MULTIPLE CLOCK DOMAIN SYNCHRONIZATION FOR NETWORK ON CHIPS

By

SOURADIP SARKAR

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science in Computer Engineering

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science

DECEMBER 2007

To the Faculty of Washington State University:

The members of the committee appointed to examine the thesis of SOURADIP SARKAR find it satisfactory and recommend that it be accepted.

Chair

ACKNOWLEDGMENT

It is time to pause and reflect back at the end of those months during which I had been working on my Master's research. Time it is to thank the people I had been working with.

First and foremost, I would thank my adviser Dr. Partha Pande. It was a pleasure working with him. I am indebted to him for his help during a critical phase of my life. Next, I would thank my co-adviser Dr. Jabulani Nyathi. It was a privilege working with him and I would always cherish the joint meeting we used to have.

I would keep fond memories of the Low Power and Robust Nanosystems Lab, WSU where I worked. The atmosphere in the lab was always cordial. I take this opportunity to thank my colleagues Mr. Amlan Ganguly, Mr. Haibo Zhu and Mr. Brett Feero.

Lastly, my parents Mr. Dipankar Sarkar and Mrs. Seema Sarkar whose words of encouragement I recall the most.

MULTIPLE CLOCK DOMAIN SYNCHRONIZATION FOR NETWORK ON CHIPS

Abstract

By Souradip Sarkar, M.S.
Washington State University
December 2007

Chair: Partha Pande

This thesis provides a new framework for the design of very high performance yet low power System on Chips (SoCs). Network on chip (NoC) is emerging as a revolutionary methodology to integrate numerous Intellectual Property (IP) blocks in a single System-on-Chip (SoC) and solving the performance limitations arising out of long interconnects. Continued advancement of NoC designs is heavily dependent on the ability to effectively communicate among the constituent Intellectual Property (IP) blocks/Embedded cores, as well as manage/reduce energy dissipation. This work first presents a low-latency, low-energy synchronization mechanism for Network on Chip architectures, which enables the network to span a system-on-chip (SoC) with multiple independent clock domains. The proposed interface scheme has been compared to another existing scheme and shown to outperform it in terms of latency and energy dissipation.

The synchronizers were introduced in the communication fabric for seamless integration of the different Intellectual Property (IP) blocks. As communication happens across clock domains, the clock distribution scheme over the entire network was redesigned for greater savings in power. It is shown that communication energy can be optimized by selecting an appropriate number of different clock regions and their relative

placement. It is demonstrated that in a mesh-based NoC the communication energy initially decreases with increasing number of clock domains, but beyond a certain threshold it shows an increasing trend due to synchronization overhead.

TABLE OF CONTENTS

LIST OF TABLES

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Systems-on-Chip Design Methodology

The notion of integrating numerous components of a computer system into a single chip has led to the miniaturization of many portable devices and an increase in their computational capabilities. Already, such systems are in place and their size, complexity and integration is increasing gradually. The network-on-Chip (NoC) design paradigm is viewed as an enabling solution for further integration of exceedingly high number of computational and storage blocks in a single chip. It is the packet switching based communications backbone that interconnects the components on a multi-core SoC. In addition to enabling the high degree of integration, the success of this new paradigm is heavily dependent on the ability to effectively communicate among the constituent functional modules called Intellectual Property (IP) blocks as well as manage/reduce energy dissipation. The key element of these Multi-processor SoC (MP-SoC) platforms is the communication fabric. Interconnect topologies implemented in ultra deep submicron (UDSM) technologies are plagued by increased latency that arises from global wire delays. Global wires carry signals across a chip, but these wires typically do not scale in length with technology scaling [1]. Though gate delays scale down with technology, global wire delays typically increase exponentially or, at best, linearly by inserting repeaters. Even after repeater insertion [3], the delay may exceed the limit of one clock cycle or even multiple clock cycles. As the system size increases, it becomes evident that the global signals require more than a single clock cycle to reach the destination from the source. Consequently synchronization of future chips with a single clock source and negligible skew will be extremely difficult, if not impossible.

