

**DESIGN AND OPTIMIZATION OF ASYNCHRONOUS
NETWORK-ON-CHIP**

by

Junbok You

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Electrical and Computer Engineering

The University of Utah

December 2011

Copyright © Junbok You 2011

All Rights Reserved

The University of Utah Graduate School

STATEMENT OF THESIS APPROVAL

This dissertation of Junbok You

has been approved by the following supervisory committee members:

Kenneth S. Stevens, Chair 03/02/2011
Date Approved

Erik Brunvand, Member 03/02/2011
Date Approved

Ganesh Gopalakrishnan, Member 03/02/2011
Date Approved

Chris Myers, Member 03/03/2011
Date Approved

Priyank Kalla, Member 03/02/2011
Date Approved

and by Gianluca Lazzi, Chair of
the Department of Electrical and Computer Engineering

and by Charles A. Wight, Dean of the Graduate School.

ABSTRACT

The bandwidth requirement for each link on a network-on-chip (NoC) may differ based on topology and traffic properties of the IP cores. Available bandwidth on an asynchronous NoC link will also vary depending on the wire length between sender and receiver. This work explores the benefit to NoC performance, area, and energy when this property is used to optimize bandwidth on specific links based on its bandwidth required by a target SoC design.

Three asynchronous routers were designed for implementing of asynchronous NoCs. Simple routing scheme and single-flit packet format lead to performance- and area-efficient router designs. Their performance was evaluated in consideration of link wire delay.

Comprehensive analysis of pipeline latch insertion in asynchronous communication links is performed in regard to link bandwidth. Optimal placement of pipeline latch for maximizing benefit to increase of bandwidth is described.

Specific methods are proposed for performance, area and energy optimization, respectively. Performance optimization is achieved by increasing bandwidth of high trafficked and high utilized links in an NoC, as inserting pipeline latches in those links. Through decrease of bandwidth of links with low traffic and low utilization by halving data-path width, reduction of wire area of an NoC is accomplished. Energy optimization is performed using wide spacing between wires in links with high energy consumption.

An analytical model for asynchronous link bandwidth estimation is presented. It is utilized to deploy NoC optimization methods as identifying adequate links for each optimization method.

Energy and latency characteristics of an asynchronous NoC are compared to a similarly-designed synchronous NoC. The results indicate that the asynchronous

network has lower energy, and link-specific bandwidth optimization has improved NoC performance.

Evaluation of proposed optimization methods by employing to an asynchronous NoC shows achievements of performance enhancement, wire area reduction and wire energy saving.

CONTENTS

ABSTRACT	iii
LIST OF FIGURES	viii
LIST OF TABLES	xi
CHAPTERS	
1. INTRODUCTION	1
1.1 Asynchronous Network-on-Chip	2
1.2 Related Work	4
1.3 Motivation	6
1.4 Dissertation Structure	9
1.5 Contributions	9
2. ASYNCHRONOUS NOC DESIGN	11
2.1 Asynchronous Router Module Designs	12
2.1.1 Switch Module Design	13
2.1.2 Merge Module Design	15
2.1.3 Asynchronous Circuit Design Methodology	17
2.2 Asynchronous Router Design	19
2.2.1 Router Performance Evaluation with Link Wire Length	22
2.2.1.1 Performance Evaluation of Asynchronous Router D1	23
2.2.1.2 Performance Evaluation of Asynchronous Router D2	25
2.2.1.3 Performance Evaluation of Asynchronous Router D3	27
3. PIPELINE LATCH IN ASYNCHRONOUS NOC	29
3.1 Design of 2-phase Linear Controller	29
3.2 Pipeline Latch Impact on Link Bandwidth	29
3.3 Optimal Position of One Pipeline Latch Placement	32
3.3.1 Optimal Position of One Pipeline Latch with Router D1	33
3.3.1.1 Maximum Bandwidth Range of D1_PL1 with Optimal PL	35
3.3.1.2 Estimation of Optimal PL Position in D1_PL1	36
3.3.2 Optimal Position of One Pipeline Latch with Router D2	39

3.3.2.1	Maximum Bandwidth Range of D2_PL1 with Optimal PL	40
3.3.2.2	Estimation of Optimal PL Position in D2_PL1	40
3.3.3	Optimal Position of One Pipeline Latch with Router D3	43
3.3.3.1	Maximum Bandwidth Range with Optimal PL in D3_PL1	43
3.3.3.2	Estimation of Optimal PL Position with Router D3	44
3.3.4	Results of One Pipeline Latch Insertion	44
3.4	Optimal Positions of Two Pipeline Latches Placement	47
3.4.1	Optimal Position of Two Pipeline Latch with Router D1	47
3.4.2	Optimal Position of Two Pipeline Latches with Router D2	52
3.4.3	Optimal Position of Two Pipeline Latches with Router D3	54
3.5	Link BW Comparison with Different PL Configurations	55
3.6	Summary	58
4.	ASYNCHRONOUS NOC OPTIMIZATON	60
4.1	Analytical Model for Link BW Estimation	60
4.2	Performance-Critical Link Optimization: PL Insertion	68
4.3	Area Critical Link Optimization: Narrow Data-Path	71
4.4	Energy-Critical Link Optimization: Double Spacing	78
4.5	Summary	79
5.	EVALUATION	81
5.1	Evaluation Methodologies	81
5.2	Evaluation of Asynchronous NoC with MPEG4 SOC	82
5.2.1	Synchronous Router Design	82
5.2.2	Comparison of Asynchronous and pSELF NoC with MPEG4 Design	83
5.3	TI Design	91
5.3.1	Asynchronous NoC for TI Design	91
5.3.2	Asynchronous NoC Optimization for TI Design	97
5.3.2.1	Performance-critical Link Optimization for TI Design	97
5.3.2.2	Area-critical Link Optimization for TI Design	102
5.3.2.3	Energy-critical Link Optimization for TI Design	106
5.3.2.4	Results of Optimized NoCs for TI Design	107
5.4	Summary	112

6. CONCLUSION AND FUTURE WORK	113
6.1 Conclusion	113
6.2 Future Work	114
REFERENCES	116

LIST OF FIGURES

1.1	Typical asynchronous handshake protocol.	3
1.2	Router dynamic energy per flit, including idle-cycles, with various flit transfer rates.	3
1.3	Link wire length effect on asynchronous communication links throughput compared with synchronous links.	6
2.1	Architecture of a three-port asynchronous router.	11
2.2	Design of switch module.	13
2.3	Timing diagram of 2-to-4 phase converter.	14
2.4	Petri-Net specification of 4-phase linear controller.	14
2.5	Circuit implementation of 4-phase linear controller.	14
2.6	Design of merge module.	15
2.7	Design of MUTEX.	16
2.8	Petri-Net specification of merge controller.	16
2.9	Implementation of merge controller.	17
2.10	Asynchronous circuit design flow.	18
2.11	Router D1.	19
2.12	Router D2.	19
2.13	Router D3.	20
2.14	Handshake cycles in asynchronous communication link.	22
2.15	Handshake cycles in D1 router.	23
2.16	Impact of link wire delay on link BW with router D1.	25
2.17	Handshake cycles in D2 router.	26
2.18	Impact of link wire delay on link BW with router D1 and router D2.	26
2.19	Handshake cycles in router D3 design.	27
2.20	Impact of link wire delay on link BW with router D1, D2 and D3.	28
3.1	Design of 2-phase linear controller.	30
3.2	PL insertion and handshake cycles.	30
3.3	Link of D1 router with a PL: <i>D1_PL1</i>	31

3.4	Impact of link wire delay on link BW with router D1 and one PL.	31
3.5	Wire length of <i>hc_2</i> and <i>hc_3</i> in <i>D1_PL1</i>	33
3.6	PL impact on link throughput in total 2.0 mm link wire.	34
3.7	PL impact on link throughput in total 1.0 mm link wire.	36
3.8	Link BW improvement in a link with D1 routers and one optimal PL. .	39
3.9	Link of D2 router with one PL: <i>D2_PL1</i>	40
3.10	Link BW improvement of <i>D2_PL1</i> with optimal PL placement.	42
3.11	Link of D3 router with a PL: <i>D3_PL1</i>	43
3.12	Link BW improvement of <i>D3_PL1</i>	45
3.13	Link BW of three PLno and three PL1_opt links.	46
3.14	Link of D1 router with two PLs: <i>D1_PL2</i>	47
3.15	Three PL2 Cases depending on Total WL.	49
3.16	Link BW improvement of a link with router D1 and two optimal positioned PLs.	51
3.17	Link of D2 router with two PLs: <i>D2_PL2</i>	52
3.18	Link BW improvement with two optimal PLs in D2 link.	53
3.19	Link of D3 router with two PLs: <i>D3_PL2</i>	54
3.20	Link BW improvement of <i>D3_PL2</i>	55
3.21	BW comparison of three links of <i>TYPE 2</i>	56
3.22	BW comparison of three links of <i>TYPE 3</i>	58
4.1	Flows of <i>input i</i> and other two related inputs, <i>input k</i> and <i>input j</i> in a three-port router	61
4.2	NoC example with traffic pattern for BW estimation model without stall condition	65
4.3	Simulation result of <i>R0_C_O</i> link BW without stall condition.	66
4.4	NoC example for BW estimation with stall conditions.	66
4.5	Simulation result of BW estimation with stall condition	68
4.6	NoC example with PL insertion for performance optimization: NoC_PL	69
4.7	NoC performance comparison between NOC_Init and NOC_PL.	70
4.8	Usage of NDP modules.	72
4.9	NDP_NW.	73
4.10	NDP_WN.	73
4.11	Link BW reduction by NDP module insertion.	75
4.12	NoC example for NDP optimization method: NOC_NDP_PLno	76

4.13	Simulation result for BW estimation	77
5.1	Implementation of switch and merge modules for pSELF router design	83
5.2	MPEG4 CTG graph. Edge weights are in MBytes/s.	84
5.3	MPEG4 network topology.	85
5.4	Available BW (<i>avBW</i>) and traffic load (<i>Load</i>) of 14 links in the asynchronous, pSELF 1.78G and pSELF 2.07G NoCs in 4× offered traffic load.	86
5.5	Link utilization in the asynchronous, pSELF 1.78G and pSELF 2.07G NoCs in 4× offered traffic load. <i>acBW</i> is an achievable link BW, and <i>Load</i> is traffic load of each link labeled on X-axis.	87
5.6	Average latency comparison between the asynchronous and pSELF networks in various offered loads.	88
5.7	Energy distribution at 1×, 2×, 3× and 4× offered loads.	89
5.8	EDP comparison between four NoC designs in various offered loads.	91
5.9	TI example network topology. PEs are in rounded-square boxes and routers in square boxes, numbers are link wire lengths in μm	92
5.10	Comparison of asynchronous NoCs in energy and average latency with TI example.	93
5.11	EDP of asynchronous NoCs with TI example.	94
5.12	Available BW and achievable BW of the most utilized links in <i>Type 2</i> designs.	95
5.13	Performance-critical links in <i>D3_PLno</i>	98
5.14	Strategy of PL insertion in <i>D3_PL_OPT</i>	99
5.15	PL insertion in <i>D3_PL_OPT</i> design.	99
5.16	<i>D3_PL_OPT</i> design improvement in acBW and path average latency.	100
5.17	<i>D3_PL_OPT</i> design improvement in energy, latency and EDP.	101
5.18	AvBW and acBW reduction by NDP module in five sender links and five receiver links.	105
5.19	AcBW comparison between all D3 designs in the most utilized and the least utilized links	109
5.20	Five D3 designs comparison.	111

LIST OF TABLES

2.1	Design results of three asynchronous routers	21
3.1	Optimal PL position of <i>D1_PL1</i> link up to 2000 μm total wire length . .	38
3.2	Optimal PL position of <i>D2_PL1</i> link up to 2000 μm total wire length . .	42
3.3	Comparison of three PLno and three PL1_opt links.	47
3.4	Eight asynchronous link designs with different routers and PL numbers.	55
4.1	Design summary of NDP modules: 32-bit data and 2-bit address in wide data-path.	74
4.2	Reduction of wire area and acBW by NDP modules.	76
4.3	Comparison of SSPACE and DSPACE links with 34-bit link width. . . .	79
5.1	Asynchronous D1 router and synchronous router design summary.	83
5.2	17 Paths which most contribute NoC average latency.	97
5.3	<i>D3_PL_OPT</i> design result comparison.	102
5.4	Routing area of links with NDP module.	104
5.5	<i>D3_PL_NDP</i> design result comparison.	106
5.6	Wire energy ratio of 23 DS links to total wire energy consumption. . . .	107
5.7	Design summary of five NoCs.	108
5.8	Design summary: wire area and energy comparison.	110

CHAPTER 1

INTRODUCTION

More multicore and heterogeneous IP cores can be integrated in a System-on-Chip (SoC) thanks to the ever decreasing feature size in deep sub-micron (DSM) technology. Likewise, core to core communication is getting more complicated and is now a dominant factor in determining SoC performance [1].

Network-on-Chips (NoCs) are segmented where shared on-chip interconnect systems are based on a network topology. Because of their multiple concurrent communications, NoCs have become a preferable interconnect solution for many SoC designs by replacing the traditional SoC interconnect structures of shared buses and point-to-point links, which are limited by their scalability. NoCs provide increased communication capability, such as low latency, high throughput and power efficiency [2, 3].

An NoC primarily consists of routers, network adapters and physical links. The router is an intermediate node in the path of data from a source IP to a destination IP. Network adapters are interface circuits adapting communication between IP cores and an NoC through protocol conversion, synchronization and packetization. Physical links are global wires of communication links. The NoC architecture and implementation are influenced by several design parameters which include topology (mesh, butterfly, torus, tree, and irregular), routing (centralized, source, and distributed) and switching (circuit, packet) schemes, flow control (buffering, virtual channel), and others.

Design-time specialization is one of distinct facets of NoC designs. Unlike micro-networks which focus more on general-purpose communication and modularity, NoC designs can be specialized with their own design restrictions. SoCs can be classified by their application domains into general-purpose on-chip multiprocessor (MPSoC),

application-specific SoCs, and platform SoCs [1]. The MPSoC is commonly built with a homogeneous set of processing units and memory systems to support various applications, usually with no domain boundary. The application-specific SoCs are dedicated to a specific application. They are composed of domain-specific hardware accelerators along with processors and controllers. The platform SoCs are intended for a family of application in a specific domain. Thus, the platform SoCs can be used in a larger variety of applications. Meanwhile, they also contain some domain-specific coprocessors like application-specific SoCs. These specialized application domains, in particular application-specific SoCs and platform SoCs, lead to specific traffic patterns. Therefore bandwidth (BW) requirements of all or some links are available when an NoC is designed. This prior information can be used effectively to design NoCs for SoCs.

Globally Asynchronous and Locally Synchronous (GALS) design is being increasingly used for SoC implementation. In a GALS system, each timing domain is locally clocked and an asynchronous communication scheme is used for communication between timing domains. This trend of asynchronous communication in NoCs is derived from three important factors [4, 5]. First, in the emerging era of nanotechnology, the clock frequency is increasing which exacerbates the difficulty of achieving global clocking across the entire chip. Second, each IP in a SoC has its own optimal operating frequency, so redesigning the IPs for a global clock frequency is inefficient for SoC performance. Finally, increasing energy consumption of the clock buffer and clock tree is a growing concern. Additionally, asynchronous NoCs have several profitable aspects, such as no need for global clock distribution, zero dynamic power consumption, fast forward latency, and robustness to variations when compared to synchronous NoCs.

1.1 Asynchronous Network-on-Chip

Data communication in an asynchronous link is based on a handshake protocol between a sender and receiver. Without global signaling for a data validity, like a clock signal in synchronous systems, asynchronous communication is executed by special signals, typically *request* and *acknowledge* signals, seen in Figure 1.1, for representing

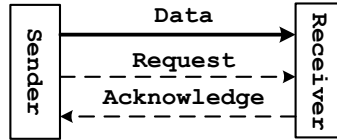


Figure 1.1: Typical asynchronous handshake protocol.

the data validity. The sender generates the request signal to notify that new data are ready, and the receiver responds with the acknowledge signal indicating that the data have been safely stored and to let the sender start a new operation with the next data.

There are two traditional types of the asynchronous handshake protocols: 2-phase Non-Return to Zero (NRZ) and 4-phase Return to Zero (RZ). The 2-phase NRZ protocol uses signal transitions, and this protocol does not necessitate that the two control signals to be returned to zero after each transfer. Two signal transitions, one on the request and the other on the acknowledge signal, are necessary for one data transfer. In the 4-phase RZ protocol, control signals are detected based on their levels and all signals need to be returned to zero after each data transfer.

Fig. 1.2 reports dynamic energy per flit, including the idle-cycle energy, used to send the same amount of data at various rates. One asynchronous router, *Async*, is compared against one synchronous router operating with a 1 GHz, 2 GHz, and 2.90 GHz clock. As can be seen, the asynchronous router uses the same dynamic

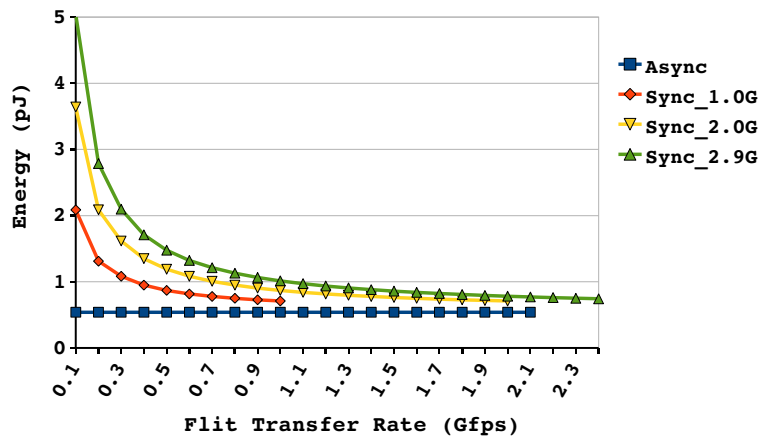


Figure 1.2: Router dynamic energy per flit, including idle-cycles, with various flit transfer rates.

energy per flit regardless of the link idle times. However, dynamic energy per flit in clocked routers is sensitive to the ratio of active versus idle cycles. As clock frequency increases and the flit transfer rate decreases, more aggregate energy is consumed by the clock gating logic in the idle time. When the flit transfer rate equals the clock frequency of the synchronous router, there is no energy consumed by idle time clocking. Simulation results are based on the design of the asynchronous router and synchronous router presented in Section 2.2 and Section 5.2.1.

1.2 Related Work

NoC designs are presented mainly in terms of their router architecture, support for Guaranteed Service (GS) and Best Effort (BE), virtual channel (VC) [6] implementation, handshake protocol, and performance evaluation.

MANGO is an asynchronous router which supports both GS and BE packet transfer [7, 8]. Connection-oriented GS transfer is accomplished through VC links, while BE traffic uses connectionless transfers with source-routing. BE packets are also used for programming GS connections. Two types VC implementation, lock-based and credit-based, are proposed [9, 10]. The lock-based VC is used in a GS connection due to its simple circuitry, while BE connections employ a credit-based protocol for high performance. A MANGO router with 5-bidirectional port and 33 bit data-path width was implemented using the 4-phase bundle-data (BD) asynchronous handshake protocol in 130 nm technology. The maximum BW is 650 Mflits/s (Mfps).

CHAIN was developed at the University of Manchester and uses delay insensitive 1-of-4 encoding [11]. It services BE packets with source routing and wormhole switching. Its network fabric is composed of steering blocks and arbiter blocks with separate command and response paths in an irregular network topology. The startup company, Silistix, sells NoC design solutions including circuits and EDA tools based on the CHAIN implementation.

QNOC is also an asynchronous router design aimed at quality of service with multiple service levels using a 2-dimensional, priority VC implementation and dynamic VC allocation [12, 13, 14]. A 4-phase BD protocol is used for the router's internal

and external links. A router implementation in 180 nm technology has throughput of 205 Mfps.

Both ANOC and DSPIN are designed for the FAUST architecture, which is a SoC platform for telecommunications [15]. They commonly use wormhole switching and a 5-port router architecture and target a mesh topology. The ANOC is an asynchronous NoC with source routing, and was implemented using the STMicroelectronics standard cell and the TIMA TAL library [16]. A quasi delay-insensitive 4-phase protocol is applied to its asynchronous circuit design. DSPIN is a synchronous router using an x -first routing algorithm. To handle clock phase skew in communication between routers, bisynchronous FIFOs are used [17]. Comparison between the two implementations shows that DSPIN has 33% less area than ANOC, but ANOC consumes 37% less power.

The AETHREAL NOC developed by Phillips provides both GS and BE. GS packets have connection-oriented guaranteed throughput and latency by a time-division multiplexed circuit switching approach, while BE packets are transferred through a round-robin arbitration scheme [18].

Communication link properties are critical issues in NoC designs in that the wire-delay of links increases relative to gate-delay and becomes more significant for communication performance as technology scales down. Power and performance of communication protocols have been modeled and compared, including 4-phase and 2-phase asynchronous handshake protocols, delay-insensitive encodings, and clocked communications [19, 20]. Using analytical models, the properties of each protocol are compared in energy and bandwidth. A simple latency model for the asynchronous handshake channel with long wires and twin request/acknowledge control scheme to increase throughput of asynchronous communication links are presented in [21]. In [22], a link capacity allocation algorithm of application-specific NoCs is presented. An analytical packet delay model for wormhole switching is developed and its realization using QNOC router through controlling the number of VC is described as well.

1.3 Motivation

Asynchronous communication links inherently possess an unfavorable property when used in NoC design, especially in view of performance. In a network chip, the largest delay is commonly associated with the time-of-flight of a signal down the communication link. Therefore, the maximum cycle time of a 2-phase asynchronous handshake on a communication link is limited in performance by the propagation delay of the request and acknowledge signals. This can reduce the bandwidth of that link by almost a factor of two. In the 4-phase RZ protocol, four times flight for signal exchange are inevitable for one data transfer, and this protocol has a four times wire-delay penalty. Consequently, selecting asynchronous links in NoC designs could be the wrong design decision since they easily show less communication link performance, due to their two- or four-times wire-delay penalty when compared to synchronous links. Figure 1.3 depicts how asynchronous communication performance degrades with increasing wire length due to handshake control signal propagation delay. The degradation is due to the overhead of the transit time of acknowledgment signal from the receiver to the sender in a 2-phase protocol. On the other hand, the throughput of the synchronous routers is not changed by the link wire length since it is determined only by its clock frequency. However, the synchronous links do have

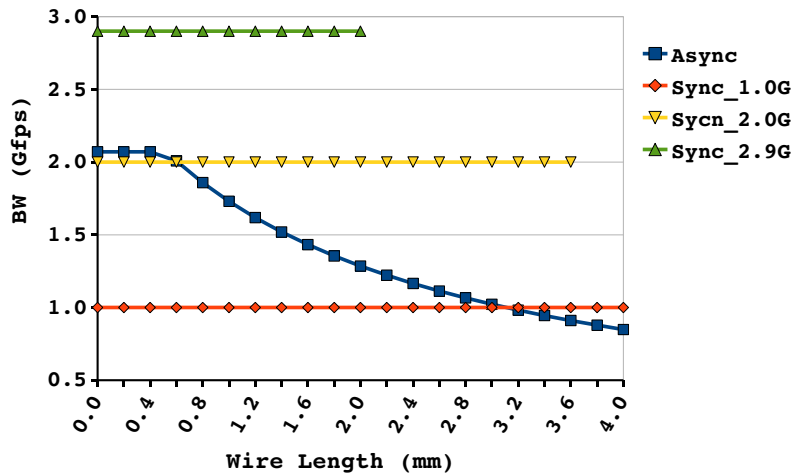


Figure 1.3: Link wire length effect on asynchronous communication links throughput compared with synchronous links.

a maximum wire length that can be supported for any given frequency with which they can operate without link pipelining.

Nevertheless, this wire-delay property of asynchronous communication links gives NoC designers a new distinct design flexibility. As mentioned previously, the BW of an asynchronous link is impacted by the link wire-delay. In other words, the available BW of an asynchronous link can be controlled by the link wire-delay as inversely proportional to the wire-delay between two asynchronous controllers. Placing controllers in closer proximity increases the link BW by reducing the link wire-delay. Therefore, the BW of each asynchronous link can be adequately specified based on its respective requirement by simply adjusting controller locations. This enables one to design a link BW optimized NoC exploiting specialized link BW requirements, and leads to an NoC design with the multibandwidth link property, which synchronous NoCs are unable to easily provide.

In synchronous NoCs, link BW is primarily determined by clock frequency, so all links basically have the same BW in accordance with the global clock frequency. Although it is possible to realize synchronous NoC links with different bandwidths, e.g. using a different data-path width per each link or a different clock frequency in each router, the realization might be neither simple nor efficient enough to expect any benefit.

Customizing link BW in an asynchronous NoC may be best leveraged when link BW requirements are known at NoC design time. Specialized functionality of application-specific SoCs makes it possible for the BW requirements of all links to be known at NoC design time. In most cases, the required BW of one link is not identical to the others. Some highly trafficked links need to have much higher BW, while other links have very low network traffic. Thus, if each link BW of an NoC is set respectively as much as required, it is substantially beneficial to the effective NoC design by meeting the performance requirement while minimizing hardware resources (power and area).

From a different point of view, the multibandwidth property of asynchronous links can be considered as an efficient method for traffic congestion resolution in NoCs. Traffic congestion occurs when the requested data transfer rate exceeds the

capacity of the shared resource, the available link BW, and significantly affects NoC performance. Increasing available BW on congested links can relieve congestion which results in better NoC performance. The ability to control an asynchronous link BW can address the congestion problem at the physical link level in a much simpler way than with synchronous NoCs. On the contrary, because all links in synchronous NoCs operate at the global clock speed, the frequency may become escalated by the most highly congested link. This causes wasteful usage of design resources. The majority of links in an NoCs are much less congested than the highest congested link, so it is of little value to increase the BW of those links.

Interestingly enough, congestion resolution can be seen as an example of the average-case operation of asynchronous circuits versus the worst-case operation of synchronous circuits. The average-case operation of an asynchronous circuit design is generally recognized as one of its advantages, as compared with a worst-case operation of the synchronous design where a clock frequency is limited by the critical, longest logic delay of a design. This average case feature has previously been applied only to functional logic block designs, and not directly applied to communication links in NoC designs since there is no functional block (combinational logic) in the links. However, the worst-case operation characteristic of synchronous systems also exists in synchronous NoCs which originates from network traffic congestion rather than any functional block design complexity, due to the fact that all link BWs are set to the same value as the highest congested links. Contrarily, asynchronous links can realize the average-case performance in NoCs by tuning the BW of only the congested point without affecting other points unnecessarily.

Many recently proposed asynchronous NoC prototype implementations have focused on the performance of an asynchronous router architecture itself, but there is no known published research addressing the multibandwidth property of asynchronous links and exploiting its distinct benefits.

The primary goal of my dissertation research is to analyze and characterize the properties of asynchronous communication links and consequently, to achieve efficient NoC design implementations in which link bandwidths are respectively optimized.

1.4 Dissertation Structure

Chapter 2 presents designs of a proposed asynchronous NoC. Three different asynchronous router designs are introduced and their performance compared against link wire delay.

Chapter 3 introduces the benefits of pipeline latches in asynchronous communication links. Optimal placement of pipeline latches for further improvement of link bandwidth is discussed.

Chapter 4 describes asynchronous NoC design optimization. Three optimization methods are presented for performance, area, and energy improvement respectively. An analytical model for link bandwidth estimation is also presented.

Chapter 5 demonstrates advantages of asynchronous NoCs by the evaluation of two example NoC designs. Features of the asynchronous NoC are presented by comparison with a similarly-designed synchronous NoC. The optimization methods proposed in Chapter 4 are applied to an NoC design and their benefits are presented.

Finally, Chapter 6 summarizes optimization of the asynchronous NoC design and results, and areas of further research are discussed.

1.5 Contributions

The major contributions of this dissertation include the following:

1. An asynchronous NoC was designed for efficiency and simplicity. Thanks to simple routing scheme and single-flit packet format, it achieves performance- and energy-efficient asynchronous router design.
2. Three distinctive asynchronous routers were designed and presented their properties in conjunction with link wire delay impact on their performance. Many asynchronous routers have been implemented in several research studies and generally the maximum throughput of router designs were discussed. However, the maximum throughput of an asynchronous router is only valid with no link wire delay and thus, restrictive. Designing asynchronous routers with consideration of link wire delay is necessary and results in better router performance in wider range of operation condition (link wire length).

3. The primary advantage of asynchronous communication, that is, customizing individual link BW based on its requirement by simply adjusting controller location, was exploited for asynchronous NoC design optimization.
4. Benefits of pipeline latch insertion on asynchronous communication links are presented. Usefulness of pipeline latches in asynchronous communication link is widely known in asynchronous circuit domain. But, this work is distinctive in that optimal position of pipeline latches for maximizing its benefit was proposed, and detailed analysis of advantages of pipeline latches in regard to managing link BW of asynchronous NoCs was presented.
5. Analytical link bandwidth model was proposed for an NoC composed of three-port routers. NoC design optimization can be expected only when adequate optimization method is applied to proper links. Accordingly, it is necessary to identify properties of each link in an NoC.
6. Three specific optimization methods were proposed for performance improvement, wire area reduction and saving wire energy consumption, respectively. The realization of NoC design optimization was presented using an SoC design.

CHAPTER 2

ASYNCHRONOUS NOC DESIGN

The NoC design introduced here is intended for *efficiency through simplicity*. To achieve this, a somewhat unconventional set of parameters is chosen including: a) simple source-routing, b) single-flit packet and c) simple high throughput and low latency network router design. The router, the main component of the NoC, is composed of three switch and three merge modules, as shown in Fig. 2.1. Each switch and merge module has one set of latches providing a 1-flit buffer on each input and output port. Note that from the second design parameter, a single-flit packet, there is no difference between a ‘flit’ and a ‘packet’ and they are used interchangeably in this thesis.

The switch directs a flit to one of two output ports. With bidirectional channels, this results in a three-ported “T” router. The packet format consists of a single flit containing source-routing bits in parallel, on separate wires, with the data bits. The packet is switched through a simple demultiplexer controlled by the most-significant routing bit. The address bits are simply rotated, or swizzled, for the output packet to place the next routing bit in the most significant position. The number of required routing bits is determined by the maximum hop count of a network generated for a specific SoC design. The flit width must be determined based on required throughput,

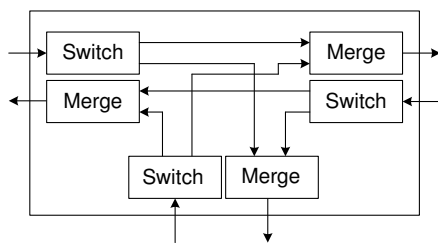


Figure 2.1: Architecture of a three-port asynchronous router.

power, and area constraints. This format has the overhead of requiring routing bits with every flit.

The merge module arbitrates between two input channels to an output channel, granting access to the first-to-arrive request signal. This effectively alternates between the two input channels, assuming each provides the next packet within an output channel’s cycle-time.

This simplicity of the router produces some interesting trade-offs. The simple routing logic has such a low latency that single-flit packets may be advantageous. These packets greatly reduce buffering requirements at each router node. There is no need to have extra logic to calculate packet lengths, and no need to set up or free routes beforehand. Links will only be blocked if all the buffers are full at a router, and streams sharing links will be interleaved.

2.1 Asynchronous Router Module Designs

Asynchronous protocols normally fall into two categories: quasi delay-insensitive (QDI) and bundled-data (BD). Generally, QDI is more robust to variations while BD allows simpler circuits. BD has a lower wire count compared to QDI’s common encodings (e.g., 1-of-4 and dual-rail). This is potentially more energy-efficient due to reduced wire repeater leakage, especially with wide links [20]. The choice of 4-phase or 2-phase protocol impacts performance and circuit complexity.

The throughput across long links is limited by link wire delay, and thus a 2-phase protocol achieves almost twice the throughput as a 4-phase protocol thanks to half the total time-of-flight link delay per transition. However, a 4-phase, level-sensitive protocol typically allows more simple circuits. In particular, MUTEX elements for arbitrating the shared output channels are level sensitive 4-phase circuits [23, 24]. Thus, the internal control logic of these asynchronous NoC routers is best designed using a 4-phase protocol.

With this in mind, the asynchronous router was designed to internally operate using a BD 4-phase protocol, while a BD 2-phase protocol is used on links between routers.

2.1.1 Switch Module Design

The design of the router's switch module is shown in Figure 2.2. A 2-to-4 phase converter ($2\text{-}4\text{ conv}$) is implemented on the input control channel (signals lr and la). This handshakes with a BD 4-phase burst-mode asynchronous controller (LC_{4p}) to pipeline the data with a data latch (DL). The output request is steered to one of two channels ($rr1$ or $rr2$) based on the most significant route bit with a demultiplexer (sw_demux). The route-bits are rotated and passed to the merge module of the router. The routing logic occurs concurrently with the handshake.

The 2-to-4 phase converter was designed manually and its timing diagram is shown in Figure 2.3. The 2-phase signals, lr and la are converted to a 4-phase protocol on wires lr_m and la_m which are inputs to the 4-phase linear controller.

The linear controller connected with the 2-to-4 phase converter has the same specification and timing assumptions as the one used in [25]. Its specification is shown in Eq. 2.1 as a CCS process logic [26] and Figure 2.4 with Petri-Net [27] where the RTC indicates a *relative timing constraint* that enables a specific timing optimization for this asynchronous circuit [25]. The circuit implementation of the linear controller is presented in Figure 2.5.

$$\begin{aligned}
 \text{LEFT} &= \underline{l_r}.c1.la.c2.\underline{l_r}.la. & \text{LEFT} \\
 \text{RIGHT} &= \underline{c1}.rr.c2.\underline{ra}.rr.\underline{ra}.\text{RIGHT} \\
 \text{LC} &= (\text{LEFT}|\text{RIGHT}) \setminus \{c1, c2\} & (2.1)
 \end{aligned}$$

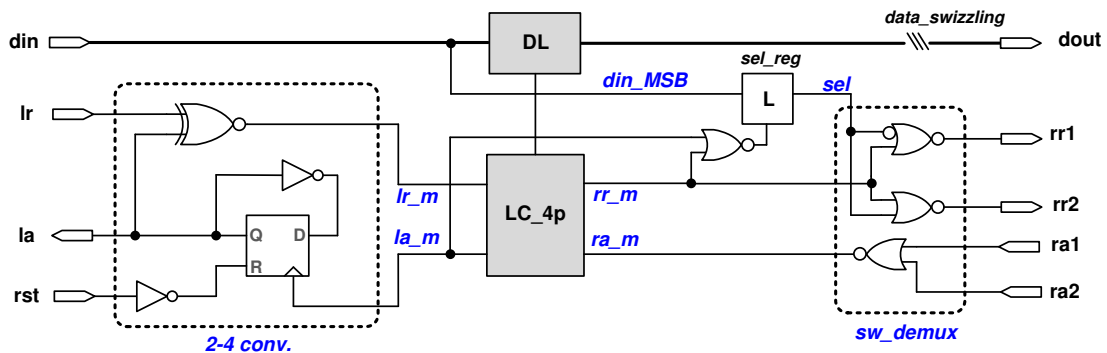


Figure 2.2: Design of switch module.

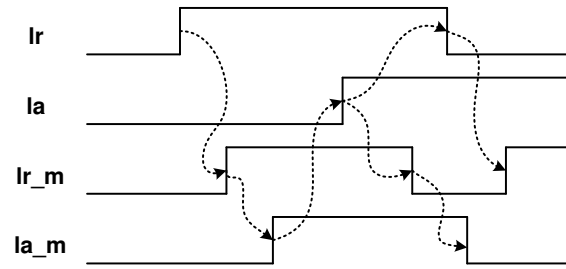


Figure 2.3: Timing diagram of 2-to-4 phase converter.

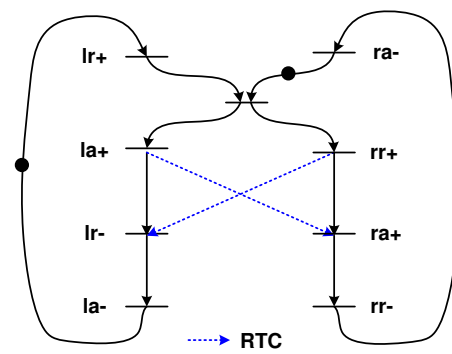


Figure 2.4: Petri-Net specification of 4-phase linear controller.

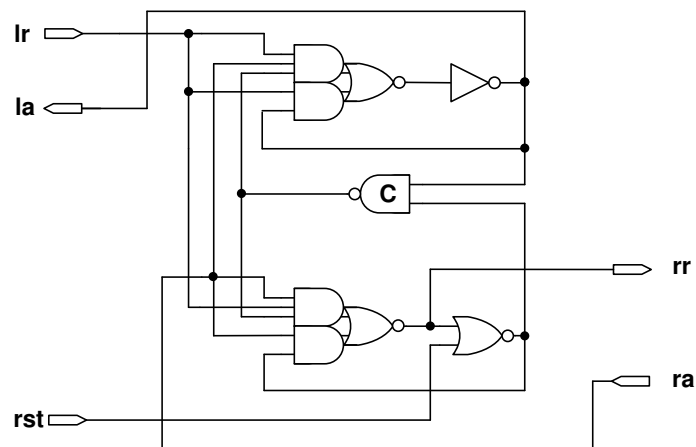


Figure 2.5: Circuit implementation of 4-phase linear controller.

2.1.2 Merge Module Design

The merge module is composed of the arbitration circuit (*ar_ckt*) and merge controller (*mg cont*) shown in Figure 2.6. The arbitration circuit contains a MUX element that serializes requests to the shared output channel. The output of the MUX element also controls a MUX that selects which input data to store in the output latch. Each transaction of the arbitration circuit requests a data transfer via the 4-phase handshake signal *lr_m*. This request passes through the merge controller to generate the 2-phase network link handshake on signals *rr* and *ra*, as well as store the data in a data latch.

