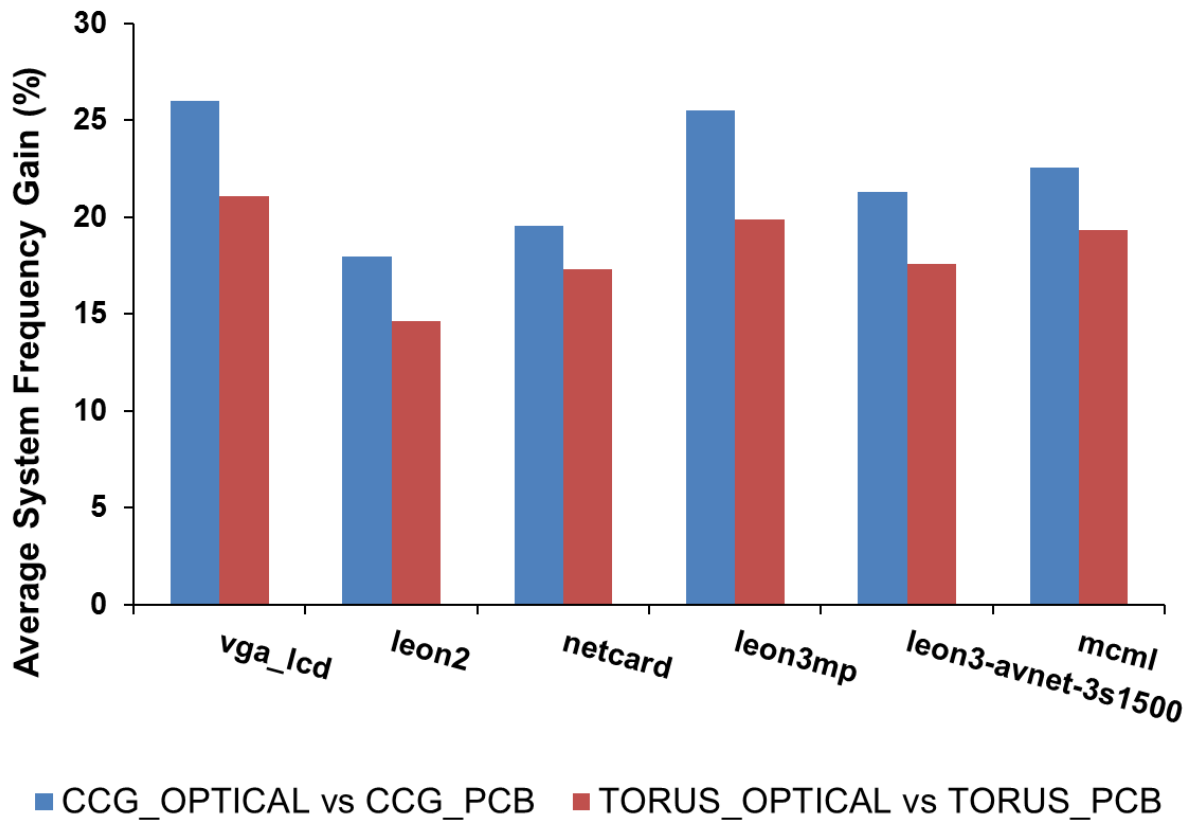


Figure 7.5: Average System Frequency Gain 2D Optical vs Conventional CCG & TORUS

7.4. Comparison of Proposed 3D Optical & Conventional MFS

7.4.1. Critical path Delay

This section summarizes the encouraging impacts of exploring the third dimension in MFS. Proposed 3D architectures with optical interface are presented in Chapter 5. Since third dimension reduces the wirelengths by half as compared to planar MFS, that's why the

benefit of employing optical interface becomes twofold. Earlier in the chapter, Table 7.4 presented the critical path delays at different levels for 2D conventional MFS. The $mux_{threshold}$ for the 3D CCG and TORUS routing architectures is the same as their 2D counterparts. Since 6X1 has a different routing structure as compared to CCG and TORUS, that's why it also has different $mux_{threshold}$ value for each benchmark circuit. Table 7.6 present the critical path delays before partitioning, after partitioning and after routing for 3D platforms. The table also shows the $mux_{threshold}$ value and the number of route-throughs incurred in CPD_PR in 6X1 routing architecture.

Table 7.6: Critical Path Delays (in nanoseconds) at Different Levels of Circuit Implementation for 3D Optical MFSs

Circuit	CPD	CPD_PP	$mux_{threshold}$ (6X1)	CPD_PR			RT in CPD_PR (6X1)
				3X2		6X1	
				CCG	TORUS		
vga_lcd	5.4	9.315	365	9.315	19.975	9.315	0
leon2	8.775	21.705	1889	21.705	43.445	32.575	1
netcard	16.875	17.66	854	17.66	28.53	17.66	0
leon3mp	26.325	27.845	521	27.845	71.325	60.455	3
leon3- avnet- 3s1500	23.675	24.9	1125	24.9	46.64	46.64	2
mcml	22.275	34.44	796	34.44	56.18	77.92	4

Longer inter-FPGA links and use of PCB connections contributed to the post-routing delays in 2D architectures as shown earlier. Whereas, direct shorter connections in 3X2 CCG architectures diminished the routing penalties significantly as compared to 2D CCG

and 3X2 TORUS where route-through latencies dominated the CPD_PR value above $mux_{threshold}$. That's why the average 3X2 TORUS post-routing latency is higher than 3X2 CCG in the proposed architectures.

Latency-optimized 3D MFS showed approximately 18% average decrease in critical path delay values in CCG, whereas, in best case scenario the decrease is up to 19.5%. In TORUS the average decrease is nearly 14% in all the benchmark circuits above the threshold multiplexing ratio, whereas in best case scenario the decrease is up to 15%. In 3D platforms, 6X1 showed the highest routing delays due to increased number of route-throughs. However, in *vga_lcd* and *netcard*, 6X1 showed same post-routing delay as 3X2 CCG because appropriate placement in 6X1 eliminated the route-throughs in CPD_PR resulting in the same system frequency as 3X2_CCG with the advantage of even smaller footprint area. 3X2_CCG architecture exhibited best results in all cases, owing to very low off-chip delays and direct shorter interconnections among all pairs of FPGAs.

7.4.2. System Frequency

Figure 7.6(a)-(f) show the system frequency of the six benchmark circuits after they have been routed across the five architectural models. The results clearly show the encouraging impact of exploiting the third dimension and high speed optical interface. The proposed 3D CCG and TORUS architectures performed better than their 2D counterparts in all benchmark circuits. As discussed in Chapter 5, any number of planes cannot be stacked together in 3D architectural models to achieve better performance than 2D MFS. Experimental results indicate that there is an optimal choice of planes and FPGAs per plane in 3D topologies that can deliver frequency improvement. That's why 6X1 performed better only where the placement eliminated the route-throughs. Increase in the multiplexing ratio reduced the system frequency above the threshold multiplexing ratio in all six benchmark circuits. This phenomenon is consistent across all five architectural models.