One of the principal characteristics of the NoC architectures is that the functional blocks communicate with one another with the help of intelligent switches. Switches have storage buffers either at the input or output and this resource can be exploited for the purpose of efficient synchronization. The inherent idea in this research is to use these buffers to manage multiple clock domain synchronization. In addition we distribute these buffers along the channel to reduce interconnect delays, clock loading and to foster reliable communication in the presence of environmental variations.

The main objective of this work is to provide a low energy synchronization mechanism for Network on Chip (NoC) architectures to enable the network to span a SoC containing many IP Blocks or groups of blocks with completely independent clock domains. At the same time, our approach should alleviate long interconnect delays and be robust under environmental uncertainties.

1.2 Multiple Clock Domains

Multiple clocks are necessary for communication among IPs firstly because different IP cores on a single chip have different functions and may run at different frequencies. Consequently, NoC architectures should be designed to support multiple clock domain synchronization. Even though having multiple clock domains has many benefits, but the approach also presents some great challenges that must be dealt with early in the design cycle. The greatest challenge in this regard is the asynchronous transfer, as data flow rate is different in the interacting components. Capturing ~~a~~ wrong data would correspond to either sampling the irrelevant data or failure to sample at the required time. Another serious issue about crossing clock domains is the problem of metastability. This happens especially when the clock transition and the data transition occur nearly simultaneously, which results in the set-up and hold time violations of the flip-flops. This indeterminate state is statistical in nature and results in the sampled values to be undefined for a short but unknown period of time. Since there is no timing relationship between source and destination domain, there may be a case where both data and clock reach the destination flip-flop at the same time. In such a situation, the flip-flop output might go to a metastable state. Metastability poses reliability issues, hence the need for synchronization schemes come up, which can help in reducing this problem. The clock distribution in such multiple clock domain NoCs is still a great challenge both from the power and the seamless integration perspective.

1.3 The Network on Chip Paradigm & Clock Distribution

The evolution of this design paradigm resulted from the communication requirements and the constraints of designing very large and complex Systems on Chip (SoC). The design of the communication infrastructure which includes the network architecture and the interacting switches has been established in [2]. Developing test infrastructures and techniques to support the NoC design paradigm has also received some attention [7, 8]. The design tools to incorporate this new methodology are also being explored. But, the aspect of clock distribution and synchronization for NoCs has not received noticeable attention. Due to the regular structure of the NoCs, shown in Figure 1, the skew can be divided into one horizontal and one vertical component. If the vertical clock lines are placed equidistant from each other, the horizontal difference in skew between two neighboring vertical lines becomes close to constant. Furthermore, the horizontal skew between two neighboring nodes on different vertical heights also becomes almost a constant. The basic idea is to divide the chip into clock regions, where the difference in arrival time of the clock signal between any two neighboring clock regions can be controlled and/or calculated beforehand due the regular structure of the NoCs. The principal limitation of these approaches is that distribution of a single synchronous clock with differing phases all along the chip was considered. The phase difference is calculated assuming a MESH or Folded Torus-like regular NoC structure. But in reality there would be IP blocks running at different frequencies in a single SoC. Consequently the above assumption has very limited applicability. Instead of depending on the architectural regularity of NoC architectures for clock synchronization we suggest designing the NoC switch blocks in such a way that they can handle communication of signals between different clock domains.