The MUX element is a special cell which is not part of the standard cell library used for the circuit implementation. Thus, a specific MUX design in [24] (shown in Figure 2.7) was characterized as a separate library cell through manual layout and HSPICE simulation.

The merge controller was specified in CCS (Eq. 2.2) and by Petri-Net (Figure 2.8). The circuit implementation is shown in Figure 2.9.

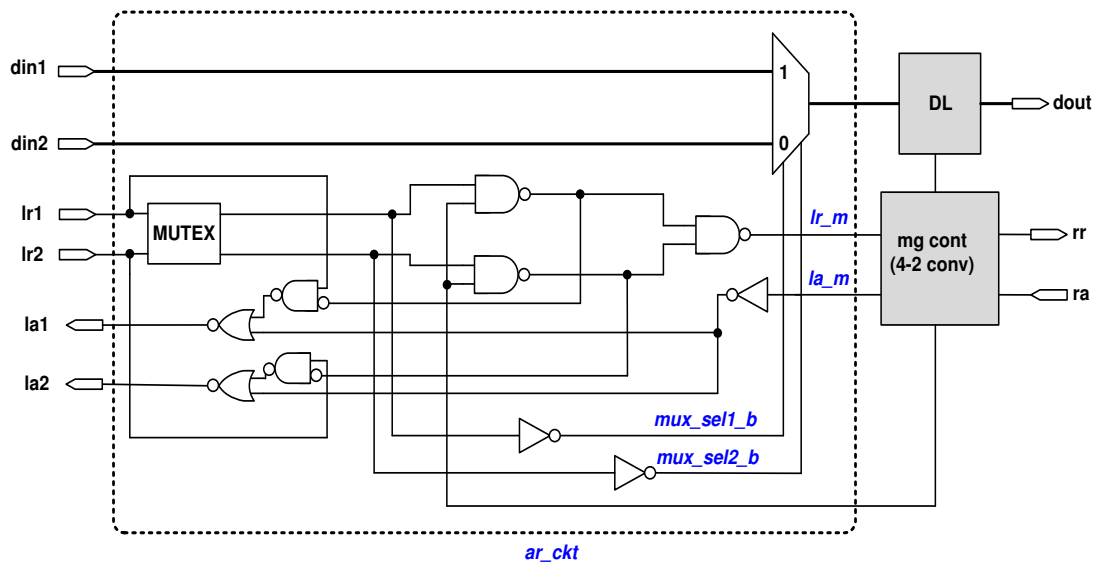


Figure 2.6: Design of merge module.

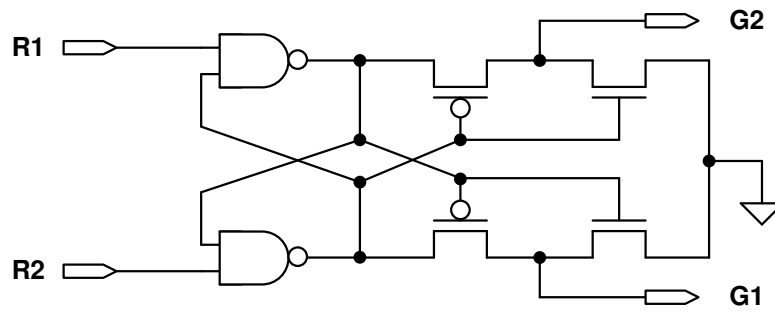


Figure 2.7: Design of MUTEX.

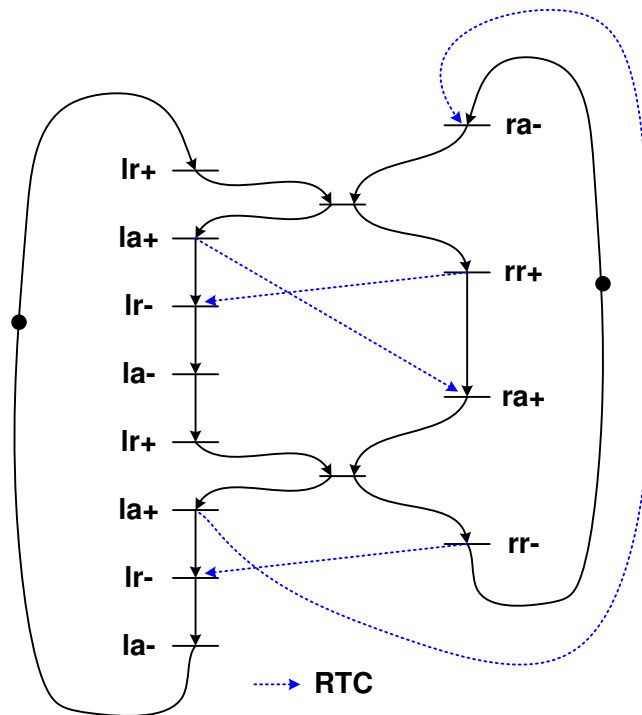


Figure 2.8: Petri-Net specification of merge controller.

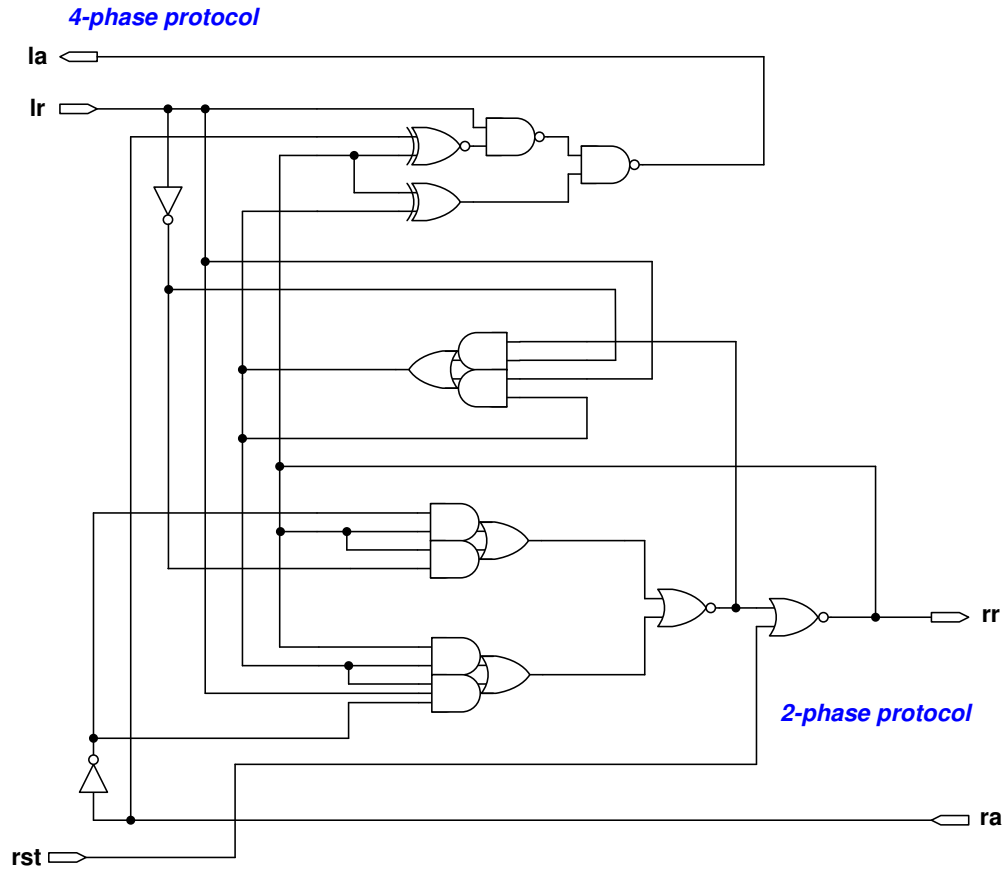


Figure 2.9: Implementation of merge controller.

$$\begin{aligned}
 \text{LEFT} &= \underline{l_r}.c1.la.c2.\underline{l_r}.la.\text{LEFT} \\
 \text{RIGHT} &= \underline{c1}.rr.\underline{c2}.ra.\text{RIGHT} \\
 \text{MG_CON} &= (\text{LEFT}|\text{RIGHT})\setminus\{c1, c2\}
 \end{aligned} \tag{2.2}$$

2.1.3 Asynchronous Circuit Design Methodology

All of the circuits were designed with the static, regular V_{th} , Artisan cell library on IBM's 65nm 10sf process. The asynchronous circuit design process uses a clocked CAD flow in a methodology similar to [28], and it is shown in Figure 2.10.

Implementation - Circuit implementation of asynchronous modules was done with *Petrify* [27] or manual design. The input to *Petrify* are Petri-Nets, which are equivalent to the process-based specification such as CCS.

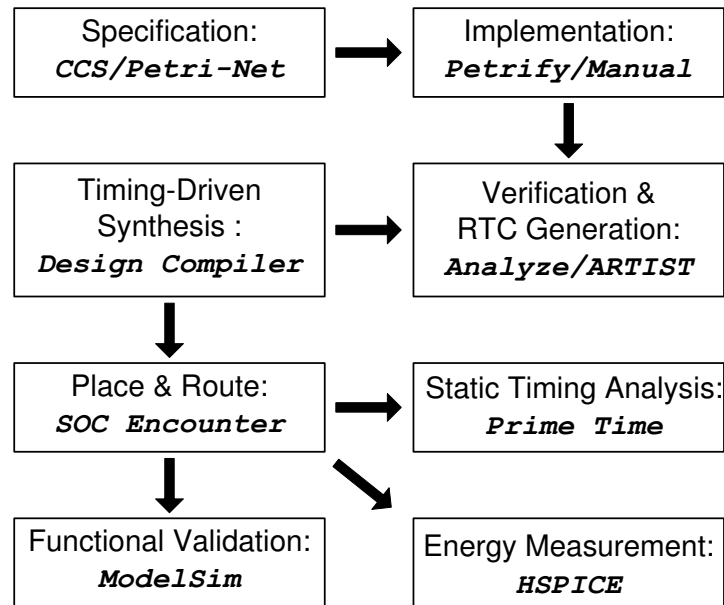


Figure 2.10: Asynchronous circuit design flow.

Verification and RTC Generation - The implemented circuits were verified using the Asynchronous Formal Verification tool, *Analyze* [29]. Another tool, *ARTIST* [30], generated the relative timing constraints (RTCs) that allow the circuit to be proven conformant to its specification, and thus operate correctly.

Timing-Driven Synthesis - The RTCs from *ARTIST* were converted into Synopsys Design Constraints (SDC) format, and the asynchronous modules and full asynchronous router design were synthesized with Synopsys Design Compiler.

Place and Route - The synthesized asynchronous router was physically placed and routed with Cadence SOC Encounter.

Static Timing Analysis - The placed and routed designs were timing-verified by Static Timing Analysis with Synopsys PrimeTime against the constraints generated by the verification tools.

Functional Validation - Functionality and performance were validated in the design with ModelSim using back annotated pre- and post-layout delays.

Energy Measurement - Energy was measured using HSPICE simulations of the design's spice netlist using parasitic extraction from Mentor Graphics Calibre PEX.

2.2 Asynchronous Router Design

Three different asynchronous routers, D1, D2 and D3, were designed with the identical architecture shown in Figure 2.1 using the switch and merge module designs of the previous section. They are shown in Figure 2.11, Figure 2.12 and Figure 2.13 where numbers in parentheses are cycle times of the corresponding handshake cycles.

The asynchronous routers consist of three switch and three merge modules shown in Figure 2.1. However, for the sake of simplicity, each router design is presented with only one switch and merge module. The other two switch and merge modules are identical to those shown in the figure.

The merge modules are identical in all three routers and their architecture is shown in Figure 2.6 whereas the switch modules are distinctive in each router design. Meanwhile, the submodules inside the switch modules, *2-4 conv.*, *sw_demux*, and *LC_4p* are identical in the three different switch module designs. In other words, each switch design distinguishes itself by the different placement of the submodules.

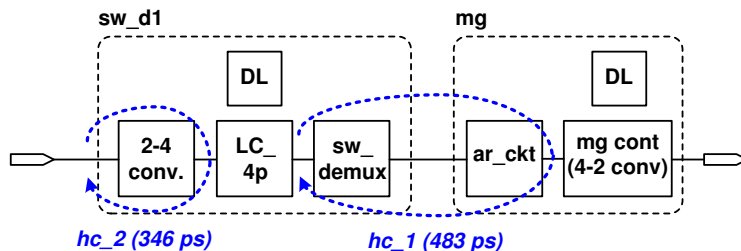


Figure 2.11: Router D1.

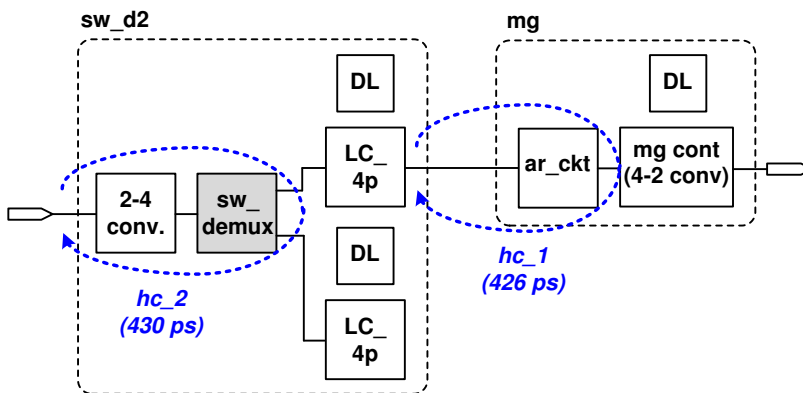


Figure 2.12: Router D2.

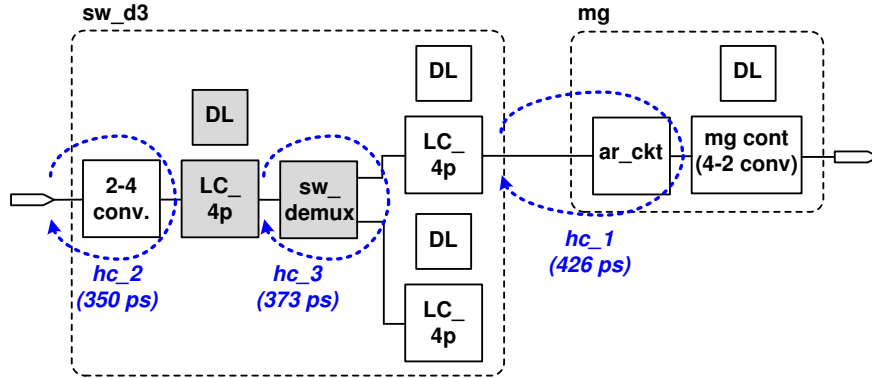


Figure 2.13: Router D3.

Asynchronous communication transfers data by a handshake protocol and hence there is one handshake cycle between any two connected data latches. So, the D1 and D2 routers have two handshake cycles, while the D3 router has three handshake cycles. The maximum throughput of each router design is determined by the handshake cycle which has the longest cycle time.

The three different router designs were intended to improve the router throughput by reducing handshake cycle time, using the placement of the submodules or adding one more data latch. The D1 router is the base design which uses the initial design of the switch module as in Figure 2.2 and the other routers were improved designs based on the D1 router.

In the switch module of the D1 router, the *sw_demux* is located after the *LC_4p* and connected with the *ar_ckt* in the merge module. It leads to the connection of two combinational blocks in the *hc_1*. Consequently, the long cycle time of the *hc_1* (483 ps) limits the throughput of the D1 router as 2.07 Gflits/s (Gfps).

The D2 router achieves better router throughput by separating the *sw_demux* and the *ar_ckt* with the pipeline latch. The *sw_demux* is located in front of the *LC_4p* and only the *ar_ckt* exists in the *hc_1* cycle of the D2 router. As a result, the D2 router has a smaller cycle time in the *hc_1* cycle (426 ps) than that of the D1 router. However, shifting the location of the pipeline latch leads to a different connection of two combinational blocks, *2-4 conv.* and *sw_demux* in the *hc_2* cycle, resulting in the increase of its cycle time to (430 ps). In consequence, the throughput of the D2 router is limited by the *hc_2* handshake cycle at 2.32 Gfps. Meanwhile, one more

data latch is required after the *sw_demux* for storing packets for different output ports separately.

Another data latch stage is inserted into the D3 router design between the two combinational blocks in the *hc_2* cycle of the D2 router to reduce the cycle time of *hc_2*. As a result, the *hc_1* cycle determines the router throughput at 2.35 Gfps.

Table 2.1 summarizes design results for the three routers. Router area and dynamic energy consumption per flit were measured with a 44-bit link width; 32-bit data-path and 12-bit routing address. The D1 router used the fewest resources in both area and energy, but it shows the lowest throughput. The D2 design has higher performance but with larger area than the D1 router. Dynamic energy dissipated per flit is not very different between the D1 and D2 router, because the majority of dynamic energy is consumed by data latches, and each flit passes through two data latches inside the routers equally in both router designs. The D3 router shows the highest router throughput. However, the performance benefit comes at the expense of the largest area and highest energy consumption per flit.

Area is dominated by data latches and the data MUXes used in the merge modules. The controllers (*LC_4p* in the switch modules and *mg cont* in merge modules) make a very small contribution to the total area. Dynamic energy is consumed when one data word passes a router from an input port to an output port. Energy is measured using HSPICE simulations with the spice netlist generated from the design using parasitic extraction from Mentor Graphics Calibre PEX. The same simulation was used in both HSPICE and ModelSim. The HSPICE control file was generated by converting a vcd file generated from the ModelSim simulation. This allowed us to more easily validate switching activity on the data and control paths. A 25% data switching activity factor was applied to the data bits for the energy simulations.

Table 2.1: Design results of three asynchronous routers

	Max. Throughput (Gfps)	Area (μm^2)	Energy/flit (pJ)
D1	2.07	3136	1.127
D2	2.32	4043	1.158
D3	2.35	4990	1.575

The maximum throughput of the routers is measured by inserting data into an input port at the maximum rate while alternating packet output port, and allowing two output ports to communicate with other routers with no wire delay. The router's low power and area are due to its simple architecture and the use of latches, rather than flip-flops, for storage elements. Latches are about half the size and use less power than flip-flops. Since much of the area and power of many router architectures derives from memory elements, this advantage makes a significant difference. Furthermore, the simplicity of the control circuits also contributes to high throughput. These routers employ a bundled data protocol rather than delay insensitive codes which results in fewer wires per channel and efficient use of standard cell libraries. However, the cost to this is that the circuit timing must be carefully specified and controlled, similar to clocked design, to ensure correct operation.

2.2.1 Router Performance Evaluation with Link Wire Length

A path from a source to a destination in an asynchronous NoC is normally composed of several routers and several links. Subsequently, there exist several handshake cycles other than the handshake cycle inside a router, and the maximum path throughput is determined by the longest handshake cycle time among several handshake cycles.

Figure 2.14 shows a link from router R0 to R1 in an NoC. (One more handshake cycle exists inside R1 if it is a D3 router.) One handshake cycle (hc_1) exists between the switch and merge modules inside the router R1. The other handshake cycles (hc_2) are external, between routers. The cycle time of the internal handshake cycle, hc_1 , is not changed after the router design is fixed. On the other hand, the cycle time

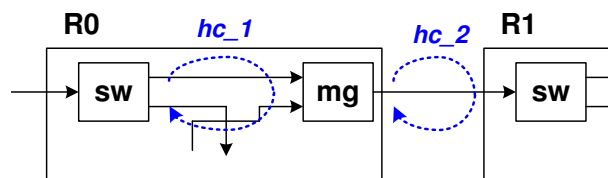


Figure 2.14: Handshake cycles in asynchronous communication link.

of the external handshake cycle is affected by their link wire length, determined by the placement of adjacent two routers. Simply, if the link wire length is long enough so that the external handshake cycle has longer cycle time than that of the internal handshake cycle due to the link wire delay, the router performance is decided by the external handshake cycle, rather than the internal handshake cycle. Therefore, it is required to take into account of the impact of link wire delay on the router performance together, particularly in asynchronous communication links.

In this section, the router performance with link wire delay is evaluated and its properties are presented. For simplicity of explanation in following sections, some terms are defined:

- ICT = Initial cycle time of a handshake cycle with no link wire delay,
- DCT = Delayed cycle time of a handshake cycle from it ICT , by link wire delay,
- WL = Wire length of a link,
- WD = Wire delay of a link.

Link wire delay is estimated using a linear regression equation, Eq. 2.3, which is driven based on the simulation results presented in [31]:

$$WD = 0.1 \times WL + 16 \quad (2.3)$$

2.2.1.1 Performance Evaluation of Asynchronous Router D1

A link which connects two D1 routers is depicted in Figure 2.15 with two handshake cycles. Cycle hc_1 is the internal handshake cycle of the D1 router and hc_2 is an external cycle between two routers.

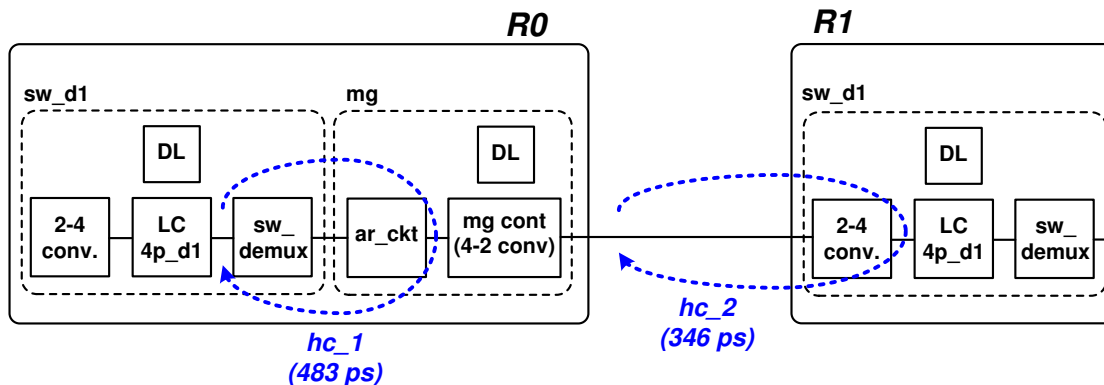


Figure 2.15: Handshake cycles in D1 router.

Link BW is determined the longer cycle time of either hc_1 or hc_2 . First, the maximum link BW is achieved with no consideration of link wire length between two routers. So, hc_1 determines the maximum link BW, 2.07 Gfps, with the longer cycle time of 483 ps than that of hc_2 (346 ps). The maximum link BW is exactly the maximum throughput of the D1 router presented in Table 2.1.

As link wire length increases, the cycle time of the hc_2 is affected by the link wire delay and consequently increases proportionally. Meanwhile, the cycle time of the hc_1 is unaffected by the link wire delay as it is inside the router. Link BW begins to decrease when the delayed cycle time (DCT) of the hc_2 is greater than the initial cycle time (ICT) of hc_1 .

Accordingly, there is a link length range where the link BW is determined by the ICT of hc_1 , rather than the DCT of hc_2 , because the ICT of hc_2 is smaller than that of hc_1 . The link length range can be calculated with Eq. 2.4 using Eq. 2.3:

$$ICT_{hc_1} \leq DCT_{hc_2} \quad (2.4)$$

$$ICT_{hc_1} \leq ICT_{hc_2} + 2 \times WD$$

$$WD \leq (ICT_{hc_1} - ICT_{hc_2})/2$$

$$\text{with } ICT_{hc_1} = 483 \text{ and } ICT_{hc_2} = 346$$

$$WD \leq 68.5 \text{ ps}$$

$$WL = (WD - 16) \times 10$$

$$WL = 525 \mu\text{m}$$

where the external router handshake uses a 2-phase protocol, so that $2 \times$ wire delay is applied in the calculation of DCT_{hc_2} .

Consequently, up to a wire length of $525 \mu\text{m}$, the ICT_{hc_1} determines the link BW at 2.07 Gfps, while DCT_{hc_2} is still less than the ICT_{hc_1} . Above $525 \mu\text{m}$, the link BW is degraded by the longer DCT_{hc_2} influenced by the link wire delay.

Such a link length range is different according to the router designs in a link, as it depends on the relation between the longest ICT and the ICTs of external handshake cycles. So, it can be used as one of the characteristics of a router design and link BW and hereafter is referred to the Maximum Bandwidth wire length Range (MBR). The

MBR of a link exists only when there is a difference between the ICT of the longest handshake cycle and the wire delay sensitive handshake cycle. Furthermore, as the difference is larger, the size of range increases.

Figure 2.16 shows simulation results measuring link BW by varying link length from $0\ \mu\text{m}$ to $4000\ \mu\text{m}$. The link length is measured from the output of R0 to the input of R1. Link BW begins to decrease only after the MBR of the link.

2.2.1.2 Performance Evaluation of Asynchronous Router D2

Similarly with the D1 router, the D2 router has two handshake cycles: hc_1 is an internal handshake cycle and hc_2 is an external handshake cycle, as depicted in Figure 2.17. However, the cycle time of each handshake cycle of the D2 router is different from that of the corresponding handshake cycle of the D1 router.

Unlike the D1 router, the longest ICT of D2 is for hc_2 , an external handshake cycle, rather than the internal handshake cycle, hc_1 , in the D1 design. This difference leads to distinctive characteristics of wire delay impact on link BW with the D2 router, and it is shown in Figure 2.18 along with the link BW of the D1 router. There is no MBR where the maximum link BW is maintained while unaffected by the link wire delay. This is because the hc_2 cycle determines the maximum throughput with no wire delay, as well as is the external handshake cycle which is sensitive to the link wire delay. Therefore, the DCT_{hc_2} is always the one which determines link BW

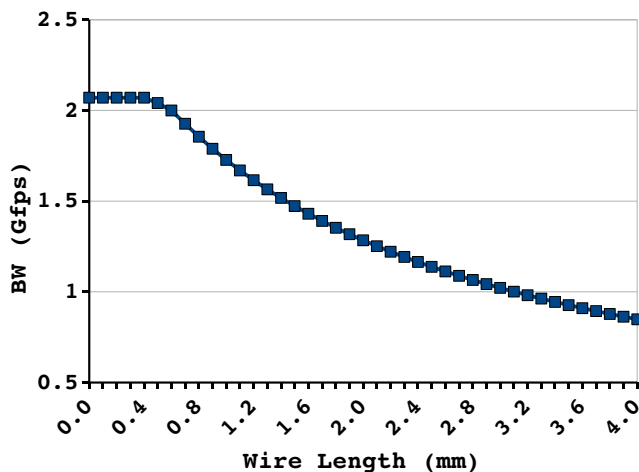


Figure 2.16: Impact of link wire delay on link BW with router D1.

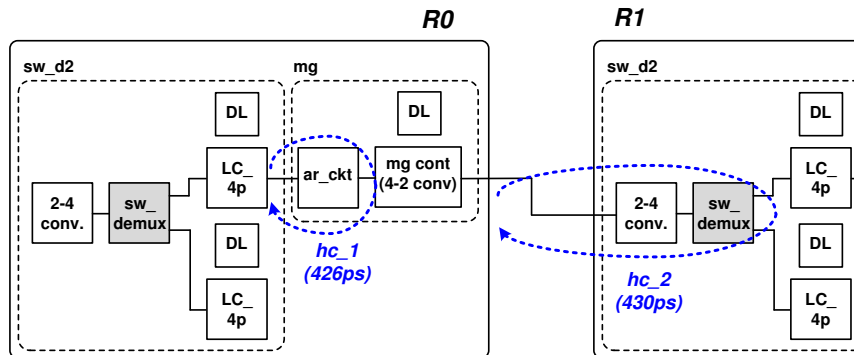


Figure 2.17: Handshake cycles in D2 router.

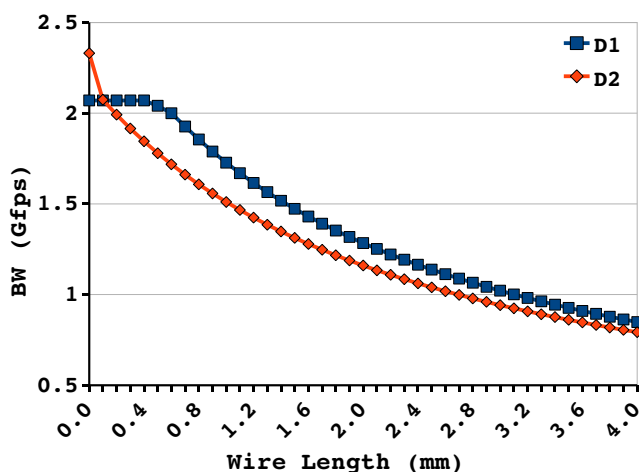


Figure 2.18: Impact of link wire delay on link BW with router D1 and router D2.

regardless of link wire length. Link BW is degraded by even a very small length of link wire. Moreover, the ICT of cycle hc_2 of the D2 router is greater than that of the D1 router and subsequently, the DCT_{hc_2} of D2 is always longer than that of the D1. As a result, the link BW with the D2 router is worse than that of D1 router for the full range of wire length in the simulation, except for wire length less than $100\ \mu\text{m}$. Thanks to the higher maximum throughput of the D2 router, the link BW is better than that of D1 router until $100\ \mu\text{m}$ wire length.

Although the design change in the D2 router achieves higher maximum throughput, it makes the router more vulnerable to the link wire delay penalty, and thereby provides worse throughput than the D1 design.

2.2.1.3 Performance Evaluation of Asynchronous Router D3

Instead of two handshake cycles in a link with D1 or D2 router, the D3 router has three handshake cycles in a connection of two routers as shown in Figure 2.19.

Cycle hc_1 determines the maximum throughput of the D3 router with the longest ICT. Cycle hc_2 is the external handshake cycle affected by the link wire delay. Meanwhile, cycle hc_3 does not impact router performance in any condition, since it has a smaller ICT than hc_1 and it is not influenced by the link wire delay as an internal handshake cycle.

Similarly to the D1 design, the D3 router has an MBR, and it can be calculated as $220 \mu\text{m}$ in Eq. 2.5.

$$ICT_{hc_1} \leq DCT_{hc_2} \quad (2.5)$$

$$ICT_{hc_1} \leq ICT_{hc_2} + 2 \times WD$$

$$WD \leq (ICT_{hc_1} - ICT_{hc_2})/2$$

$$\text{with } ICT_{hc_1} = 426 \text{ and } ICT_{hc_2} = 350$$

$$WD = 38 \text{ ps}$$

$$WL = (WD - 16) \times 10$$

$$WL = 220 \mu\text{m}$$

Figure 2.20 compares link BW of the three router designs. The D3 router has the highest link BW and can maintain it until $220 \mu\text{m}$. In addition, the impact of

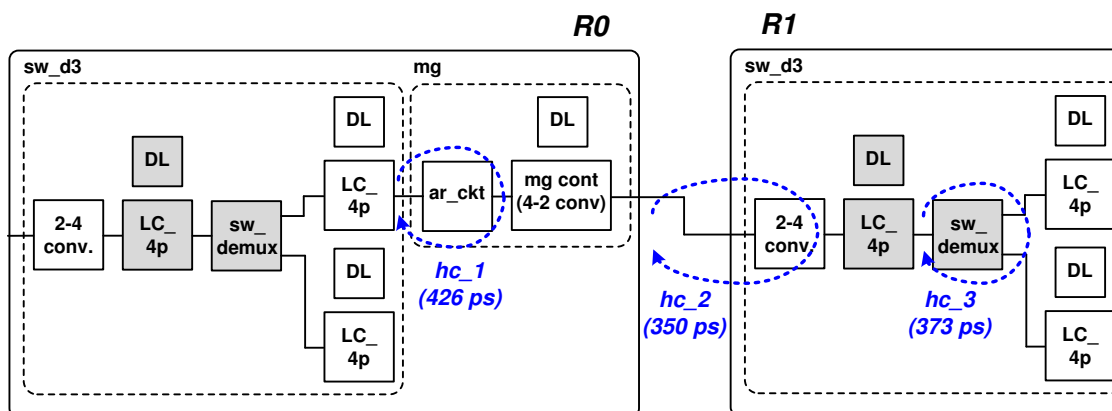


Figure 2.19: Handshake cycles in router D3 design.

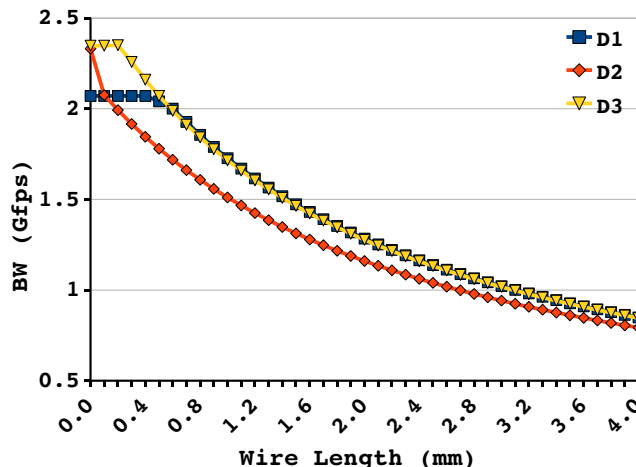


Figure 2.20: Impact of link wire delay on link BW with router D1, D2 and D3.

link wire delay on the link BW of the D3 router is very similar to that of the D1 router, especially, after $500 \mu\text{m}$ of link length. This is because the wire delay sensitive handshake cycle, *hc_2*, of both D1 and D3 routers, are almost identical to each other.

As link wire length increases, the link wire delay is a dominating factor in determining the DCT of the wire delay sensitive handshake cycle in all three designs and subsequently, the link BW gets closer each other.

Overall, given the distinctive characteristics of each router design, the selection of the best router design depends on link wire length. With zero wire length, the D2 router is the best, since it performs as well as D3, while consuming less energy than D3, similar to D1. But, its link BW degrades rapidly and it becomes the worst design only after $100 \mu\text{m}$ of wire length. The D3 router would be a preferable design when high link BW is required with short wire length, like under $500 \mu\text{m}$. It, however, consumes more energy than the others. Above $500 \mu\text{m}$ of link wire length, the D1 router is the most energy- and area-efficient design with the same link BW with the D3, better than the D2, as well as the least energy consumption.

CHAPTER 3

PIPELINE LATCH IN ASYNCHRONOUS NOC

Pipeline latches (PLs) in asynchronous communication links are more beneficial than in synchronous ones, since they act not only as data buffers but also to improve link BW, whereas they are for buffering only in synchronous links.

3.1 Design of 2-phase Linear Controller

The three routers in the previous section were designed to handshake externally with a 2-phase protocol. Thus, a PL should use the same 2-phase protocol. A 2-phase linear controller was designed, following the asynchronous circuit design procedure described in the Section 2.1.3. The CCS specification and circuit implementation of the 2-phase linear controller are shown in Eq. 3.1 and Figure 3.1, respectively.

$$\begin{aligned} \text{LEFT} &= \underline{l.r.c1.la}. \text{ LEFT} \\ \text{RIGHT} &= \underline{c1.rr.ra}. \text{ RIGHT} \\ \text{LC_2p} &= (\text{LEFT}|\text{RIGHT})\backslash\{c1\} \end{aligned} \tag{3.1}$$

3.2 Pipeline Latch Impact on Link Bandwidth

A PL in an asynchronous link divides one handshake cycle (hc_0) into two other handshake cycles, hc_1 and hc_2 , as it adds an additional data latch stage between two routers, depicted in Figure 3.2. Subsequently, the link wire length is also divided into two short lengths and therefore, the link BW is determined by the DCT of either cycle hc_1 or hc_2 with shorter wire lengths, rather than the DCT of hc_0 with the whole link wire length. For instance, if the PL is inserted into the center of the link, the total wire length is evenly divided into two half length wires, and the link BW is affected by only half of the link wire delay.

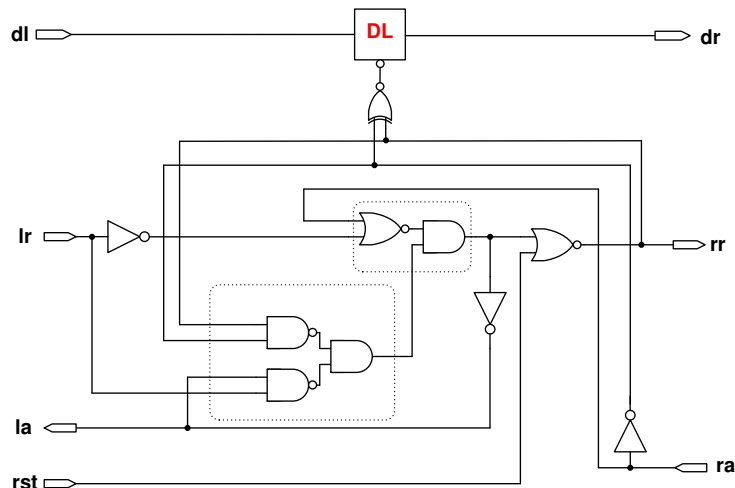


Figure 3.1: Design of 2-phase linear controller.

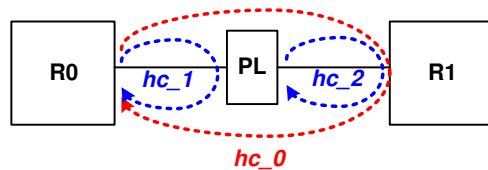


Figure 3.2: PL insertion and handshake cycles.

Figure 3.3 shows a link, $D1_PL1$, with one PL between two D1 routers. It is identical to the link shown in Figure 2.15 except the PL. For the brevity of explanation, hereafter, the link in Figure 2.15 is referred as $D1_PLno$, a link between two D1 routers with no PL.

The $D1_PL1$ link has three handshake cycles, one more than $D1_PLno$, as the PL in $D1_PL1$ divides cycle hc_2 of $D1_PLno$ into two handshake cycles, hc_2 and hc_3 . Hence, the $D1_PL1$ link has two external handshake cycles of which DCTs are affected by link wire lengths. Meanwhile, inserting a PL in a link does not affect the ICTs of the handshake cycles inside routers, so hc_1 is identical in the two links. The ICT of cycle hc_1 is the longest one and determines the maximum link BW with zero wire length.