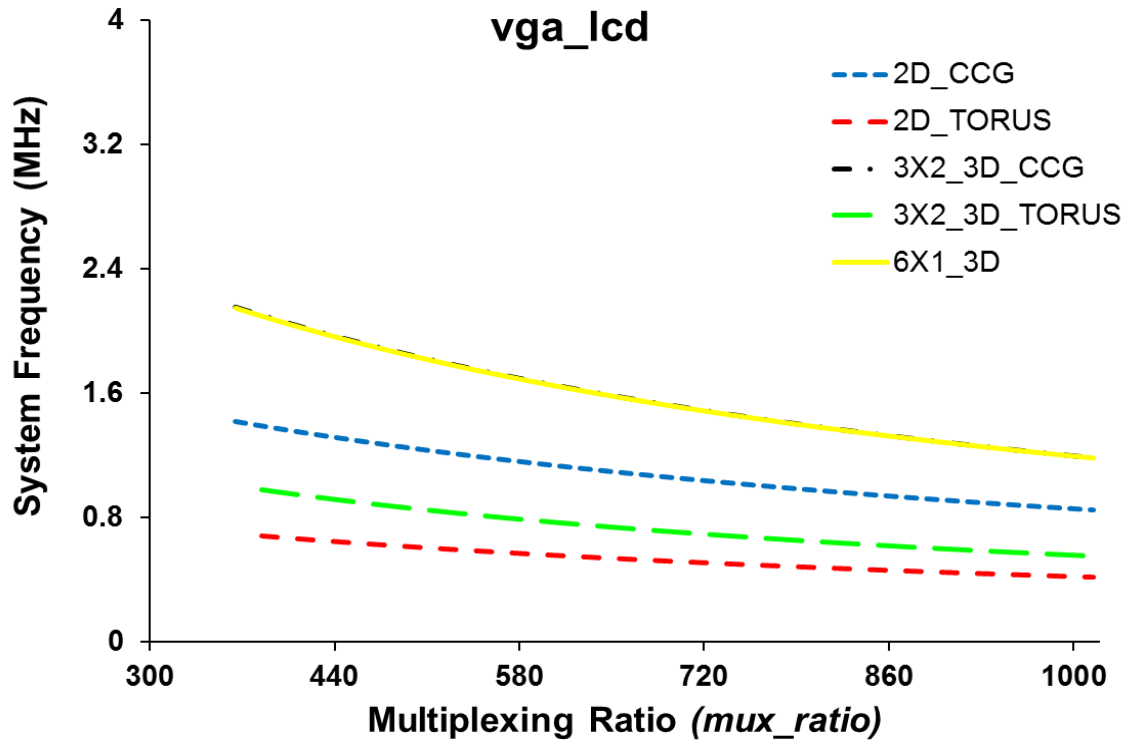


Figure 7.6(a): System Frequency 2D vs 3D with MGT Multiplexing Scheme

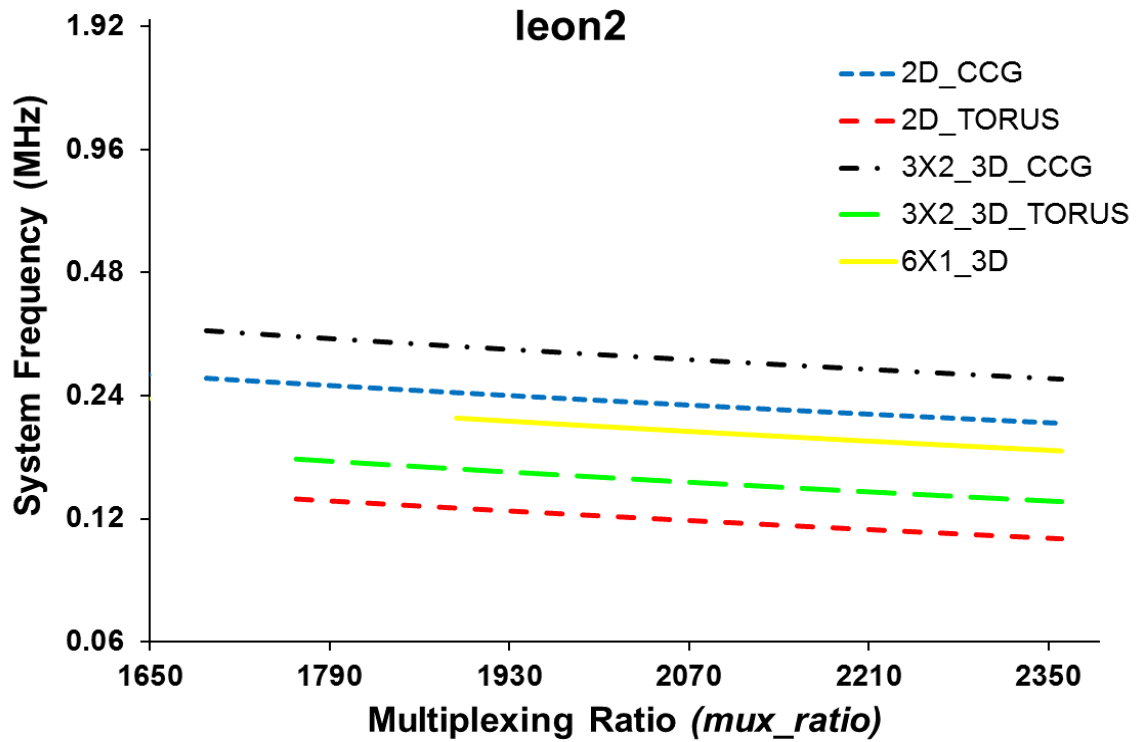


Figure 7.6(b): System Frequency 2D vs 3D with MGT Multiplexing Scheme

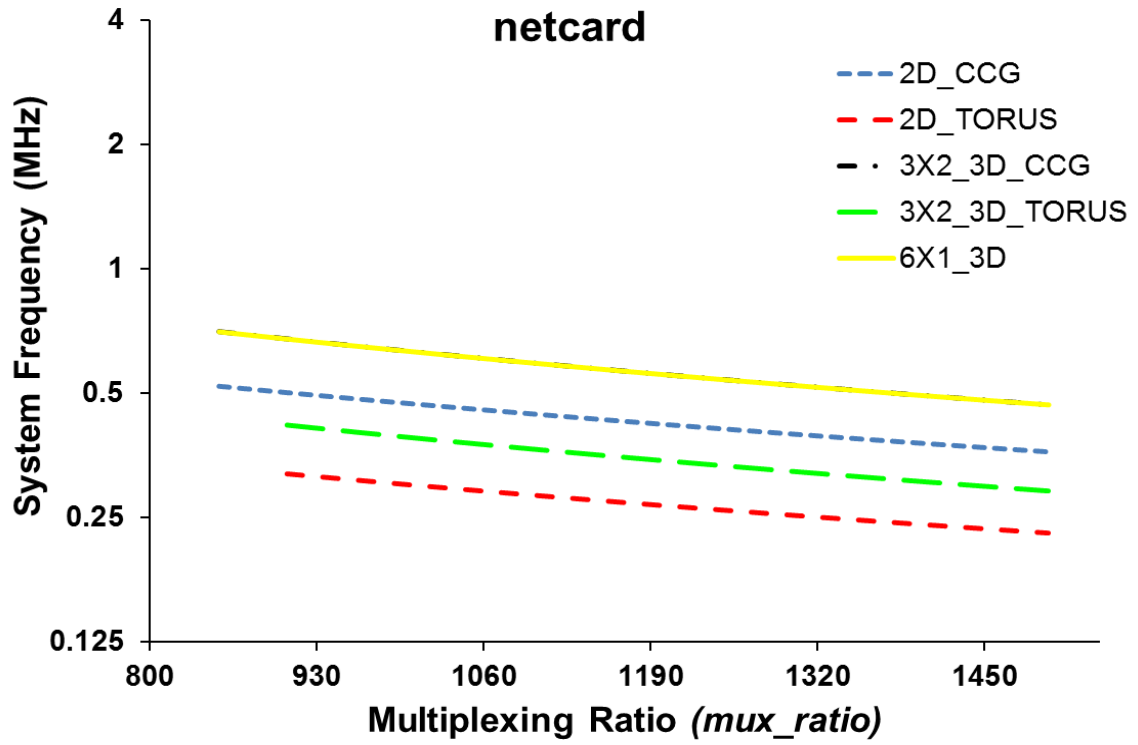


Figure 7.6(c): System Frequency 2D vs 3D with MGT Multiplexing Scheme

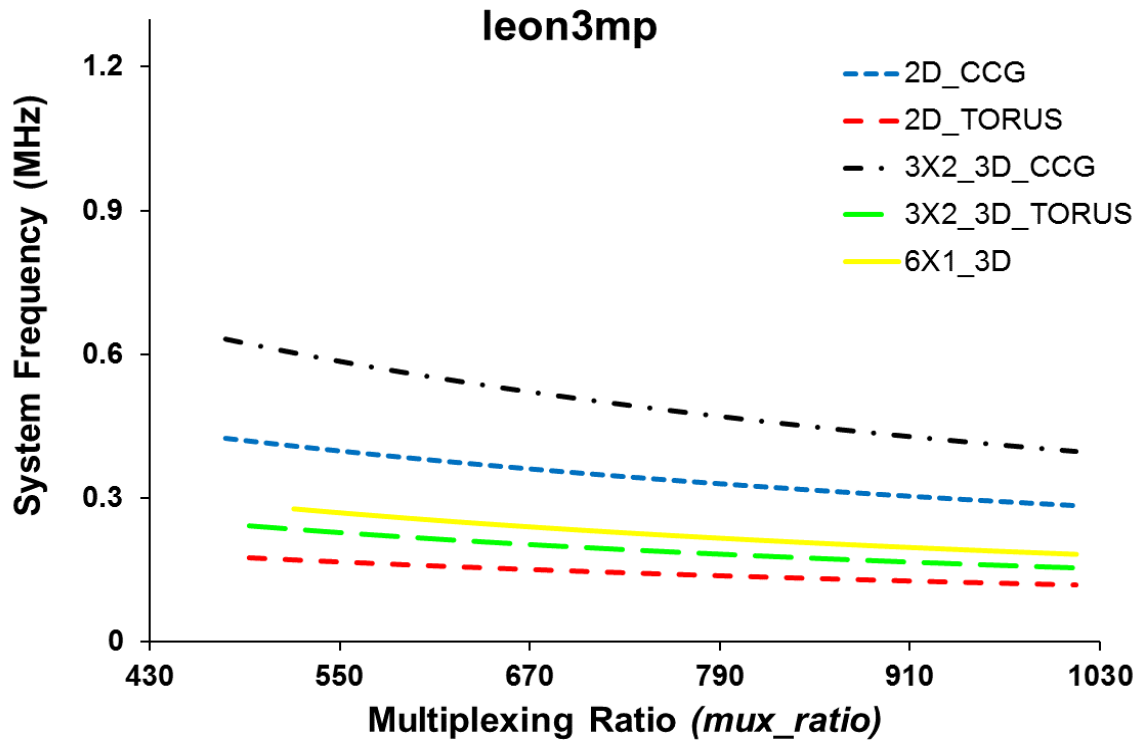


Figure 7.6(d): System Frequency 2D vs 3D with MGT Multiplexing Scheme

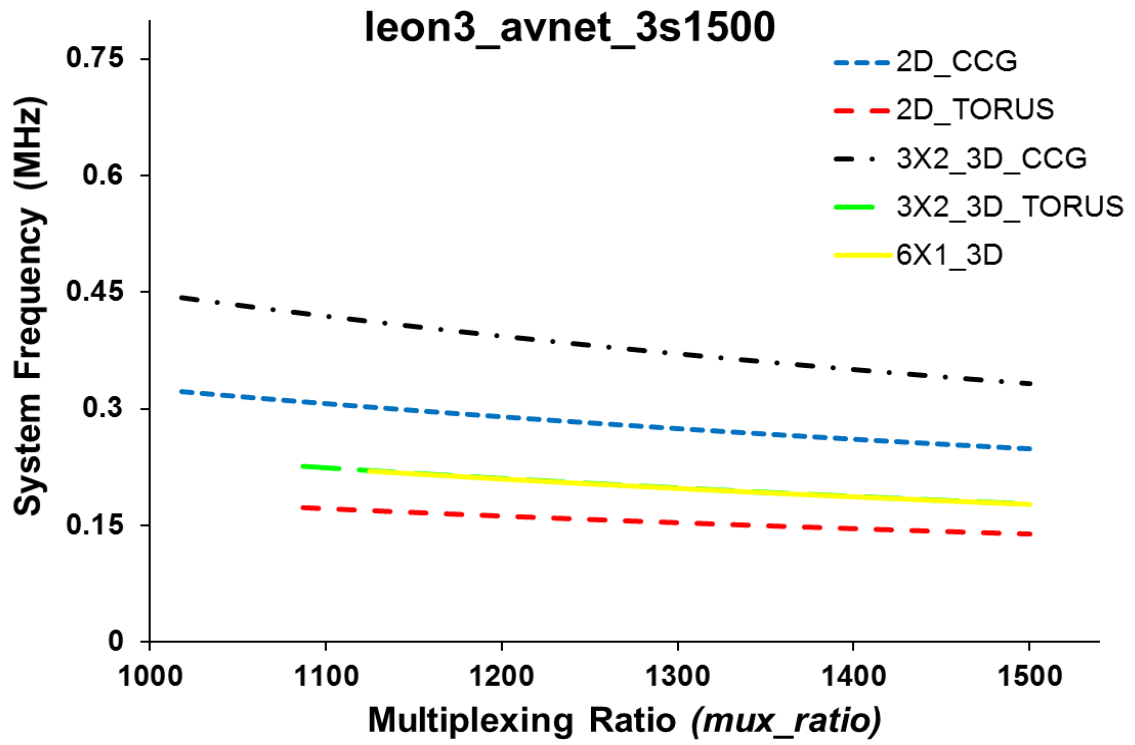


Figure 7.6(e): System Frequency 2D vs 3D with MGT Multiplexing Scheme

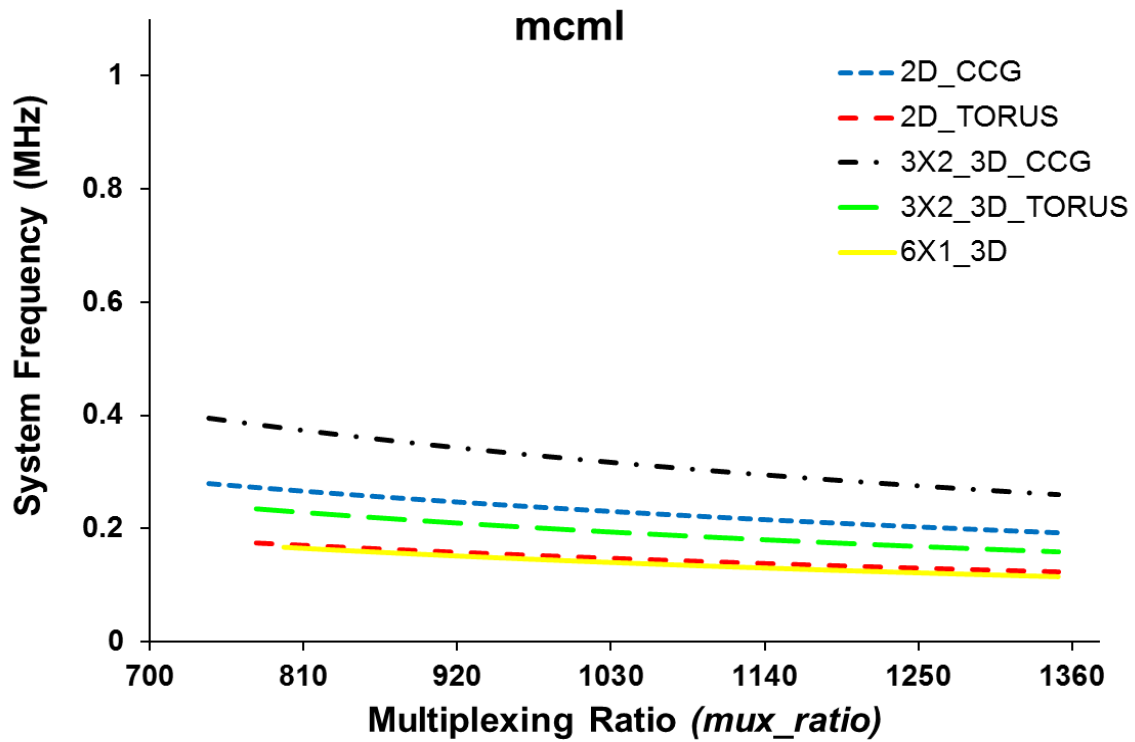


Figure 7.6(f): System Frequency 2D vs 3D with MGT Multiplexing Scheme

Figure 7.7(a)-(f) present the frequency gain in 3D platforms when compared to their 2D counterparts for multiplexing ratios above threshold value in all six benchmark circuits. As the results indicate all 3D CCG platforms exhibited a positive frequency gain over the entire range of multiplexing ratio as compared to 2D CCG architectures. 3D_3X2_TORUS also showed positive improvement as compared to 2D_TORUS however, couldn't perform better than CCG because of route-through penalties. 6X1 showed improvement only in medium-sized circuits (*vga_lcd* & *netcard*) where placement tool managed to eliminate the route-throughs. In larger circuits, the number of route-throughs increased due to limited routing capacity of 6X1, and resulted in poor frequency gain as compared to other 2D and 3D architectural models.