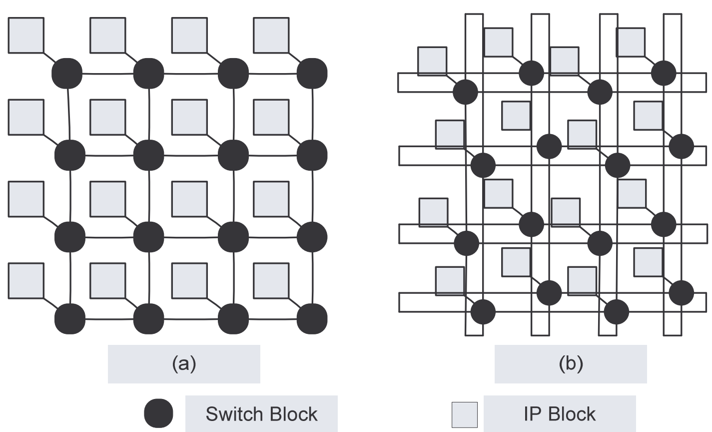

Figure 1.1: The regular structure of (a) Mesh and (b) Folded Torus

1.4 Synchronization Techniques in NoC

Since, multiple clock domains are an essential part of the SoC design, handling the communication between different modules operating at different frequencies is one of the most important design issues in NoCs. Various methods of multiple clock domain synchronization have been proposed in the context of general VLSI design. Some of the most commonly used synchronization mechanisms include (i) a double flip-flop technique, in which the output of the destination flip-flop is sent directly to another flip-flop in the destination domain [2], (ii) an asynchronous control-signal synchronization technique, in which the actual data crossing the clock domains is not synchronized, and (iii) a first-in-first-out (FIFO) based approach, in which a FIFO memory serves as a form of elastic buffer [21].

In the FIFO based design approach, synchronization latches are added at the sender and receiver modules to handle the overflow and underflow issues. The FIFO buffer is primarily to enable the two modules to queue and retrieve data from the queue at their individual operating frequencies. The scheme requires a number of handshake signals to properly execute the data transfer protocol [21]. We propose to have data travel with its associated clock in the communicating interface which implies that each time there is a request for a transaction, there must be valid data. This way, no extra clock cycles are required for the synchronization circuitry, thus saving power. At the same time, this approach aims to reduce the power dissipated in distributing the global clock signal, as the clocks would be locally generated. We have evaluated our design in some of the very regular network topologies like the Mesh and the Folded-Torus. In both of these architectures, the wire-lengths are equal throughout the network. In Mesh, a switch is connected to four of its neighboring switches and the ones at the edges to three of its immediate neighbors. The switches on the corners are connected to two of its neighbors as shown in Figure 1. Similarly, in Folded-Torus there is a wrap around connectivity between the switches at the edges. The wire lengths in the latter are a little longer than the former, but still small enough that single cycle communication is achieved between the neighboring switches. The main reason for considering these two architectures were the regularity in the length of interconnects and single cycle data transfer among neighboring switches. We extend the scheme to architectures with longer interconnects by using the concept of pipelining and repeater insertion. This objective is achieved by distributing the FIFO buffers along the way from source to destination. The scheme is illustrated in Figure 2.

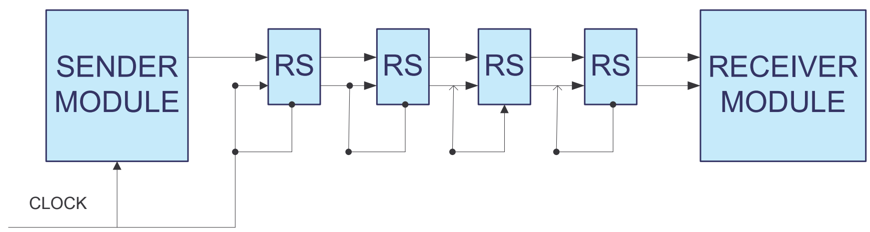

Figure 1.2: Pipelined clock and data interface circuit

This scheme allows the data to travel with its associated clock pulse and has potential for power savings (as redundant toggling is reduced). The pipelined clock also accommodates a phase shift relationship i.e. it easily allows for clocks with the same frequency but differing phases to drive the different IPs. The relay stations (RS) are FIFO buffers that would be used for pipelining both clock and data signals from the sender to the receiver end [4].