The benefit of inserting a PL in $D1_PLno$ is shown in Figure 3.4 where link BW variation of two links, $D1_PLno$ and $D1_PL1$, are compared as a function of varying

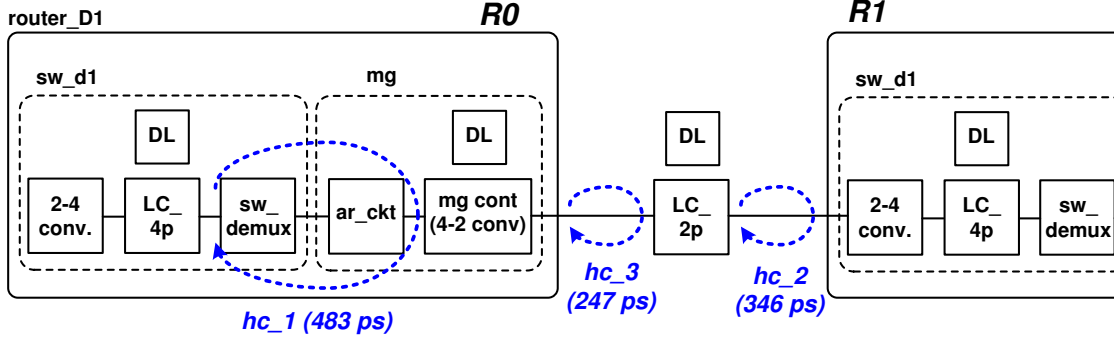


Figure 3.3: Link of D1 router with a PL: $D1_PL1$.

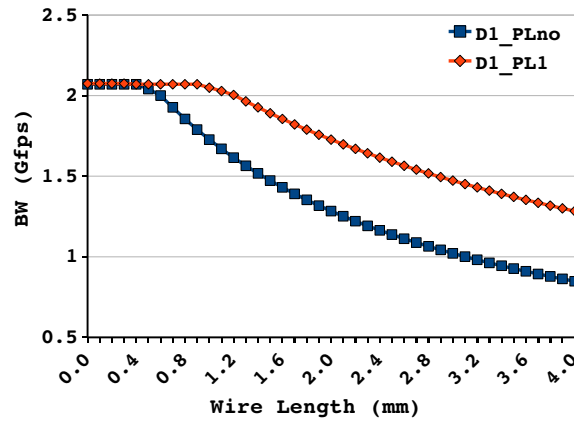


Figure 3.4: Impact of link wire delay on link BW with router D1 and one PL.

total link wire length. In $D1_PL1$, the PL is placed exactly in the middle of the link so that the divided link wire length is half of the total wire length.

Comparing the DCTs of the two external handshake cycles, hc_2 and hc_3 , of the link $D1_PL1$, the DCT of hc_2 is always longer than that of hc_3 and subsequently determines the link BW, since the total link wire length is evenly divided in two and the ICT of hc_2 is longer than the ICT of hc_3 .

Thanks to the PL insertion, the effective wire length of hc_2 of $D1_PL1$ is cut in half, compared to $D1_PLno$. So, the link BW of the $D1_PL1$ is always better than that of $D1_PL0$. The link BW of $D1_PL1$ with a $2000\ \mu\text{m}$ wire length is 1.73 Gfps which is exactly same with that of $D1_PLno$ at $1000\ \mu\text{m}$. Compared to the link BW of the $D1_PLno$ at $2000\ \mu\text{m}$, 1.28 Gfps, it is improved by 35%.

Furthermore, the MBR of the $D1_PL1$ link is determined by the relation between the ICT of hc_1 and the DCT of hc_2 and calculated in Eq. 3.2 and Eq. 3.3. It is twice as far as that of the $D1_PLno$ link at $525\ \mu\text{m}$.

$$ICT_{hc_1} \leq DCT_{hc_2} \quad (3.2)$$

$$ICT_{hc_1} \leq ICT_{hc_2} + 2 \times WD_{MBR}$$

$$WD_{MBR} \leq (ICT_{hc_1} - ICT_{hc_2})/2$$

$$\text{with } ICT_{hc_1} = 483 \text{ and } ICT_{hc_2} = 346$$

$$WD_{MBR} \leq 68.5 \text{ ps}$$

$$WL_{MBR}/2 \leq (WD_{MBR} - 16) \times 10 \quad (3.3)$$

$$WL_{MBR} \leq 525 \times 2 = 1050\ \mu\text{m}$$

3.3 Optimal Position of One Pipeline Latch Placement

As aforementioned, the BW of a $D1_PL1$ link is always determined by the DCT of cycle hc_2 , rather than hc_3 , as hc_2 has a larger ICT for the same link wire length. Eq. 3.4 calculates the DCT of two handshake cycles and their link BW when a PL is inserted at the center of a link with a $2000\ \mu\text{m}$ long wire.

$$DCT_{hc_2} = ICT_{hc_2} + 2 \times WD \quad (3.4)$$

$$= 346 + 2 \times (0.1 \times 1000 + 16)$$

$$= 578 \text{ ps} \rightarrow 1.78 \text{ Gfps}$$

$$DCT_{hc_3} = ICT_{hc_3} + 2 \times WD$$

$$= 247 + 2 \times (0.1 \times 1000 + 16)$$

$$= 479 \text{ ps} \rightarrow 2.08 \text{ Gfps}$$

The link BW of $D1_PL1$ is 1.78 Gfps due to the larger DCT of hc_2 . Meanwhile, the higher throughput of cycle hc_3 , 2.08 Gfps, is limited by hc_2 and can not be utilized. The unbalanced throughput between two handshake cycles comes from the difference of ICTs with the same wire length.

Instead of dividing total link wire equally, if the PL is inserted in consideration of the unbalanced ICTs of two handshake cycles, by giving more wire length to the shorter ICT of cycle hc_3 , the DCT of hc_2 would be reduced with a shorter wire length which results in further link BW improvement by the PL insertion.

Overall, if two handshake cycles, which handshake through a PL inserted in a link, have different ICTs, the handshake cycle with smaller ICT has the capability to handle more wire delay than the other. Therefore, there exists an optimal position of a PL where the DCTs of two handshake cycles are balanced resulting in the maximum link BW that can be achieved by PL insertion.

3.3.1 Optimal Position of One Pipeline Latch with Router D1

In order to see how the link BW is affected by the PL position in $D1_PL1$, a simulation was performed. Figure 3.5 depicts the $D1_PL1$ link again with only link wire length variables, n and m , and Figure 3.6 shows the link BW variance by sweeping the PL position between router R0 to R1 with $2000\ \mu\text{m}$ of total wire length. The x -axis is the distance of the PL from R0, which is n (WL of hc_3) in Figure 3.5. If the distance is $0\ \mu\text{m}$, the PL is placed just after the output of the R0 router without any link length assigned to the hc_3 .

The worst link BW is achieved when the PL is placed at $0\ \mu\text{m}$, closest to R0 and farthest from R1, such that the total wire length is assigned only to cycle hc_2 while no wire delay penalty is given to cycle hc_3 . The link BW of 1.28 Gfps is determined by the DCT of cycle hc_2 at 778 ps.

As the PL moves from R0 to R1, the wire length of hc_2 decreases. Subsequently, the DCT of hc_2 is reduced and hence, the link BW is improved. But, overall improvement occurs only until the position of $1250\ \mu\text{m}$. After that position, link

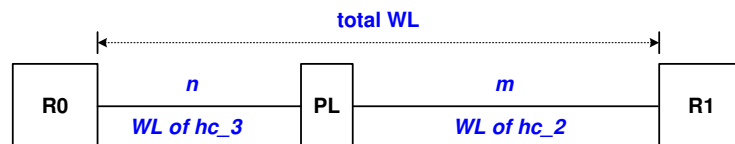


Figure 3.5: Wire length of hc_2 and hc_3 in $D1_PL1$.

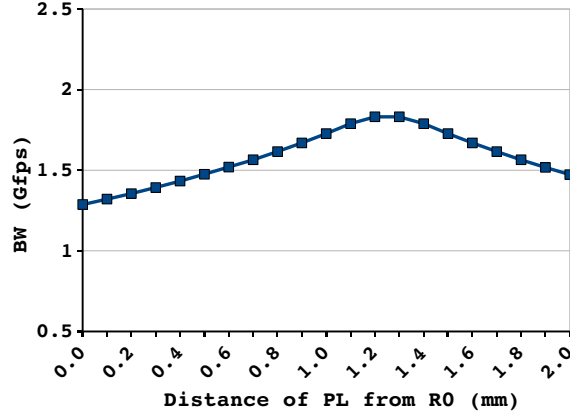


Figure 3.6: PL impact on link throughput in total 2.0 mm link wire.

BW begins to decrease as too much link wire length is assigned to cycle hc_3 . So, the DCT of cycle hc_3 becomes greater than that of hc_2 and it determines the link BW.

Accordingly, link BW is maximum, 1.89 Gfps, when the PL is located 1250 μm from R0 or 750 μm from R1 with a 2000 μm long wire. The position with the maximum link BW is where the DCTs of the two handshake cycles are equal. They can be calculated with Eq. 3.5.

$$\begin{aligned}
 DCT_{hc_2} &= ICT_{hc_2} + 2 \times (0.1 \times WL_{hc_2} + 16) & (3.5) \\
 &= 346 + 2 \times (0.1 \times 750 \mu\text{m} + 16) \\
 &= 528 \text{ ps} \rightarrow 1.89 \text{ Gfps} \\
 DCT_{hc_3} &= ICT_{hc_3} + 2 \times (0.1 \times WL_{hc_3} + 16) \\
 &= 247 + 2 \times (0.1 \times 1250 \mu\text{m} + 16) \\
 &= 529 \text{ ps} \rightarrow 1.89 \text{ Gfps}
 \end{aligned}$$

Since cycle hc_3 can handle more wire delay than hc_2 , thanks to its shorter ICT, the optimal PL position is biased to R1 by assigning a longer wire length to hc_3 which has the shorter ICT.

In consequence, link BW varies according to the position of a PL in a link and there is an optimal position of the PL where the link BW is its maximum. At the optimal position, the DCTs of two handshake cycles, handshaking with each other through the PL, are balanced and equal. If a PL is located in any other position, rather than the optimal one, the balance of two DCTs is broken and one of the DCTs

is longer than the other. Accordingly, the link BW can not achieve its maximum value.

3.3.1.1 Maximum Bandwidth Range of D1_PL1 with Optimal PL

Since both the external handshake cycles, hc_2 and hc_3 , of a $D1_PL1$ link have shorter ICTs than that of the internal handshake cycle, hc_1 , each handshake cycle has its own MBR in relation to the internal cycle hc_1 . Similarly with Eq. 3.2, the MBRs of hc_2 and hc_3 can be calculated with Eq. 3.6 and Eq. 3.7, respectively.

$$ICT_{hc_1} \leq DCT_{hc_2} \quad (3.6)$$

$$ICT_{hc_1} \leq ICT_{hc_2} + 2 \times WD_{MBR}$$

$$WD_{MBR} \leq (ICT_{hc_1} - ICT_{hc_2})/2 = 68.5 \text{ ps}$$

$$\text{with } ICT_{hc_1} = 483 \text{ and } ICT_{hc_2} = 346$$

$$MBR_{hc_2} \leq (WD_{MBR} - 16) \times 10 = 525 \mu\text{m}$$

$$ICT_{hc_1} \leq DCT_{hc_3} \quad (3.7)$$

$$ICT_{hc_1} \leq ICT_{hc_3} + 2 \times WD_{MBR}$$

$$WD_{MBR} \leq (ICT_{hc_1} - ICT_{hc_3})/2$$

$$\text{with } ICT_{hc_1} = 483 \text{ and } ICT_{hc_3} = 247$$

$$WD_{MBR} \leq 118 \text{ ps}$$

$$MBR_{hc_3} \leq (WD_{MBR} - 16) \times 10 = 1020 \mu\text{m}$$

The MBR of hc_2 is $525 \mu\text{m}$ and hc_3 has a MBR of $1020 \mu\text{m}$. Therefore, a $D1_PL1$ link can maintain maximum BW of 2.07 Gfps with up to $1545 \mu\text{m}$ of aggregate link wire length, if one pipeline latch is placed optimally with the wire length for cycle hc_2 being under $525 \mu\text{m}$ and that of hc_3 less than $1020 \mu\text{m}$.

Figure 3.7 shows another simulation result with $1000 \mu\text{m}$ total link wire length. Like Figure 3.6, the link BW increases as the PL position is moved from R0 to R1. However, after passing around $500 \mu\text{m}$, the link BW reaches the maximum throughput of the D1 router, 2.07 Gfps and maintains it until $1000 \mu\text{m}$. When the PL is placed

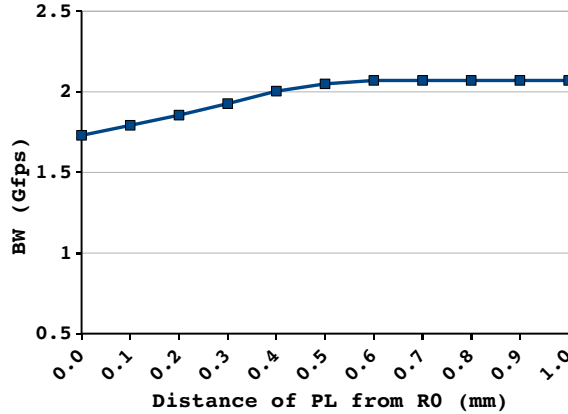


Figure 3.7: PL impact on link throughput in total 1.0 mm link wire.

over $500 \mu\text{m}$, the wire length of cycle hc_2 becomes less than its MBR, $525 \mu\text{m}$, so that the DCT of hc_2 is shorter than hc_1 , 483 ps . In regard to cycle hc_3 , the total wire length of $1000 \mu\text{m}$ is less than the MBR of hc_3 at $1020 \mu\text{m}$. Thus, the DCT of hc_3 is always less than the ICT of hc_1 for this length of wire. Subsequently, the link BW is maintained as 2.07 Gfps , the max throughput, with any location of PL after $500 \mu\text{m}$ from R0.

3.3.1.2 Estimation of Optimal PL Position in D1_PL1

The optimal PL position varies with respect to the total wire length of a link. It can be calculated by Eq. 3.8 and Eq. 3.9, using the fact that DCTs of two handshake cycles are identical at the optimal position.

$$n + m = WL \quad (3.8)$$

$$DCT_{hc_2} = DCT_{hc_3} \quad (3.9)$$

where n is wire length of hc_3 , m is that of hc_2 , and WL is the total wire length of a link, as shown in Figure 3.5. Eq. 3.9 is transformed using wire length variables n and m in Eq. 3.10.

$$\begin{aligned}
DCT_{hc_2} &= DCT_{hc_3} \\
ICT_{hc_2} + 2 \times WD_{hc_2} &= ICT_{hc_3} + 2 \times WD_{hc_3} \\
ICT_{hc_2} + 2 \times (0.1 \times m + 16) &= ICT_{hc_3} + 2 \times (0.1 \times n + 16) \\
n - m &= (ICT_{hc_2} - ICT_{hc_3})/0.2 = 495 \quad (3.10) \\
&\text{where } ICT_{hc_2} = 346 \text{ and } ICT_{hc_3} = 247
\end{aligned}$$

From two equations, Eq. 3.8 and Eq. 3.10,

$$\begin{aligned}
n &= (WL + 495)/2 \quad (3.11) \\
m &= WL - n
\end{aligned}$$

Table 3.1 presents the optimal positions of a pipeline latch in a *D1_PL1* link of up to 2000 μm wire length. *WL* is total link wire length. *Cal. OP* is the optimal position calculated by the Eq. 3.11, while *Act. OP* is the actual optimal position with which the DCTs of *hc_3* and *hc_2* are calculated. *Link CT* is the longest cycle time among ICT of *hc_1* and the two DCTs *hc_2* and *hc_3*. As indicated in the first row of the table, with zero wire length, the ICT of *hc_1*, 483 ps, is the longest one and determines the link BW of 2.07 Gfps.

Up to 500 μm of wire length, the *Cal. OP* is longer than the total wire length. So, *Act. OP* is determined by placing the PL at the far end of the link from R0, so that all link wire length is assigned to cycle *hc_3* with shorter ICT than *hc_2*. No wire length is assigned to cycle *hc_2* and therefore, the DCT of *hc_2* is its ICT under 500 μm . Two DCTs are not balanced each other and subsequently, *Act. OP* is not the optimal position. In addition, the link BW is its maximum since the DCTs of both *hc_2* and *hc_3* are less than the ICT of *hc_1* in this range of wire length.

With over 500 μm in wire length, the DCTs of *hc_3* and *hc_2* are equal. It shows that the optimally positioned PL balances the DCTs of the two handshake cycles which leads to the maximum link BW for the corresponding link wire length.

In addition, the link BW can maintain the maximum throughput of the D1 router of 2.07 Gfps up to a wire length of 1500 μm . Due to the optimal position of the PL, the assigned wire length for the *hc_2* and *hc_3* cycles, respectively, is less than the MBR of each handshake cycle: 1020 μm for *hc_3* and 525 μm for *hc_2*. In other words,

Table 3.1: Optimal PL position of $D1_PL1$ link up to $2000\ \mu\text{m}$ total wire length

WL (μm)	Cal. OP (μm)	Act. OP (μm)	DCT _{<i>hc_3</i>} (ps)	DCT _{<i>hc_2</i>} (ps)	Link CT (ps)	Link BW (Gfps)
0	0	0	247	346	483	2.07
100	297.5	100	299	346	483	2.07
200	347.5	200	319	346	483	2.07
300	397.5	300	339	346	483	2.07
400	447.5	400	359	346	483	2.07
500	497.5	500	379	346	483	2.07
600	547.5	547.5	388.5	388.5	483	2.07
700	597.5	597.5	398.5	398.5	483	2.07
800	647.5	647.5	408.5	408.5	483	2.07
900	697.5	697.5	418.5	418.5	483	2.07
1000	747.5	747.5	428.5	428.5	483	2.07
1100	797.5	797.5	438.5	438.5	483	2.07
1200	847.5	847.5	448.5	448.5	483	2.07
1300	897.5	897.5	458.5	458.5	483	2.07
1400	947.5	947.5	468.5	468.5	483	2.07
1500	997.5	997.5	478.5	478.5	483	2.07
1600	1047.5	1047.5	488.5	488.5	488.5	2.05
1700	1097.5	1097.5	498.5	498.5	498.5	2.01
1800	1147.5	1147.5	508.5	508.5	508.5	1.97
1900	1197.5	1197.5	518.5	518.5	518.5	1.93
2000	1247.5	1247.5	528.5	528.5	528.5	1.89

neither the DCT of hc_3 nor hc_2 is longer than the ICT of hc_1 if the total wire length is less than $1500\ \mu\text{m}$ and the PL is optimally placed.

Simulated link BW with optimal PL insertion ($D1_PL_{opt}$) is shown in Figure 3.8 with two other links: $D1_PL_{no}$ and $D1_PL_{mid}$. The $D1_PL_{no}$ represents a link without PL insertion as in Figure 2.15 and the $D1_PL_{mid}$ is the one where a PL is inserted at the middle of the link and it was shown in Figure 3.4.

By placing a PL in the optimal position rather than in the center of a link, link BW is further improved. The MBR of the $D1_PL1_{opt}$ is extended to $1525\ \mu\text{m}$ from its $1050\ \mu\text{m}$ value for $D1_PL_{mid}$ and from $525\ \mu\text{m}$ of $D1_PL_{no}$. The link BW of the $D1_PL1_{opt}$ at $2000\ \mu\text{m}$ is $1.89\ \text{Gfps}$ which is a 6% improvement from that of $D1_PL1_{mid}$ and 48% from a $D1_PL_{no}$ link. The difference in link BW comes from

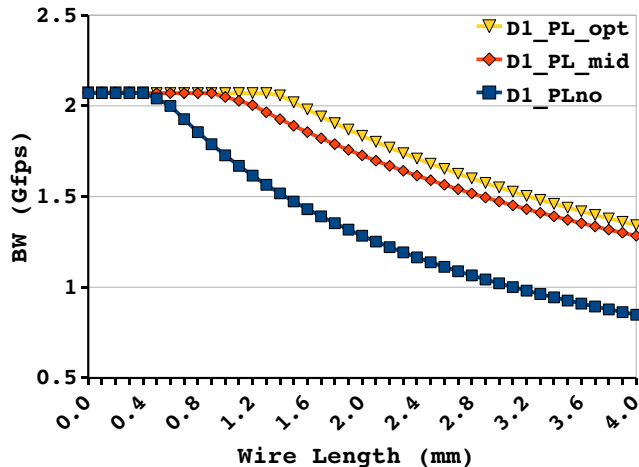


Figure 3.8: Link BW improvement in a link with D1 routers and one optimal PL.

the fact that the link wire length assigned to hc_2 is different in all three links at the same total link wire length.

Note that the simulated link BW is a little bit worse than the values calculated in Table 3.1, especially for wire lengths over $1400 \mu\text{m}$. The calculated BW is estimated based on the handshake cycle times which were measured independently, whereas in circuit simulation, the hc_2 and hc_3 cycles can interfere each other across the PL and therefore, neither one can achieve the cycle time as fast as the calculated values.

All values of the optimal PL position and the MBR of the links are valid only for the D1 router as they depend on the ICTs of handshake cycles and the ICTs are specific to a particular router design.

3.3.2 Optimal Position of One Pipeline Latch with Router D2

Figure 3.9 depicts $D2_PL1$, a link between two D2 routers with one PL inserted in the link. Variables n and m represent the wire length of the hc_3 and hc_2 cycles respectively. Similar to the $D1_PL1$ link in Figure 3.3, this design has one internal handshake cycle, hc_1 and two external handshake cycles, hc_2 and hc_3 . The ICTs of the three handshake cycles, however, are different from those of $D1_PL1$ which results in different characteristics of the MBR, optimal PL position, and link BW.

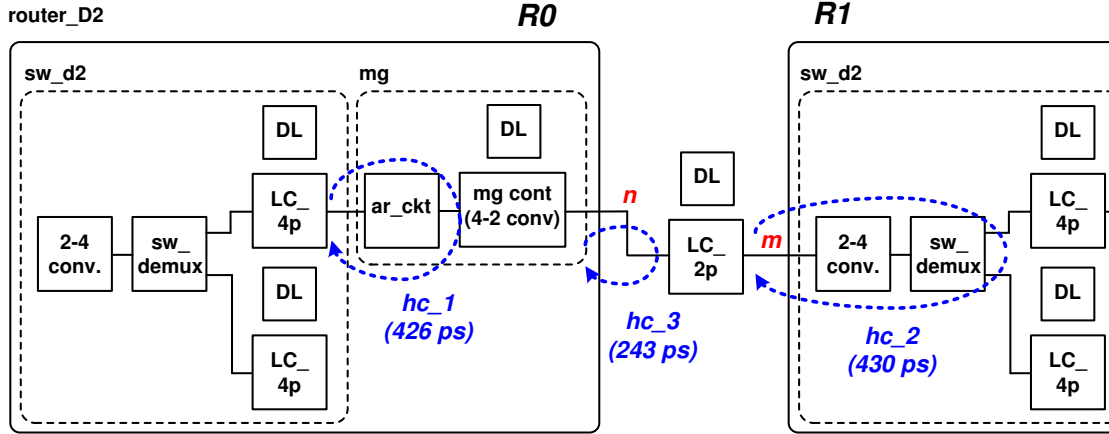


Figure 3.9: Link of D2 router with one PL: $D2_PL1$.

3.3.2.1 Maximum Bandwidth Range of $D2_PL1$ with Optimal PL

The MBR of the link $D2_PL1$ can be estimated similar to link $D1_PL1$. However, as already explained in Section 2.2.1.2, there is no MBR for external handshake cycle hc_2 since it has the longest ICT by itself. Therefore, a small fraction of link wire length will degrade the link BW. On the contrary, the other external handshake cycle, hc_3 , has an ICT which is less than that of hc_2 . Thus, it has an MBR of $775 \mu\text{m}$ as calculated in Eq. 3.12.

$$ICT_{hc_2} \leq DCT_{hc_3} \quad (3.12)$$

$$ICT_{hc_2} \leq ICT_{hc_3} + 2 \times WD_{MBR}$$

$$WD_{MBR} \leq (ICT_{hc_2} - ICT_{hc_3})/2 = 93.5$$

$$\text{with } ICT_{hc_2} = 430 \text{ and } ICT_{hc_3} = 243$$

$$MBR \leq (WD_{MBR} - 16) \times 10 = 775 \mu\text{m}$$

In consequence, the $D2_PL1$ link can maintain its maximum BW of 2.33 Gfps (the maximum throughput of the D2 router) with up to $775 \mu\text{m}$ of link wire length when a PL is adequately placed.

3.3.2.2 Estimation of Optimal PL Position in $D2_PL1$

The optimal PL position of the $D2_PL1$ link is where the DCTs of cycles hc_2 and hc_3 are identical with each other. This can be calculated by Eq. 3.13 and Eq. 3.14,

which is similar to Eq. 3.8 and Eq. 3.10 for the $D1_PL1$ link.

$$n + m = WL \quad (3.13)$$

$$n - m = (ICT_{hc.2} - ICT_{hc.3})/0.2 = 935 \quad (3.14)$$

$$\text{where } ICT_{hc.2} = 430 \text{ and } ICT_{hc.3} = 243$$

From Eq. 3.13 and Eq. 3.14, the optimal PL position in a $D2_PL1$ link is:

$$n = (WL + 935)/2 \quad (3.15)$$

$$m = WL - n$$

Table 3.2 presents the optimal positions of one PL in a $D2_PL1$ link of up to 2000 μm wire length. At zero wire length, the longest ICT of $hc.2$, 430 ps, determines the link BW as 2.32 Gfps.

Until 900 μm , the *Cal. OP* is longer than its corresponding WL , so that all wire length is assigned to cycle $hc.3$ with the shorter ICT. In this range of WL , the *Act. OP* is not the actual optimal position since the two DCTs of $hc.2$ and $hc.3$ are not balanced. After 1000 μm , the two DCT values are equal and they determine the link BW.

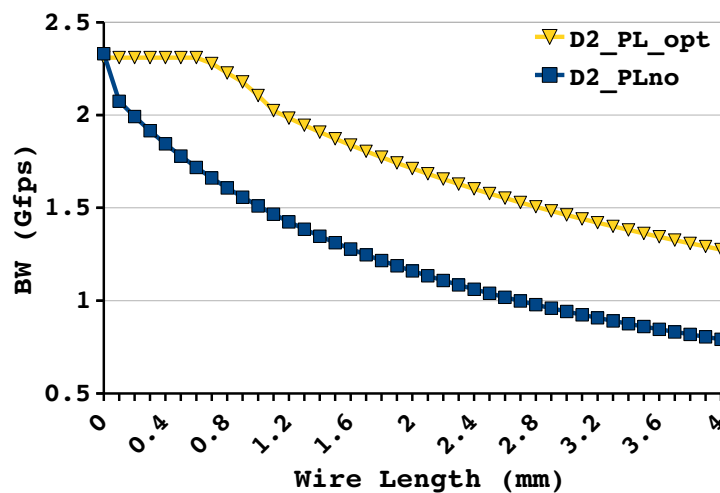
The MBR of cycle $hc.3$ is 775 μm , and the link BW maintains its maximum up to approximately 800 μm . After 900 μm , the link BW begins to decrease.

Figure 3.10 presents the simulated link BW of $D2_PLno$ and $D2_PL1_opt$ with wire length ranging from 0 mm to 4.0 mm. $D2_PLno$ is a link with the D2 routers without a PL in the link as in Figure 2.17. $D2_PL1_opt$ has a PL placed in the optimal PL position.

As already shown in Figure 2.20, $D2_PLno$ is the design least robust to link wire delay and shows the worst link BW, compared with the D1 or D3 routers. Such a weakness of the D2 router is mitigated considerably with optimally inserted one PL. Especially, the link BW improvement of the $D2_PL1_opt$ from the $D2_PLno$ is noticeable when the link wire length is less than 700 μm due to the MBR of the $D2_PL1_opt$ link which does not exist in the $D2_PLno$ link.

Table 3.2: Optimal PL position of $D2_PL1$ link up to $2000\ \mu\text{m}$ total wire length

WL (μm)	Cal. OP (μm)	Act. OP (μm)	$\text{DCT}_{hc.3}$ (ps)	$\text{DCT}_{hc.2}$ (ps)	Link CT (ps)	Link BW (Gfps)
0	0	0	243	430	430	2.32
100	525	100	295	430	430	2.32
200	575	200	315	430	430	2.32
300	625	300	335	430	430	2.32
400	675	400	355	430	430	2.32
500	725	500	375	430	430	2.32
600	775	600	395	430	430	2.32
700	825	700	415	430	430	2.32
800	875	800	435	430	435	2.30
900	925	900	455	430	455	2.20
1000	975	975	470	470	470	2.13
1100	1025	1025	480	480	480	2.08
1200	1075	1075	490	490	490	2.04
1300	1125	1125	500	500	500	2.00
1400	1175	1175	510	510	510	1.96
1500	1225	1225	520	520	520	1.92
1600	1275	1275	530	530	530	1.89
1700	1325	1325	540	540	540	1.85
1800	1375	1375	550	550	550	1.82
1900	1425	1425	560	560	560	1.79
2000	1475	1475	570	570	570	1.75

**Figure 3.10:** Link BW improvement of $D2_PL1$ with optimal PL placement.

3.3.3 Optimal Position of One Pipeline Latch with Router D3

A link with two D3 routers and one PL, $D3_PL1$, is shown in Figure 3.11. The $D3_PL1$ link is identical with a link in Figure 2.19 ($D3_PLno$), except for the PL in the link. The PL makes the external handshake cycle, hc_2 , of the $D3_PLno$ into two separate handshake cycles, hc_2 and hc_4 , and it results in four handshake cycles in the $D3_PL1$ design. The longest ICT of cycle hc_1 determines the maximum link BW with zero wire length and two external handshake cycles, hc_2 and hc_4 , are affected by link wire delays.

3.3.3.1 Maximum Bandwidth Range with Optimal PL in $D3_PL1$

Since two external handshake cycles, hc_2 and hc_4 , are less than the longest ICT, 426 ps of hc_1 , there exist MBRs of each handshake cycle which are estimated in Eq. 3.16 and Eq. 3.17.

$$ICT_{hc_1} \leq DCT_{hc_2} \quad (3.16)$$

$$ICT_{hc_1} \leq ICT_{hc_2} + 2 \times WD_{MBR}$$

$$WD_{MBR} \leq (ICT_{hc_1} - ICT_{hc_2})/2 = 38$$

$$\text{with } ICT_{hc_1} = 426 \text{ and } ICT_{hc_2} = 350$$

$$MBR_{hc_2} \leq (WD_{MBR} - 16) \times 10 = 220 \mu\text{m}$$

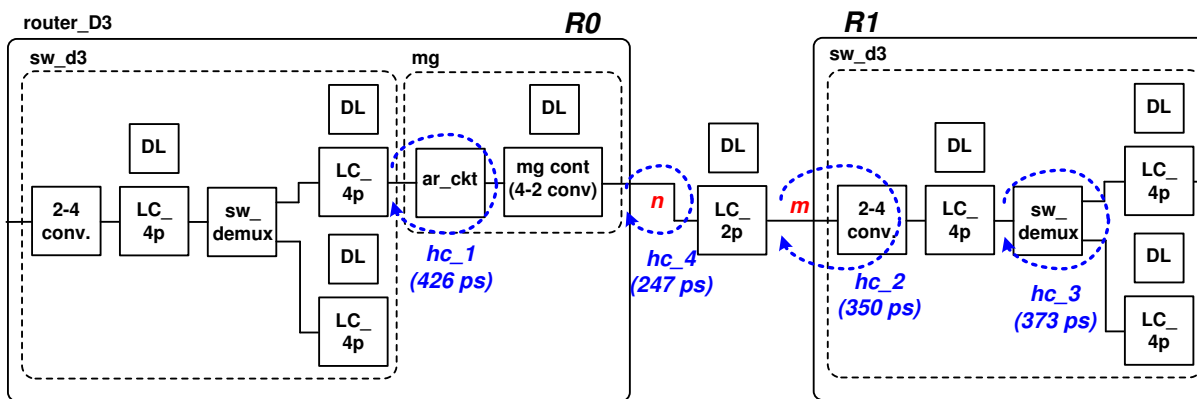


Figure 3.11: Link of D3 router with a PL: $D3_PL1$.

$$\begin{aligned}
ICT_{hc-1} &\leq DCT_{hc-4} & (3.17) \\
ICT_{hc-1} &\leq ICT_{hc-4} + 2 \times WD_{MBR} \\
WD_{MBR} &\leq (ICT_{hc-1} - ICT_{hc-4})/2 = 89.5 \\
&\text{with } ICT_{hc-1} = 426 \text{ and } ICT_{hc-4} = 247 \\
MBR_{hc-4} &\leq (WD_{MBR} - 16) \times 10 = 735 \mu\text{m}
\end{aligned}$$

Consequently, a *D3_PL1* link can maintain its maximum bandwidth of 2.35 Gfps up to the sum of MBRs of two handshake cycles with 955 μm wire length, if a PL is placed at the optimal position corresponding to the total wire length.

3.3.3.2 Estimation of Optimal PL Position with Router D3

Optimal PL positions in a *D3_PL1* link, where the DCTs of *hc-2* and *hc-3* are identical with each other, are calculated in Eq. 3.18, Eq. 3.19 and Eq. 3.20.

$$n + m = WL \quad (3.18)$$

$$n - m = (ICT_{hc-2} - ICT_{hc-4})/0.2 = 515 \quad (3.19)$$

$$\text{where } ICT_{hc-2} = 350 \text{ and } ICT_{hc-4} = 247$$

$$n = (WL + 515)/2 \quad (3.20)$$

$$m = WL - n$$

The benefit of an optimally placed PL is shown in Figure 3.12 by comparing link BW variance of the *D3_PLno* and the *D3_PL1_opt* designs. The link BW is improved substantially including extension of the link MBR from 220 μm for *D3_PLno* to 955 μm for *D3_PL1_opt*.

3.3.4 Results of One Pipeline Latch Insertion

Figure 3.13 shows link BW of PLno links and PL1_opt links for the three different router designs. The PLno links in Figure 3.13(a) are links with no PL insertion and it is the same with Figure 2.20 with different link names. The PL1_opt links have one optimally placed PL, shown in Figure 3.13(b). Moreover, Table 3.3 compares properties of the six links in area, energy/flit and MBRs. Area and energy/flit are

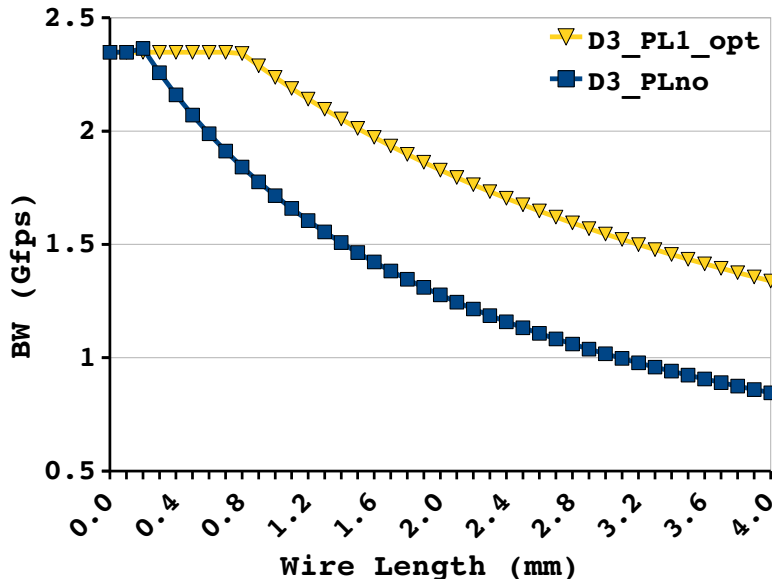


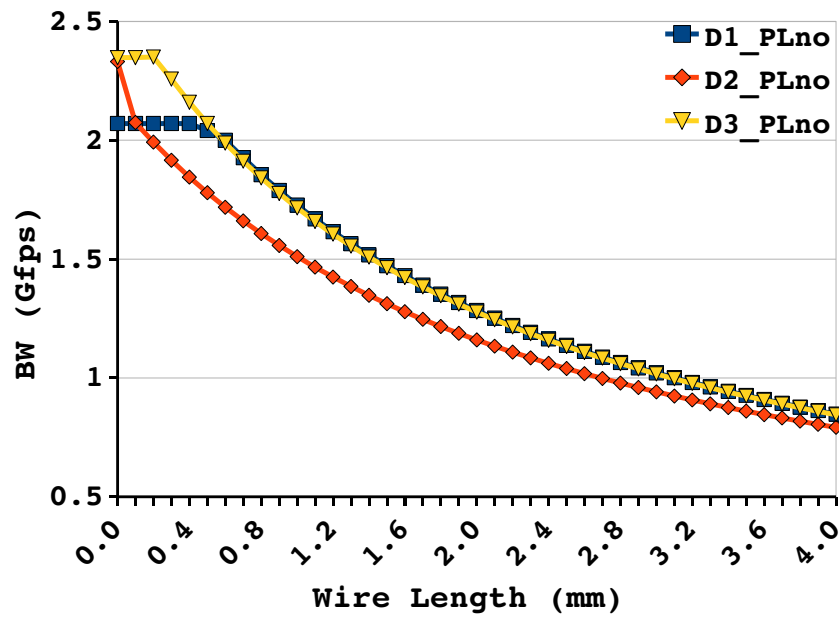
Figure 3.12: Link BW improvement of $D3_PL1$.

the values for a 44-bit wide link in the router designs. The area and energy/flit are the sum of corresponding values of one router and one PL in PL1_opt links.