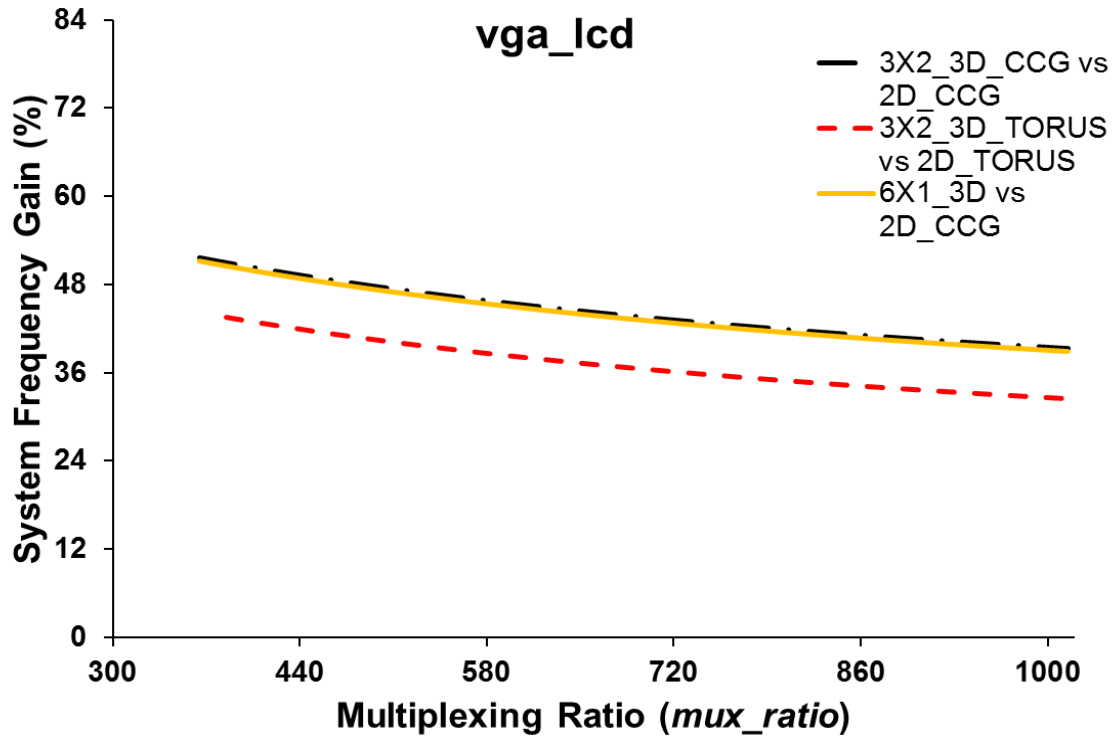


Figure 7.7(a): System Frequency Gain 2D vs 3D with MGT Multiplexing Scheme

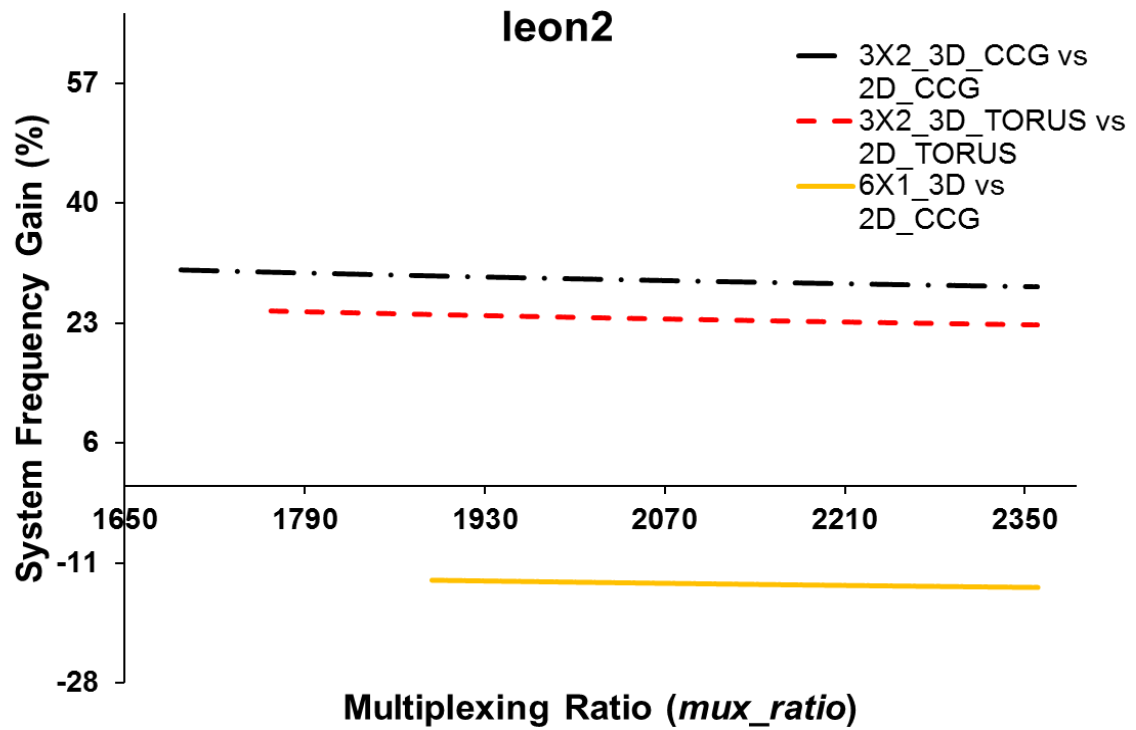


Figure 7.7(b): System Frequency Gain 2D vs 3D with MGT Multiplexing Scheme

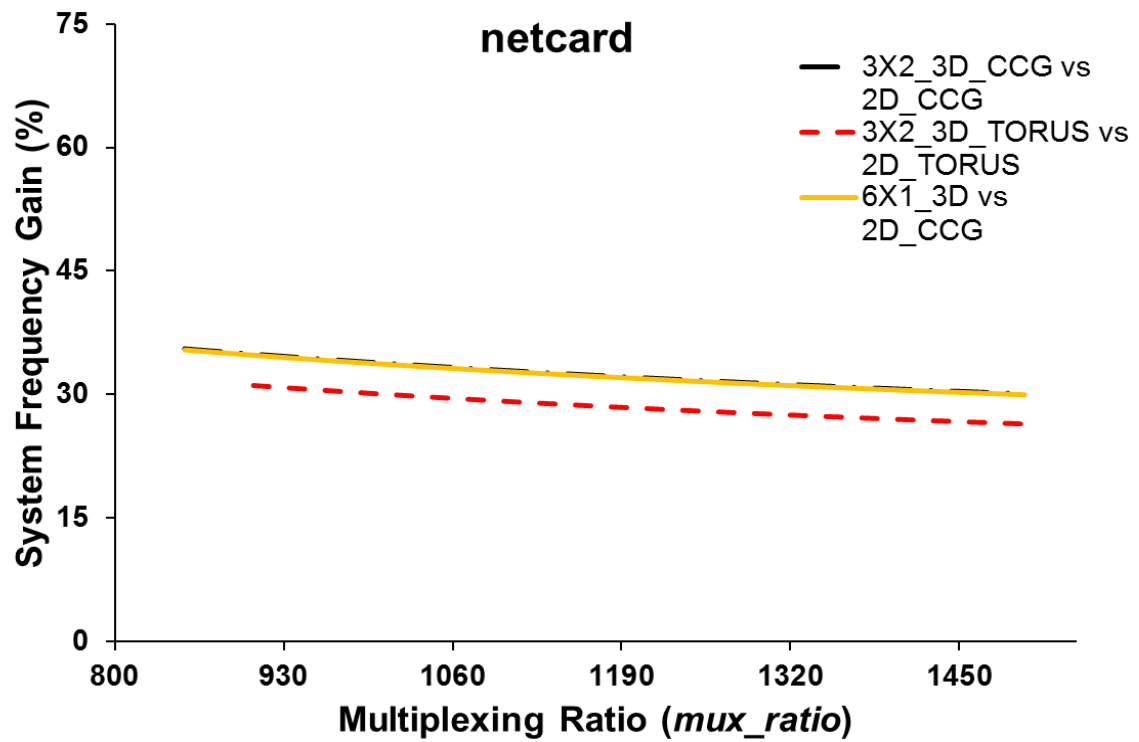


Figure 7.7(c): System Frequency Gain 2D vs 3D with MGT Multiplexing Scheme

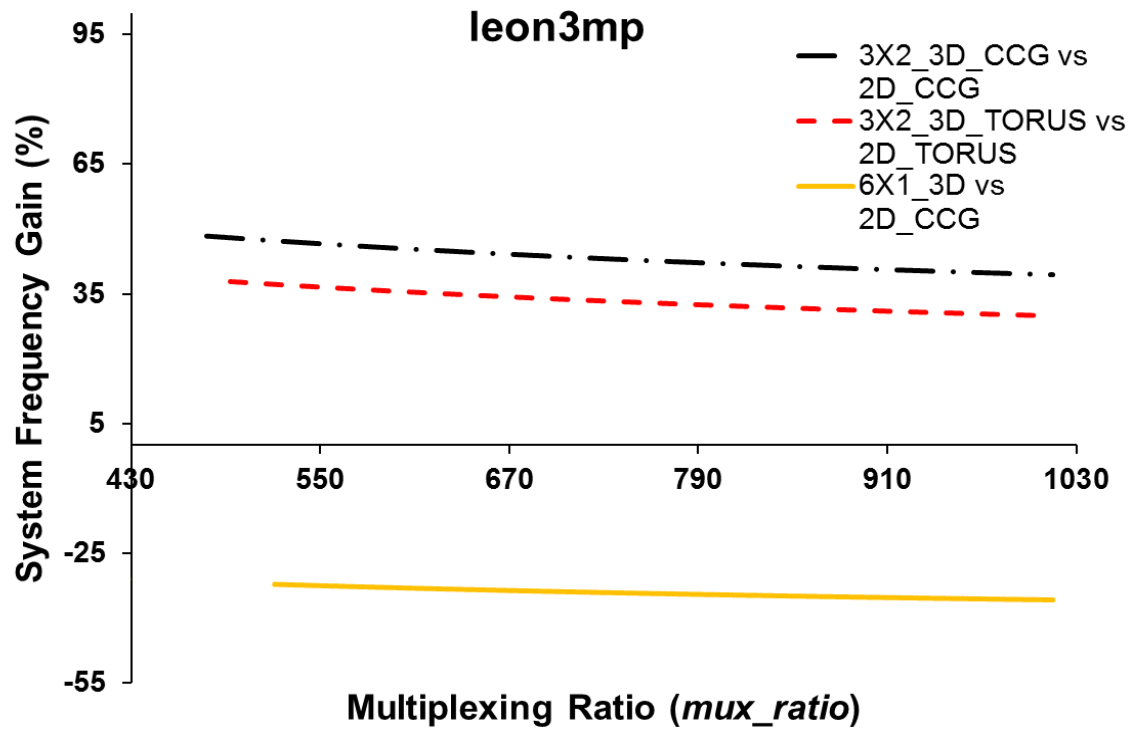


Figure 7.7(d): System Frequency Gain 2D vs 3D with MGT Multiplexing Scheme

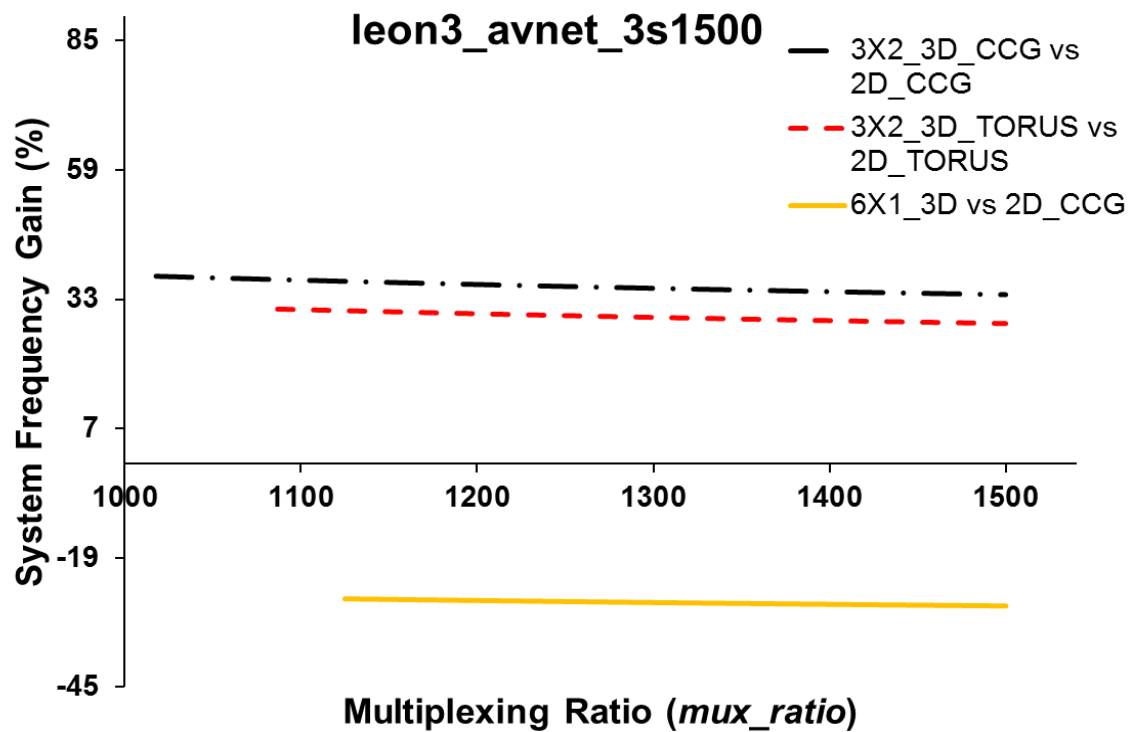


Figure 7.7(e): System Frequency Gain 2D vs 3D with MGT Multiplexing Scheme

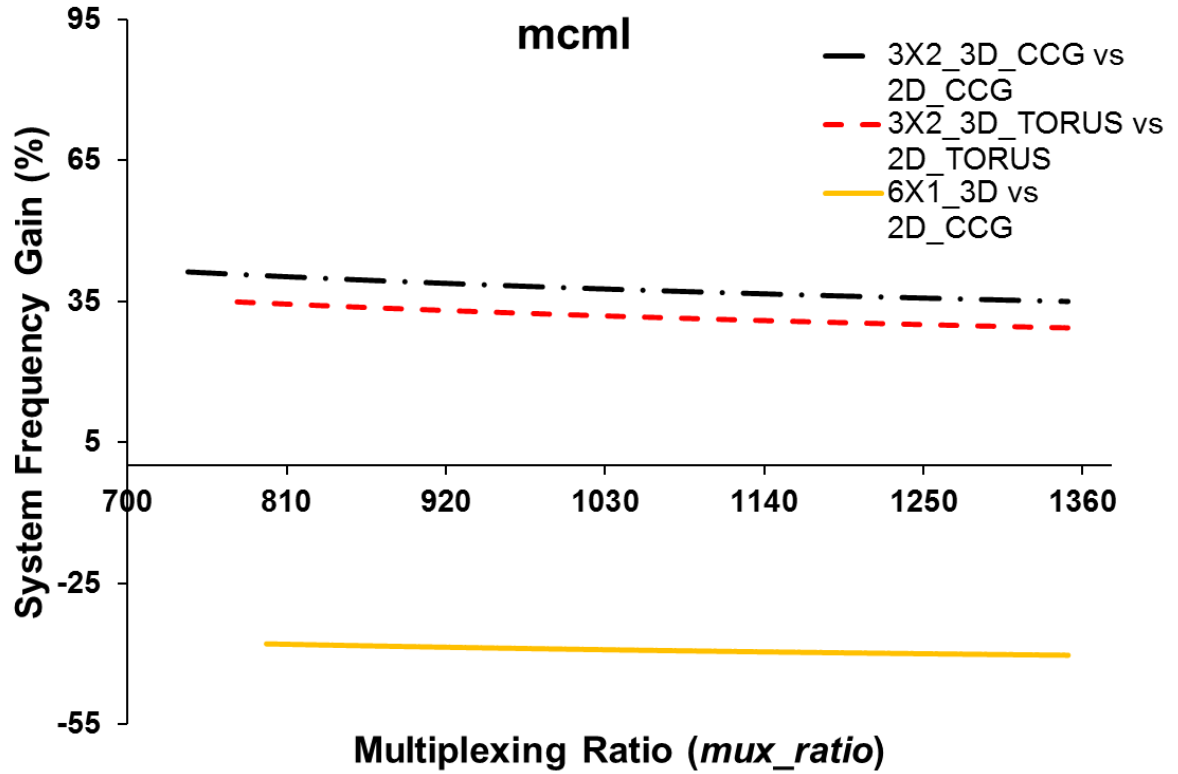


Figure 7.7(f): System Frequency Gain 2D vs 3D with MGT Multiplexing Scheme