Our primary concern is whether the latency, throughput, energy dissipation and data integrity are meeting the required standards.

1.5 Contributions

The principal contribution of this work can be summarized as below:

- Implementation of a novel synchronization scheme for NoC architectures and comparing its performance with existing synchronization schemes.
- Design of the clock distribution network in a mesh-based NoC.

- Characterizing energy dissipation profile for multiple clock domain Networks on Chip.

1.6 Thesis Organization

The thesis is organized in five chapters. The first chapter introduces the complexity of the problem and some of the existing methods of addressing those issues. The second chapter presents the related work done in this regard and a detailed literature survey is provided. In the third chapter, a detailed system and circuit level design for the proposed synchronizer to communicate between two modules that are independent and running at arbitrary frequencies is provided. In this chapter design and implementation of another existing synchronizer that was used for benchmarking the performance of the proposed scheme was elaborated. A detailed comparative analysis of the performances of the two synchronization methods in NoC fabrics is also provided. The fourth chapter investigates ~~on~~ the energy dissipation profiles for regular NoCs with regard to single and multiple domain clock generation and distribution. In this fourth chapter, it is also demonstrated how shuffling the clock domains spatially results in the change in the total energy. The results for the synchronizer circuit and the energy numbers are presented here. Finally the last chapter summarizes the important contributions made and points out the direction of future research.

CHAPTER 2

2.1 Related Work

To meet the communication requirements of large SoCs, the network-on-a-chip (NoC) paradigm is emerging as a new design methodology. In this chapter we discuss some of the established work in the area of data synchronization and clock distribution in design of complex VLSI systems.

There have been significant efforts by different research groups dealing with many aspects of NoC. Pande et al [5] have discussed the design trade-offs and performance evaluation of NoC architectures. Design of switch blocks for NoCs [6] has been addressed by others. Developing test infrastructures and techniques to support the NoC design paradigm has also received some attention [7, 8]. Moreover there have been considerable efforts in developing CAD tools [9] to support this new paradigm. The aspect of clock distribution and synchronization for NoCs has not received noticeable attention. One of the several possible synchronization schemes was presented by Hataminian and Cash [10]. They noted that for very regular structures, the skew can be divided into one horizontal and one vertical component. If the vertical clock lines are placed equidistant from each other, the horizontal difference in skew between two neighboring vertical lines becomes close to constant. Furthermore, the horizontal skew between two neighboring nodes on different vertical heights also becomes almost a constant. As shown in Figure 1(a) the difference in skew between the nodes A and B and the skew between the nodes C and D, is almost the same.

Due to the regular structure of the NoCs, shown in Figure 1(b) it is proposed that the Hataminian and Cash solution can be easily extended to these. As shown in Fig 2 the clock is either generated or entered onto the chip in the upper left corner of the NoC. However, there will be one significant difference between the two structures. In the Hataminian-Cash distribution, data was only flowing in one direction. In the NoC case, data will be flowing in four directions. In [11], the authors describe a method of distributing a Quasi-synchronous clock, i.e., a synchronous clock with the same frequency but with a constant phase difference, across the entire Network-on-Chip.