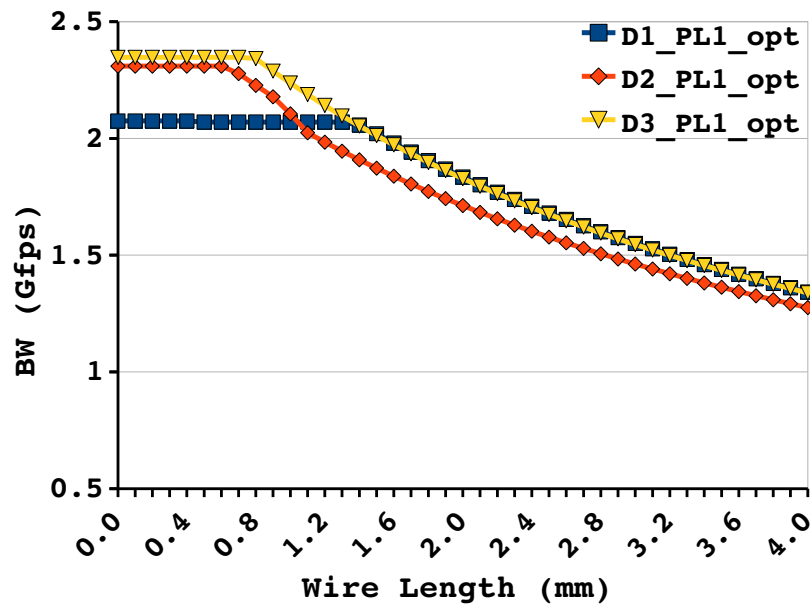
All three PL1_opt links have extended MBRs and better link BW across the whole range of wire lengths. Especially, the most beneficiary of the optimal PL insertion is the link with D2 routers. The $D2_PLno$ link shows the worst link BW in almost the entire range of wire length, due to its least robustness to link wire delay penalty. But, the $D2_PL1_opt$ link can be the most performance- and energy-efficient link particularly for under $1000\ \mu\text{m}$ of wire length. In this wire length range, it shows better link BW than $D1_PL1_opt$ link and comparable to that of $D3_PL1_opt$ link with less energy consumption per flit than the $D3_PL1_opt$ link.

The $D1_PL1_opt$ and the $D3_PL1_opt$ have very similar ICT with two external handshake cycles that handshake through the inserted PL. Therefore, they show identical link BW after $1500\ \mu\text{m}$, like the wire length range after $500\ \mu\text{m}$ in Figure 3.13(a).

The area overhead of PL1_opt links is insignificant with consideration of their link BW benefit. For instance, the $D2_PL1_opt$ link uses 10% more area, compared to the $D2_PLno$. Meanwhile, the energy overhead can be considerable since 42% more energy is consumed per flit in the $D2_PL1_opt$ link than $D2_PLno$.



(a) PLno links



(b) PL1_opt links

Figure 3.13: Link BW of three PLno and three PL1_opt links.

Table 3.3: Comparison of three PLno and three PL1_opt links.

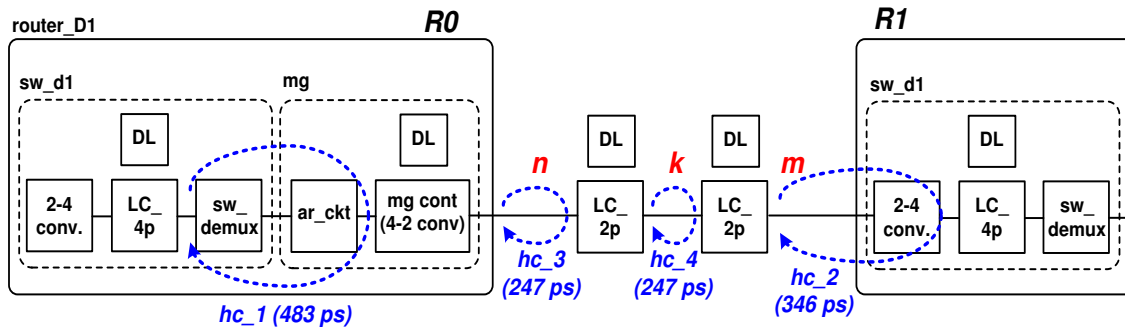
	Area(μm^2)	Energy/flit(pJ)	MBR(μm)
D1_PLno	3136	1.127	525
D1_PL1_opt	3537	1.620	1545
D2_PLno	4043	1.158	0
D2_PL1_opt	4444	1.651	775
D3_PLno	4990	1.575	220
D3_PL1_opt	5391	2.068	955

3.4 Optimal Positions of Two Pipeline Latches Placement

Inserting two PLs in a link is still beneficial when the link suffers from insufficient link BW compared to its traffic load, due to long link wire length which cannot be effectively covered by just one PL insertion. With similar concepts and methods of the one PL insertion, it is possible to calculate the optimal positions of two PLs in a link.

3.4.1 Optimal Position of Two Pipeline Latch with Router D1

Figure 3.14 shows $D1_PL2$ link which connects two D1 routers with two PLs. When one more PL is added, the $D1_PL2$ link has one more external handshake cycle, hc_4 , than the $D1_PL1$ design in Figure 3.3. Except for the new handshake cycle, hc_4 , the other three handshake cycles are identical with those of the link

**Figure 3.14:** Link of D1 router with two PLs: $D1_PL2$.

D1_PL1. In Figure 3.14, m , n and k represent link wire length of hc_2 , hc_3 and hc_4 , respectively.

First, the MBR of *D1_PL2* is extended by the new handshake cycle, hc_4 , and the MBR of hc_4 is calculated with the ICT of hc_1 in Eq. 3.21.

$$\begin{aligned}
 ICT_{hc_1} &\leq DCT_{hc_4} & (3.21) \\
 ICT_{hc_1} &\leq ICT_{hc_4} + 2 \times WD_{MBR} \\
 WD_{MBR} &\leq (ICT_{hc_1} - ICT_{hc_4})/2 \\
 &\text{with } ICT_{hc_1} = 483 \text{ and } ICT_{hc_4} = 247 \\
 WD_{MBR} &\leq 118 \text{ ps} \\
 MBR_{hc_4} &\leq (WD_{MBR} - 16) \times 10 = 1020 \mu\text{m}
 \end{aligned}$$

As a result, if two PLs are optimally placed in a *D1_PL2* link, the link BW can be maintained at its maximum, up to a wire length of $2565 \mu\text{m}$. This is the sum of the three MBRs of hc_2 , hc_3 and hc_4 of the *D1_PL2* link. The MBRs of cycles hc_2 ($1020 \mu\text{m}$) and hc_3 ($525 \mu\text{m}$) of *D1_PL2* are the same with those of the *D1_PL1*, as they have identical ICTs in both links.

There are three possible cases to calculate the optimal position of two PLs in a link, according to the relation between total link wire length and the MBR of the link. This is shown in Figure 3.15.

- *PL2_CASE 1*: Link WL is shorter than the MBR of the PL1 link, Figure 3.15(a).
- *PL2_CASE 2*: Link WL is longer than the MBR of the PL1, but shorter than the MBR of the PL2 link, Figure 3.15(b).
- *PL2_CASE 3*: Link WL is longer than the MBR of the PL2 link, Figure 3.15(c).

PL2_CASE 1 is the case where the total link wire is short enough so that the link BW maintains its maximum with one PL insertion. Thus, as in Figure 3.15(a), one of two PLs is placed at the output of the R0, such that PL1 responds to R0 as fast as possible and subsequently, it enables R0 to handle the next packet earlier giving better NoC performance. In this case, the wire length n is set to zero and k and

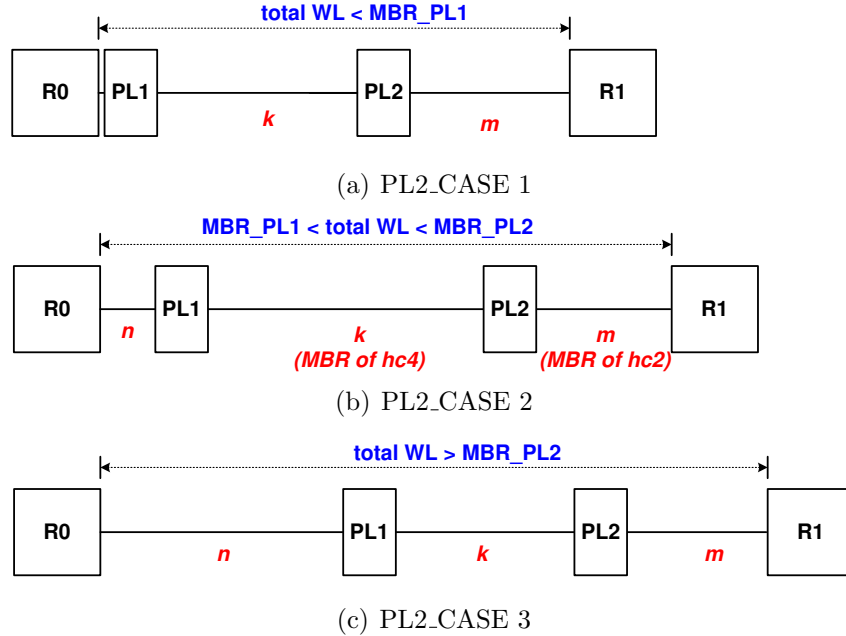


Figure 3.15: Three PL2 Cases depending on Total WL.

m are determined by the optimal position of one PL with corresponding total WL. In consequence, the link BW is maintained its maximum by the optimally positioned PL2. In addition, there is no BW benefit from inserting the second PL in a link as the second PL acts for just data buffering.

In *PL2_CASE 2*, the link wire length is longer than the MBR of a link with one PL, so the link BW increases when another PL is inserted. In order to maintain the maximum link BW, k and n are set with the MBR of each handshake cycle, as shown in Figure 3.15(b). Only n varies according to the total wire length and it is set by the residual wire length fraction after subtracting the MBR of PL1 from the total wire length. For instance, in a *D1_PL2* link with a $2000\ \mu\text{m}$ long wire, k is fixed to $1020\ \mu\text{m}$, m is set at $525\ \mu\text{m}$ and subsequently n is $455\ \mu\text{m}$. As a result, DCTs of all three external handshake cycles are equal to or less than the longest ICT of cycle *hc_1*, 483 ps, and therefore, the link BW is the maximum at 2.07 Gfps.

In *PL2_CASE 3* shown Figure 3.15(c), the total link wire length is longer than the MBR of PL2, so link BW begins to decrease. The optimal positions of the two PLs in *PL2_CASE 3* can be determined by a similar way used for calculating the optimal position of one PL in the previous section: the DCTs of three external handshake

cycles which handshake through two PLs are balanced and identical when two PLs are optimally placed.

For a *D1_PL2* link, the optimal positions of two PLs is determined by Eq. 3.22, Eq. 3.23 and Eq. 3.24.

$$n + k + m = WL \quad (3.22)$$

$$DCT_{hc-3} = DCT_{hc-4} \quad (3.23)$$

$$DCT_{hc-3} = DCT_{hc-2} \quad (3.24)$$

Eq. 3.23 can be transformed to Eq. 3.25 with wire length variables, n and k .

$$\begin{aligned} DCT_{hc-3} &= DCT_{hc-4} \\ ICT_{hc-3} + 2 \times WD_{hc-3} &= ICT_{hc-4} + 2 \times WD_{hc-4} \\ ICT_{hc-3} + 2 \times (0.1 \times n + 16) &= ICT_{hc-4} + 2 \times (0.1 \times k + 16) \\ n - k &= (ICT_{hc-4} - ICT_{hc-3})/0.2 = 0 \quad (3.25) \\ &\text{where } ICT_{hc-4} = 247 \text{ and } ICT_{hc-3} = 247 \end{aligned}$$

Obviously, the ICTs of cycles $hc-3$ and $hc-4$ are equal to each other so that wire length n and k are same when their DCTs are balanced.

Eq. 3.24 can be transformed similarly with wire length variables, n and m as in Eq. 3.26.

$$\begin{aligned} DCT_{hc-3} &= DCT_{hc-2} \\ ICT_{hc-3} + 2 \times WD_{hc-3} &= ICT_{hc-2} + 2 \times WD_{hc-2} \\ ICT_{hc-3} + 2 \times (0.1 \times n + 16) &= ICT_{hc-2} + 2 \times (0.1 \times m + 16) \\ n - m &= (ICT_{hc-2} - ICT_{hc-3})/0.2 = 495 \quad (3.26) \\ &\text{where } ICT_{hc-2} = 346 \text{ and } ICT_{hc-3} = 247 \end{aligned}$$

Finally, from Eq. 3.22, Eq. 3.25 and Eq. 3.26, the optimal position of two PLs in a *D1_PL2* link are determined by Eq. 3.27:

$$n = WL/3 + 165 \quad (3.27)$$

$$k = WL/3 + 165$$

$$m = WL/3 - 330$$

Figure 3.16 shows link BW improvement using two optimally placed PLs in a D1 link, as comparing with $D1_PLno$ and $D1_PL1_opt$. There is no link BW benefit in $PL2_CASE 1$ region since it can be covered enough by one PL. As seen in the $PL2_CASE 2$ region, the MBR of $D1_PL2_opt$ is extended to $2500\mu m$ from the $1500\mu m$ length of $D1_PL1_opt$. Note that there is a little inconsistency between the simulated and calculated MBR values. In the $PL2_CASE 3$ region, the link BW of $D1_PL2_opt$ begins to decrease as the link wire length is out of the MBR of $D1_PL2_opt$.

Through $PL2_CASE 2$ and $PL2_CASE 3$ regions, $D1_PL2_opt$ achieves 0.35 Gfps and 0.87 Gfps more link BW on average than the $D1_PL1_opt$ and $D1_PLno$ links, respectively. At $4000\mu m$ wire length, $D1_PL2_opt$ (1.69 Gfps) shows twice the link BW of the $D1_PLno$ design (0.85 Gfps).

This link BW benefit comes at the expenses of more energy consumption. As the D1 router has two data latches from one input to an output internally, two PLs in a link consume a similar amount of energy to what the router consumes. The $D1_PL2_opt$ link consumes 2.112 pJ per flit transfer which is almost twice energy than $D1_PLno$ link, at 1.127 pJ with 44-bit flit size. However, the energy overhead could be insignificant when compared to the energy dissipation by link wires. ORION estimates energy consumed by link wires 44-bits wide, with a 25% switching activity,

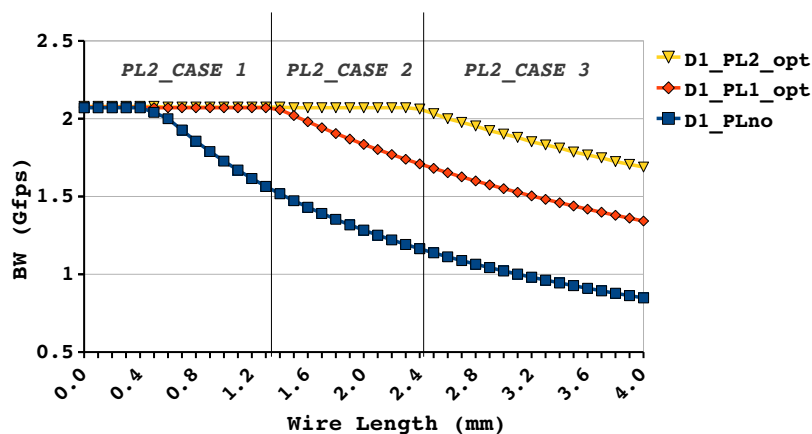


Figure 3.16: Link BW improvement of a link with router D1 and two optimal positioned PLs.

and $2000 \mu\text{m}$ long link at 43.2 pJ per flit which is $20\times$ the energy consumed by the $D1_PL2_opt$ design.

3.4.2 Optimal Position of Two Pipeline Latches with Router D2

A $D2_PL2$ link is shown in Figure 3.17 with one internal and three external handshake cycles. Cycle hc_4 is newly created by the additional PL from the $D2_PL1$ in Figure 3.9. Similar to the $D1_PL2$ link, the new hc_4 cycle has its own MBR in relation with the ICT of hc_2 , and is calculated in Eq. 3.28.

$$ICT_{hc_2} \leq DCT_{hc_4} \quad (3.28)$$

$$ICT_{hc_2} \leq ICT_{hc_4} + 2 \times WD_{MBR}$$

$$WD_{MBR} \leq (ICT_{hc_2} - ICT_{hc_4})/2 = 93.5$$

$$\text{with } ICT_{hc_2} = 430 \text{ and } ICT_{hc_4} = 243$$

$$MBR_{hc_4} \leq (WD_{MBR} - 16) \times 10 = 775 \mu\text{m}$$

Consequently, the size of MBR of $D2_PL2$ is $1550 \mu\text{m}$, the sum of MBRs of each external handshake cycle. The MBR of hc_2 and hc_3 are not affected by the additional PL so they are identical with those of the $D2_PL1$ link: $0 \mu\text{m}$ for cycle hc_2 and $775 \mu\text{m}$ for hc_3 .

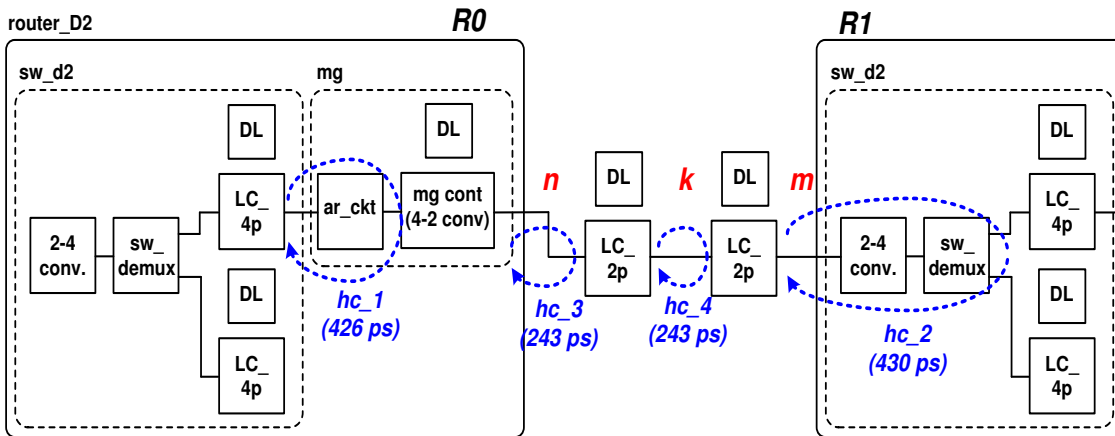


Figure 3.17: Link of D2 router with two PLs: $D2_PL2$.

Optimal positions of two PLs in a $D2_PL2$ link can be calculated similar to the $D1_PL2$ link. The MBR of $D2_PL1$ is $775\ \mu\text{m}$ and $1550\ \mu\text{m}$ for $D2_PL2$. In $PL2_CASE\ 1$ of $D2_PL2$, link wire length is under $775\ \mu\text{m}$. Variables k and m are assigned by the optimal position of one PL while n is set to zero. If wire length is between $775\ \mu\text{m}$ and $1550\ \mu\text{m}$, k and m are fixed at their MBR sizes of $775\ \mu\text{m}$ and $0\ \mu\text{m}$, respectively. The remaining wire length is assigned to n , as in $PL2_CASE\ 2$ of $D1_PL2$. For $PL2_CASE\ 3$, the optimal positions of two PLs in a $D2_PL2$ link is calculated by Eq. 3.29 which is driven using Eq. 3.22, Eq. 3.25 and Eq. 3.26 and ICTs of handshake cycles in $D2_PL2$ link.

$$n = WL/3 + 311.5 \quad (3.29)$$

$$k = WL/3 + 311.5$$

$$m = WL/3 - 623$$

Link BW improvement of two PLs in a $D2_PL2$ link is presented in Figure 3.18. Simulation results show that the MBR of a D2 link is extended to $1400\ \mu\text{m}$ and link BW is improved in the entire range of wire length.

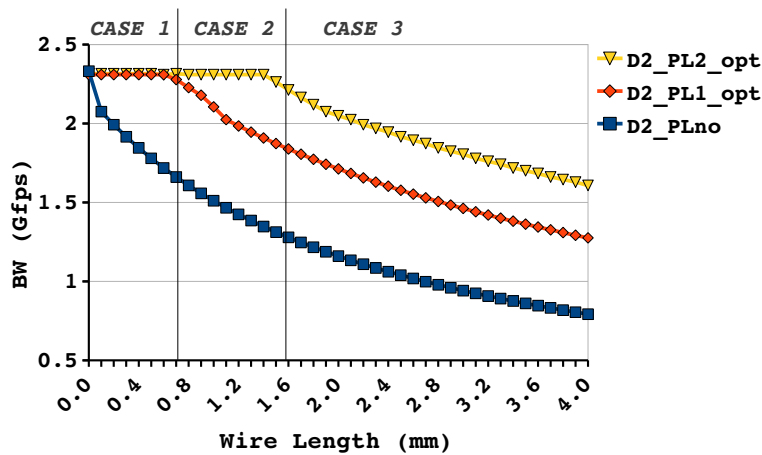


Figure 3.18: Link BW improvement with two optimal PLs in D2 link.

3.4.3 Optimal Position of Two Pipeline Latches with Router D3

Figure 3.19 shows a $D3_PL2$ link with D3 routers and two PLs. Given the ICTs in Figure 3.19 and Eq. 3.30 for that calculates the MBR of the new handshake cycle, hc_5 , the MBR of D3_PL2 design is shown to be $1690 \mu\text{m}$ with $735 \mu\text{m}$ of hc_4 and $220 \mu\text{m}$ of hc_2 .

$$DCT_{hc_1} = DCT_{hc_5} \quad (3.30)$$

$$ICT_{hc_1} = ICT_{hc_5} + 2 \times WD$$

$$WD = (ICT_{hc_1} - ICT_{hc_5})/2 = 89.5$$

$$\text{with } ICT_{hc_1} = 426 \text{ and } ICT_{hc_5} = 247$$

$$WL_{hc_5} = (WD - 16) \times 10 = 735 \mu\text{m}$$

In addition, the optimal positions of the two PLs in D3_PL2 link is estimated through Eq. 3.31 which is driven similarly to Eq. 3.29.

$$n = WL/3 + 165 \quad (3.31)$$

$$k = WL/3 + 165$$

$$m = WL/3 - 330$$

The BW improvement of a $D3_PL2$ link compared with other two D3 links is shown in Figure 3.20.

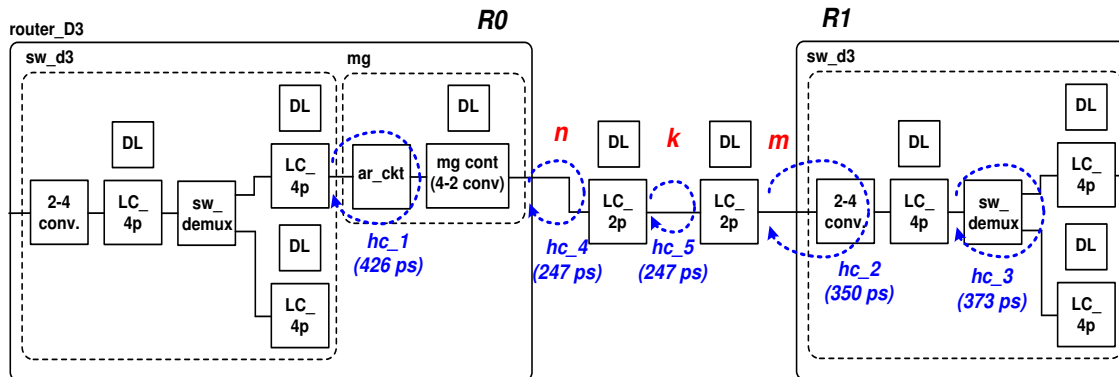


Figure 3.19: Link of D3 router with two PLs: $D3_PL2$.

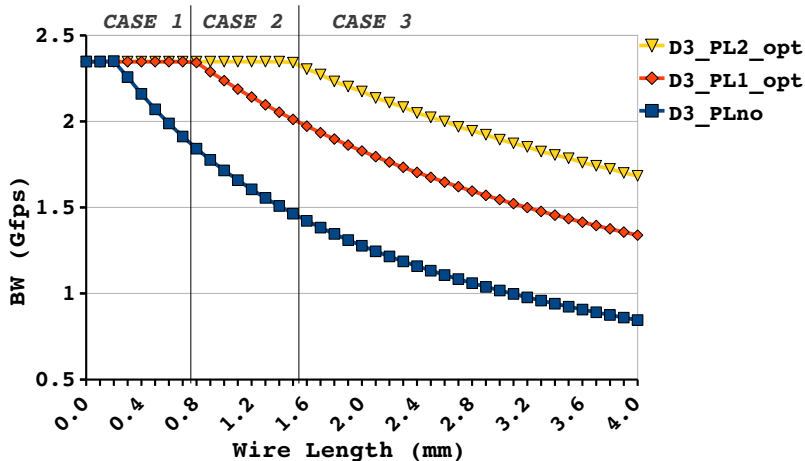


Figure 3.20: Link BW improvement of $D3_PL2$.

3.5 Link BW Comparison with Different PL Configurations

Eight distinctive asynchronous links have been implemented, according to their router designs and the number of PLs inserted in a link. Table 3.4 shows these eight links and classifies them into three types based on the number of pipelined data latch stages ($\#$ of DL) in each link. *TYPE 1* links have two data latches inside the routers without any link pipeline latches. *TYPE 2* links have three data latches: $D1_PL1$ and $D2_PL1$ have two internal data latches and one PL externally, while all three data latches are located inside the D3 router in a $D3_PLno$ link. Similarly, *TYPE 3* links have four data latches in each link.

Table 3.4: Eight asynchronous link designs with different routers and PL numbers.

	Figure	Router	$\#$ of PL	$\#$ of DL	Energy/flit(nJ)	TYPE
D1_PLno	2.15	D1	0	2	1.127	1
D2_PLno	2.17	D2	0	2	1.158	
D1_PL1	3.3	D1	1	3	1.620	2
D2_PL1	3.9	D2	1	3	1.651	
D3_PLno	2.19	D3	0	3	1.575	
D1_PL2	3.14	D1	2	4	2.113	3
D2_PL2	3.17	D2	2	4	2.144	
D3_PL1	3.11	D3	1	4	2.068	

Hence, links belonging to the same *TYPE* consume very analogous total dynamic energy per flit, as shown in the column of Energy/flit in Table 3.4. Energy/flit is measured with a 44-bit link width and 25% activity factor. PLs are inserted in their optimal position. The *Figure* column in the table represents the reference number of the figure corresponding to the particular link design.

In Figure 3.16, Figure 3.18 and Figure 3.20, different PL configurations were already compared, but they were performed with the identical router design, and thereby the energy consumption of each link was different: energy dissipation per flit of PL2 links are normally twice that of of corresponding PLno links. Thus, they might be not fair comparisons when based solely on link BW.

Accordingly, in this section, link BW is compared while equalizing energy consumption of all links under comparison.

Links of *TYPE 1* has already been compared with each other through Figure 2.18. The *D2_PLno* link has higher maximum BW, thanks to the more balanced two internal handshake cycles, *hc_1* and *hc_2*, than *D1_PLno*. But, it results in the longer ICT of *hc_2* which is a highly wire delay sensitive handshake cycle and hence produces worse link BW when wire delay penalty is included. Consequently, *D1_PLno* shows better performance in the whole range of link wire length except below $100\ \mu\text{m}$ long.

Figure 3.21 depicts the link BW variance of the three *TYPE 2* links in the range of link wire length up to 4.0 mm. Two PL1 links, *D1_PL1* and *D2_PL1*, effectively show

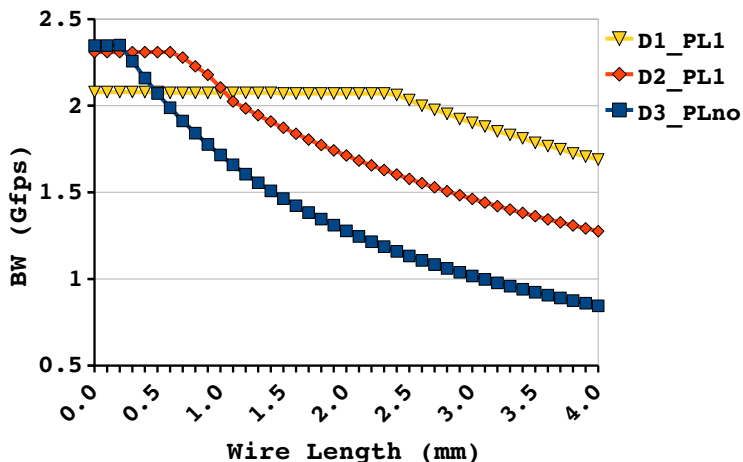


Figure 3.21: BW comparison of three links of *TYPE 2*.

better link BW than the *D3_PLno* link across the board. Actually, the comparison between two PL1 links and one PLno link can be considered as a comparison of the different impact of internal buffering and external buffering. The *D3_PLno* link has three data latches internally. Instead, the two PL1 links have one external data latch (PL) placed in the optimal position for maximizing link BW. The external data latch in the two PL1 links can be compared to one of the three internal data latches of the D3 router in the *D3_PLno* if it were placed external to the router. Consequently, the result indicates that external buffering is a more effective link design than internal buffering, given equal depth of buffering in links and routers, when considering the delay effect of wire delay on link BW. Additionally, it shows how one PL efficiently diminishes the effect of wire delay to improve link BW.

In regard to NoC performance, *D2_PL1* links can be recognized as the best design among three *TYPE 2* links, assuming that all links of an NoC are implemented with a single identical link design. The *D2_PL1* has higher link BW than the *D1_PL1* link up to a 1000 μm wire link, whereas *D1_PL1* provides better link BW after links that are 1000 μm long. Link BW improvement by PL insertion is employed generally after the NoC topology and floor plan are fixed through an NoC floor-planning tool. Ideally, most of the high traffic links have been optimized to have relatively short wire lengths, while low traffic links would allow long link wires. Furthermore, the NoC performance saturation primarily relies on the link BW of the high traffic links. The impact of low traffic links is usually insignificant. In consequence, the higher BW of the *D2_PL1* design in short wire lengths is preferable to the better BW in long wire length of the *D1_PL1* link for NoC performance.

Link BW of the three *TYPE 3* links is shown in Figure 3.22. All three links improve their BW with one more PL in a link than the corresponding *TYPE 2* links. With the same reason as in the *TYPE 2* links, it is expected that both *D2_PL2* and *D3_PL1* link designs will be the best design for NoC power and performance, since the two designs maintain similarly higher link BW in the range of short wire length than the *D1_PL2* link.

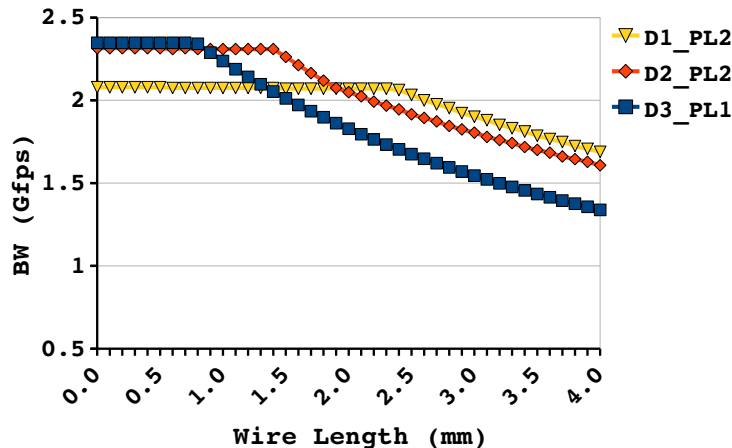


Figure 3.22: BW comparison of three links of *TYPE 3*.

3.6 Summary

PLs in asynchronous communication links are substantially beneficial for improving link BW by mitigating the link wire delay effect on link BW. Inserting one PL into a link increases the link BW simply and effectively. Furthermore, inserting two PLs in a link can achieve up to twice the link BW, compared to a corresponding link with no PL. In addition, the extension of the MBRs of a link by PL insertion can give more flexibility to an NoC floor plan. If any two controllers (router/PL) are placed within the MBR of the link, there is no BW degradation due to the link wire delay. In other words, any two controllers can be placed freely without considering a decrease of the link BW within the MBR.

In fact, the PL insertion is not the only way of increasing link BW when necessary. The simpler way is to widen the data path width of a link. If the data-path width of a link is doubled, then twice the link BW can be achieved. Moreover, even though energy dissipation per flit in the wide data-path is doubled, total energy consumption of the link does not increase, since the total number of packets of the link is cut in half, thanks to the doubly wide data-path. However, area overhead by the wide data-path link is substantial. Total wire routing area of the whole NoC will be approximately twice and thereby, total leakage power consumption is twofold as well. In addition, it may require a new NoC topology or floor plan. In consequence, it can hardly be an efficient way of improving one link BW as much as the PL insertion.

On the other hand, it is possible to use a wide data-path only in a specific link . However, due to inconsistency of the data-path width between the link and routers at the both ends of the link, extra circuitry is necessary for converting the data-path width: narrow to wide in a sending router and wide to narrow in a receiving router. Subsequently, these additional circuits diminish the efficiency of the wide data-path link. The doubly wide data-path link might not achieve as much as expected, twice link BW, since the logic delays of two converters are not insignificant. Moreover, energy dissipation by the converters is not much less than two PLs. The delay and energy overhead of such path width converts can be seen later in Section 4.3. Those converters in Section 4.3 were designed for a different purpose, but the design concepts are almost equivalent. Additionally, the wide data-path link still uses twice the wire routing area of the link.

In contrast, as shown through this section, inserting PL can control only a specific link BW with increasing some energy dissipation, without requiring any additional change of other design parameters, such as router designs, NoC topology or floor plan. Moreover, the link BW can be improved as much as necessary through adjusting the number of PLs inserted, and the inserted PL operates seamlessly with existing routers, as all of them are asynchronous. In consequence, PL insertion in an asynchronous communication link is a promising solution for not only relieving negative impact of link wire delay on link BW but controlling individual link BW effectively.

The PL insertion can be exploited as an effective way of enhancing link BW of only such links which have limited BW by link wire delay, compared to their BW requirements.

CHAPTER 4

ASYNCHRONOUS NOC OPTIMIZATON

For the optimization of asynchronous NoC designs, links of an NoC are classified into three types according to their properties: performance-critical, area-critical and energy-critical links. The performance-critical links are the links which play the most important role in determining NoC performance. They are normally highly utilized links with high traffic loads. Area-critical links are those which have excess link BW compared to their BW requirement. So, the wire routing area of these links can be saved through adjusting the excessive link BW by means of narrowing the data-path width of the link. The energy critical links are those links where wire energy dissipation contributes significantly to the total energy consumption of an NoC.

Three optimization methods, PL insertion, Narrow Data-Path (NDP) and Double-Spacing (DS), are presented for each type of links, respectively, in the following sections.

4.1 Analytical Model for Link BW Estimation

In order to employ a suitable optimization method, the type of each link in an NoC should be identified, prior to the actual NoC optimization process. In particular, two link types, performance- and area-critical links, distinguish themselves mainly based on their link utilization. Therefore, it is required to know the assigned link BW of each link to calculate link utilization and given BW requirement of each link, according to the communication characteristic of a target SoC system.

In addition, two different types of link BW exist: available BW (avBW) and achievable BW (acBW). The avBW of a link is what the link can provide maximally with no consideration of the packet contention. So, the avBW of a link can be

estimated simply with the router design of the link and the link wire length. Thus, all link BW mentioned in previous sections mean the avBW.

The acBW is a BW which a link can actually achieve with consideration of possible packet contention with other flows. Sharing physical links with multiple packets flows is the fundamental feature of NoC designs. Hence, contention between packet flows is inevitable. For instance, in a three-port router, two input flows share one output link. When both input flows are trying to transfer packets to the shared output link at the same time, only one of two inputs can access the output link. Meanwhile, the other input flow is stalled and has to await until the prior flow is completed.

So, the acBW is apparently a more correct value than the avBW, for computing the link utilization. However, the estimation of the acBW of a link is not as simple as that of the avBW, since it requires one to consider the possibility of packet contention which depends on the packet transfer rate of all packet flows related to the flow of the link. Thus, an analytical model is required in order to accurately estimate the acBW of each link of an NoC. The analytical model for link acBW was derived based on [22] which presents analytical packet delay model in virtual channeled wormhole networks. The analytical model in this work is for a specific network composed of bidirectional three-port routers with fair arbitration.

Figure 4.1 illustrates a flow of *input i* and two other interrelated input flows, *input j* and *input k* in a three-port router. The *input i* flow is divided into two internal

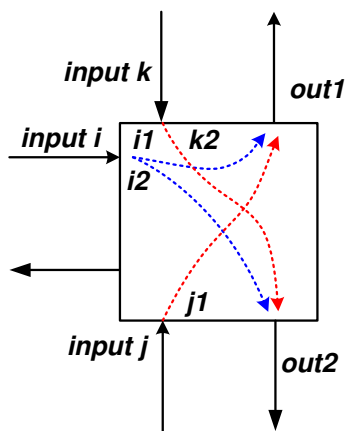


Figure 4.1: Flows of *input i* and other two related inputs, *input k* and *input j* in a three-port router

flows, $i1$ and $i2$: $i1$ is a flow from *input* i to *out1*, while $i2$ is from *input* i to *out2*. Flow $i1$ shares output link *out1* with another input flow *input* j and $i2$ uses the *out2* link with the *input* k flow.