Figure 7.8 presents the average frequency gain in 3D platforms when compared to their 2D counterparts. All 3D CCG platforms exhibited an average frequency gain of nearly 37% as compared to 2D CCG architectures, whereas in best case scenario i.e. *vga_lcd* the gain was up to 44%. 3D_3X2_TORUS showed an improvement of 30% on average as compared to 2D_TORUS whereas the best frequency gain was up to 36% in case of *vga_lcd*. 6X1 average performance improvement is conditional to the number of route-throughs and needs intensive and efficient placement in order to prove its significance.

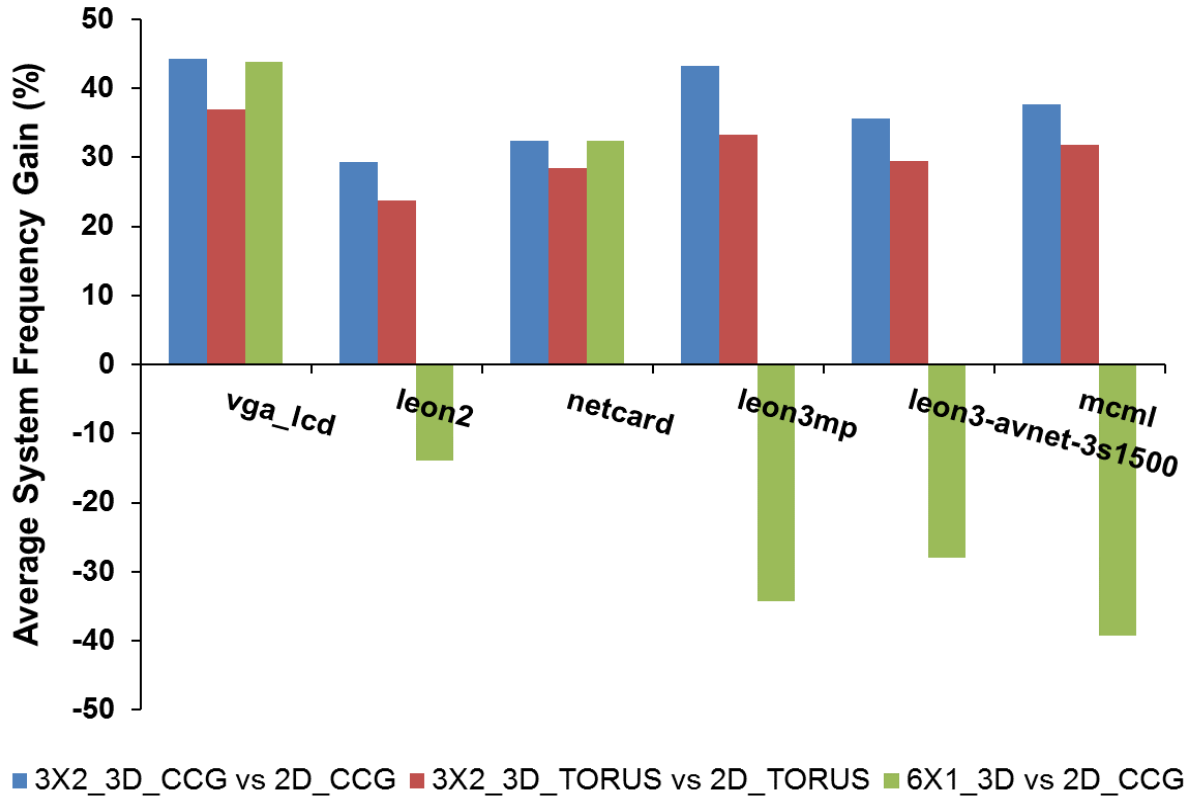


Figure 7.8: Average System Frequency Gain 2D vs 3D in CCG & TORUS Routing Architectures