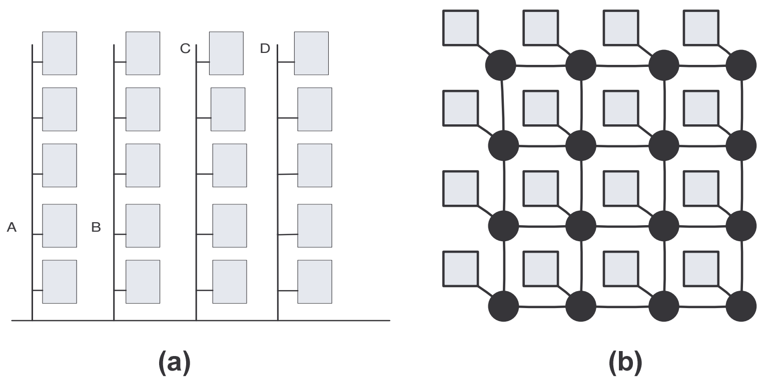

Figure 3.1: (a) Hatamanian-Cash's Clock Distribution (b) Mesh Based NoC

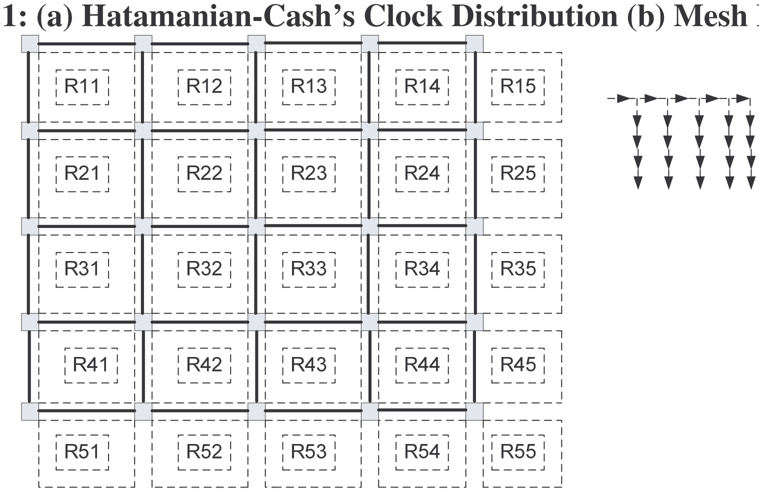

Figure 2.2: Clock distribution in NoC with clock entry/source in the upper left corner

The basic idea in all the above mentioned schemes is to divide the chip into clock regions, where the difference in arrival time of the clock signal between any two neighboring clock regions can be controlled and/or calculated beforehand due the regular structure of the NoCs. One of the possible extensions of this Quasi-Synchronous method is the Mesochronous clock distribution. The principal limitation of all these approaches is that the authors assume to distribute a single synchronous clock with differing phases all along the chip. The phase difference is calculated assuming a MESH-like regular NoC structure. But in reality there would be IP blocks running at different frequencies in a single SoC. Consequently the above assumption has very limited applicability. In addition this MESH like regular structure is only one of the many possible NoC architectures. In case of custom-built irregular NoC topology the method of estimating phase difference between different clock regions will not work. Instead of going for customized techniques for different topologies, to get around the problem, the modification of the network switches needs to be done in order to smoothly integrate the discrepancy arising out of different wire lengths and variation in the operating frequency. The only constraint being that data transfer should take one clock cycle when communication is between neighboring switches. This is easily achieved in MESH or Folded-Torus architectures as the wire lengths are small. In case of an irregular architecture like the Butterfly-Fat-Tree (BFT), the wire lengths are not equal in all the levels of the tree. This results in different delays among the communicating modules from different levels. Thus, a synchronizer is required as the data flow rate is asynchronous in nature.

A number of approaches have been proposed for synchronizing different clock domains. They address issues like clock skew, drift and jitter [12]. One such method is by using plausible and stretchable clocks [13]. These interfaces temporarily pause and stretch the receiver's clock. This approach requires the modification of the receiver design. Thus, the design is not suitable for reuse as our work requires consistency in both the sending and the receiving ends. Another alternative approach attempts to synchronize data and control signals of the receiver and the sender modules [14]. The simple two-latch synchronizers were proposed by Seitz [15], to synchronize data signals. Seizovic [16] robustly interfaces *asynchronous* with *synchronous* environments through a "synchronization FIFO." However, the latency of his design is proportional to the number of FIFO stages. The most important implementation overhead of his work is complex synchronizers which would be responsible for high energy dissipation. In [17], an introduction to several FIFO synchronizer designs is presented and their properties critically examined. Ginosar [18] provides an excellent overview of the most common limitations and failures in interfacing mixed-clock domains. A patent from Intel [19] proposes a highly-optimized mixed-clock FIFO. But, this design requires one synchronizer *per FIFO cell* (in all *(N + 1)* synchronizers if there are *N FIFO cells*). So, power is compromised for robustness.