In order to formalize the analytical model for the acBW estimation of the *input* i link, the following notation will be used:

$$\begin{aligned}
\lambda_{i1} &= \text{average packet transfer rate of } \textit{input } i \text{ to } \textit{out1}, \\
\lambda_{i2} &= \text{average packet transfer rate of } \textit{input } i \text{ to } \textit{out2}, \\
\lambda_i &= \text{average packet transfer rate of } \textit{input } i, \lambda_i = \lambda_{i1} + \lambda_{i2}, \\
\lambda_{j1} &= \text{average packet transfer rate of } \textit{input } j \text{ to } \textit{out1}, \\
\lambda_{k2} &= \text{average packet transfer rate of } \textit{input } k \text{ to } \textit{out2}, \\
R_{i1} &= \text{packet transfer ratio of flow } i1 \text{ to flow } i, R_{i1} = \lambda_{i1}/\lambda_i, \\
R_{i2} &= \text{packet transfer ratio of flow } i2 \text{ to flow } i, R_{i2} = \lambda_{i2}/\lambda_i, \\
R_{s1} &= \text{stalled packet ratio of flow } i1 \text{ to flow } j1, R_{s1} = \lambda_{j1}/\lambda_{i1}, \\
R_{s2} &= \text{stalled packet ratio of flow } i2 \text{ to flow } k2, R_{s2} = \lambda_{k2}/\lambda_{i2}, \\
avBW_i &= \text{avBW of } \textit{input } i \text{ link}, \\
BW_{out1} &= \text{acBW of } \textit{out1}, \\
BW_{out2} &= \text{acBW of } \textit{out2}, \\
BW_{i1} &= \text{BW which the flow } i1 \text{ can utilize from the total BW of } \textit{out1}, BW_{out1}, \\
&\quad \text{in consideration of packet contention with flow } j1, \\
BW_{i2} &= \text{BW which the flow } i2 \text{ can utilize from the total BW of } \textit{out2}, BW_{out2}, \\
&\quad \text{in consideration of packet contention with flow } k2.
\end{aligned}$$

Eq. 4.1 models the acBW of *input* i which is determined by the packet transfer ratio to each output link, R_{i1} and R_{i2} , and the BW of the two output links assigned to *input* i , BW_{i1} and BW_{i2} .

$$\text{acBW of } \textit{input } i = R_{i1} \times BW_{i1} + R_{i2} \times BW_{i2} \quad (4.1)$$

where BW_{i1} and BW_{i2} can be estimated based on three different stall conditions of the $i1$ and $i2$ flows, respectively. Equations for the estimation of the BW_{i1} follow with three stall conditions, represented by R_{s1} . The estimation of BW_{i2} can be done in the identical way.

- **Condition 1: $R_{s1} = 0$** - Packets of the flow $i1$ do not contend, as there is no packet transfer in the flow $j1$ which shares output link *out1*. So, the flow $i1$ can exploit

the whole link BW of the *out1*, BW_{out1} and consequently, the BW_{i1} is determined by the smaller of $avBW_i$ and BW_{out1} :

$$BW_{i1} = \min(avBW_i, BW_{out1}) \quad (4.2)$$

• **Condition 2: $0 < R_{s1} < 1$** - The flow *i1* is possibly contending with the other flow *j1*. R_{s1} is less than 1, which means that the packet transfer rate of the flow *i1* (λ_{i1}) is higher than that of the related flow *j1* (λ_{j1}). Therefore, some packets of the flow *i1* are stalled by the flow *j1*, whereas others can be transferred to the *out1* link without contention. BW_{i1} can be written as

$$BW_{i1} = (1 - R_{s1}) \times \underbrace{\min(avBW_i, BW_{out1})}_a + R_{s1} \cdot \underbrace{\min(avBW_i, \frac{BW_{out1}}{2})}_b \quad (4.3)$$

where term *a* is the BW of nonstalled packets which is determined to be the lesser of $avBW_i$ and BW_{out1} , and term *b* is the BW of stalled packets, the lesser BW of either $avBW_i$ or $BW_{out1}/2$. The term $BW_{out1}/2$ is the packet transfer rate of two contending packets in an *out1* link. As MUTEX element is used in arbitrating flow contention, two flows are served alternatively in contention.

• **Condition 3: $R_{s1} \geq 1$** - If the packet transfer rate of the flow *j1* (λ_{j1}) is equal or greater than that of the flow *i1* (λ_{i1}), all packets from the *input i* to the *out1* are always stalled by the flow *j1* stochastically. Consequently, the flow *i1* can only utilize half of the link BW of *out1*, $BW_{out1}/2$, as in Eq. 4.4. In addition, Eq. 4.4 is the case of Eq. 4.3 with setting R_{s1} to 1.

$$BW_{i1} = \min(avBW_i, \frac{BW_{out1}}{2}) \quad (4.4)$$

A complete form of the equations for estimating the BW_{i1} is shown in Eq. 4.5. Equally, equations for the BW_{i2} can be written as Eq. 4.6 with corresponding variables for the flow *i2*.

$$BW_{i1} = \begin{cases} \min(avBW_i, BW_{out1}) & R_{s1} = 0 \\ (1 - R_{s1}) \times \min(avBW_i, BW_{out1}) + \\ \quad R_{s1} \cdot \min(avBW_i, \frac{BW_{out1}}{2}) & 0 < R_{s1} < 1 \\ \min(avBW_i, \frac{BW_{out1}}{2}) & R_{s1} \geq 1 \end{cases} \quad (4.5)$$

$$BW_{i2} = \begin{cases} \min(avBW_i, BW_{out2}) & R_{s2} = 0 \\ (1 - R_{s2}) \times \min(avBW_i, BW_{out2}) + \\ R_{s2} \cdot \min(avBW_i, \frac{BW_{out2}}{2}) & 0 < R_{s2} < 1 \\ \min(avBW_i, \frac{BW_{out2}}{2}) & R_{s2} \geq 1 \end{cases} \quad (4.6)$$

Two examples are presented to demonstrate the accuracy of the analytical model of link acBW estimation. The first example was performed without stall conditions, while the second one was experimented with stall conditions.

Figure 4.2 illustrates an NoC system used for the first example. The example NoC is composed of four Processing Elements (PEs) connected with two D1 routers. Numbers inside a pair of links represent link wire length (μm) and avBW (Gfps) in parenthesis of the links. Numbers in percentage are the packet transfer ratios of an input flow to one of two output links. In simulation, only PE0 sends out packets to two other PEs, PE2 and PE3: 20% of packets to PE2 and 80% to PE3. The acBW of link *R0_C_O* is a link of interest. Therefore, the *R0_C_O* link corresponds to *input i* in Figure 4.1, and *R1_B_O* and *R1_A_O* are the *out1* and *out2* links, respectively, while *R1_A_I* is *input j* and *R1_B_I* is *input k* with respect to the *input i*, *R0_C_O*. The actual parameters values for estimating the acBW of *R0_C_O* follow, based on the simulation conditions:

$$\begin{aligned} \lambda_i &= \text{average packet rate of flow in } R0_C_O, \\ \lambda_{i1} &= 0.8 \times \lambda_i, \\ \lambda_{i2} &= 0.2 \times \lambda_i, \\ \lambda_{j1} &= 0, \\ \lambda_{k2} &= 0, \\ R_{i1} &= 0.8, \\ R_{i2} &= 0.2, \\ R_{s1} &= 0.0, \\ R_{s2} &= 0.0, \\ avBW_i &= 1.62 \text{ Gfps}, \\ BW_{out1} &= 1.47 \text{ Gfps}, \\ BW_{out2} &= 1.29 \text{ Gfps}, \\ BW_{i1} &= \min(1.62, 1.47) = 1.47 \text{ Gfps} \\ BW_{i2} &= \min(1.62, 1.29) = 1.29 \text{ Gfps} \end{aligned}$$

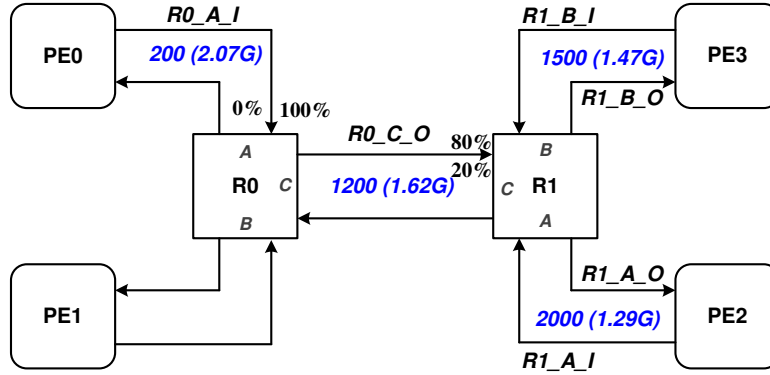


Figure 4.2: NoC example with traffic pattern for BW estimation model without stall condition

where λ_i is identical to the packet injection rate of PE0, since all packets from PE0 pass through link $R0_C_O$. This example is for the case without stall condition, so no packet is from $R1_A_I$ and $R1_B_I$ and subsequently, R_{s1} and R_{s2} are zero. The $avBW_i$ of $R0_C_O$ is 1.62 Gfps which is determined by the $1200 \mu\text{m}$ link wire length with D1 router. Variables BW_{out1} and BW_{out2} give the acBW of each output link, $R1_B_O$ and $R1_A_O$, and they are equal to their avBW because the two links are connected directly with the receiver of corresponding PE, respectively. In such links, no stall occurs as the links are not shared with any other flow and it is assumed that all receivers have infinite packet buffers inside.

The variables BW_{i1} and BW_{i2} are calculated using Eq. 4.5 and Eq. 4.6. Both are limited by the lower link BW of the output links, rather than the avBW of input link. Finally, by substituting actual values into Eq. 4.1, the acBW of link $R0_C_O$ is estimated as:

$$\begin{aligned} acBW_{R0_C_O} &= R_{i1} \times BW_{i1} + R_{i2} \times BW_{i2} \\ &= 0.8 \times 1.47 + 0.2 \times 1.29 = \mathbf{1.43} \text{ Gfps} \end{aligned} \quad (4.7)$$

Figure 4.3 presents simulation results as a function of varying the packet injection rate of PE0. Sim_BW represents link BW of $R0_C_O$ measured in the simulation. Avg_L is average latency of packets and it is aligned to the right-hand side Y-axis of the figure. It can be seen that the analytical model estimation closely predicts acBW of $R0_C_O$. When the packet injection rate is over the estimated acBW, 1.43 Gfps,

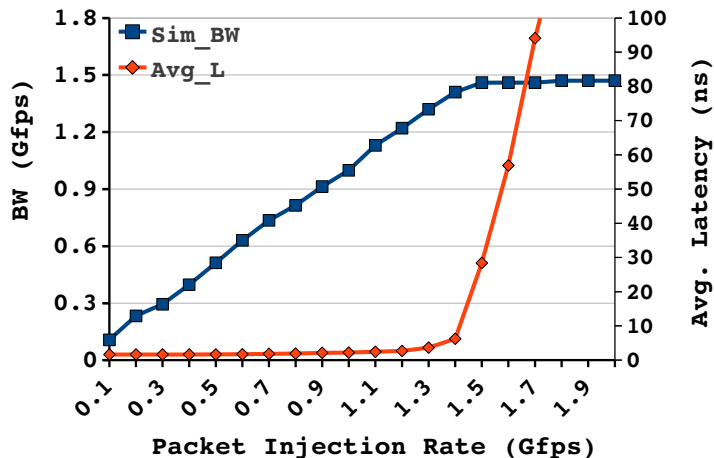


Figure 4.3: Simulation result of $R0_C_O$ link BW without stall condition.

the link $R0_C_O$ begins to be fully utilized and thereby Avg_L increases dramatically. Sim_BW is also saturated to 1.47 Gfps which approximately conforms to the estimated acBW of $R0_C_O$.

The second example with stall conditions was performed with some modification of the simulation in the first example. As shown in Figure 4.4, two additional packet flows are generated in $R1_A_I$ and $R1_B_I$, while other parameters are not changed from Figure 4.2. PE2 sends packets to PE3 (λ_{j1}) with 40% of the packet rate of PE0, in order to cause contention to the flow from $R0_C_O$ to $R1_B_O$. Similarly, $R1_B_I$ link has a packet flow directed to PE2 (λ_{k2}) with 10% of the packet rate of PE0.

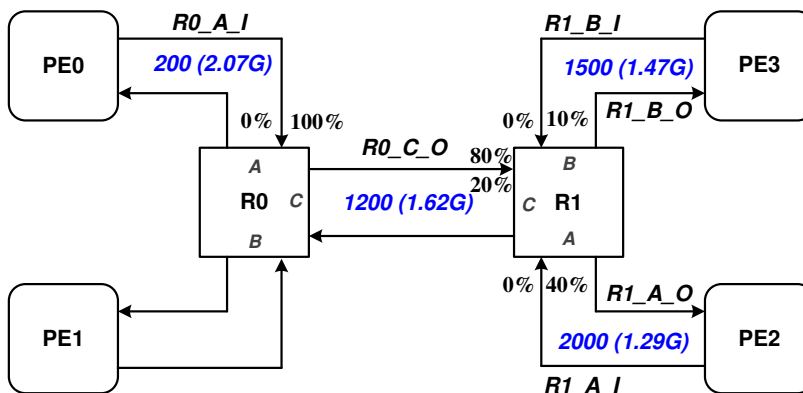


Figure 4.4: NoC example for BW estimation with stall conditions.

Both new flows make 50% of packets of the flow in $R0_C_O$ to experience contention in their output links, as represented by the stall rates, R_{s1} and R_{s2} .

Parameters for estimating the acBW of $R0_C_O$ are below:

$$\begin{aligned}
\lambda_i &= \text{average packet transfer rater of flow in } R0_C_O, \\
\lambda_{i1} &= 0.8 \times \lambda_i, \\
\lambda_{i2} &= 0.2 \times \lambda_i, \\
\lambda_{j1} &= 0.4 \times \lambda_i, \\
\lambda_{k2} &= 0.1 \times \lambda_i, \\
R_{i1} &= 0.8, \\
R_{i2} &= 0.2, \\
R_{s1} &= 0.4/0.8 = \mathbf{0.5}, \\
R_{s2} &= 0.1/0.2 = \mathbf{0.5}, \\
avBW_i &= 1.62 \text{ Gfps}, \\
BW_{out1} &= 1.47 \text{ Gfps}, \\
BW_{out2} &= 1.29 \text{ Gfps}, \\
BW_{i1} &= (1 - 0.5) \times \min(1.62, 1.47) + 0.5 \times \min(1.62, 1.47/2) = \mathbf{1.11} \text{ Gfps} \\
BW_{i2} &= (1 - 0.5) \times \min(1.62, 1.29) + 0.5 \times \min(1.62, 1.29/2) = \mathbf{0.964} \text{ Gfps}
\end{aligned}$$

where the $avBW_i$, BW_{out1} and BW_{out2} are identical to those of the first example, since they are determined by link wire lengths, regardless of the packet transfer rate of flows. The BW_{i1} variable is reduced to 1.11 Gfps from 1.47 Gfps and BW_{i2} is only 0.964 Gfps, rather than 1.29 Gfps, as 50% of packets of the $i1$ and $i2$ flows are stalled and can utilize only half of their output link BW.

Eq. 4.8 estimates the acBW of $R0_C_O$ for the simulation parameters.

$$\begin{aligned}
acBW_{R0_C_O} &= R_{i1} \times BW_{i1} + R_{i2} \times BW_{i2} \\
&= 0.8 \times 1.11 + 0.2 \times 0.964 = \mathbf{1.08} \text{ Gfps}
\end{aligned} \tag{4.8}$$

The simulation results of the second example, in Figure 4.5, show that Avg_L grows substantially and Sim_BW maintains 1.11 Gfps, when the packet injection rate is over 1.1 Gfps. In consequence, the analytical model can adequately estimate the acBW of a link with stall condition as well.

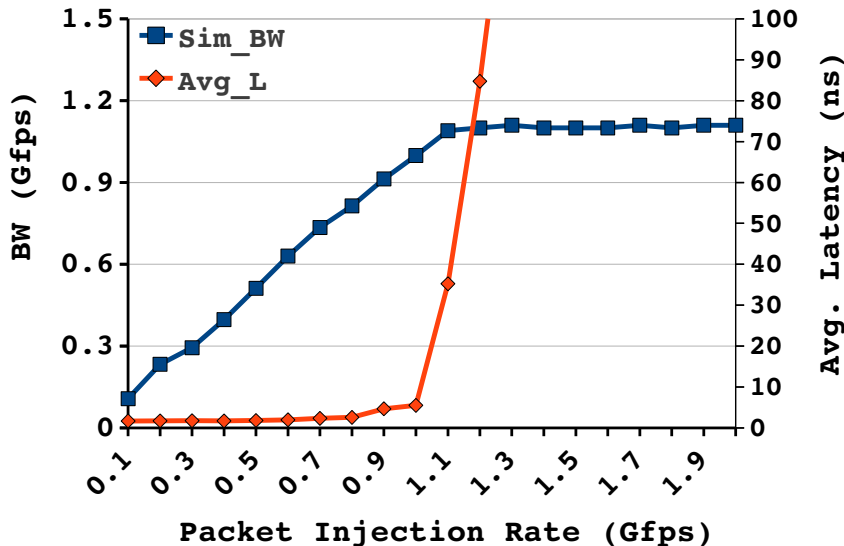


Figure 4.5: Simulation result of BW estimation with stall condition

4.2 Performance-Critical Link Optimization: PL Insertion

The performance-critical links in an NoC are highly utilized with higher traffic loads than other links and subsequently, their impact on the NoC performance is substantial. Therefore, increasing the BW of such links can lead to noticeable enhancement of the NoC performance.

As presented through Section 3, inserting PLs in an asynchronous link can increase the link BW as diminishing link wire delay effect. So, the PL insertion on performance-critical links can be employed as an NOC design optimization method, especially for NOC performance improvement.

In order to present the NoC performance benefit from the PL insertion optimization, additional simulation was performed with an NoC design, NoC_PL, illustrated in Figure 4.6. In fact, the NoC_PL is the one with identical packet flows and simulation conditions with the NoC in Figure 4.4, except two PLs, $P1$ and $P2$, inserted into the $R1_B_O$ and $R0_C_O$ links. This enables a performance comparison between two NoC designs in the same condition excluding the PL insertion. For the brevity of explanation, hereafter the NoC in Figure 4.4 is referred to as NoC_Init as indicating that it is the initial NoC before any optimization is performed.

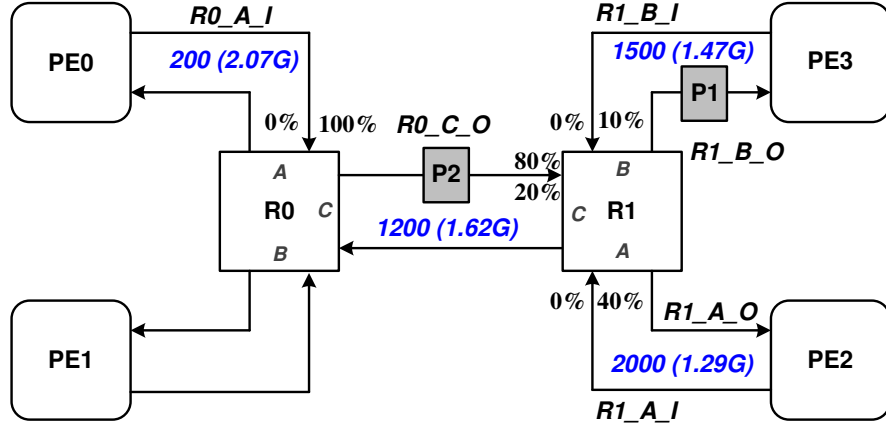


Figure 4.6: NoC example with PL insertion for performance optimization: NoC.PL

The performance of the NOC_Init design was limited by the low acBW of $R0_C_O$, 1.11 Gfps, which was determined primarily by the low acBW of the $R1_B_O$ link. So, one PL, $P1$, is inserted to the link $R1_B_O$ which increases its link BW to 2.07 Gfps, the maximum throughput of the D1 router, as eliminating link wire delay impact on the link BW. As presented in Section 3.3, a link with D1 routers has $1545 \mu\text{m}$ MBR with one optimally placed PL. As a result, the acBW of the $R0_C_O$ increase to 1.26 Gfps from 1.08 Gfps as in Eq. 4.9 where identical parameters with the NOC_Init example are not shown.

$$BW_{out1} = 1.47 \rightarrow \mathbf{2.07} \text{ Gfps}, \quad (4.9)$$

$$BW_{i1} = (1 - 0.5) \times \min(1.62, \mathbf{2.07}) + \\ 0.5 \times \min(1.62, \mathbf{2.07}/2) = \mathbf{1.33} \text{ Gfps}$$

$$\text{acBW}_{R0_C_O} = R_{i1} \times BW_{i1} + R_{i2} \times BW_{i2} \\ = 0.8 \times \mathbf{1.33} + 0.2 \times 0.964 = \mathbf{1.26} \text{ Gfps}$$

Inserting $P1$ in the link $R1_B_O$ results in link $R0_C_O$ becoming the performance bottleneck link in the path from PE0 to PE3. Nonstalled packets of the $R0_C_O$ flow directed to PE3 are limited by the low avBW of $R0_C_O$, 1.62 Gfps, as seen in the first term of the equation for BW_{i1} in Eq. 4.9. Thus, another PL, $P2$, is inserted into the $R0_C_O$ link leading to increasing the avBW of the link to 2.07 Gfps from 1.62 Gfps. Inserting the second PL enables the nonstalled packet of $R0_C_O$ to be transferred

at the maximum BW, 2.07 Gfps, and consequently, increases BW_{i1} to 1.55 Gfps from 1.33 Gfps as in Eq. 4.10 with estimated acBW of $R0_C_O$, 1.43 Gfps. On the contrary, BW_{i2} is not affected by the increase of avBW of $R0_C_O$ since it is still limited by the lower acBW of the output link, 1.29 Gfps of $R1_A_O$.

$$avBW_i = 1.6 \rightarrow \mathbf{2.07} \text{ Gfps}, \quad (4.10)$$

$$BW_{i1} = (1 - 0.5) \times \min(\mathbf{2.07}, 2.07) + 0.5 \times \min(\mathbf{2.07}, 2.07/2) = \mathbf{1.55} \text{ Gfps}$$

$$BW_{i2} = (1 - 0.5) \times \min(\mathbf{2.07}, 1.29) + 0.5 \times \min(\mathbf{2.07}, 1.29/2) = 0.964 \text{ Gfps}$$

$$\begin{aligned} acBW_{R0_C_O} &= R_{i1} \times BW_{i1} + R_{i2} \times BW_{i2} \\ &= 0.8 \times \mathbf{1.55} + 0.2 \times 0.964 = \mathbf{1.43} \text{ Gfps} \end{aligned}$$

Figure 4.7 compares average packet latency of the NoC_Init (Avg_L) and the NoC_PL (Avg_L_PL). The performance benefit from PL insertion can be seen in that the performance saturation point of the NOC_PL is extended to 1.4 Gfps and the average packet latency is enhanced drastically, especially, after the packet injection rate 1.1 Gfps, the saturation point of NOC_Init.

Clearly, PL insertion will cause an increase in NoC energy dissipation. Considering that a D1 router has two data latches internally from an input to an output port,

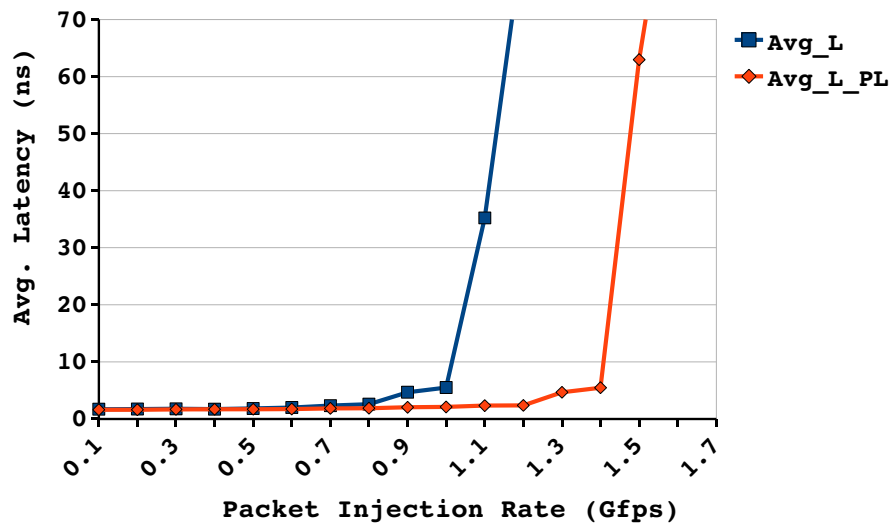


Figure 4.7: NoC performance comparison between NOC_Init and NOC_PL

inserting one PL in a link increase router logic energy dissipation per flit in the link by approximately 50%. In general, however, energy dissipation by the NoC components, like routers and PLs, is a relatively small component of the total NoC energy, when compared to the energy expended in the link wires. For instance, energy consumption of a link with 500 μm long and 34-bit link width is 8.876 pJ with 25% switching activity. This is 23 times the energy consumption by one PL with the same flit width. So, the additional energy overhead by the PL insertion method might not be significant.

4.3 Area Critical Link Optimization: Narrow Data-Path

Normally, an NoC design has the same data-path width in all links, and the selection of the size of data-path width is one of the critical NoC design parameters. If the data-path width is wider than necessary, an NoC performs well but wastes lots of resources, especially wire routing area and leakage power. On the contrary, if the data-path width is narrower than the BW requirements of the target SoC system, many links in the NoC suffer from deficient link BW and the NoC performance will be unacceptable.

In consideration of the NoC performance, the BW of high trafficked links are the main determining factor for the size of the data-path width of the whole NoC design. Inevitably, however, there always exist such links which have low BW requirements, but for which link BW is designed to be much greater than required, due to the wider data-path width chosen for high trafficked links.

In such a link with excessive link BW, the over-invested resource, in particular, wire routing area, can be saved by adjusting link BW properly by means of halving the data-path width. As a result, the wire routing area of the link is simply cut in half. This is allowable only when the reduced link BW is still sufficient to handle its BW requirement properly, as the link BW is reduced by half as well, due to the half size data-path width.

The NDP (Narrow Data-Path) method is employed to optimize an NoC design, by leveraging the slack of link BW in low trafficked links for saving wire routing

area. The BW reduction of low trafficked links usually does not affect the whole NoC performance. Moreover, since a topology generation tool should optimize an NoC floor plan focusing mainly on the high traffic links, it produces a floor plan of an NoC where high traffic links have short wire lengths because this will reduce link wire energy, whereas low traffic links may have relatively long wire lengths. Accordingly, the reduction of wire routing area in low trafficked links through the NDP method can contribute considerably to saving total wire routing area.

For the NDP optimization method, NDP_NW (Narrow-Wide) and NDP_WN (Wide-Narrow) modules were designed. Figure 4.8 depicts different usages of two modules. The NDP_NW is the one which converts a data-path width from narrow to wide. It can be used when the data-path width of a link is narrower (16-bit) than a connected router input (32-bit), such as a link between a sender (SEND) and a router (R0). Meanwhile, the NDP_WN is inserted in such a link between an output of R1 and a receiver (RCV), as converting 32-bit data into 16-bit data before injecting it into a narrow data-path link.

Designs of two NDP modules are presented in Figure 4.9 and Figure 4.10 where the wide data-path width is 32-bit, the narrow data-path width is 16-bit and the routing address is a 2-bit wide signal. The NDP_NW module requires two 16-bit data latches (DL) in order to temporarily store the 16-bit data, before forwarding them to the 32-bit data-path simultaneously. The LHS channel (lr and la) of the NDP_NW needs two handshake cycles, in order to pass a 32-bit data to its RHS channel (rr and ra) which runs only one handshake cycle. The NDP_WN uses a 16-bit data MUX, as it forwards half of the 32-bit data at a time. One handshake cycle of the LHS channel (lr and la) of the NDP_WN is completed in conjunction with two handshake cycles of its RHS channel (rr and ra).

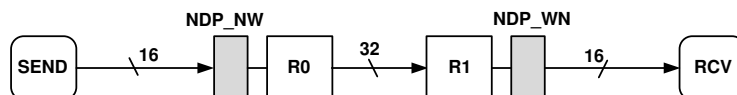


Figure 4.8: Usage of NDP modules.

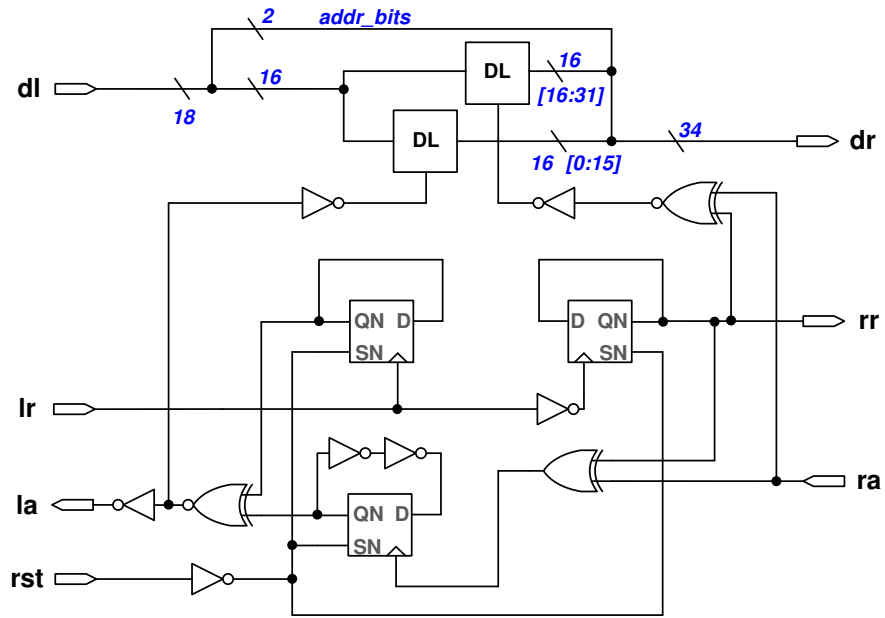


Figure 4.9: NDP_NW.

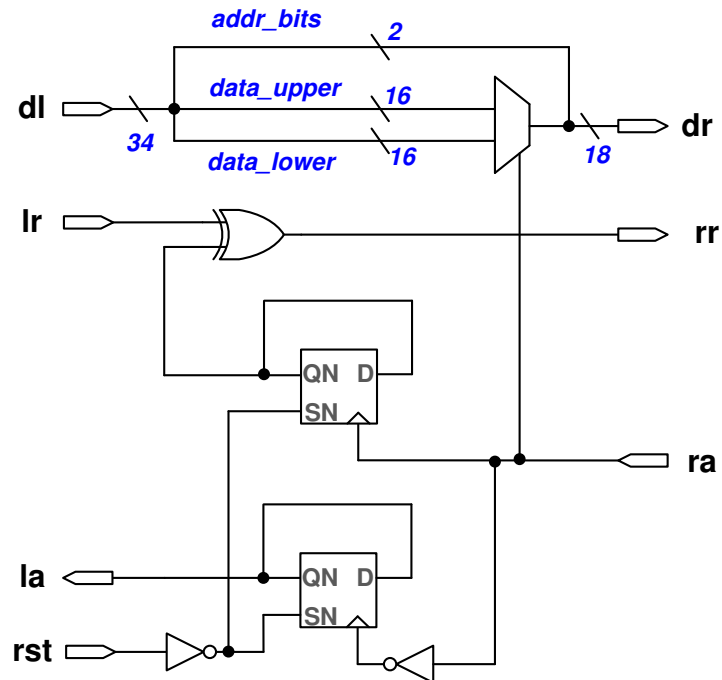


Figure 4.10: NDP_WN.

Table 4.1 summarizes design results of the two NDP modules. Area overhead of both NDP modules are negligible compared to the area of one 34-bit D1 router, $2423 \mu\text{m}^2$. Energy/flit of the NDP_NW is 41% of that of the D1 router, while the NDP_WN consumes very low energy per flit. However, energy overhead of the NDP modules might be insignificant, in consideration of the total NoC energy consumption, because the NDP modules are employed only in low trafficked links, that is, small number of packets. Delay, in the third column of Table 4.1, is a logic delay for each module to send two narrow size data. This logic delay is an extra performance overhead of the NDP method. Simply, it is expected that the link BW with an NDP module is reduced to half of the wide data-path link BW, by the half sized data-path. However, the BW degradation with the NDP modules is worse than the expectation, because of the logic delay of the NDP modules. The handshake cycle time in a link where an NDP module is employed is increased more as much as the logic delay of the NDP module.

Figure 4.11 shows the BW reduction of four different links with the NDP modules, as compared with a normal link, *Normal*, which represents a link without an NDP module. The *NDP_NW* and *NDP_WN* are links with an NDP module, while the *NDP_NW_PL* and *NDP_WN_PL* are links with an NDP module as well as one PL. In comparing the *NDP_NW* and *NDP_WN* with the *Normal* link, for short wire length ranges (under 2.0mm), the BW of two NDP links is less than 50% of the *Normal* link BW, due to the logic delay of NDP modules. The logic delay of the *NDP_NW* is greater than that of *NDP_WN*. Hence, the BW of the *NDP_NW* link is further reduced than that of the *NDP_WN*. As the link wire length increases, the link wire delay dominates in determining the link BW, while the logic delay overhead of the NDP modules relatively decreases. In the end, at a 4.0mm wire length, the BW of *NDP_NW* and *NDP_WN* is approximately half of the *Normal* link BW.

Table 4.1: Design summary of NDP modules: 32-bit data and 2-bit address in wide data-path.

	Area(μm^2)	Energy/flit(pJ)	Delay (ps)
NDP_NW	387	0.358	548
NDP_WN	172	0.051	301

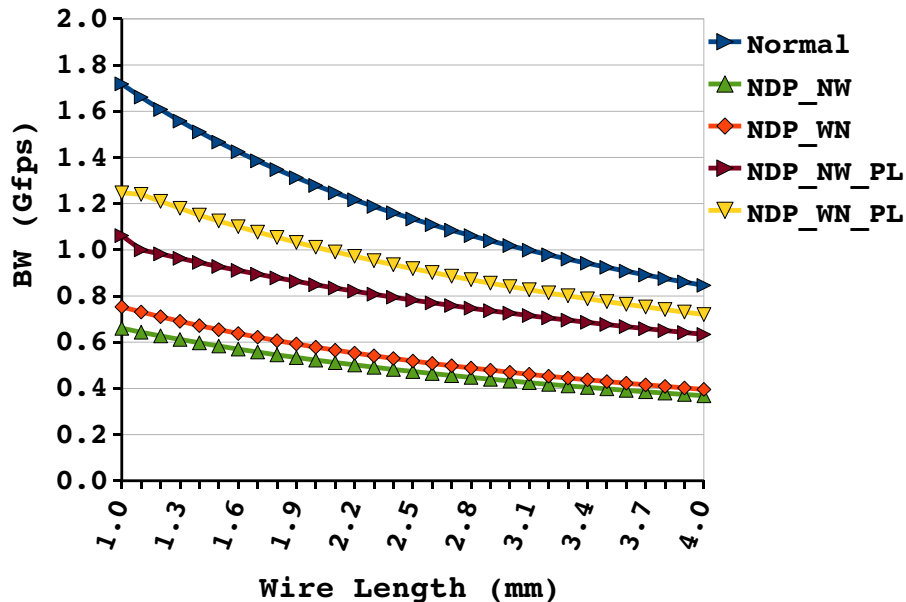


Figure 4.11: Link BW reduction by NDP module insertion.

This logic delay overhead can be relieved by means of inserting PLs in a link. Two other links with the NDP modules and one PL, *NDP_NW_PL* and *NDP_WN_PL*, show much better BW than their counterparts with no PL, *NDP_NW* and *NDP_WN*, as one PL noticeably reduces the BW reduction penalty. In addition, the link BW of the *NDP_NW_PL* and *NDP_WN_PL* are greater than the half of the *Normal* link BW. With the low traffic in NDP links, additional energy consumption by PL insertion may be small enough to be ignored.

Figure 4.12 illustrates an NoC example with two NDP modules inserted into links with the least traffic loads, *R1_B_I* and *R1_A_O*. The NDP module in the link *R1_B_I* is *NDP_NW*, while *NDP_WN* is inserted in the *R1_A_O* link. Other conditions are identical with those of the NoC_PL in Figure 4.6.

Table 4.2 shows the reduction of wire routing area and link acBW in the two links with the NDP modules. Wire area estimation was performed by ORION 2.0 wire models [32] and acBW is estimated using the analytical model for link BW estimation. In both links, wire routing area with NDP module is 47% less than without NDP.

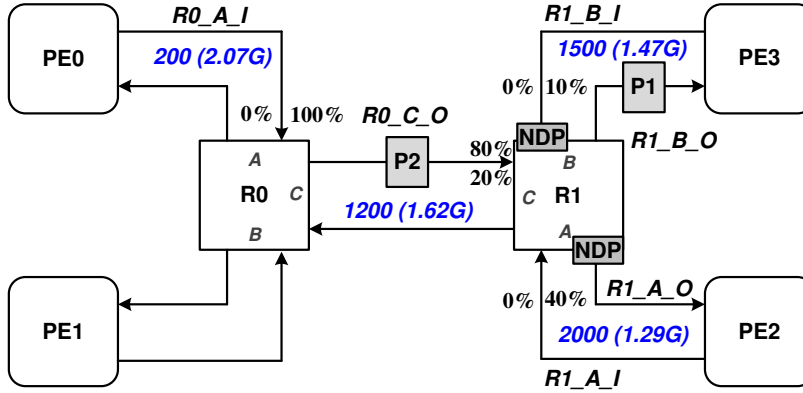


Figure 4.12: NoC example for NDP optimization method: NOC_NDP_PLno

Table 4.2: Reduction of wire area and acBW by NDP modules.

		$R1_B_I$	$R1_A_O$
Wire Length(μm)		1500	2000
Area(μm^2)	w/o NDP	46908	62308
	w/ NDP	25116	33363
acBW(Gfps)	w/o NDP	0.645	1.290
	w/ NDP	0.285	0.570

However, the benefit of the NDP method in wire routing area should be carefully accomplished with consideration of the NoC performance degradation. Simulation results with the NDP modules are shown in Figure 4.13 along with the previous two simulation results: Avg_L is of NOC_Init and Avg_L_PL is the latency of the NoC optimized by the PL insertion method, NoC_PL. They were already shown in Figure 4.7. $Avg_L_NDP_PLno$ is a simulation result with NDP modules, that is, the NoC shown in Figure 4.12, whereas $Avg_L_NDP_PL$ is average packet latency of an NoC which has two PLs in the $R1_A_O$ link to compensate for the BW reduction by the NDP module.