7.5. Summary

Several time-multiplexed MFS routing architectures were evaluated and compared in this chapter. CAD tools were employed to map real sequential large benchmark circuits. The architectures were compared on the basis of post-routing critical path delay and system frequency metrics. The first section compared the three multiplexing schemes (Logic Multiplexing, SERDES and MGT) for the two routing architectures (CCG and TORUS). SERDES performed better than the other two schemes, whereas CCG showed superior results as compared to TORUS. Next, we compared the proposed 2D latency-optimized MFS and conventional MFS with MGT multiplexing. Based on the presented results, it was shown that the proposed 2D MFS with optical interface exhibited significant performance improvement over the range of multiplexing ratios above the threshold value. Lastly, we evaluated the performance of 3D architectures and compared them with their

2D counterparts. 3D platforms performed better than both 2D conventional and optical MFS routing architectures.

Chapter 8 Conclusions & Future Work

8.1. Dissertation Summary

In this dissertation we proposed 2D and 3D latency-optimized time-multiplexed MFS routing architectures. We used rigorous experimental approach and real sequential benchmark circuits to evaluate and compare the proposed and existing MFS routing architectures. This research provided a new insight into the encouraging effects of using off-chip optical interface and three dimensional MFS routing architectures. New proposed MFS routing architectures using optical links have shown superior performance as compared to the existing architectures.

In Chapter 2, we discussed the different types of inter-FPGA connections and the MFS routing architectures used in this research. The basic assumptions and architectural details of CCG and TORUS routing architectures were presented in detail. We have also thoroughly covered the previous research done on different types of MFS routing architectures in this chapter.

In Chapter 3, the concept of time-multiplexed MFS routing architectures was described and the three multiplexing schemes i.e. Logic Multiplexing, SERDES and MGT were also discussed in detail. We have drawn a comparison among the three multiplexing schemes and presented the previous research.

In Chapter 4, we described the concept of short-ranged optical interface in multi-FPGA systems. The architectural requirements of MFS serial optical interface were discussed in detail. In this chapter we presented the new latency-optimized proposed 2D MFS routing architectures with optical interface. The chapter also covered the previous work done on MFS serial optical interface.

In Chapter 5, optical 3D MFS routing architectures were presented. We have shown that 3D architectures perform better than planar MFS routing architectures based on their interconnection length distribution, asymptotic wire-length behavior and structural

distribution. This chapter also described the proposed 3D time-multiplexed MFS routing architectures. Previous research done in exploring the third dimension in MFS was also presented in this chapter.

In Chapter 6, the experimental evaluation framework and the CAD tools employed to map real sequential benchmark circuits to different 2D and 3D MFS routing architectures were described. The architecture evaluation metrics (post-routing critical path delay and system frequency) were discussed and the benchmark circuits used were also presented. A static timing analyzer developed for measuring the critical path delays of benchmark circuits mapped to different MFS routing architectures was described. Time-multiplexed inter-FPGA router was also described.

Finally, in Chapter 7, evaluation and comparison results and their analysis were presented for the existing and proposed time-multiplexed MFS routing architectures. First, we presented a comparison among the three time-multiplexed MFS built in CCG and TORUS routing architectures. It was shown that SERDES performed better than Logic Multiplexing and MGT, however, for very high multiplexing ratio, the performance of SERDES and MGT became comparable. Next, we compared the proposed latency-optimized 2D optical MFS routing architectures with the existing MFS routing architectures. Post routing critical path delay and system frequency improvements in the proposed MFS architectures were reported in this chapter. Latency-optimized 2D MFS showed approximately 12% average decrease in critical path delay values in CCG, whereas, in TORUS the average decrease is nearly 9% all the benchmark circuits above the threshold multiplexing ratio. Furthermore, all 2D optical CCG platforms exhibited an average frequency gain of 22% as compared to 2D CCG architectures with electrical interconnects. In best case scenario, the performance gain was up to 26%. 2D TORUS optical MFS showed an improvement of up to 18% on average as compared to 2D TORUS with electrical interconnects. In best case scenario 2D TORUS optical MFS system frequency gain was up to 21%. Lastly, we compared the proposed 3D optical MFS routing architectures with conventional planar MFS and showed that achieved system frequency gain is very encouraging. Latency-optimized 3D MFS showed approximately 18% average decrease in critical path delay values in CCG, whereas, in TORUS the average decrease is

nearly 14% all the benchmark circuits above the threshold multiplexing ratio. All 3D CCG platforms exhibited an average frequency gain of nearly 37% as compared to 2D CCG architectures whereas in best case scenario the gain was up to 44%. 3D_3X2_TORUS showed an improvement of up to 30% on average as compared to 2D_TORUS whereas the best frequency gain was up to 36%. 6X1 average performance improvement is conditional to the number of route-throughs and needs intensive and efficient placement in order to prove its significance.

8.2. Principal Contributions

Performance of existing MFS routing architectures is limited by many factors such as limited pin resources, inter-FPGA communication strategy and off-chip interface selection. In order to resolve the problems stated above, the major contributions of this thesis include the following:

- We proposed novel scalable 3D MFS architectures which showed improved system performance compared to conventional 2D MFS architectures. The vertical stacking resulted in shorter off-chip links improving the overall system frequency with the additional advantage of smaller footprint area.
- The proposed 3D architectures employed serialized interconnect between intra-plane and inter-plane FPGAs to address the pin limitation problem. Additionally, all off-chip links are replaced by optical fibers that exhibited latency improvement and resulted in faster MFS. Results indicated that exploiting third dimension provided latency and area improvements as compared to 2D MFS. The experimental results have shown 37% improvement in average system frequency as compared to planar MFS with electrical interconnects whereas the best frequency gain was up to 44%.
- We also proposed latency-optimized planar 2D MFS architectures in which electrical interconnections are replaced by optical interface in same spatial distribution. Performance evaluation and comparison have shown that the proposed architectures had reduced critical path delay and system frequency improvement as compared to conventional MFS. 2D optical platforms exhibited an average frequency gain of 22% as compared to 2D MFS with electrical interconnects whereas the best frequency gain was up to 26%.

- Achieved performance of three time multiplexing schemes; Logic Multiplexing, SERDES and MGT, is compared for a given range of serialization ratio using different routing architectures in planar MFSs with PCB connections.

8.3. Future Directions

FPGAs are used for high performance computing (HPC) and to accelerate high-performance applications on custom computing machines. An acceleration board is an FPGA-based platform which is capable of implementing complex computing tasks with low-latency and high-bandwidth. Many FPGA-based computational acceleration boards are commercially available that offer HPC solutions with Tera-Flop capabilities and scalability capacity for Peta-Flop performance and beyond. “FPGA-based boards are used to accelerate real-time processing in Computational Finance, High Frequency Trading, Computational Physics, Computational Biology, Data Analytics, Encryption/Decryption, Real-time Image, Video Processing, and others” [87].

However, multi-FPGA based acceleration boards are extremely limited and can be considered as a promising and emergent field of research in near future. Advanced Processing Platform: Nallatech 510T [88] is an example of latest application of MFS in acceleration boards. It comprises of 2 Intel Xeon E5-2620 v4 processors and claims to provide a “high-performance multi-FPGA compute accelerator platform for high-performance, low latency, large design capacity, memory bandwidth, and programmability applications” [88].

Employing MFS as computational accelerator for HPC can be explored as an area for future research which comes with multiple challenges. One of them is exploiting suitable MFS routing architectures that can serve as an efficient, fast and low cost target platform for high-level synthesis tools such as Intel SDK for OpenCL. Additionally, in these multi-FPGA acceleration boards, the application of optical interface for inter-FPGA communication can be evaluated for increased system performance. Finally, the aspect of using 3D MFS in accelerator boards for even smaller foot-print and latency-optimization can also be an intriguing subject for future research.