Chakraborty and Greenstreet [20] propose a family of interface circuits, which mediate between mixed-clock domains. They start with a basic design, controlled by two identical clocks; the latency of this design is two clock cycles. This basic design is shown to handle clock jitter, while introducing no additional latency penalties in communication. The authors then propose two interesting extensions, which handle rational clock-frequency multiples and plesiochronous clocks. These exhibit an average latency of half a clock cycle, with a worst-case penalty of two clock cycles. Finally, the authors also discuss briefly extensions to arbitrary clock frequencies and FIFO interfaces.

The most comprehensive and varied work in this area is that of Chelcea and Nowick [21]. They discuss a number of low-latency mixed timing FIFO designs that interface System-on-Chip modules running at different frequencies. The work of Chelcea and Nowick has a wealth of designs to choose from, but of interest to our research is the synchronous to synchronous interface. They require detectors in order to compute the current state of the FIFO (full or empty). We have borrowed one of their interface designs as a benchmark for comparative study of our scheme in terms of latency and energy dissipation. Our scheme in comparison relies on the simplicity of the controller design. We use less number of gates (for the controller) and instead of having separate detectors for monitoring the status of the FIFO cells, we do not initiate data transfer unless there is a ready signal from both the sending and receiving ends. The most important feature of our design is probably the ability to synchronize different modules operating at independent and arbitrary frequencies. The throughput of single cycle data transfer among neighboring modules is achieved by suitably pipelining the data path.

From the system level perspective, low power and high speed are the most important driving factors. To achieve these requirements, the individual IPs need to operate at their maximum operating speeds. This is possible when fine limitations like clock skew, clock power and synchronization power are minimized. As the chip complexity increases, the share of clock power in the total power also increases. The maximum power dissipation in the clock circuitry ranges from 40-50 % of the entire chip power [22]. In [23], the clock power related issues in multiple clock domain SoCs have been discussed. But, it assumes that a single clock source is responsible for feeding the various PLLs located at the different clock domain sources. This has a limitation in the sense that the global clock network still remains large.

This work aims to addresses the power savings and performance improvements which can be achieved by efficiently designing the clock distribution network and data synchronizers for future NoCs.

2.2 Conclusion

In the next chapter, we first present the design and implementation details of an efficient synchronizer for multiple clock domains. It is subsequently followed by the comparative study of the same with an existing model in terms of power and latency. Then we present induction of the synchronizer into the NoC communication fabrics to observe its impact on the communication energy consumption. Finally, we explore how this synchronizer can be used in reducing the power dissipation of the clock network.

CHAPTER 3

3.1 The Synchronization Techniques

Multiple clock domains are becoming an essential part of the NoC design, and handling the communication between different modules operating at different frequencies is the most important design issue of modern NoCs. Among the various methods of synchronization, we adopt the FIFO based approach for communication in the NoCs. The communication switches in a NoC already have FIFO buffers and we propose to reuse this existing infrastructure. The schematic representation of the communication infrastructure in NoC is shown in Figure 3.1 where the functional modules involved in data transfer, communicate via a set of switches.