$Avg_L_NDP_PLno$ shows the degraded NoC performance from the Avg_L_PL due to the BW reduction by insertion of the NDP modules in two links. The BW reduction in $R1_B_I$ has little effect on NoC performance, since it does not influence any other link BW, as it is a link connecting a sender (PE3) and the first router (R1). In contrast, the BW reduction in $R1_A_O$ can degrade NoC performance. The $R1_A_O$

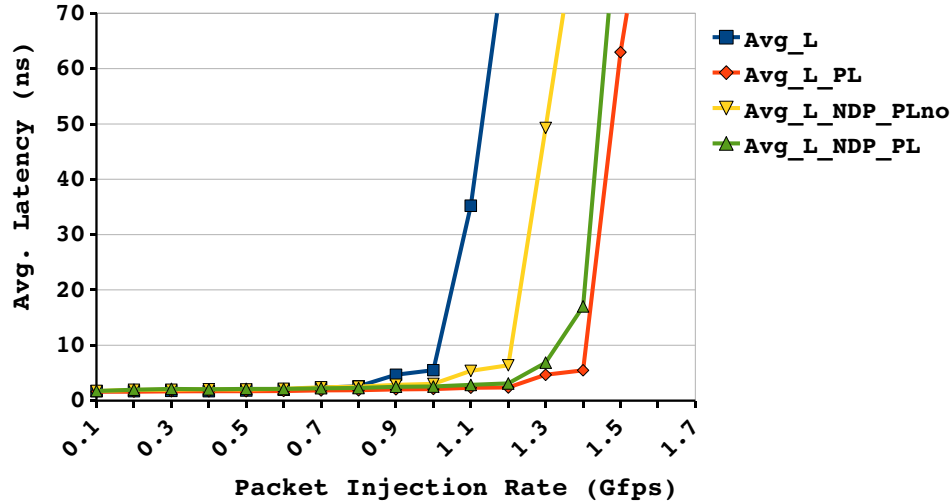


Figure 4.13: Simulation result for BW estimation

is one of the output links of the $R0_C_O$ link, the most performance-critical link in the example. Thus, $R0_C_O$ experiences its link BW reduction, even though the packet transfer rate to $R1_A_O$ from $R0_C_O$ is relatively small (20%), and this results in NoC performance degradation. The acBW of $R0_C_O$ can be estimated as in Eq. 4.11, in which the reduction of the acBW of $R1_A_O$ (BW_{out2}) is applied.

$$BW_{out2} = 1.29 \rightarrow \mathbf{0.570} \text{ Gfps} \quad (4.11)$$

$$avBW_i = 2.07 \text{ Gfps}$$

$$BW_{i1} = (1 - 0.5) \times \min(2.07, 2.07) + 0.5 \times \min(2.07, 2.07/2) = 1.55 \text{ Gfps}$$

$$BW_{i2} = (1 - 0.5) \times \min(2.07, \mathbf{0.570}) + 0.5 \times \min(2.07, \mathbf{0.570}/2) = \mathbf{0.427} \text{ Gfps}$$

$$\begin{aligned} acBW_{R0_C_O} &= R_{i1} \times BW_{i1} + R_{i2} \times BW_{i2} \\ &= 0.8 \times 1.55 + 0.2 \times \mathbf{0.427} = \mathbf{1.32} \text{ Gfps} \end{aligned}$$

In consequence, the acBW of $R0_C_O$ decreases to 1.32 Gfps from 1.43 Gfps of NoC_PL, the one without NDP modules and it leads to the performance degradation as shown in Figure 4.13.

However, the performance degradation by the usage of NDP module in $R1_A_O$ can be relieved through the PL insertion method. As already shown in Figure 4.11,

PL insertion in a link with an NDP module can considerably compensate the BW reduction penalty originated from the half size data-path width and the logic delay overhead of the NDP modules. So, two PLs are inserted into the link *R1_A_O*. This makes the acBW of the link 1.11 Gfps and subsequently increases the acBW of *R0_C_O* as estimated in Eq. 4.12.

$$\begin{aligned}
 BW_{out2} &= 0.57 \rightarrow \mathbf{1.11} \text{ Gfps}, & (4.12) \\
 avBW_i &= 2.07 \text{ Gfps}, \\
 BW_{i1} &= (1 - 0.5) \times \min(2.07, 2.07) + \\
 &\quad 0.5 \times \min(2.07, 2.07/2) = 1.55 \text{ Gfps} \\
 BW_{i2} &= (1 - 0.5) \times \min(2.07, \mathbf{1.11}) + \\
 &\quad 0.5 \times \min(2.07, \mathbf{1.11}/2) = 0.435 \text{ Gfps} \\
 acBW_{R0_C_O} &= R_{i1} \times BW_{i1} + R_{i2} \times BW_{i2} \\
 &= 0.8 \times 1.55 + 0.2 \times \mathbf{0.832} = \mathbf{1.40} \text{ Gfps}
 \end{aligned}$$

The acBW of *R0_C_O* with PL insertion is almost similar with that of the NoC_PL (1.43 Gfps) which has no NDP module. Consequently, *Avg-L-NDP-PL* presents an NoC optimization using the NDP method in conjunction with PL insertion, achieving comparable performance to *Avg-L-PL* while saving 47% of link wire routing area.

4.4 Energy-Critical Link Optimization: Double Spacing

Link wire dynamic energy is proportional not only to the number of packets transferred down a link, but also to link wire length. Generally, high trafficked links in an NoC should be optimized with short wire lengths in order to minimize total wire energy consumption by the NoC topology and router placement. In such a design the energy critical links will be the medium trafficked links with relatively long wire lengths.

Energy dissipation by link wires is related to wire resistance and capacitance, where the wire capacitance is composed of ground capacitance and coupling capacitance. Double Spacing(DS) is a method of optimizing an NoC design through

reduction of energy consumption in the link wires, by diminishing the wire coupling capacitance by means of separating any adjacent two wires to twice of the required minimum wire spacing.

Using ORION with parameters of the IBM 65 nm technology library, Table 4.3 presents properties of *R1_B_O* and *R0_C_O* links in the previous NoC example, Figure 4.12, in two different wire spacing configurations: single-spaced (SSPACE) and double-spaced (DSPACE). Link width is 34 bits on the global layer of wires with a 25% switching activity used for estimation of link wire energy.

The DSPACE links consumes 29% less energy than the SSPACE links, whereas their link routing areas increase by 40%, due to wider spacing. Interestingly, the total wire routing overhead by the DS method is $34440 \mu\text{m}^2$ which is less than the saved routing area by the NDP method in the previous section, $50573 \mu\text{m}^2$. In other words, the saved routing area through the NDP optimization method can be exploited effectively for saving energy consumption in the link wires.

4.5 Summary

Three methods, pipeline latch (PL) insertion, narrow data path (NDP), and double spacing (DS), are developed for optimizing asynchronous NoCs. The PL insertion method is used to improve the NoC performance by increasing the bandwidth of performance-critical links of the NoC. Strategically inserting PLs where necessary can enhance the NoC performance while minimizing NoC design costs. The energy overhead by PL insertion is not significant, compared to the link wire energy. The NDP method was proposed to save wire routing area leveraging excess link BW in low trafficked links in an NoC. In particular, the performance degradation by the NDP

Table 4.3: Comparison of SSPACE and DSPACE links with 34-bit link width.

		<i>R1_B_O</i>	<i>R0_C_O</i>
Wire Length(μm)		1500	1200
Energy/flit(pJ)	SSPACE	25.56	20.88
	DSPACE	18.27	14.95
Area(μm^2)	SSPACE	46908	37669
	DSPACE	66060	52957

method can be effectively resolved in collaboration with the PL insertion method. Further optimization can be performed by using the DS method, especially for saving total wire energy consumption.

The PL insertion method specialized to asynchronous NoCs only and there is no simple way of controlling individual link BW in synchronous NoCs. Meanwhile, the NDP and DS methods are NoC optimization techniques not unique for asynchronous NoCs. Both can be similarly employed to any synchronous NoCs as well. However, the NDP method in conjunction with the PL insertion is still specific to asynchronous NoCs. Predictably, usage of the NDP method in synchronous NoCs might be more limited or cause performance degradation.

An analytical model for link acBW estimation was developed for the three port router and single-flit packet format. In order to identify candidate links for each optimization method, the utilization of each link needs to be known. The analytical model accurately predicts link acBW and thereby gives useful information for the optimization process.

CHAPTER 5

EVALUATION

Two SoC examples were used to evaluate asynchronous NoCs and their optimization. The first example is an MPEG4 decoder described in [33] and used in several other NoC research projects. The MPEG4 example was used especially for comparing between one asynchronous NoC with the D1 router and one synchronous NoC in terms of performance and energy consumption.

The second example is an abstraction of a SOC design of Texas Instruments[34] which was provided in collaborating with our research group. The TI example was particularly used for demonstrating the asynchronous NoC optimization methods, presented in Section 4.

5.1 Evaluation Methodologies

A custom CAD tool, ANetGen, is used to generate the topology and router placement of the NoCs [35]. ANetGen was developed in our research group for generating optimized topology for asynchronous NoCs with the three-port routers. ANetGen takes an input format that defines expected traffic bandwidth as well as the core dimensions. The core floor plan is specified prior to ANetGen, which then determines physical placement of the routers and their logical topology. This tool reduces the length of high traffic links to save wire energy. For the asynchronous NoC, this artifact also increases avBW on the links that need it most. The cores were floor planned with the Parquet tool [36].

A SystemC-based simulator was developed for asynchronous and synchronous NoCs to model packet latency. The simulations were made as accurate as possible to the physical design by back-annotating the delays extracted from layout into the

ModelSim Verilog-SystemC co-simulation. A traffic generator injects packets by Poisson process according to the BW requirements of each IP core.

The wire delays for each link are modeled using an interpolation of simulation values [31]. Wire energy per link is estimated with the Orion 2.0 models [32]. The Orion implementation was improved in this work to use more accurate sizing of the buffer driving the first wire segment.

5.2 Evaluation of Asynchronous NoC with MPEG4 SOC

5.2.1 Synchronous Router Design

An asynchronous NoC was evaluated with the MPEG4 example by comparing its properties with those of a synchronous NoC. In order to compare two NoCs fairly, a synchronous router was design using a specific latency insensitive protocol, called pSELF (phase Synchronous ELastic Flow) [37]. This protocol is similar to the SELF protocol [38]. Latency insensitive protocols (LIP) are an adoption of asynchronous handshaking for a clocked system, and thus operate with a similar flow control method as the asynchronous protocols [39]. The similarity results in analogous LIP router architectures that use handshake signals, as well as a clock, for timing and sequencing. This allows a generally fair comparison of the effect of the communication links on NoC performance by minimizing other factors which may come from the flow control and router designs.

The architecture of the synchronous router is almost identical to that of the asynchronous router, shown in Figure 2.1. The pSELF switch and merge modules are shown in Figure 5.1(a) and Figure 5.1(b). Their operation is basically identical to that of their asynchronous counterparts. The arbitration circuit of the pSELF merge uses a round-robin scheme when two valid inputs ($vl1$ and $vl2$) are contended. The pSELF switch uses a half buffer latch $pEHB_H$ active on the high clock phase while the $pMerge_L$ latch operates in the low phase of the clock. Clock gating is inherently implemented in the pSELF as part of the protocol since the data latch is clocked only when the valid signal (vl) is active.

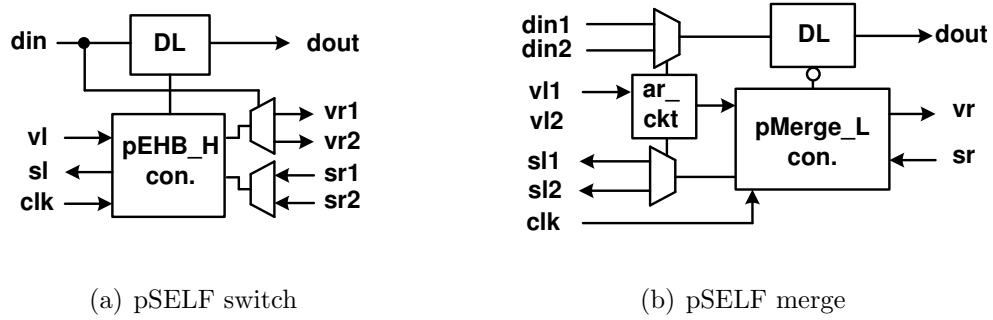


Figure 5.1: Implementation of switch and merge modules for pSELF router design

Table 5.1: Asynchronous D1 router and synchronous router design summary.

	Async.	pSELF
Max. Throughput (Gfps)	2.07	2.90
Dynamic Energy/flit (pJ)	0.54	0.71
Dynamic Idle Energy/clock (pJ)	0.00	0.16
Area (μm^2)	1829	1974

Table 5.1 summarizes design results of the asynchronous D1 router and the pSELF router. The two routers use a 21-bit flit width: 16-bit data-path and 5-bit routing address. The data-path width was determined in consideration of the BW requirements of the MPEG4 example and the routing address bit was decided by the maximum hop count of the topology, generated by the ANetGen.

The pSELF router has better maximum throughput, while the asynchronous router uses less energy. Almost equal areas of two routers comes from similar architecture and identical latch-based data storage inside routers.

Dynamic idle energy per clock is the energy consumed by transitions of a gated clock when there is no valid flit transfer. There is no such energy consumption in the asynchronous router.

5.2.2 Comparison of Asynchronous and pSELF NoC with MPEG4 Design

The MPEG4 example consists of 12 IP cores and each IP core communicates with a subset of all other IPs with different BW requirements. Communication properties of the design are represented with a *Communication Trace Graph* (CTG), shown in

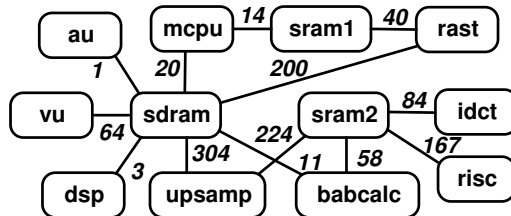


Figure 5.2: MPEG4 CTG graph. Edge weights are in MBytes/s.

Figure 5.2, where nodes are IP cores and weights show the required average BW between communicating pairs. Note that the weights have been modified from those originally provided in [33].

Asynchronous and clocked pSELF NoCs for the MPEG4 example were implemented using the same topology and router placement, illustrated in Figure 5.3. The 12 IP cores are connected with 10 three-port routers and 42 total links. Link wire lengths are represented in μm , with the numbers between two related links. ANetGen generates the topology such that high traffic links are assigned relatively short wire length for increasing the link’s avBW and reducing wire dynamic energy. As a result, IP cores with higher traffic, such as *SDRAM*, *upsamp* and *rast* in Figure 5.2, have short link wire length. Meanwhile, the IP core *au* has the longest link wire length with the lowest BW requirement.

Three different global clock frequencies are employed for the clocked pSELF NoC: 1.78 GHz, 2.07 GHz and 2.90 GHz. The asynchronous NoC consists of D1 routers of which the maximum throughput is 2.07 Gfps. The 1.78 GHz frequency for the pSELF design was selected because it has the same aggregate avBW as the sum of all the links in the asynchronous network. Thus, the asynchronous and 1.78 GHz pSELF design have the same average link avBW. The 2.07 GHz pSELF router has the same avBW as the asynchronous D1 router if there were zero wire delay between network nodes. The 2.90 GHz is the maximum clock frequency of the pSELF router.

The MPEG4 design was simulated with different BW requirements. The default bandwidth ($1\times$) implements the communication bandwidth values shown in the specification in Figure 5.2. Traffic load is increased by multiplying the base value of each

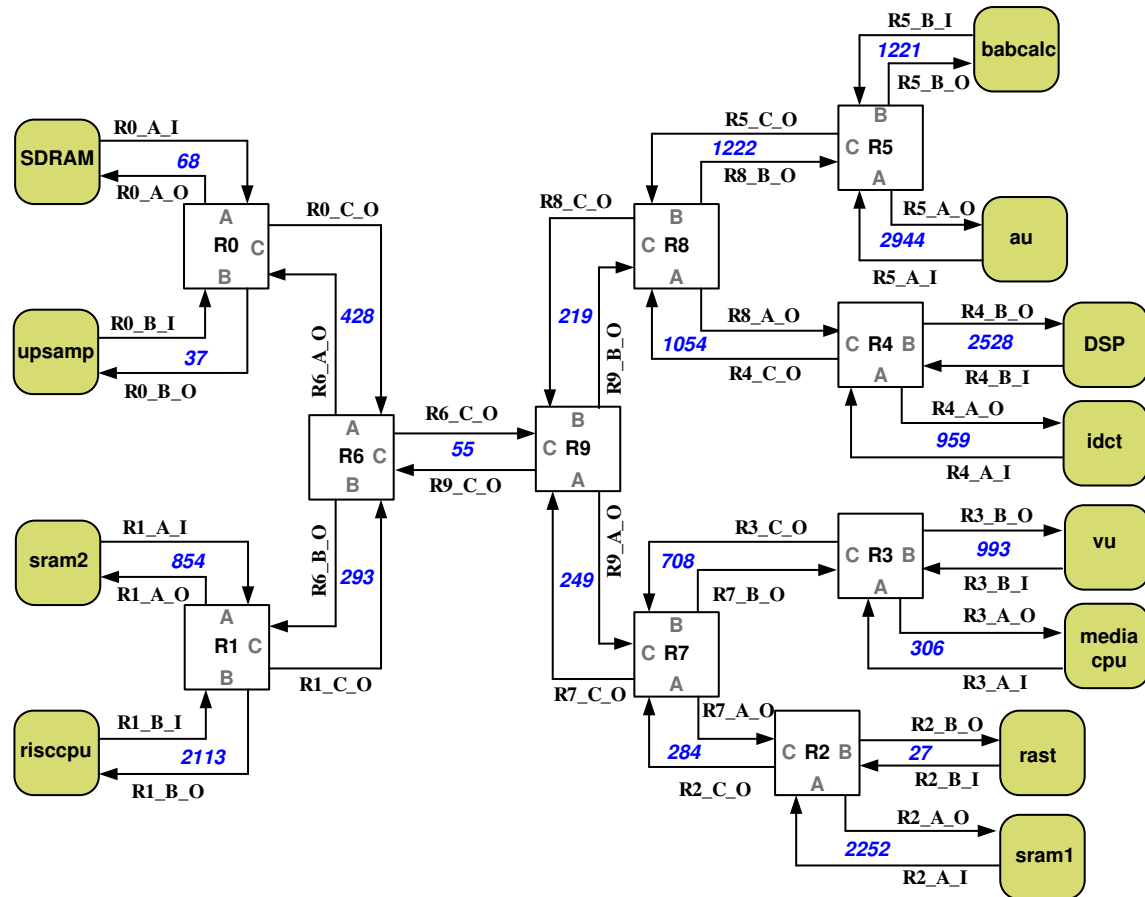


Figure 5.3: MPEG4 network topology.

path by the same factor, resulting in three times the load for a $3\times$ network, by five times for $5\times$, and so on. This gives a comparison at increased traffic loads.

Figure 5.4 shows avBW and load on 14 links of the asynchronous, pSELF 1.78G and pSELF 2.07G NoCs with a $4\times$ offered load. The first seven links are those with the greatest loads, while the last seven links carry the smallest traffic loads. The different properties of link avBW assignment can be clearly seen between asynchronous and clocked NoCs. The two pSELF NoCs have identical avBW on all links, regardless of the link's traffic loads, due to their synchronous nature and global clock frequency. On the contrary, the avBW of each asynchronous link differs based on its individual link wire length determined by the network topology and router placement with consideration of traffic loads of each link. Therefore, high trafficked links have higher

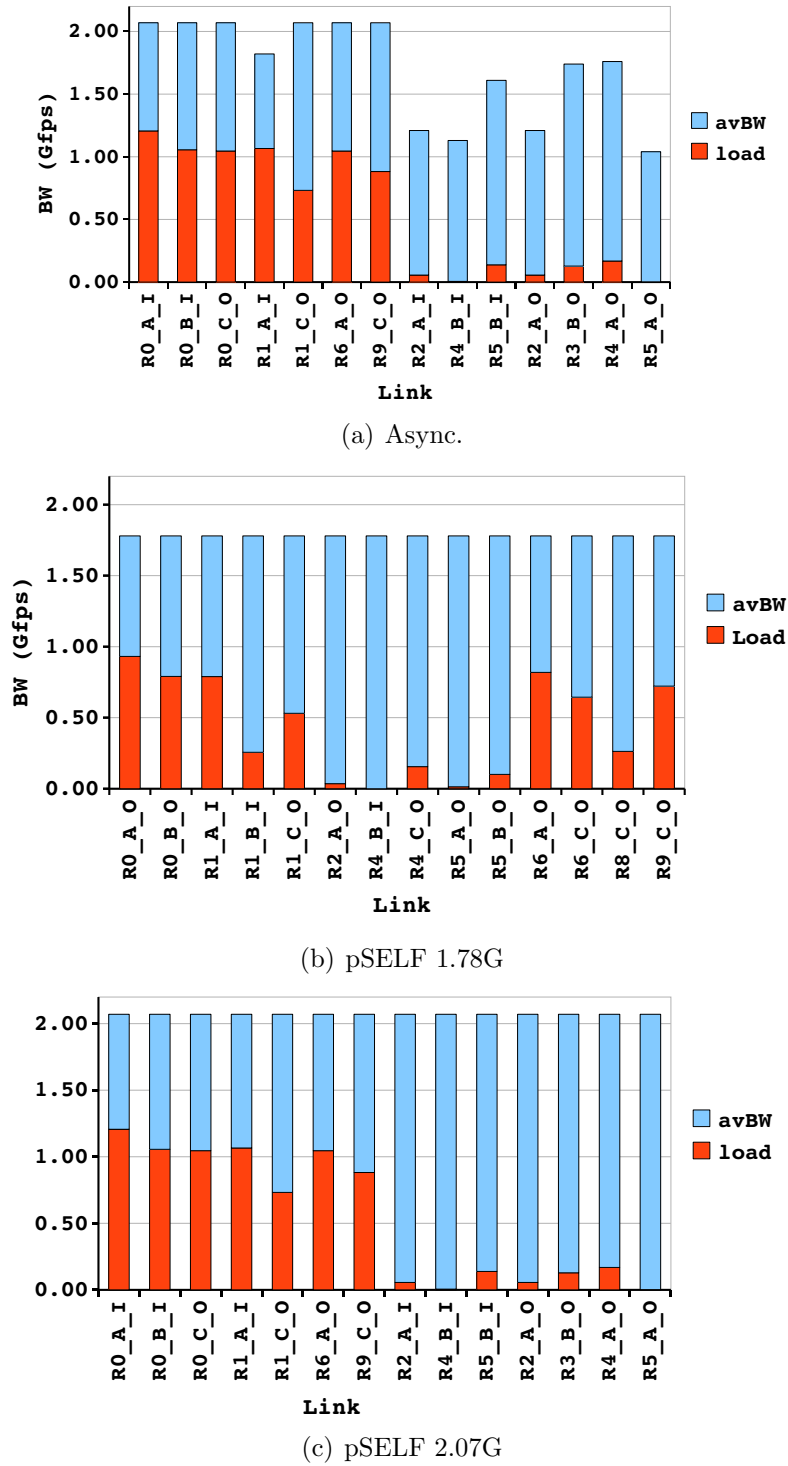


Figure 5.4: Available BW (*avBW*) and traffic load (*Load*) of 14 links in the asynchronous, pSELF 1.78G and pSELF 2.07G NoCs in $4\times$ offered traffic load.

acBW, whereas the seven low trafficked links are assigned relatively low avBW with long link wire length.

In fact, link acBW is the determining factor for NoC performance, rather than the avBW, as acBW takes into account packet contention as well as the BW of all subsequent links in the network. Figure 5.5 shows link utilization of the 14 links with the acBW and the traffic load of each link in three NoCs with $4\times$ offered loads.

In the high trafficked links, all three NoCs have less acBW than corresponding avBW because of packet contention and limits of subsequent links. In particular, the pSELF 1.78G NoC has the most limited acBW and thus, it will become congested earlier with increasing offered traffic. Three links, *R0_A_I*, *R0_B_I* and *R1_A_I*, of pSELF 1.78G NoC are already fully utilized with $4\times$ offered loads.

The asynchronous and pSELF 2.07G shows similar link acBW. The clock frequency of 2.07 GHz was selected to match with the maximum throughput of the control logic of the asynchronous router. Note that this frequency of operation is only achieved with zero wire delay in the asynchronous network. However, thanks to the optimized network floor plan generated from the ANetGen considering traffic load of each link, all performance-critical links have such short wire length that those links are not affected by their link wire delay. For example, six links out of the seven

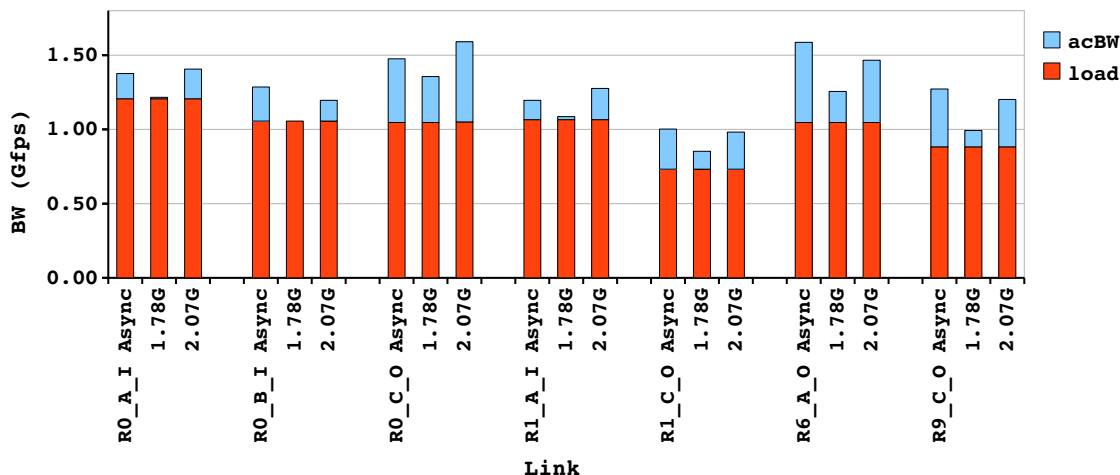


Figure 5.5: Link utilization in the asynchronous, pSELF 1.78G and pSELF 2.07G NoCs in $4\times$ offered traffic load. *acBW* is an achievable link BW, and *Load* is traffic load of each link labeled on X-axis.

high trafficked links have link length shorter than $525\mu\text{m}$, the MBR of the D1 link with no PL. Subsequently, there is no link BW degradation due to link wire delay in these six links, and their avBW are the maximum throughput of the D1 router, 2.07 Gfps. In addition, low link avBW in the low traffic links in the asynchronous NoC do not significantly affect the acBW of performance-critical links. As a result, it is expected that the asynchronous and the pSELF 2.07G NoC will be very similar in their NoC performance.

Figure 5.6 compares the average latency of the asynchronous network and three pSELF networks with varying offered traffic loads. An increase of latency as offered traffic load rises shows that traffic paths contend for switch and link resources for long periods of time. The pSELF design clocked at 1.78 GHz has longer latency at a light traffic load than the other three NoCs. Here, packet latency is determined mainly by the clock period since the network is largely uncongested. This is larger at 1.78 GHz than the asynchronous network and the two other higher frequency networks. Furthermore, its saturation point is at $4\times$ load as expected from the link utilization of the highest traffic links shown in Figure 5.5. Meanwhile, the asynchronous network and pSELF 2.07G network show almost identical average packet latency, due to similarly assigned acBW in high trafficked links. The pSELF 2.90G network shows the lowest average latency. This design is not fully congested

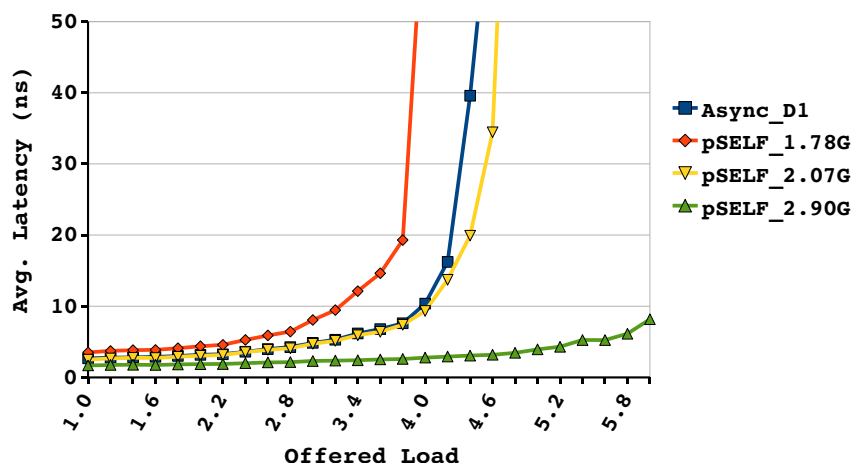


Figure 5.6: Average latency comparison between the asynchronous and pSELF networks in various offered loads.

even at the highest offered load examined, due to the sufficient BW in all links. However, this advantage in latency comes at the expense of the higher energy usage of a faster clock.

Energy usage is reported in Figure 5.7 for each network at four different offered loads: 1×, 2×, 3× and 4×. The asynchronous NoC energy consists of the routers' dynamic energy (*RTR_Dyn_E*) and the wire energy (*Wire_Dyn_E*). The pSELF NoC energy includes another component, the idle clock energy (*RTR_I_Clk_E*), which is from the cycles in which routers do not switch flits. In addition, *EHB_I_Clk_E* is energy dissipated by synchronous PLs, the half buffer latch (*EHB*), that are required for the pSELF 2.90G NoC. As previously presented in Figure 1.3, the pSELF 2.90G NoC has a 2100 μm link wire length limit. Wires longer than this require a pipeline latch to support the 2.90 GHz clock frequency. In the network topology for the MPEG4 example, a total of 8 links are longer than this wire length limit of the pSELF 2.90G and subsequently, eight PLs are inserted. Most of these links with long link wire length are low traffic links so the energy consumed by these PLs is mainly by idle clocking. Therefore, only the idle clock energy of synchronous PLs is included in the energy comparison. The other two pSELF NoCs have much longer wire length limits thanks to longer clock periods, so that there is no need to add link pipelining.

The router dynamic energy is the total energy used by all 10 routers in the networks. Because of their architectural similarity, the router dynamic energy is very

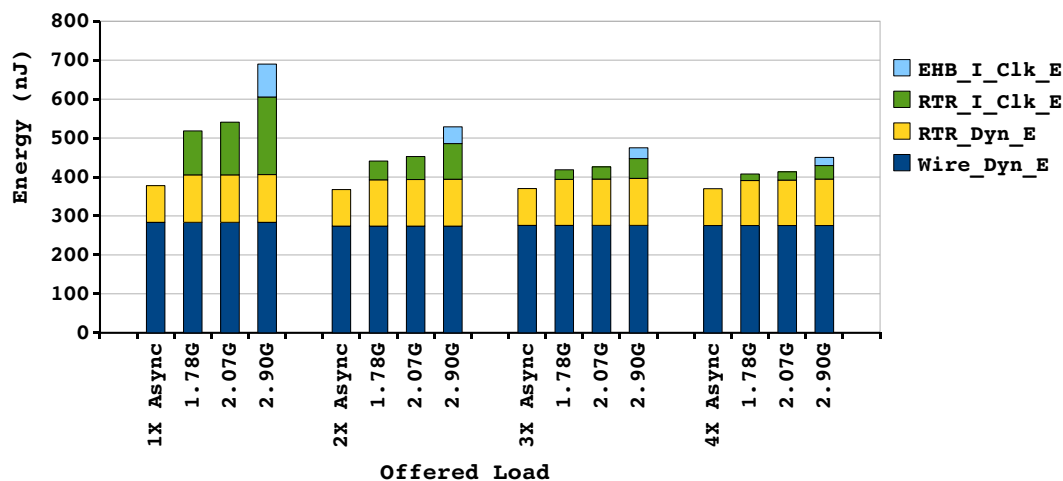


Figure 5.7: Energy distribution at 1×, 2×, 3× and 4× offered loads.

similar between the asynchronous router and pSELF router. Wire energy is the sum of energy used by the wires composing the links and their drivers. Each link energy was calculated based on its length and carried traffic volume. The asynchronous and pSELF networks used the same topology and router placement, and thus the link wire energy is identical in all networks.

As a consequence, idle clock energy (*RTR_I_Clk_E* and *EHB_I_Clk_E* of the pSELF 2.90G) is the primary differentiator for the total NoC energy between networks. The asynchronous network consumes less energy than all other pSELF networks by as much as the idle clock energy of each pSELF network. The portion of the idle-to-total energy increases as the offered load is lowered, and as the clock frequency is increased, both of which lead to more idle cycles. Higher operating frequency is beneficial for low packet latency, and it also improves the capability to handle higher traffic load. However, it has more idle cycles on the low traffic links, which wastes considerable energy from idle clocking. Accordingly, the asynchronous network is more energy-efficient compared to the pSELF of high frequency, particularly when the offered load onto the network is low.

The asynchronous network consumes 30%, 19%, 13% and 10% less energy than the pSELF 2.07G (which has the similar average packet latency) in $1\times$, $2\times$, $3\times$ and $4\times$ offered loads, and 45% less than pSELF 2.90G in $1\times$ offered load.

For a fairer comparison between different NoC designs, the *Energy-Delay Product* (EDP) metric is used where the delay term is the average latency of an NoC design. The lesser value of EDP is more preferable. Figure 5.8 compares EDP values of the four NoCs. In computing EDP values, wire energy consumption was excluded as it is identical in all NoC designs.

The pSELF 1.78G is the worst design in the whole range of offered load due to the lowest performance. The pSELF 2.07G is worse than the asynchronous NoC by the extra energy dissipation of idle clocking, in spite of similar NoC performance. The EDP difference of two designs is getting closer as the offered load increases by the reduction of the idle clock energy portion in total energy consumption. Compared to the pSELF 2.09G, the asynchronous NoC shows much better EDP in low traffic loads, less than $3\times$. Meanwhile, over $3\times$ offered loads, the pSELF 2.90G is the most

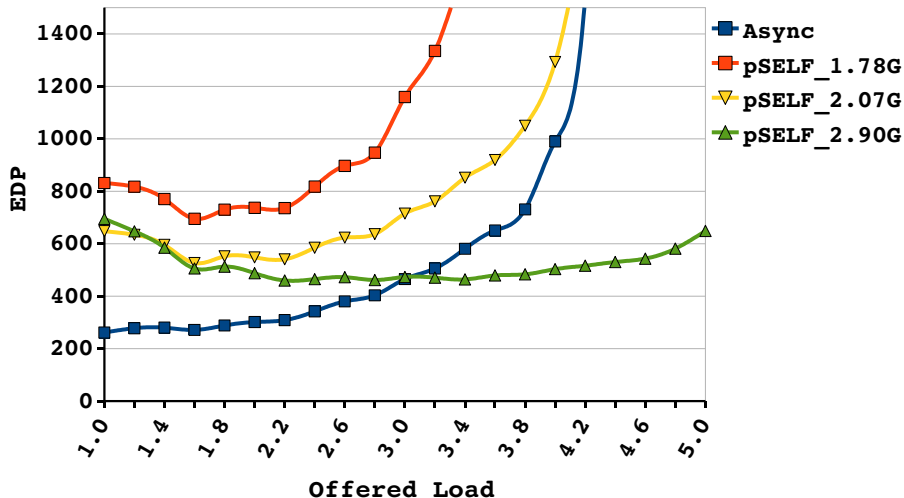


Figure 5.8: EDP comparison between four NoC designs in various offered loads.

efficient as it gains benefit from the lower average latency than the asynchronous NoC. Note that energy consumption by a clock distribution network for the pSELF NoCs is not included in the energy computation. The EDP values of three pSELF NoC will increase when clock tree energy is considered.

Overall, the optimization of individual link BW of the asynchronous NoC, through topology and router placement based on traffic loads of each link, makes it possible to adequately overcome the disadvantage of the asynchronous communication links, that is, link BW reduction by wire delay. Consequently, properly balanced link BW assignment in the asynchronous NoC accomplishes comparable NoC performance to its synchronous counterpart.

5.3 TI Design

5.3.1 Asynchronous NoC for TI Design

The TI example is composed of 35 PEs and has 354 communication paths among 1190 possible paths between PEs. The topology for the TI example generated from ANetGen consists of 33 routers and 134 links and is shown in Figure 5.9.