REFERENCES

- [1] Khalid, M. A. S., Rose, J, “A novel and efficient routing architecture for multi-FPGA systems”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 8 Issue 1, pp.30-39, Feb. 2000.
- [2] Dini Group DNVUF4A Product Brief, Ver. 1.02, March 2016 [Online] Available: http://www.dinigroup.com/product/data/DNVUF4A/files/DNVUF4A_brief_v102_lo.pdf , Last Visited: January 2017.
- [3] [Online] Available: <http://sciengines.com/products/computers-and-clusters/riviera-s6-lx150.html>, Last Visited: January 2017.
- [4] Strauch, T. 2011, “Multi-FPGA System With Unlimited and Self-Timed Wave-Pipelined Multiplexed Routing”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 19, Issue 9, pp. 1549-1558, Sept. 2011.
- [5] Amos, D., Lesea, A., Richter, R., “FPGA-Based Prototyping Methodology Manual”, Synopsys, 2011.
- [6] Wayne Wolf, “FPGA-Based System Design”, Pearson Education, 2004
- [7] Li, Mike Peng, “Overcome Copper Limits with Optical Interfaces”, Altera White Paper, 2011.
- [8] Tang, Q., “Methodology of Multi-FPGA Prototyping Platform Generation”, Doctoral dissertation, Université Pierre et Marie Curie-Paris VI, 2015.
- [9] Cadence Protium Rapid Prototyping Platform, 2014 [Online] Available: https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/tools/system-design-verification/protium-rapid-prototyping-ds.pdf, Last Visited: January 2017.
- [10] S2C Quad KU115 Prodigy Logic Module, 2016 [Online] Available: http://www.s2cinc.com/assets/files/Quad_KU115_Prodigy_LM_Datasheet.pdf, Last Visited: January 2017.
- [11] “FPGA Based Rapid Prototyping Platforms for Telecommunication”, BEECube Brochure 2016 [Online] Available: http://www.beecube.com/uploads/6/3/4/9/63495763/beecube_brochure_web.pdf, Last Visited: January 2017.

- [12] HyperSiliconVeriTiger-DH2000TQ Overview [Online] Available: <http://www.hypersilicon.com/ProductShow.asp?id=206>, Last Visited: January 2017.
- [13] proFPGA quad V7Prototyping System, Product Brief [Online] Available: http://www.prodesign-europe.com/profpga/PDF/proFPGA_quadV7_product_brief.pdf, Last Visited: January 2017.
- [14] Asaad, S., et al. "A cycle-accurate, cycle-reproducible multi-FPGA system for accelerating multi-core processor simulation", Proceedings of the ACM/SIGDA International Symposium on Field Programmable Gate Arrays. ACM, pp. 153-162, Feb. 2012.
- [15] Synopsys' HAPS-70 FPGA-based Prototyping Solution [Online] Available: <https://www.synopsys.com/cgi-bin/proto/pdfdl/docsdl/haps70-brochure.pdf?file=haps70-brochure.pdf>, Last Visited: January 2017.
- [16] Van Den Bout, D. E., et al, "Anyboard: An FPGA-Based Reconfigurable System", IEEE Design and Test of Computers, pp. 21-30, June 1992.
- [17] Gokhale, M., et al, "Building and Using a Highly Parallel Programmable Logic Array", IEEE Computer, pp 81-89, January 1991.
- [18] Hauck, S., et al, "Mesh Routing Topologies For Multi-FPGA Systems", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 6, pp. 400-408, 1998.
- [19] Walters, S., "Computer-Aided Prototyping for ASIC-Based Systems", IEEE Design and Test of Computers, Vol. 8, pp. 4-10, June 1992.
- [20] Vuillemin, J.E., et al, "Programmable Active Memories: Reconfigurable Systems Come of Age", IEEE Transactions on VLSI, Vol. 4, No. 1, pp. 56-69, March 1996.
- [21] Babb, J., et al, "Logic Emulation with Virtual Wires", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol.16, No. 6, pp. 609-626, June 1997.
- [22] Guo, R., et al, "A 1024 Pin Universal Interconnect Array with Routing Architecture", Proc. of IEEE Custom Integrated Circuits Conference, pp. 4-5, 1992.

- [23] Dini Group User Manual DN7020K10, March 2012 [Online] Available: http://www.dinigroup.com/product/data/DN7020k10/files/Manual_DN7020K10_RE_V2.pdf, Last Visited June 2017.
- [24] Baxter, R., et al., “Maxwell - a 64 FPGA Supercomputer”, Proceedings of the Second NASA/ESA Conference on Adaptive Hardware and Systems, AHS '07, pp. 287–294, August 2007. IEEE Computer Society.
- [25] Mencer, O., et al., “Cube: A 512-FPGA cluster”, 5th Southern Conference on Programmable Logic, SPL, pp. 51 –57, April 2009.
- [26] A. Putnam et al., “A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services”, IEEE Micro, Vol. 35, Issue 3, pp.10-22, June 2015.
- [27] Hyder, Z., and Wawrzynek, J., “Defect tolerance in multiple-FPGA systems” IEE Proceedings-Computers and Digital Techniques, Vol. 153, Issue 3, pp. 139-145, May 2006.
- [28] Tang, Q., Habib M., and Matthieu T., “Multi-FPGA prototyping board issue: the FPGA I/O bottleneck”, IEEE International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV), 2014.
- [29] Sawyer, N., “LVDS Source Synchronous 7:1 Serialization and Deserialization Using Clock Multiplication”, Application Note: Xilinx 7 Series FPGAs, March 2015.
- [30] Kintex UltraScale+ FPGAs Data Sheet: DC and AC Switching Characteristics, May 2016, [Online], Available: http://www.xilinx.com/support/documentation/data_sheets/ds922-kintex-ultrascale-plus.pdf, Last Visited January 2017.
- [31] UltraScale Architecture GTY Transceivers, User Guide [Online], Available: http://www.xilinx.com/support/documentation/user_guides/ug578-ultrascale-gty-transceivers.pdf, Last Visited January 2017.
- [32] RocketIO Transceiver, User Guide, [Online], Available: http://www.xilinx.com/support/documentation/user_guides/ug024.pdf , Last Visited January 2017.
- [33] Athavale, A., Christensen, C., “High-Speed Serial I/O Made Simple, Designers’ Guide”, Xilinx Inc., 2005.

- [34] Liu. Y. et al., “Building a multi-FPGA-based emulation framework to support networks-on-chip design and verification”, *International Journal of Electronics*, Vol. 97 Issue 10, pp. 1241-1262, Oct. 2010.
- [35] S2C Inc. White Paper, “Choosing the best pin multiplexing method for your Multiple-FPGA partition”, July 2016, [Online], Available: http://www.s2cinc.com/assets/files/Choosing_the_best_pin_multiplexing_method_for_your_Multiple-FPGA_partition.pdf, Last Visited December 2016.
- [36] Aloisio, A., Cevenini, F., Giordano, R. and Izzo, V., “High-Speed, Fixed-Latency Serial Links With FPGAs for Synchronous Transfers”, *IEEE Transactions on Nuclear Science*, Vol. 56, pp. 2864–2873, Oct. 2009.
- [37] Posner, M., “Results from Xilinx UltraScale VU440 based HAPS-80 with HAPS ProtoCompiler”, *Breaking the Three Laws Blog*, Synopys.com, January 2016.
- [38] Jansweijer, P. P. M., Peek. H. Z., “Measuring propagation delay over a coded serial communication channel using FPGAs”, *Nuclear Instruments and Methods in Physics Research. Section A, Accelerators, Spectrometers, Detectors and Associated Equipment* 626.Suppl. 1 (2011): S169-S172.
- [39] Booth, B. et al., “On-Board Optical Interconnection”, CTR III TWG Report #3, MIT Microphotonics Center, April 2013.
- [40] Agrawal, G. P., “Fiber-Optic Communication Systems”, 4th Edition, John Wiley and Sons, New York, 2010.
- [41] Sequeira F, Duarte D, Bilro L, Rudnitskaya A, Pesavento M, Zeni L, Cennamo N., “Refractive Index Sensing with D-Shaped Plastic Optical Fibers for Chemical and Biochemical Applications”, *Sensors*, 16(12):2119, Dec 2016.
- [42] Finisar 10G/1G Dual Rate (10GBASE-SR and 1000BASE-SX) 400m Multimode Datacom SFP+ Optical Transceiver Product Specification October 2016, [Online]. Available: https://www.finisar.com/sites/default/files/downloads/finisar_ftlx8574d3bcv_1g-10g_850nm_multimode_datacom_sfp_transceiver_productspecrevb1.pdf, Last Visited January 2017.