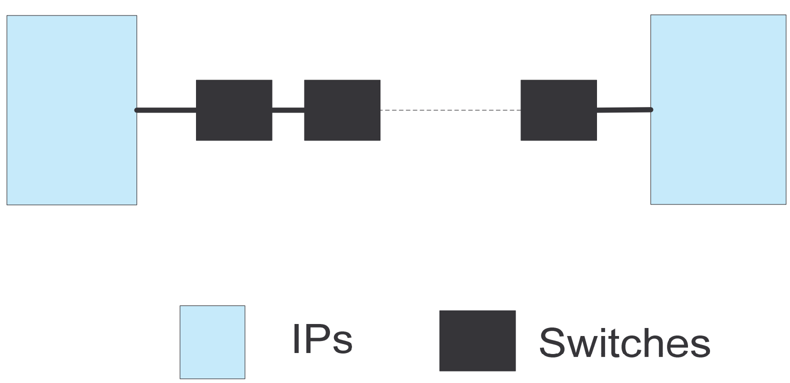

Figure 3.1: Schematic diagram of NoC communication interconnect

3.2 Circuit Level Description

3.2.1 Interconnects

The importance of interconnects in deep submicron cannot be dispensed. The device sizes are shrinking with each technology generation, and multilevel metal layer routing is dominating the landscape of integrated circuit design.

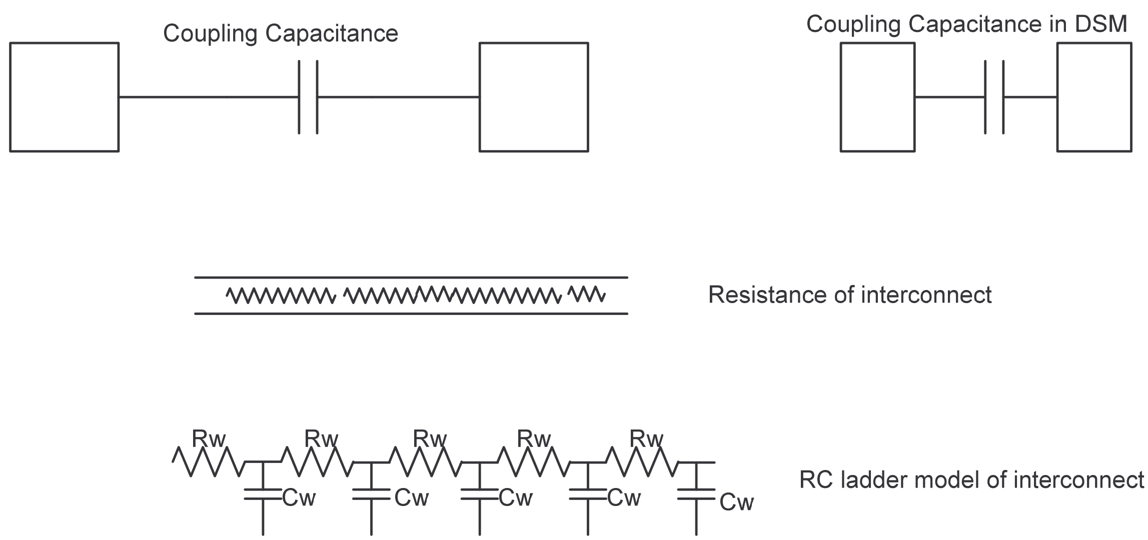

Figure 3.2: Schematic Diagram to show the importance of parasitics in Deep Submicron Designs

In Figure 3.2 we show how this effect is getting dominant in deep submicron design. An ideal interconnect would have a negligible resistive and capacitive effect, but as the width of the wires is decreased, the resistance increases. This increase in wire resistance causes an RC delay phenomenon that is increasing with technology generations. At the same time, the spacing between the wires is reducing to a point that coupling between the wires has become significant. The resulting capacitive coupling further increases the delay. Thus, an interconnect in deep submicron design can be modeled as the distributed RC ladder structure shown in Figure 3.2. Overall such effects on signal integrity are a major challenge in modern designs. No longer can these parasitic effects be neglected in modern designs. Efficient models for accurate delay calculation have become very important. In the next section, we present a standard method of modeling the wire delay.