As presented in Table 3.4, Section 3.5, eight different asynchronous communication link designs are possible with three different router designs (D1, D2, and D3). These can be combined with the number of PLs in a link. They are classified into three

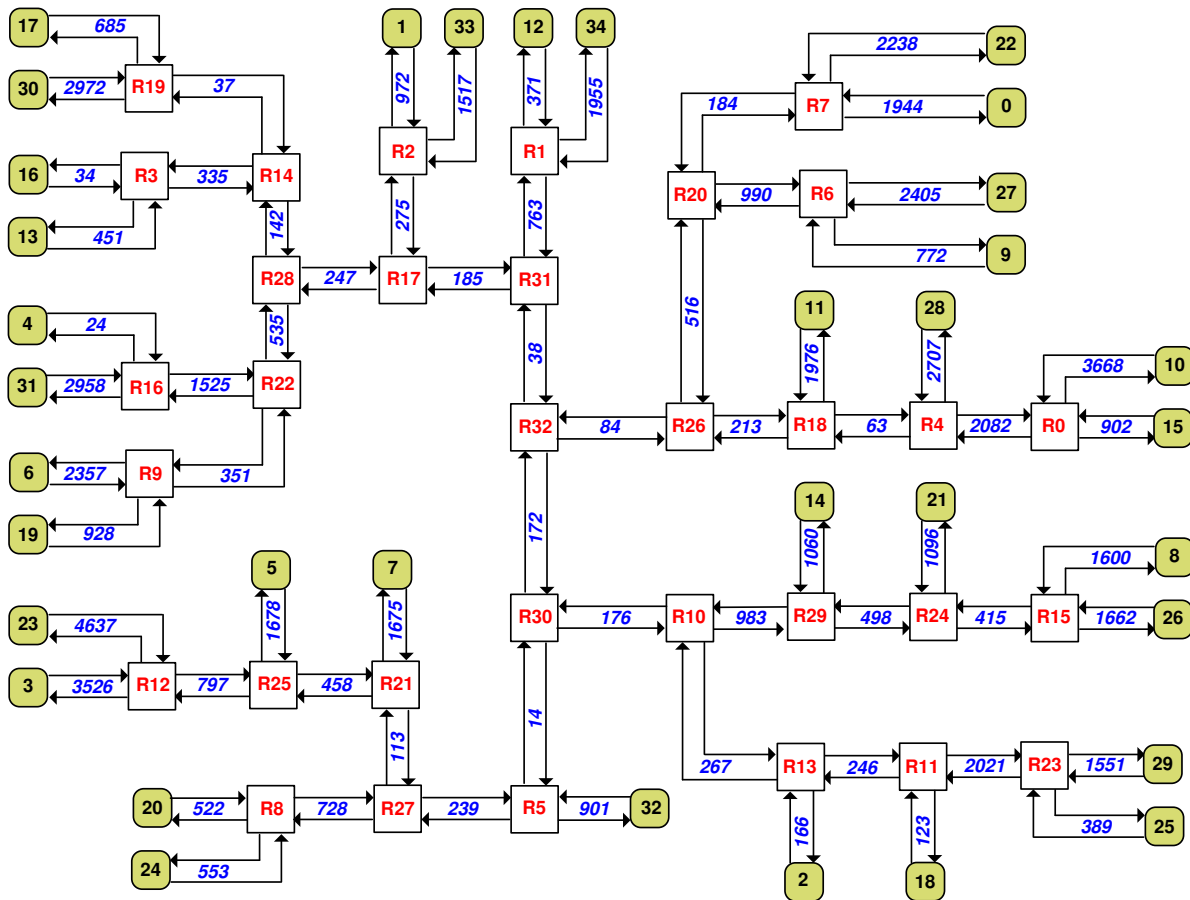


Figure 5.9: TI example network topology. PEs are in rounded-square boxes and routers in square boxes, numbers are link wire lengths in μm .

types and the properties of link BW and energy consumption were compared with each other belonging to the same type of link.

For an optimized asynchronous NoC for the TI design, the eight different NoCs were first evaluated and compared with each other. The *D3_PLno* design in *Type 2* and *D3_PL1* in *Type 3* are a little bit different from the other designs for the same type class, with regard to the number of data latches in a path from a source to a destination. For instance, *D1_PL1* and *D2_PL1* designs have one PL in all links. Therefore, if a path is connected through three routers and subsequently four links from a sender to a receiver, the total number of data latches in the path is 10, six data latches inside three routers and four external PLs in each link. But, *D3_PLno* design has only nine data latches in the same path, as each router has three internal data latches without any external PL. So, for a fair comparison between NoCs in the same

type, the $D3_PLno$ design has one PL at all receiver links. The receiver link (from the last router to a receiver in a path) is preferable to the sender link (from a sender to the first router in a path) for NoC performance, since it is the lowest downward link for which BW affects all preceding link's BW. On the contrary, increasing the BW of the sender link is not beneficial for any other link BW, except the sender itself. Consequently, $D3_PLno$ NoC has total 35 PLs in all receiver links which is the same number of IP cores of the TI design. Similarly, the $D3_PL1$ NoC has total 169 PLs: 134 PLs in all links and an additional 35 PLs in all receiver link. The comparison performed in Section 3.5 did not consider this aspect as only the link BW of two routers was of interest.

Evaluation results of total NoC energy and average latency are shown in Figure 5.10 for eight NoCs. They are also compared by EDP in Figure 5.11.

The total NoC energy is composed of wire ($wire_e$), router (rtr_e) and PL (pl_e) energy aligned to the left-hand side y -axis. The average latency uses the right-side y -axis. Total energy of the NoCs in the same type is nearly equal because energy dissipated by one packet in a path is the same for all designs, owing to the same number of data latches. Since the difference between NoC types is the number of PL

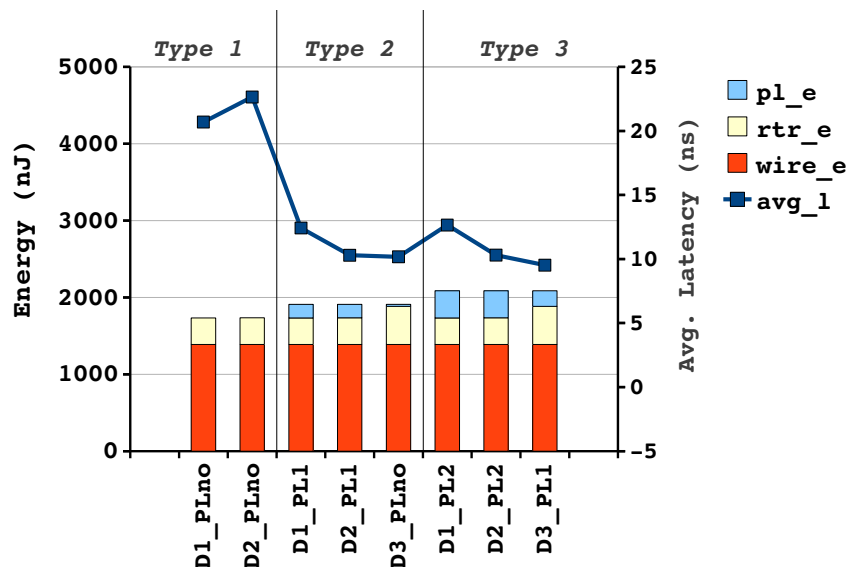


Figure 5.10: Comparison of asynchronous NoCs in energy and average latency with TI example.

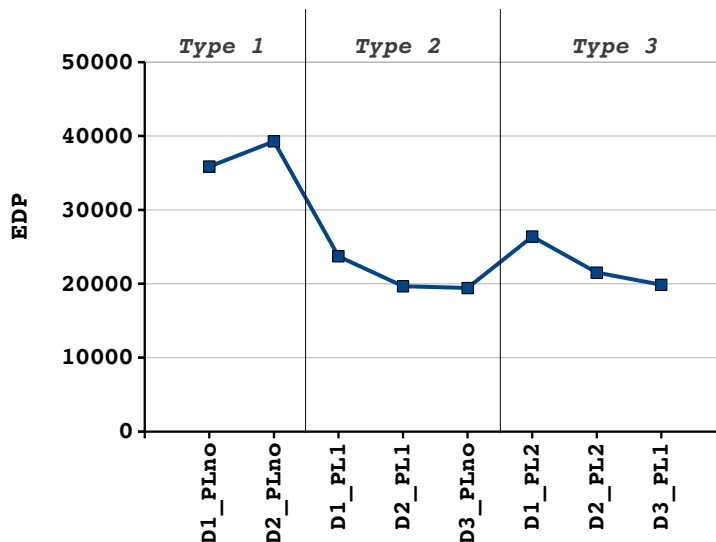


Figure 5.11: EDP of asynchronous NoCs with TI example.

in an NoC, the total energy differs between NoC types based on the energy of the PLs. The D3 NoCs in *Type 2* ($D3_PLno$) and *Type 3* ($D3_PL1$) have larger router energy but less PL energy, compared to the other NoCs in the same type.

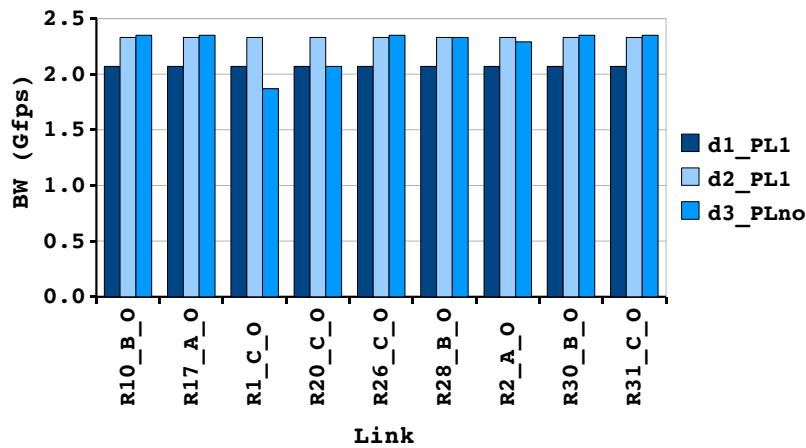
The average latency presents improvement of NoC performance through PL insertion, in particular, from *Type 1* to *Type 2* NoCs. The two *Type 1* NoCs, $D1_PLno$ and $D2_PLno$, show the worst performance. Both NoCs do not have any PL in their link so that link wire delay is fully applied to each link BW and therefore, considerably degrades BW of all links. Furthermore, $D2_PLno$ is worse than $D1_PLno$ because it is more vulnerable to link wire delay penalty (as shown in Figure 2.18).

The transition from *Type 1* to *Type 2* NoCs achieves a dramatic decrease in the average latency. In particular, the performance enhancement from $D2_PLno$ to $D2_PL1$ is much larger than the D1 case. This is because the D2 router has a higher maximum throughput than the D1 router, and one PL in all links noticeably reduces link wire delay penalty.

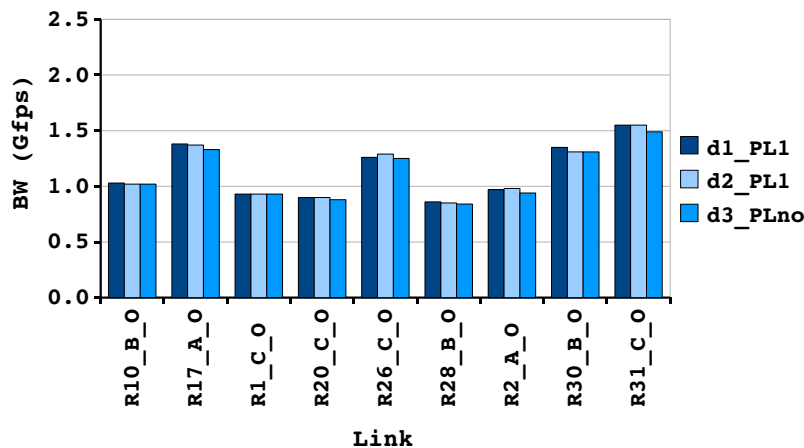
Interestingly, $D3_PLno$ NoC shows comparable performance to $D2_PL1$ and better than $D1_PL1$ design, even with no PL in all links, except its receiver links. This can be explained by two factors: First, thanks to the wire length optimized floor plan, most of the high traffic links have relatively short wire lengths. Seven out of nine highest traffic links have wire lengths less than $250\ \mu m$. This leads to no link BW reduction

by link wire delay in such high traffic links, as the D3 router MBR of $D3_PLno$ links is $220\ \mu\text{m}$ as shown in Figure 3.21. Furthermore, the maximum throughput of the D3 router is greater than that of D1 and D2 routers. Second, the PLs inserted in all receiver links in $D3_PLno$ NoC substantially improve the NoC performance.

Figure 5.12 shows the avBW and acBW of nine highest traffic links in all three NoCs in *Type 2*. The D3 design has higher avBW than the others except for the two links with relatively long link length: R1_C_O is $763\ \mu\text{m}$ and R20_C_O is $516\ \mu\text{m}$. Even so, the avBW of these two links is still comparable to its counterparts. The acBW is not distinguishable between all three NoCs. If an NoC is saturated by excessive traffic loads, the acBW of the fully utilized links primarily determines NoC



(a) Available BW



(b) Achievable BW

Figure 5.12: Available BW and achievable BW of the most utilized links in *Type 2* designs.

performance. But, all three NoCs do not have any link fully utilized with traffic loads in the TI design. Thus, the avBW of high traffic links can somewhat influence the NoC performance since packet transfer rates relies on the avBW of those links in nonstalled condition.

The performance improvement in *Type 2* NoCs, compared to *Type 1* counterparts, is obviously reflected in EDP values. The total energy increase in *Type 2* is not so significant as to be compensated by the enhanced average latency.

Unlike the NoC design transition from *Type 1* to *Type 2*, the transition from *Type 2* to *Type 3* does not show any significant benefit. In spite of an increase in total NoC energy, *D1_PL2* and *D2_PL2* NoCs have almost identical average latencies with those of *D1_PL1* and *D2_PL1*, respectively. This result means that one PL in all links of *D1_PL1* and *D2_PL1* produces sufficient link BW to handle the BW requirement of the TI design. Thus, inserting additional PLs in those NoCs merely increase total NoC energy without any benefit for the performance, resulting in deteriorated EDP.

The *D3_PL1* NoC achieves some improvement in the average latency at 9.52 ns, compared to *D3_PLno* at 10.17 ns. Nevertheless, the increased energy consumption with much more PLs outgrows the performance improvement. Hence the *D3_PL1* design has marginally worse EDP than that of *D3_PLno*.

Overall, from the comparison of eight different asynchronous NoCs, the *D3_PLno* can be considered a candidate NoC for further optimization using proposed methods in Section 4. In fact, the *D2_PL1* design shows comparable EDP value to the *D3_PLno* design. However, there was no improvement of the average latency from *D2_PL1* (10.29 ns) to *D2_PL2* (10.30 ns). It is expected that 10.29 ns or so is the best that one can achieve using the D2 router. On the contrary, it is possible to improve the average latency of *D3_PLno*, as seen by *D3_PL1*. Thus, *D3_PLno* design can be optimized further by inserting additional PLs in selected performance-critical links while achieving the same performance of *D3_PL1* as minimizing energy overhead by additional PLs. In other words, the optimal NoC design of TI design would exist between *D3_PLno* and *D3_PL1* designs.

5.3.2 Asynchronous NoC Optimization for TI Design

The three optimization methods, PL insertion, narrow data path (NDP), and double wire spacing (DS), presented in Section 4, are applied to the *D3_PLno* design in turn to implement an optimized NoC design for the TI design.

5.3.2.1 Performance-critical Link Optimization for TI Design

Through strategic PL insertion into performance-critical links in *D3_PLno*, an NoC where performance is optimized by PL insertion, *D3_PL-OPT*, was designed. The *D3_PL-OPT* NoC shows comparable performance to the *D3_PL1*, with many fewer PLs, and subsequently, less energy consumption than the *D3_PL1*.

In determining performance-critical links, the average latency contribution of each path was used. Table 5.2 presents 17 selected paths out of the 354 total paths of which path average latency contributes most highly to the NoC average latency in the *D3_PLno* design. The contribution is calculated based on the number of packets (*NP*) transferred in a path and the average latency (*Avg L*) of the path.

The 17 paths transfer 19% of the total simulated packets, contribute 20% of the total NoC average latency, and the transaction of the paths are from only 12 senders to 8 receivers. Figure 5.13 shows the PEs (gray rounded-boxes) and performance-critical links (green area) which are related in the selected paths in the *D3_PLno* design.

PL insertion in the performance-critical links was decided by maintaining the avBW of those links to be over 2.0 Gfps. Accordingly, one PL is inserted on a link of with a length is between 500 μm and 1500 μm . Two PLs are inserted in links of

Table 5.2: 17 Paths which most contribute NoC average latency.

Path	NP	Avg L	Cont.(%)	Path	NP	Avg L	Cont.(%)
PE11_PE10	480	7.03	1.10	PE20_PE34	208	14.59	0.99
PE11_PE34	304	9.93	0.98	PE21_PE11	304	12.29	1.21
PE12_PE33	608	7.03	1.39	PE2_PE33	224	15.90	1.16
PE13_PE11	272	16.46	1.45	PE2_PE34	224	17.27	1.26
PE13_PE4	384	7.04	0.88	PE33_PE0	272	9.95	0.88
PE14_PE10	656	13.08	2.79	PE33_PE12	384	10.31	1.29
PE17_PE34	320	11.97	1.24	PE34_PE1	448	6.83	0.99
PE1_PE34	288	10.20	0.95	PE9_PE21	160	17.22	0.89
PE20_PE33	192	18.57	1.16	Total	5728	-	20.60

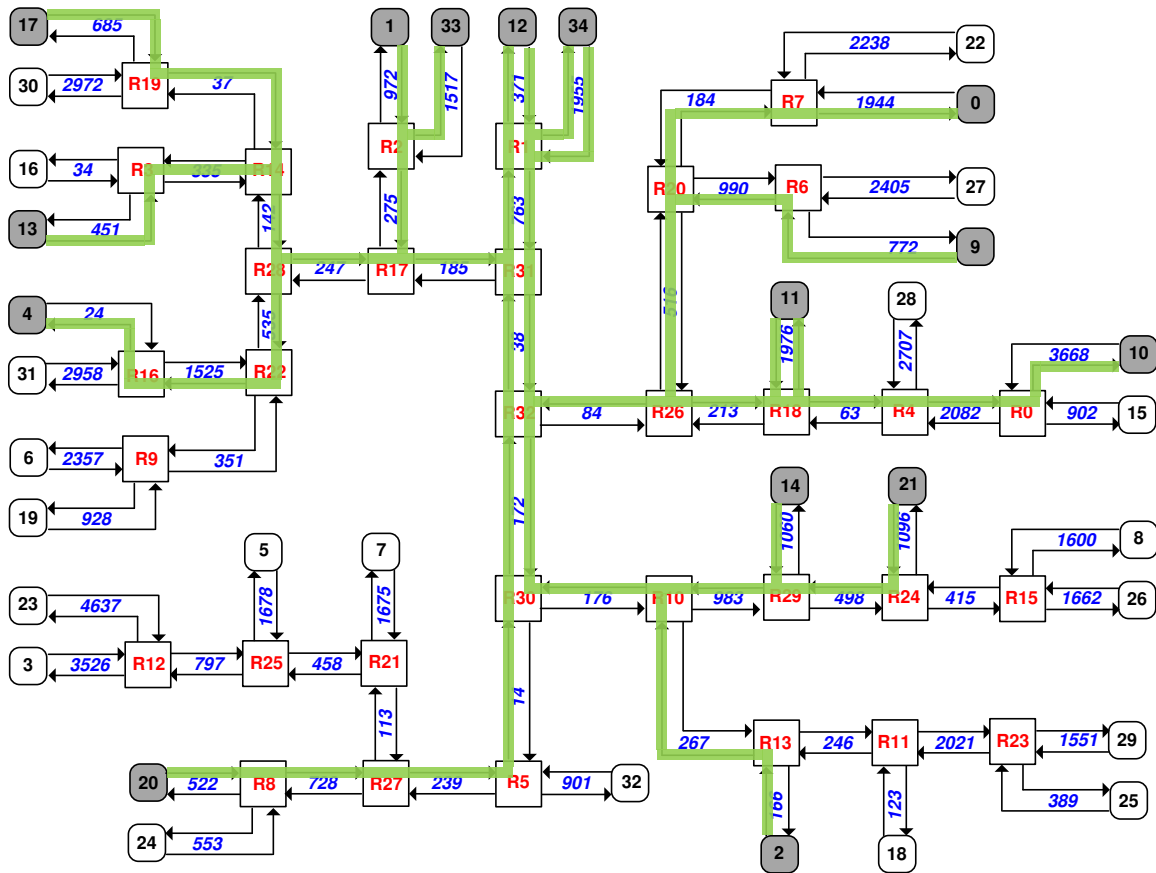


Figure 5.13: Performance-critical links in $D3_PLno$.

which the length is over $1500\ \mu\text{m}$ so that the link's avBW is maintained over 2.0 Gfps until $2500\ \mu\text{m}$. This is depicted in Figure 5.14 which is identical with Figure 3.20 showing link BW variance in D3 links with different number of PLs. In fact, it is possible to make a link avBW to be 2.35 Gfps, the maximum throughput of D3 router, by inserting PL in a link below $500\ \mu\text{m}$. However, inserting a PL increases energy consumption and such a link with short length has high traffic (due to optimizations by ANetGen).

The $D3_PL_OPT$ design has 21 additional PLs inserted from the $D3_PLno$ design. The PL placement is shown in Figure 5.15 as green boxes with a 'P'. In addition, the $D3_PLno$ NoC already has one PL in all receiver links. Thus, one additional PL is inserted into the receiver links over $1500\ \mu\text{m}$ long, such as the link from R1 router to PE34 in the top-center of Figure 5.15.

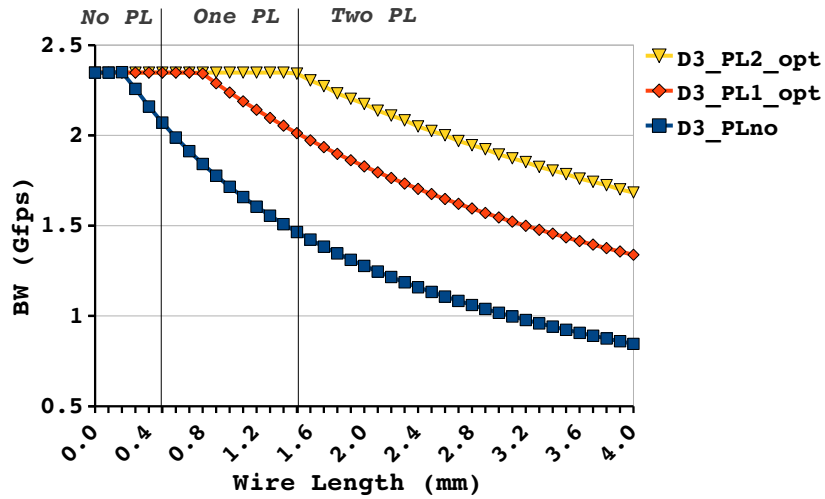


Figure 5.14: Strategy of PL insertion in $D3_PL_OPT$.

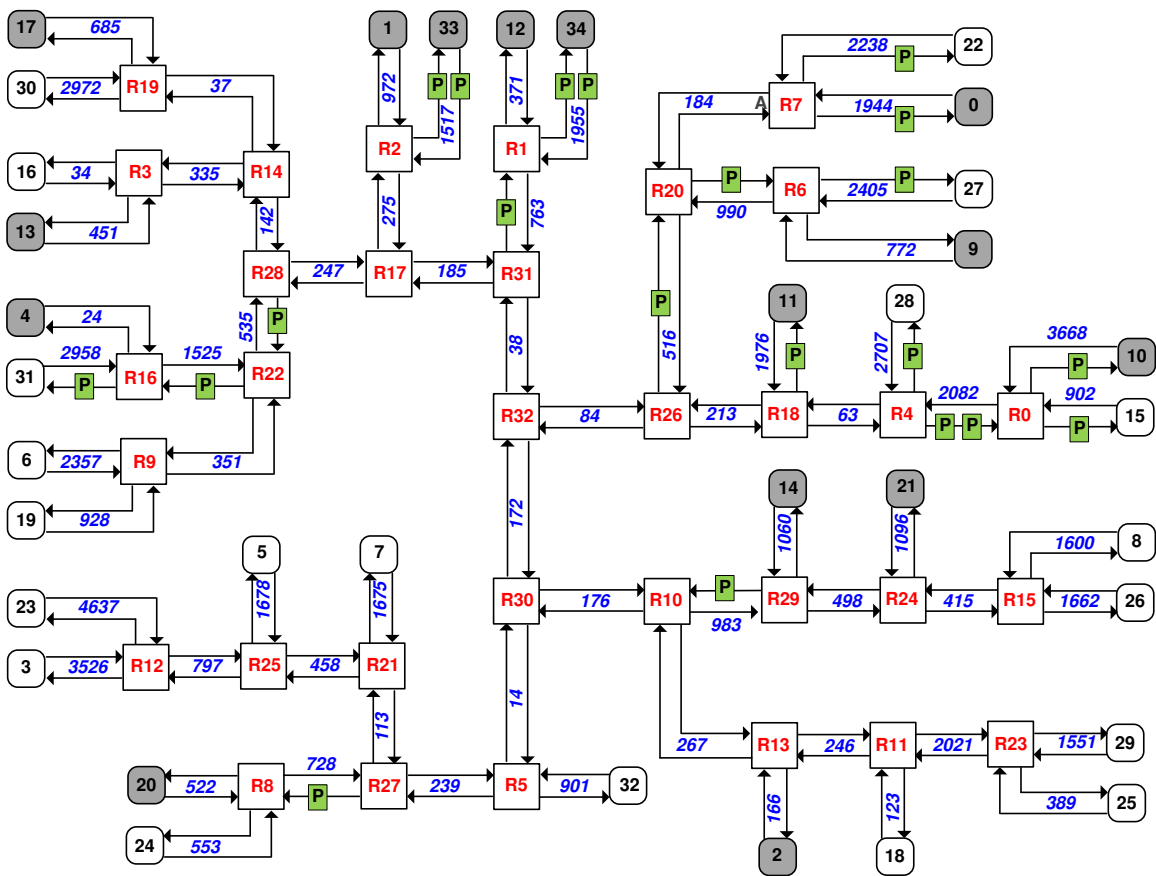
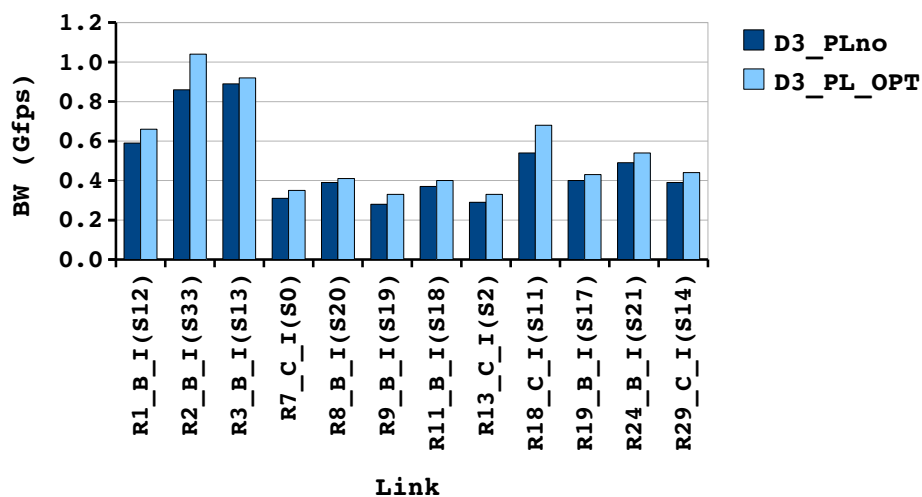
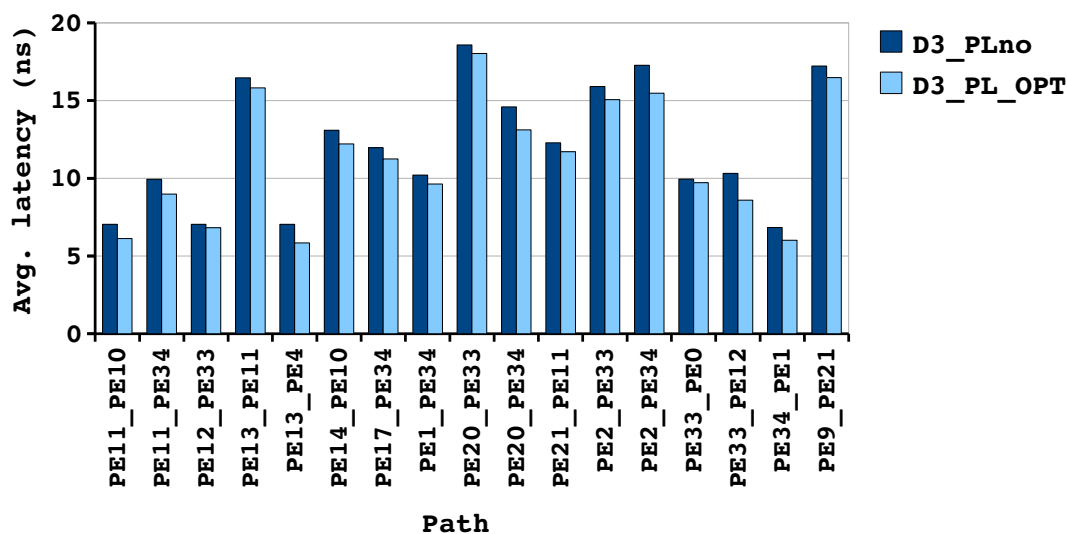


Figure 5.15: PL insertion in $D3_PL_OPT$ design.

Figure 5.16 shows design improvement of the $D3_PL_OPT$. Figure 5.16(a) presents increased acBW of the 12 sender links which are sources of the 17 selected performance-critical paths, and the reduced average latency of 17 paths is compared in Figure 5.16(b). In consequence, as shown in Figure 5.17(a), the better average latency in performance-critical links of the $D3_PL_OPT$ NoC results in an enhancement of NoC average latency to 9.58 ns, which is almost same as that of $D3_PL1$ at 9.52 ns, with 7% less energy consumption. This achieves an enhanced EDP illustrated in

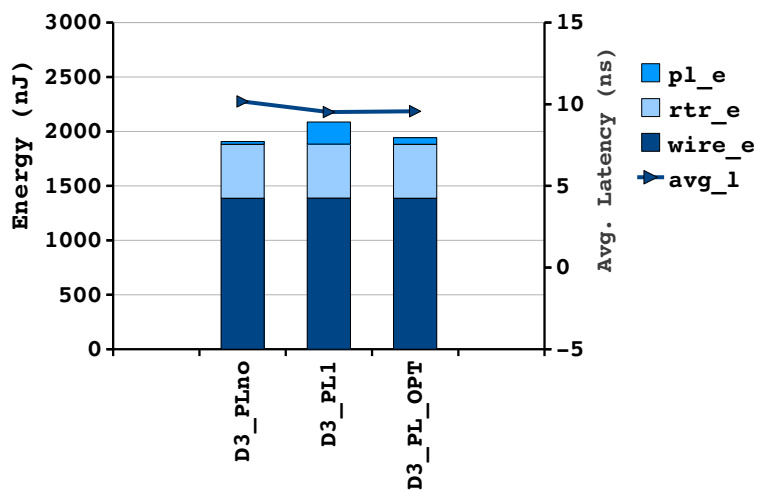


(a) Achievable BW of 12 sender links comparison with $D3_PLno$.

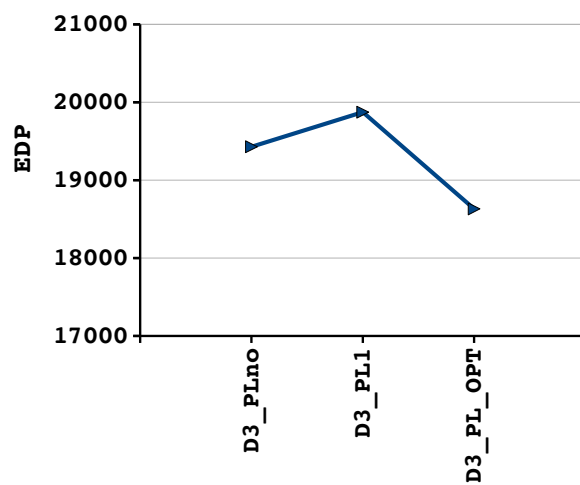


(b) Avg. Latency of 17 paths comparison with $D3_PLno$.

Figure 5.16: $D3_PL_OPT$ design improvement in acBW and path average latency.



(a) Energy and Avg. Latency comparison



(b) EDP

Figure 5.17: *D3_PL_OPT* design improvement in energy, latency and EDP.

Figure 5.17(b). The EDP of the *D3_PL_OPT* is improved 4% from *D3_PLno* and 6% from the *D3_PL1* design.

Table 5.3 summarizes the design results of *D3_PL_OPT* with the total number of PL inserted and compares it to the *D3_PLno* and *D3_PL1* NoCs. The optimally placed 21 PLs in *D3_PL_OPT* accomplished the same performance as the *D3_PL1* design that uses an additional 134 PLs than the *D3_PLno* design.

5.3.2.2 Area-critical Link Optimization for TI Design

Obviously, there are some links for which assigned BW is much greater than required in the *D3_PL_OPT* NoC design. In this section, the *D3_PL_OPT* design will be further optimized for wire routing area, leveraging the NDP (Narrow Data-Path) optimization method, presented in Section 4.3. The NDP method utilizes excessive link BW in low traffic links, for saving wire routing area as narrowing data-path width of such links. Any NoC performance improvement by the NDP method is not expected. Rather, some degradation of performance can occur due to reduction of link BW by half size data-path width even in low traffic links.

The NoCs for the TI design use a 76-bit flit size: 64-bit data-path and 12 routing address bits. So, links with NDP have a 32-bit data-path and 12 routing address bits, thereby creating a 42% reduction of the number of wires. As the wire routing area of a link is proportional to the number of wires of the link, a similar amount of wire routing area reduction is expected.

In the selection of low traffic links where NDP modules are employed, three rules were applied. First, only the lowest level of links, sender and receiver links, are considered among low traffic links. Most links with long wire length, preferable to the NDP method, are sender or receiver links in the floor plan of the TI design. Second, a receiver link is selected only when its utilization is below 5%. Receiver links can

Table 5.3: *D3_PL_OPT* design result comparison.

	Num PL	Avg L (ns)	Total E (nJ)	EDP
<i>D3_PLno</i>	35	10.17	1910	19428
<i>D3_PL1</i>	169	9.52	2087	19872
<i>D3_PL_OPT</i>	56	9.58	1944	18630

impact NoC performance even though its utilization is low, since it affects acBW of all precedent links. Reducing avBW of a receiver link might increase contention in its precedent links. Some receiver links are excluded, even though the link utilization is less than 5%, if they are directly related to the acBW of any performance-critical links of the previous section. Finally, a sender link is chosen when a link's avBW is much greater than its acBW and with very low traffic. The acBW of a sender link is largely limited by packet contention in its subsequent links, so that the acBW of sender links are generally low. In consequence, there is BW margin between avBW and acBW of a sender link. The NDP module was applied to such a sender link of which acBW was not degraded by link avBW reduction due to NDP module penalty. Unlike the receiver links, reducing avBW of sender links makes no effect on the other links' acBW because they are the lowest level of upward links.

A total of 22 links are chosen for applying NDP modules: 13 sender links and 9 receiver links. In the sender links, NDP_NW (Narrow-Wide) modules were attached on an input of router connection, while NDP_WN (Wide-Narrow) modules are used in receiver links as attached on an output of a router connection.

In addition, as shown in Figure 4.11 of Section 4.3, the NDP method becomes more effective when it is applied in conjunction with PL insertion as the BW reduction of the NDP links are relieved considerably by one PL. Thus, it was considered to insert one PL in each of all 22 NDP links. Meanwhile, the *D3_PL_OPT* design already has at least one PL in all receiver links. Note that the *D3_PLno* has one PL in all receiver links to match the other two designs of *Type 2*, and the *D3_PL_OPT* is the NoC with additional PLs inserted into the *D3_PLno*. Therefore, extra 13 PLs are inserted in the 13 sender links with NDP_NW module. The energy overhead by the newly added 13 PLs is insignificant because the NDP links are very low trafficked.

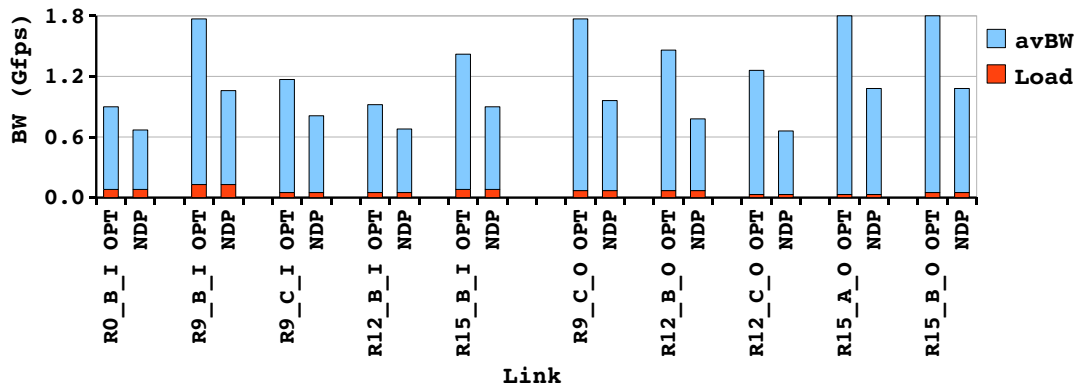
In Table 5.4, all NDP links are presented with their wire length and decreased wire routing area by NDP module. The total saved routing area is 1.38 mm^2 , or a reduction of 14.5% in the total routing area. Area and leakage overhead of the 13 NDP_NW and 9 NDP_WN is negligible. The total area of the 13 NDP_NW is $6812 \mu\text{m}^2$ and the nine NDP_WN is $5652 \mu\text{m}^2$. Both are smaller in area than one D3 router ($8619 \mu\text{m}^2$).

Table 5.4: Routing area of links with NDP module.

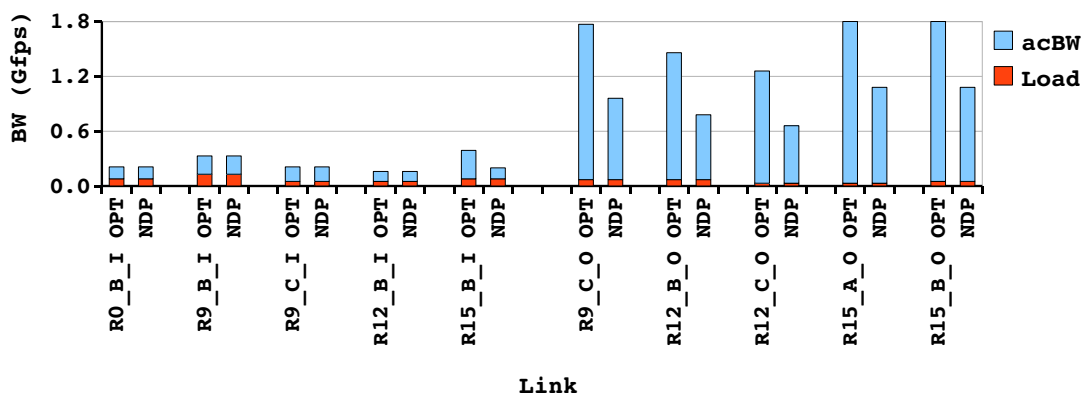
Link	wire_len(μm)	Routing Area (μm^2)	
		OPT	NDP
R0_B_I	3668	252296	146684
R3_B_O	451	31539	18335
R8_C_I	553	38672	22482
R9_B_I	928	64897	37728
R9_C_O, R9_C_I	2357	162689	94585
R12_B_O, R12_B_I	3526	242590	141041
R12_C_O, R12_C_I	4637	318527	185191
R15_A_O, R15_A_I	1662	115186	66966
R15_B_O, R15_B_I	1600	110949	64502
R16_A_I	2958	203768	118469
R19_C_O, R19_C_I	2972	204725	119025
R21_A_I	1674	116007	67443
R23_B_O, R23_B_I	389	27203	15815
R23_C_O	1551	107600	62555
R25_C_I	1678	116280	67602
Total Area		3295251	1915556

Figure 5.18 shows the reduction in avBW and acBW of the NDP links. In both Figure 5.18(a) and Figure 5.18(b), the first five links are from the 13 sender links with NDP, and the last 5 links are receiver links. In both sender and receiver links, predictably, avBW is reduced by the NDP module penalty as shown in Figure 5.18(a). The amount of BW reduction in the receiver links is larger than that of sender links. This is because the receiver links have a PL before inserting the NDP module, whereas a new PL was inserted in each sender link with the NDP module to alleviate the BW reduction by the NDP method.