- [43] Li, Mike Peng, Martinez, J., Vaughan, D., “Transferring High-Speed Data over Long Distances with Combined FPGA and Multichannel Optical Modules”, Altera White Papers, pp. 1-6, 2012.
- [44] Jansweijer, P.P.M., Peek, H.Z., “Measuring propagation delay over a 1.25 Gbps bidirectional data link”, National Institute for Subatomic Physics, Amsterdam, 2010.
- [45] Edin, K., “An FPGA Implementation for a High-Speed Optical Link with a PCIe Interface”, PhD Thesis, McGill University, 2011.
- [46] Kuzmin, A., Fey, D., Lohmann, U., “An Integrated into FPGA System for Optical Link Testing and Parameters Tuning”, International Journal on Advances in Systems and Measurements, Vol.7 no1 & 2, pp. 141-149, 2014.
- [47] Gilpatric, M., “Pluggable Optical Interfaces and Their Compatibility with Xilinx FPGAs”, Xilinx White Paper, Dec. 2012.
- [48] Ghiasi, A., “Multi-Vendor Interoperability Testing of CFP2 and QSFP28 with CEI-28G-VSR Interface During OFC 2013 Exhibition”, OIF White Paper, March 2013.
- [49] Deng, B., et al., “JTAG-based remote configuration of FPGAs over optical fibers”, Journal of Instrumentation Vol. 10, 2015.
- [50] Minami, S., Hoffmann, J., Kurz, N. and Ott, W., “Design and implementation of a data transfer protocol via optical fiber”, IEEE Transactions on Nuclear Science, Vol. 58 Issue 4, pp.1816-1819, 2011.
- [51] Verdiell, J.M., “Advances in Onboard Optical Interconnects: A New Generation of Miniature Optical Engines”, In DesignCon 2013, Samtec Inc.
- [52] Morris, K., “Staying Ahead of the Curve - Synopsys Upgrades HAPS”, Electronic Engineering Journal, November 2012.
- [53] Russo, R. L., “On the tradeoff between logic performance and circuit-to-pin ratio for LSI”, IEEE Transactions On Computers, Vol. C-21 Issue 2, pp. 147-153, Feb 1972.
- [54] Marck, H. V., Campenhout J. V., “Three-dimensional optoelectronic architectures for massively parallel processing systems”, Proceedings of the Fourth International Conference on Massively Parallel Processing Using Optical Interconnection, IEEE, pp. 178-182, 1997.
- [55] Stroobandt, D., “A Priori Wire Length Estimates for Digital Design”, Springer Science & Business Media, 2011.

- [56] Stroobandt, D., “Recent advances in system-level interconnect prediction”, IEEE Circuits and Systems Newsletter 19.9 (2000): 4-20.
- [57] Dambre, J., Van Marck, H., Van Campenhout, J., “Quantifying the impact of optical interconnect latency on the performance of optoelectronic FPGAs” in Interconnects in VLSI Design, Springer Science & Business Media, pp. 195-202, 2012.
- [58] DeCusatis, C., “Optical Backplanes Board and Chip Interconnects” In “Fiber Optic Data Communication: Technological Trends and Advances”, pp. 216-269, Academic Press, 2002.
- [59] Vanwassenhove, L., et al., “Demonstration of 2-D plastic optical fiber based optical interconnect between CMOS ICs”, Optical Fiber Communication Conference and Exhibit, Vol. 3. pp. WDD74-WDD74, 2001.
- [60] Brunfaut, M., et al., “A multi-FPGA demonstrator with POF-based optical area interconnect”, 12th Annual Meeting IEEE Lasers and Electro-Optics Society Vol. 2. 1999.
- [61] DiniGroup “LVDS Pin Multiplexing IP”, White Paper, [Online] Available: <http://www.dinigroup.com/web/pinmux.php>, Last Visited: February 2017.
- [62] “FireFly™ Application Design Guide”, Samtec Brochure, January 2017, [Online] Available: <http://suddendocs.samtec.com/ebrochures/firefly-brochure.pdf>
- [63] UltraScale Architecture PCB Design User Guide, [Online], Available: http://www.xilinx.com/support/documentation/user_guides/ug583-ultrascale-pcb-design.pdf, Last Visited August 2016.
- [64] Pavlidis, V., Friedman, E., “3-D topologies for networks-on-chip”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 15, Issue. 10, pp. 1081–1090, Oct. 2007.
- [65] Lin, M., et al., “Performance benefits of monolithically stacked 3-D FPGA”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 26, Issue. 2, pp. 216-229, 2007.
- [66] Saban, K., “Xilinx Stacked Silicon Interconnect Technology Delivers Breakthrough FPGA Capacity, Bandwidth, and Power Efficiency”, Xilinx White Paper, Dec. 2012.

- [67] Pangracious, V., Marrakchi, Z., Mehrez, H., “Physical Design and Implementation of 3D Tree-Based FPGAs” in *Three-Dimensional Design Methodologies for Tree-based FPGA Architecture*, Springer International Publishing, pp. 169-198, 2015.
- [68] Tang, Q., Tuna, M., Mehrez, H., “Design for prototyping of a parameterizable cluster-based Multi-Core System-on-Chip on a multi-FPGA board,” *Proceedings of the 23rd IEEE International Symposium on Rapid System Prototyping*, Tampere, Finland, pp. 71–77, Oct. 2012.
- [69] Khalid, M. A. S., Rose, J., “The Effect of Fixed I/O Pin Positioning on The Routability and Speed of FPGAs”, *Proceedings of The Third Canadian Workshop on Field-Programmable Devices (FPD’95)*, pp. 92-104, 1995.
- [70] Kuznar, R. et al, “Partitioning Digital Circuits for Implementation on Multiple FPGA ICs”, MCNC Technical Report 93-03, North Carolina, 1993.
- [71] “Quick Look under the Hood of ABC A Programmer’s Manual”, Berkeley Verification and Synthesis Research Center, Dec 2006.
- [72] Sanders, P., and Schulz, C., “KaHIP v2.00- Karlsruhe High Quality Partitioning-- User Guide”, 2017. [Online] Available: <https://arxiv.org/pdf/1311.1714.pdf>, Last Visited: May 2017.
- [73] Sherwani, N. A., “Placement, Floorplanning and Pin assignment” in *Algorithms For VLSI Physical Design Automation*, Springer Science & Business Media, 2013, pp. 159-181.
- [74] Sarrafzadeh, M., Wang, M., Yang, X., “DRAGON: A Placement Framework”, in *Modern Placement Techniques*, Springer Science & Business Media, pp. 57-88, 2003.
- [75] Hitchcock, R. B., Smith, G. L., & Cheng, D. D., “Timing analysis of computer hardware”, *IBM journal of Research and Development*, Vol. 26, Issue 1, pp. 100-105, 1982.
- [76] Xilinx “Vivado Design Suite User Guide, Design Analysis and Closure Techniques”, UG906 (v2014.1) May 14, 2014, [Online] Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2014_1/ug906-vivado-design-analysis.pdf, Last Visited: February 2017.

- [77] Altera “TimeQuest Timing Analyzer, Quick Start Tutorial”, 2009 [Online]
Available: https://www.altera.com/en_US/pdfs/literature/ug/ug_tq_tutorial.pdf, Last Visited February 2017.
- [78] Synopsys “PrimeTime Golden Timing Signoff Solution and Environment”, 2016 [Online]
Available: <https://www.synopsys.com/content/dam/synopsys/implementation&signoff/datasheet/s/primetime-ds.pdf>, Last Visited: February 2017.
- [79] “Breakthrough automatic partitioning tool from Flexras Technologies”, EE Times 2012 [Online] Available: http://www.eetimes.com/document.asp?doc_id=1317250, Last Visited: February 2017.
- [80] Xilinx Kintex UltraScale+ FPGAs Data Sheet: DC and AC Switching Characteristics, May 2016 [Online] Available: https://www.xilinx.com/support/documentation/data_sheets/ds922-kintex-ultrascale-plus.pdf, Last Visited: March 2017.
- [81] Xilinx Community Forums, <https://forums.xilinx.com/t5/Timing-Analysis/bd-p/TIMEANB>
- [82] Xilinx Technical Support, *private communication*, June 2016.
- [83] Christoph Albrecht, “IWLS 2005 Benchmarks”, Cadence Berkeley Labs, 2005.
- [84] Luu, J., Goeders, J., Wainberg, M., et. al, “VTR 7.0: Next generation architecture and CAD system for FPGAs”, ACM Transactions on Reconfigurable Technology and Systems (TRETS), Vol. 7, Issue 2, Article 6, June 2014.
- [85] ReFLEX CES [Online] Available: <https://www.reflexces.com/applications/finance>, Last Visited: June 2017.
- [86] Sugesti, E. S., Purnomo S. P., Ramli, K., and Budiardjo, B., “Analysis of Delay Bound in IEEE 802.11 g WLAN over Fiber Networks”, In Proceedings of International Conference on Telecommunication (ICTel), Vol. 5, Issue 1, pp. 27-30. 2009.
- [87] Gidel Computation Accelerators [Online] Available: <http://www.gidel.com/HPC-RC/HPC2-Overview.asp>

- [88] Advanced Processing Platform: Nallatech 510T, March 2017 [Online] Available:
<http://www.cmc.ca/WhatWeOffer/Prototyping/FPGABasedPlatforms/AdvancedProcessingPlatforms/Nallatech510T.aspx>

VITA AUCTORIS

Name: Asmeen Kashif

Place of birth: Lahore, Pakistan

Year of birth: February, 1983

Education: University of Engineering & Technology, Lahore, Pakistan.
(2001-2004) B.Sc. Electrical Engineering

University of Engineering & Technology, Lahore, Pakistan.
(2005-2008) M.Sc. Electrical Engineering

University of Windsor, Windsor, ON.
(2013-2017) Ph.D. Electrical Engineering