The acBW reduction in Figure 5.18(b) is different between the sender and receiver links. The acBW of the sender links is not affected at all by the avBW reduction by the NDP modules, except for one link, R15_B_I. The acBW of the sender links is mainly determined by packet contention in their subsequent links. Therefore, they normally have a large margin of the avBW and some reduction of avBW by the NDP module can hardly affect their acBW. The acBW of R15_B_I is directly affected by the avBW reduction of the next subsequent link, R15_B_O, not by itself (R15_B_O is one of the receiver links with the NDP module.)



(a) Available BW reduction.



(b) Achievable BW reduction.

Figure 5.18: AvBW and acBW reduction by NDP module in five sender links and five receiver links.

On the contrary, the acBW reduction of receiver links is exactly the same as the reduction in avBW, since they do not suffer from packet contention and therefore always have the same avBW and acBW. The reduced acBW of all receiver links is still sufficient when compared to their low traffic loads.

Table 5.5 summarizes the design results of the NoC optimized through the NDP method (*D3_PL_NDP*) from *D3_PL_OPT*, and is compared with the previous three NoCs. The number of PLs (*Num PL*) of the *D3_PL_NDP* design is 13 more than *D3_PL_OPT*. Those are inserted in the 13 sender links with the NDP modules. The NoC performance of *D3_PL_NDP* was affected by the BW reduction in the links with the NDP modules and consequently, the average latency returned back to that of

Table 5.5: *D3_PL_NDP* design result comparison.

	Num PL	Avg L (ns)	Total E (nJ)	EDP	Wire Area (mm^2)		
					Repeater	Routing	Total
<i>D3_PLno</i>	35	10.17	1910	19428	1.11	8.40	9.51
<i>D3_PL1</i>	169	9.52	2087	19872	1.11	8.40	9.51
<i>D3_PL_OPT</i>	56	9.58	1944	18630	1.11	8.40	9.51
<i>D3_PL_NDP</i>	69	10.19	1939	19761	0.95	7.18	8.13

D3_PLno NoC. Moreover, with the increased number of PLs over *D3_PLno*, the total energy increased and thereby, the EDP of *D3_PL_NDP* is a little bit worse .

However, the reduced wire area is $1.38 mm^2$, 14.5% of the total wire area of the other three NoCs. It can be considered that the reduced wire area cost is offset by the benefits from the performance improvement by the PL insertion in the OPT design. The performance gain is transformed to the routing area reduction.

It is also possible to make the average latency of the *D3_PL_NDP* optimization comparable to the *D3_PL_OPT* NoC, if the NDP method is restrictively applied into a fewer number of links, sacrificing the amount of saved routing area but preventing performance degradation by the NDP method.

5.3.2.3 Energy-critical Link Optimization for TI Design

In this section, the third optimization method, DS (Double-Spacing), is applied to the *D3_PL_NDP* NoC and presents another NoC, *D3_PL_DS*, which is optimized in wire energy consumption as exploiting the saved wire areas through the NDP method in the previous section.

Link wire energy is proportional to not only the number of packets of the link but also link wire length. As ANetGen performs optimization of topology and floor plan concentrating mainly on high traffic links, links with high traffic loads are already optimized to have short wire length. Thus, energy-critical links, candidate links for the DS method, should be links with medium traffic loads with relatively long wire length.

Based on the energy ratio of each link to total wire energy of the *D3_PL_NDP*, 23 links, which consume the most wire energy, are selected for the DS method. The number of links for the DS method was limited such that the increased total wire

routing area overhead of the DS method is not more than the wire area saved by the NDP method, 1.38 mm^2 .

The ratio of the energy consumption of the 23 selected links is shown in Table 5.6. The sum of wire energy consumption of these 23 links is 52% of the total wire energy.

As a result, by using double-spaced link wires instead of single-spaced wires in these 23 energy-critical links, the total wire energy consumption is reduced by 15.7% with a 1.13 mm^2 wire area overhead. The cost of wire energy reduction is fully offset by the benefits of the EDP optimization in the *D3_PL_DS* NoC. Detailed design results of the *D3_PL_DS* NoC are presented in the following section in summarizing all NoC designs for the TI example.

5.3.2.4 Results of Optimized NoCs for TI Design

This section compares the design results of five different NoCs: two base NoCs with a D3 router, *D3_PLno* and *D3_PL1*, and three optimized NoCs, *D3_PL_OPT*, *D3_PL_NDP* and *D3_PL_DS* which are distinguished by the applied optimization methods.

- *D3_PLno* : base design with no PL insertion.
- *D3_PL1* : base design with one PL in all links.
- *D3_PL_OPT* : performance optimized design by PL insertion from *D3_PLno*.
- *D3_PL_NDP* : design with area optimization through the NDP method from *D3_PL_OPT*.

Table 5.6: Wire energy ratio of 23 DS links to total wire energy consumption.

Link	Ratio(%)	Link	Ratio(%)	Link	Ratio(%)
R0_B_O	4.31	R4_C_O	2.10	R18_C_O	2.21
R0_C_O	1.33	R4_C_I	2.10	R18_C_I	2.55
R1_A_O	4.28	R6_A_O	1.43	R20_B_O	1.34
R1_A_I	4.13	R6_B_O	1.66	R20_C_O	1.27
R1_C_O	2.17	R6_B_I	1.82	R26_B_O	1.20
R2_B_O	3.27	R7_B_O	1.67	R29_A_O	1.82
R2_B_I	3.39	R7_B_I	1.80	R31_A_O	2.15
R4_A_O	2.75	R10_A_O	1.31	Total	52.06

- *D3_PL_DS*: design with wire energy optimization through the DS method from *D3_PL_NDP*.

Table 5.7 summarizes designs in terms of the number of PLs inserted, the total aggregated acBW (*Total acBW*) and average latency (*Avg. L*). The total aggregated acBW is the sum of acBW of all links in a design.

The *D3_PLno* has one PL in all 35 receiver links, while the *D3_PL1* has one PL in all 134 links and additional PL in all 35 receiver links. The *D3_PL_OPT* is a design which is optimized in performance by inserting PLs strategically only in performance-critical links. An additional 21 PLs are inserted in the *D3_PLno* design. More 13 PLs are inserted in *D3_PL_NDP* to minimize NDP overhead in the sender links in which the NDP module are inserted. The last design, *D3_PL_DS* has the same number of PLs from *D3_PL_NDP* design, as no more PLs are inserted.

Throughout this thesis, a PL in asynchronous communication links was intended mainly for link BW improvement, even though it provides additional buffering. So, as more PLs are inserted in an asynchronous NoC, the NoC has more link BW and hence better NoC performance is expected. Accordingly, in the comparison between the *D3_PLno* and *D3_PL1* NoC, the *D3_PL1* with 134 more PLs has more total acBW and therefore, performs better than *D3_PLno*.

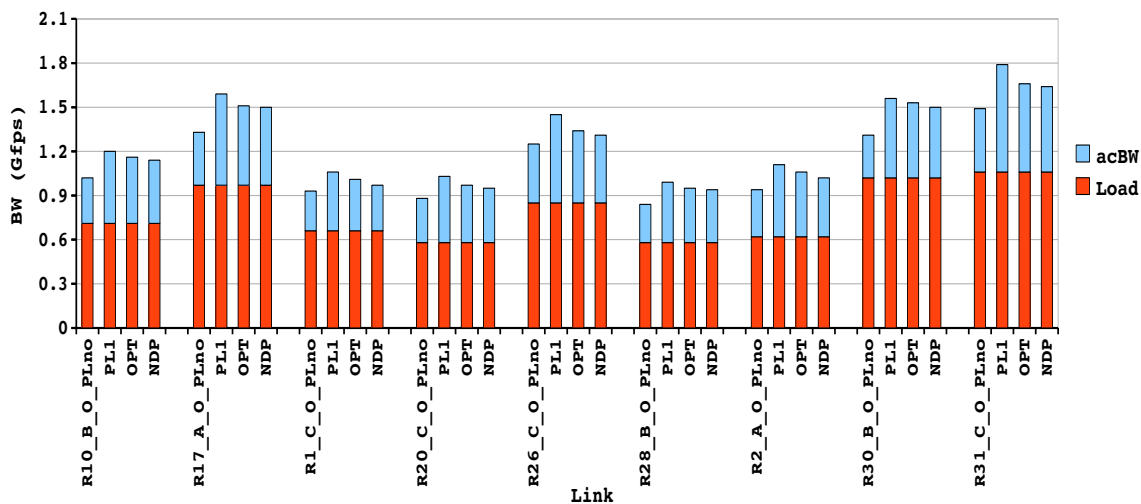
However, the PL number in an NoC is not always directly transformed into NoC performance. Rather, the effectiveness of PL insertion is more important as shown with the *D3_PL_OPT* design. *D3_PL_OPT* has only 21 more PLs than the *D3_PLno*, which is 113 less PLs than *D3_PL1*. Subsequently, the total acBW of the *D3_PL_OPT* is less than that of the *D3_PL1*. Nevertheless, it performs similarly to *D3_PL1* as the 21 PLs were inserted strategically in performance-critical links in the *D3_PL_OPT*

Table 5.7: Design summary of five NoCs.

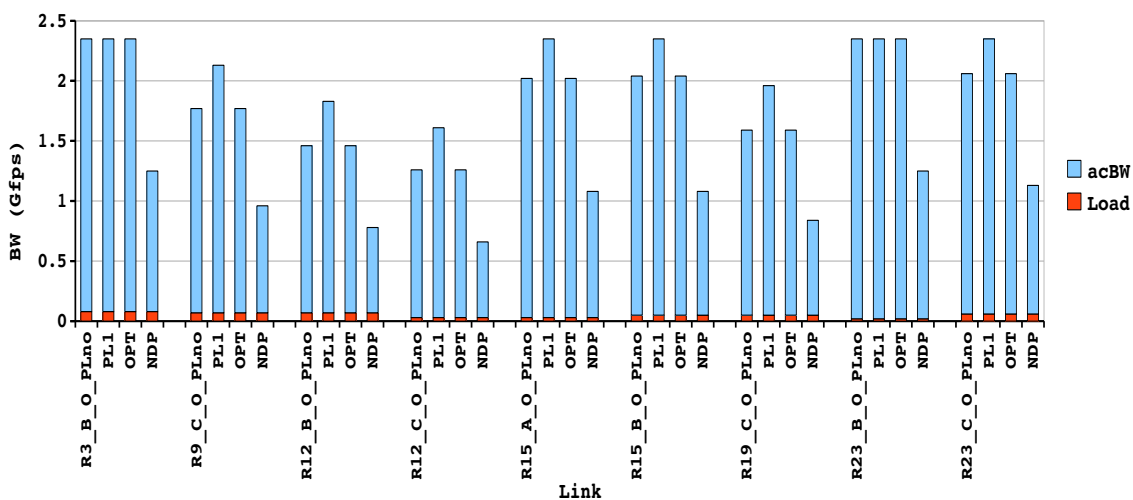
	Num PL	Total acBW (Gfps)	Avg. L(ns)
<i>D3_PLno</i>	35	165.25	10.17
<i>D3_PL1</i>	169	188.34	9.52
<i>D3_PL_OPT</i>	56	178.27	9.58
<i>D3_PL_NDP</i>	69	164.79	10.19
<i>D3_PL_DS</i>	69	164.79	10.19

design, and this accomplished a balanced link BW assignment with consideration of link traffic loads: more link BW in high trafficked links and less BW in low trafficked links.

This is clearly shown in Figure 5.19 which compares the acBW of the nine most trafficked links (Figure 5.19(a)) and the nine least trafficked links (Figure 5.19(b)) between NoCs. (The total acBW and average latency of the *D3_PL_DS* design has no difference with *D3_PL_NDP*, so it is not separately compared in Figure 5.19.)



(a) AcBW of the most utilized links.



(b) AcBW of the least utilized links.

Figure 5.19: AcBW comparison between all D3 designs in the most utilized and the least utilized links .

The *D3_PL_OPT* design has comparable acBW in most of the high trafficked links, whereas it assigned less BW in many low trafficked links compared to the *D3_PL1* NoC. The consequence is that similar performance is achieved with fewer design resources (fewer PLs) creating a more efficient NoC.

Similarly, another link BW optimized design can be seen through comparison between the *D3_PLno* and the *D3_PL_NDP*. Both NoCs show almost identical total acBW as well as average latency in Table 5.7. Furthermore, it would seem that the *D3_PL_NDP* is worse than *D3_PLno* due to more 34 PLs. However, link BW assignment of the two NoCs are significantly different from each other. The *D3_PLno* has less acBW in high trafficked links, compared to the others, while similar acBW is assigned in low trafficked links. In contrast, the acBW in high trafficked links of the *D3_PL_NDP* is almost equal to those of the *D3_PL_OPT*, whereas much lower link BW is assigned into the low trafficked links. The link BW assignment of the *D3_PL_NDP* is more balanced with link traffic loads than even the *D3_PL_OPT* design. As a result, in virtue of effectively balanced link BW assignment, the *D3_PL_NDP* accomplished the reduction of wire area compared to the *D3_PLno*.

Table 5.8 compares five NoC designs in terms of wire area and wire energy. Three NoCs, the *D3_PLno*, *D3_PL1* and *D3_PL_OPT* have identical link wire design, resulting in the same properties. By employing 22 NDP modules in low traffic links, the *D3_PL_NDP* saves 1.38 mm^2 in total wire area including wire repeater and wire routing areas, and it leads to a slight reduction in total wire energy owing to the reduced wire leakage power and fewer repeaters.

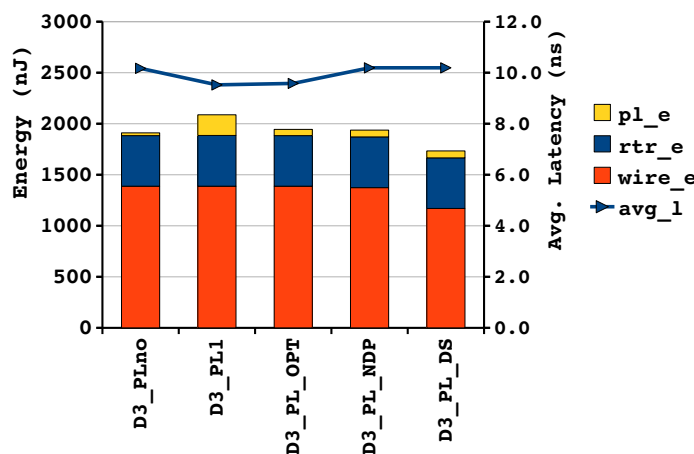
The advantage of the *D3_PL_DS* is observed in the total wire energy. Wire energy saved in the 23 energy-critical links of the *D3_PL_DS* design results in a 15.8% reduction of the total wire energy, compared to the other NoC designs. This energy

Table 5.8: Design summary: wire area and energy comparison.

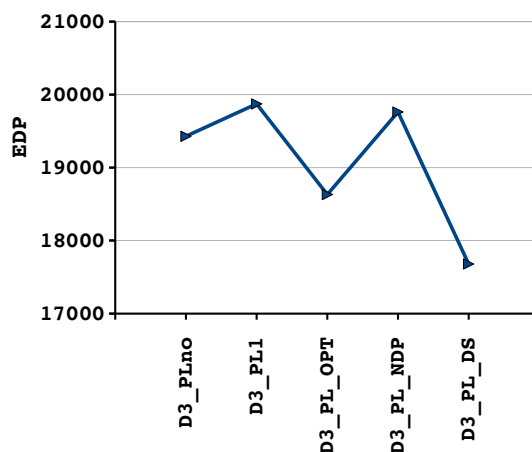
	Area (mm^2)			Wire Energy (nJ)		
	Repeater	Routing	Total	Leakage	Dynamic	Total
PLno, PL1, PL_OPT	1.11	8.40	9.51	82	1306	1388
PL_NDP	0.95	7.18	8.13	67	1303	1370
PL_DS	0.84	8.41	9.25	62	1107	1169

benefit comes at the expense of the wire area overhead. But, by exploiting the wire area saved by the NDP method, the $D3_PL_DS$ uses approximately the same wire area as the first three NoCs.

Overall, the total NoC energy and average latency is shown in Figure 5.20(a), and Figure 5.20(b) depicts the EDP of the five NoC designs. From the perspective of NoC performance, the $D3_PL_OPT$ is the best design with the lowest average latency while reducing energy consumption. Based on EDP values, the $D3_PL_DS$ design is the most performance-energy efficient NoC for the TI example. The EDP of the $D3_PL_DS$ is improved 9% over that of the $D3_PLno$ NoC.



(a) Energy and Avg. Latency



(b) EDP

Figure 5.20: Five D3 designs comparison.

5.4 Summary

Through the implementation of asynchronous NoCs for two SoC examples, the advantages and optimization of asynchronous NoCs were presented. With the first example, an MPEG4 design, the benefit of bandwidth optimization in the design of the asynchronous NoC was shown by means of comparing the asynchronous NoC with similarly designed synchronous NoCs in terms of performance and energy consumption. The topology and placement optimizations, in consideration of traffic loads of each link, create the asynchronous NoC design in which most of performance-critical links have wire length short enough to minimize the link wire delay penalty on asynchronous communication links. Furthermore, no idle clock energy is the main advantage of the asynchronous NoC design.

The optimization of an asynchronous NoC design was presented with the TI example. Three optimization methods were applied to the initial NoC design in turn. The PL insertion method achieved the best NoC performance while minimizing NoC design costs. The NDP method achieved a wire area optimized NoC, while the DS method saved considerably wire energy consumption. The three optimization methods can be applied independently according to the primary constraints of an NoC design, or they can be used all together as presented.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The primary advantage of asynchronous NoCs is the ability to customize individual link BW based on its respective requirement by simply adjusting controller locations. This work investigates the benefit of bandwidth optimization in the designs of asynchronous NoCs.

Three asynchronous routers were designed based on simple and efficient circuit. By comparing performance of three different router designs, the link wire delay impact on the router performance was presented.

The effect of pipeline latch insertion in asynchronous communication links was evaluated. Optimally placed PL can maximizing the benefit of PL insertion to link BW improvement. So, a way of computing optimal positions of PLs was proposed. Eight different asynchronous communication links were proposed, based on three router designs and number of PLs inserted. In addition, link BW variance of those links were evaluated and compared.

Three optimization methods for asynchronous NoCs were proposed performance, area and energy improvement, respectively. Improvement of link BW can be effectively controlled by the PL insertion. So, it was proposed as an optimization method for improving asynchronous NoC performance, by means of inserting PLs in performance-critical links. The NDP method can be used for saving link wire routing area as trimming excessive link BW in low traffic links by adjusting link BW of such links through narrowing link data-path. Two data-path width converters were implemented for this method. Energy consumption by link wire is normally the largest portion of total NoC energy. Controlling space of adjacent wires can result in considerable reduction in link wire energy and thereby, energy-optimized NoC. In

order to employ each optimization method to proper links in an NoC, it is required to know link properties based on link utilization. An analytical model of link BW estimation in an NoC composed of a three-port router was presented. In particular, the three optimization methods are considerably efficient in that they do not require any modification of other design parameters, such as network topology, floor plan, or router designs.

Comparison between similarly-designed asynchronous and synchronous NoC with one SoC example shows that exploiting the controllability of each link BW by link wire length of asynchronous designs results in comparable performance to the synchronous one. The topology and placement optimizations can almost obviate the link wire delay penalty on link BWs in the asynchronous NoC. In addition, no energy consumption by idle clocking and clock distribution makes the asynchronous NoC much better than the synchronous NoC.

Three optimization methods were applied to an asynchronous NoC for an SoC design and the optimization results were presented. The PL insertion method achieved improvement of NoC performance by 5.8%, compared to initial unoptimized NoC. The performance benefit comes at the expenses of total energy increase by 2.3%. The NDP method saved 14.5% of the total wire area, while performing similarly to the initial NoC. Furthermore, energy consumption by link wires are reduced by 15.8%, and it results in 9% improvement of EDP.

6.2 Future Work

Several further work is able to enhance the results of this research. First, in this work, the three-port routers were designed based on unconventional design parameters, simple source-routing and single-flit packet, which enabled simple and efficient router designs. However, these design parameters have some drawbacks. As the routing address needs to be transferred along with data, separate wires for the routing address are required. This results in more dynamic energy, leakage power and routing area of link wires. In addition, the maximum throughput of three routers are limited by the the MUTEX element for arbitration in their merge module, which has long logic delay and operates a 4-phase protocol. Every flit needs to pass the *ar_ckt*

with single-flit packet format. On the other hand, a multiflit packet with worm-hole routing scheme is widely used in other NoC designs. The multiflit packet does not need separate wires for the routing address since the address information is sent as a header flit. Furthermore, since only the header flit needs to pass the MUTEX element and body and tail flits can access an output port without the arbitration delay, performance improvement is expected. However, extra circuits are required to set up and free packet routes for supporting the multiflit format. It would be worthwhile designing an asynchronous router with the multiflit packet format and comparing the trade-off between two designs.

Second, it is desirable to implement an automatic tool in employing the optimization methods. Even though the approach used in this work was systematic using link utilization, several simulations were required to get valuable optimization. So, development of a tool which can find an optimization solution as taking an input with parameterized conditions could improve the optimization process considerably.

Third, the network adapter, another main component of NoC, is an interface circuit between IP cores and an NoC, through protocol conversion, synchronization and packetization. For employing the proposed asynchronous NoC design for real SoC design, a specific network adapter needs to be prepared preferentially.

REFERENCES

- [1] G. D. Micheli and L. Benini, *Networks on Chips*. Morgan Kaufmann, 2006.
- [2] B. Towles and W. J. Dally, “Route packets, not wires: On-chip interconnectoin networks,” *Design Automation Conference*, vol. 0, pp. 684–689, 2001.
- [3] T. Bjerregaard and S. Mahadevan, “A survey of research and practices of network-on-chip,” *ACM Computing Surveys*, vol. 38, no. 1, 2006.
- [4] S. Shukla, K. Stevens, and E. M. Kishinevsky, “Special issue globally asynchronous locally synchronous design,” in *IEEE Design & Test*. IEEE Computer Society, Sep.-Oct. 2007.
- [5] K. S. Stevens, D. Gebhardt, J. You, Y. Xu, V. Vij, S. Das, and K. Desai, “The future of formal methods and GALS design,” in *Electronic Notes in Theoretical Computer Science*, vol. 245, no. 1, 2009, pp. 115–134.
- [6] W. Dally, “Virtual-channel flow control,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 3, pp. 194–205, 1992.
- [7] T. Bjerregaard and J. Sparsø, “Implementation of guaranteed services in the MANGO clockless network-on-chip,” *IEEE Proceedings: Computing and Digital Techniques*, vol. 153, no. 4, pp. 217–229, 2006.
- [8] —, “A scheduling discipline for latency and bandwidth guarantees in asynchronous network-on-chip,” in *Proceedings of the 11th IEEE International Symposium on Asynchronous Circuits and Systems*, 2005, pp. 34–43.
- [9] T. Bjerregaard, S. Mahadevan, R. G. Olsen, and J. Sparsø, “An OCP compliant network adapter for GALS-based SoC design using the MANGO network-on-chip.” in *Proceedings of International Symposium on System-on-Chip 2005*. IEEE, 2005.
- [10] T. Bjerregaard and J. Sparsø, “Virtual channel designs for guaranteeing bandwidth in asynchronous network-on-chip,” in *Proceedings of the IEEE Norchip Conference (NORCHIP 2004)*. IEEE, 2004.
- [11] W. J. Bainbridge and S. B. Furber, “CHAIN: A delay insensitive chip area interconnect,” *IEEE Micro Special Issue on Design and Test of System on Chip*, vol. 142, No.4., pp. 16–23, Sep. 2002.
- [12] D. R. Rostislav, V. Vishnyakov, E. Friedman, and R. Ginosar, “An asynchronous router for multiple service levels networks on chip,” *Asynchronous Circuits and Systems, International Symposium on*, vol. 0, pp. 44–53, 2005.

- [13] R. Dobkin, R. Ginosar, and A. Kolodny, "QNOC Asynchronous router," *VLSI Journal*, vol. 42, pp. 103–115, Feb. 2009.
- [14] E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "QNoC: QoS architecture and design process for network on chip," *Journal of Systems Architecture, Special Issue on Network on Chip*, vol. 50, pp. 105–128, Feb. 2004.
- [15] I. Miro-Panades, F. Clermidy, P. Vivet, and A. Greiner, "Physical implementation of the DSPIN network-on-chip in the FAUST architecture," in *NOCS '08: Proceedings of the Second ACM/IEEE International Symposium on Networks-on-Chip*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 139–148.
- [16] P. Maurine, J. Rigaud, F. Bouesse, G. Sicard, and M. Renaudin, "Static implementation of QDI asynchronous primitives," in *13th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS2003)*, Sep. 2003, pp. 181–191.
- [17] I. M. Panades and A. Greiner, "Bi-synchronous FIFO for synchronous circuit communication well suited for network-on-chip in GALs architectures," May. 2007.
- [18] K. Goossens, J. Dielissen, and A. Rădulescu, "The Æthereal network on chip: Concepts, architectures, and implementations," *IEEE Design and Test of Computers*, vol. 22, no. 5, pp. 414–421, - 2005.
- [19] K. S. Stevens, "Energy and performance models for clocked and asynchronous communication," in *the 9th IEEE International Symposium on Asynchronous Circuits and Systems*. IEEE, May. 2003, pp. 56–66.
- [20] K. S. Stevens, P. Golani, and P. A. Beerel, "Energy and performance models for synchronous and asynchronous communication," *IEEE Trans. on VLSI Systems*, 2010.
- [21] R. Ho, J. Gainsley, and R. Drost, "Long wires and asynchronous control," in *The 10th IEEE International Symposium on Asynchronous Circuits and Systems*, 2004, pp. 240–249.
- [22] Z. Guz, I. Walter, E. Bolotin, I. Cidon, R. Ginosar, and A. Kolodny, "Network delays and link capacities in application-specific wormhole nocs," *VLSI Design*, vol. 2007, 2007.
- [23] C. Mead and L. Conway, *Introduction to VLSI Systems*. Addison-Wesley, 1980.
- [24] J. Sparso and S. B. Furber, *Principles of Asynchronous Circuit Design: A Systems Perspective*. Springer, 2001.
- [25] K. S. Stevens, R. Ginosar, and S. Rotem, "Relative timing," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 11, no. 1, pp. 129–140, 2003.
- [26] R. Milner, *Communication and Concurrency*. London. U.K.:Prentice-Hall, 1989.

- [27] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, and A. Yakovlev, "Petrify: A tool for manipulating concurrent specifications and synthesis of asynchronous controllers," *IEICE Transactions on Information and Systems*, vol. E80-D, no. 3, pp. 315–325, Mar. 1997.
- [28] K. S. Stevens, Y. Xu, and V. Vij, "Characterization of asynchronous templates for integration into clocked cad flows," in *15th International Symposium on Asynchronous Circuits and Systems*, May. 2009, pp. 151–161.
- [29] K. S. Stevens, "Practical verification and synthesis of low latency asynchronous systems," Ph.D. dissertation, University of Calgary, Sep. 1994.
- [30] Y. Xu and K. S. Stevens, "Automatic synthesis of computation interference constraints for relative timing verification," in *Proc. of the 26th Intl. Conf. on Computer Design (ICCD)*, Oct. 2009, pp. 16–22.
- [31] L. Carloni, A. Kahng, S. Muddu, A. Pinto, K. Samadi, and P. Sharma, "Accurate predictive interconnect modeling for system-level design," *IEEE Trans. on VLSI Systems*, vol. 18, no. 4, pp. 679–684, Apr. 2010.
- [32] A. Kahng, B. Li, L.-S. Peh, and K. Samadi, "ORION 2.0: A fast and accurate NoC power and area model for early-stage design space exploration," in *DATE*, Apr. 2009, pp. 423–428.
- [33] E. B. V. D. Tol and E. G. T. Jaspers, "Mapping of MPEG-4 decoding on a flexible architecture platform," in *Media Processors*, 2002, pp. 1–13.
- [34] "Texas Instruments Inc." [Online]. Available: www.ti.com
- [35] D. Gebhardt, J. You, and K. S. Stevens, "Comparing energy and latency of asynchronous and synchronous nocs for embedded SoCs," in *4th IEEE International Symposium on Network-on-Chips*, May 2010.
- [36] S. Adya and I. Markov, "Fixed-outline floorplanning: Enabling hierarchical design," *IEEE Trans. on VLSI*, vol. 11, no. 6, pp. 1120–1135, Dec. 2003.
- [37] J. You, Y. Xu, H. Han, and K. S. Stevens, "Performance evaluation of elastic GALS interfaces and network fabric," *Electron. Notes Theor. Comput. Sci.*, vol. 200, no. 1, pp. 17–32, 2008.
- [38] J. Cortadella, M. Kishinevsky, and B. Grundmann, "Synthesis of synchronous elastic architectures," in *Proceedings of the Digital Automation Conference (DAC06)*. IEEE, Jul. 2006, pp. 657–662.
- [39] L. P. Carloni, K. L. McMillan, and A. L. Sangiovanni-Vincentelli, "Theory of latency-insensitive design," *IEEE Transaction on Computer aided design of integrated circuits and systems*, vol. 20, pp. 1059–1076, 2001.
- [40] E. Beigne, F. Clermidy, P. Vivet, A. Clouard, and M. Renaudin, "An asynchronous noc architecture providing low latency service and its multi-level design framework," in *the 11th IEEE International Symposium on Asynchronous Circuits and Systems*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 54–63.

- [41] W. J. Dally, "Virtual-channel flow control," in *Proc. of the 17th Annual International Symposium on Computer Architecture (ISCA)*, Seattle, Washington, May 1990, pp. 60–68.
- [42] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, 2003.
- [43] T. Felicijan, "Quality-of-service (QoS) for asynchronous on-chip networks," Ph.D. dissertation, Department of Computer Science, University of Manchester, 2004.
- [44] D. Gebhardt and K. S. Stevens, "Elastic flow in an application specific network-on-chip," *Electron. Notes Theor. Comput. Sci.*, vol. 200, no. 1, pp. 3–15, 2008, Proc. Int'l Workshop on Formal Methods for GALS.
- [45] C. A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M. S. Yousif, and C. R. Das, "Vichar: A dynamic virtual channel regulator for network-on-chip routers," *Microarchitecture, IEEE/ACM International Symposium on*, vol. 0, pp. 333–346, 2006.
- [46] N. Muralimanohar and R. Balasubramonian, "Interconnect design considerations for large nuca caches," in *Proceedings of the 34th Annual International Symposium on Computer Architecture*, ser. ISCA '07. ACM, 2007, pp. 369–380.
- [47] R. Balasubramonian, N. Muralimanohar, K. Ramani, L. Cheng, and J. B. Carter, "Leveraging wire properties at the microarchitecture level," *IEEE Micro*, vol. 26, pp. 40–52, Nov. 2006.
- [48] R. Ho, K. Mai, and M. Horowitz, "The future of wires," *Proceedings of the IEEE*, vol. 89, no. 4, pp. 490–504, Apr. 2001.
- [49] D. Lattard, E. Beigne, F. Clermidy, Y. Durand, R. Lemaire, P. Vivet, and F. Berens, "A reconfigurable baseband platform based on an asynchronous network-on-chip," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 1, pp. 223–235, Jan. 2008.
- [50] V. Soteriou, H. Wang, and L. Peh, "A statistical traffic model for on-chip interconnection networks," in *Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006. MASCOTS 2006. 14th IEEE International Symposium on*, Sep. 2006, pp. 104 – 116.
- [51] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Performance evaluation and design trade-offs for network-on-chip interconnect architectures," *IEEE Transactions on Computers*, vol. 54, pp. 1025–1040, 2005.
- [52] L.-S. Peh and W. Dally, "A delay model for router microarchitectures," *Micro, IEEE*, vol. 21, no. 1, pp. 26–34, Jan./Feb. 2001.
- [53] Z. Yu and B. Baas, "A low-area multi-link interconnect architecture for GALS chip multiprocessors," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 5, pp. 750–762, May. 2010.

- [54] D. Bertozzi, A. Jalabert, S. Murali, R. Tamhankar, S. Stergiou, L. Benini, and G. De Micheli, “Noc synthesis flow for customized domain specific multiprocessor systems-on-chip,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 16, no. 2, pp. 113 – 129, Feb. 2005.
- [55] F. Feliciian and S. Furber, “An asynchronous on-chip network router with quality-of-service (qos) support,” in *SOC Conference, 2004. Proceedings. IEEE International*, Sep. 2004, pp. 274 – 277.
- [56] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tien-syrja, and A. Hemani, “A network on chip architecture and design methodology,” in *VLSI, 2002. Proceedings. IEEE Computer Society Annual Symposium on*, 2002, pp. 105 –112.
- [57] F. Angiolini, P. Meloni, S. M. Carta, L. Raffo, and L. Benini, “A layout-aware analysis of networks-on-chip and traditional interconnects for mpsoCs,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 3, pp. 421 –434, Mar. 2007.
- [58] U. Ogras and R. Marculescu, “Analytical router modeling for networks-on-chip performance analysis,” in *Design, Automation Test in Europe Conference Exhibition, 2007. DATE '07*, Apr. 2007, pp. 1 –6.
- [59] T. Ye, L. Benini, and G. De Micheli, “Analysis of power consumption on switch fabrics in network routers,” in *Design Automation Conference, 2002. Proceedings. 39th*, 2002, pp. 524 – 529.
- [60] E. Rijpkema, K. Goossens, A. Radulescu, J. Dielissen, J. van Meerbergen, P. Wielage, and E. Waterlander, “Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip,” *Computers and Digital Techniques, IEEE Proceedings -*, vol. 150, no. 5, pp. 294–302, Sep. 2003.
- [61] A. Lines, “Asynchronous interconnect for synchronous SoC design,” *Micro, IEEE*, vol. 24, no. 1, pp. 32 – 41, Jan.-Feb. 2004.
- [62] K. Banerjee and A. Mehrotra, “A power-optimal repeater insertion methodology for global interconnects in nanometer designs,” *Electron Devices, IEEE Transactions on*, vol. 49, no. 11, pp. 2001 – 2007, Nov. 2002.
- [63] B. Quinton, M. Greenstreet, and S. Wilton, “Asynchronous IC interconnect network design and implementation using a standard ASIC flow,” in *Computer Design: VLSI in Computers and Processors, 2005. ICCD 2005. Proceedings. 2005 IEEE International Conference on*, Oct. 2005, pp. 267 – 274.
- [64] A. Tran, D. Truong, and B. Baas, “A GALS many-core heterogeneous DSP platform with source-synchronous on-chip interconnection network,” in *Networks-on-Chip, 2009. NoCS 2009. 3rd ACM/IEEE International Symposium on*, May 2009, pp. 214 –223.
- [65] T. Chelcea and S. Nowick, “Robust interfaces for mixed-timing systems with application to latency-insensitive protocols,” in *Design Automation Conference, 2001. Proceedings*, 2001, pp. 21 – 26.

- [66] K. Lee, S.-J. Lee, and H.-J. Yoo, "Low-power network-on-chip for high-performance SoC design," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 14, no. 2, pp. 148 –160, Feb. 2006.
- [67] P. Pande, C. Grecu, A. Ivanov, and R. Saleh, "High-throughput switch-based interconnect for future SoCs," in *System-on-Chip for Real-Time Applications, 2003. Proceedings. The 3rd IEEE International Workshop on*, Jun.- Jul. 2003, pp. 304 – 310.
- [68] V. Soteriou, N. Eisley, H. Wang, B. Li, and L.-S. Peh, "Polaris: A system-level roadmap for on-chip interconnection networks," in *Computer Design, 2006. ICCD 2006. International Conference on*, Oct. 2006, pp. 134 –141.
- [69] M. Amde, T. Felicijan, A. Efthymiou, D. Edwards, and L. Lavagno, "Asynchronous on-chip networks," *Computers and Digital Techniques, IEEE Proceedings -*, vol. 152, no. 2, pp. 273 – 283, Mar. 2005.
- [70] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70 –78, Jan. 2002.
- [71] K. Srinivasan and K. Chatha, "Layout aware design of mesh based NoC architectures," in *Hardware/Software Codesign and System Synthesis, 2006. CODES+ISSS '06. Proceedings of the 4th International Conference*, Oct. 2006, pp. 136 –141.
- [72] S. Pestana, E. Rijpkema, A. Radulescu, K. Goossens, and O. Gangwal, "Cost-performance trade-offs in networks on chip: A simulation-based approach," in *Design, Automation and Test in Europe Conference and Exhibition, 2004. Proceedings*, vol. 2, Feb. 2004, pp. 764 – 769 Vol.2.
- [73] H. Wang, L.-S. Peh, and S. Malik, "A technology-aware and energy-oriented topology exploration for on-chip networks," in *Design, Automation and Test in Europe, 2005. Proceedings*, Mar. 2005, pp. 1238 – 1243 Vol. 2.