3.2.2 RC delay in long wires and repeater insertion

For long wires, the RC propagation delay can be computed in terms of the total resistance and capacitance. If the total wire length is L and each small segment is ΔL in length, then $L = n\Delta L$, where n is the number of segments. If the wire resistance and capacitance per unit length are R_{int} and C_{int}, the total resistance and capacitance is given by $R_{wire=}\ R_{int}L$ and $C_{wire=}\ C_{int}L$. Using Elmore delay [24] calculation and assuming n segments, we obtain

$$T = (R_{int}\Delta L)(C_{int}\ \Delta L) + 2(R_{int}\Delta L)(C_{int}\ \Delta L) + \ldots + n(R_{int}\Delta L)(C_{int}\Delta L)$$

$$= (R_{int}C_{int})(\Delta L)^2\ (1 + 2 + \ldots + n)$$

$$= (R_{int}C_{int})(\Delta L)2\,(n)\,(n+1)/2$$

$$\approx (R_{int}C_{int})(\Delta L)^2\ (n)^2\ /2 \quad = (R_{int}C_{int})(L)2\,/2 = (R_{wire} * C_{wire})/2$$

The actual delay is closer to $0.38\ R_{wire}C_{wire}$. There are some other lumped RC models that are used for modeling the wire delay, namely L model, T model and the Π model [24]. Compared to the L model, the Π and T models are closer to the real scenario. It is evident from the above derivation that, the delay of a wire segment increases as the square of the length. For long global wires, the quadratic delay characteristics cannot be tolerated in the design. A standard solution to reduce the quadratic delay is to insert repeaters or buffers along the wire. This method is shown in Figure 3.3 where a wire of length L has N buffer inserted. The result is smaller segments with each segment of length L/N between consecutive buffers.

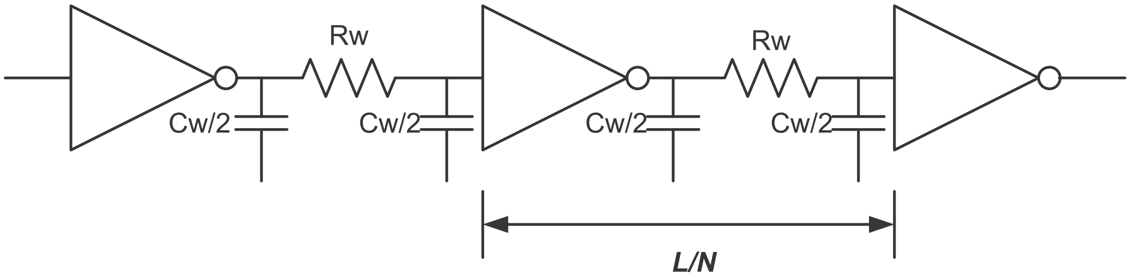

Figure 3.3: Schematic representation of Buffer insertion

The net delay through the buffers and interconnect is given by:

$$tp = N * [R_{eff}(C_{self} + C_{W/2}) + (R_{eff} + R_W)(C_{W/2} + C_{fanout})] \text{ where}$$

$$R_{eff} = R_{eqn}/M \qquad C_{self} = Cj3W * M$$

$$C_{fanout} = Cg3W * M$$

This reduces the quadratic delay to a more linear delay as each of the buffers is driving a much smaller segment. Figure 3.4 shows the change in delay from quadratic to linear as a result of buffer insertion. It clearly shows how the delay associated with smaller segments can be approximated as being linearly dependent on the length of the segment. Having too many buffers increases the delay due to the buffers whereas having lesser number results in the quadratic effect. The optimum number of buffers minimizes the total delay and this methodology is known as buffer or repeater insertion [24]. The optimal number of buffers (*N*) can be computed as

$$\partial tp / \partial N = 0 \Rightarrow N = \sqrt{0.4 R\text{int}\, C\text{int}\, L2 / tpbuf}$$

The size of the buffers (*M* times the minimum size) can be calculated as

$$\partial tp / \partial M = 0 \Rightarrow M \sqrt{(\text{Re}\,qn / Cg3W)(C\text{int}/ R\text{int})}$$

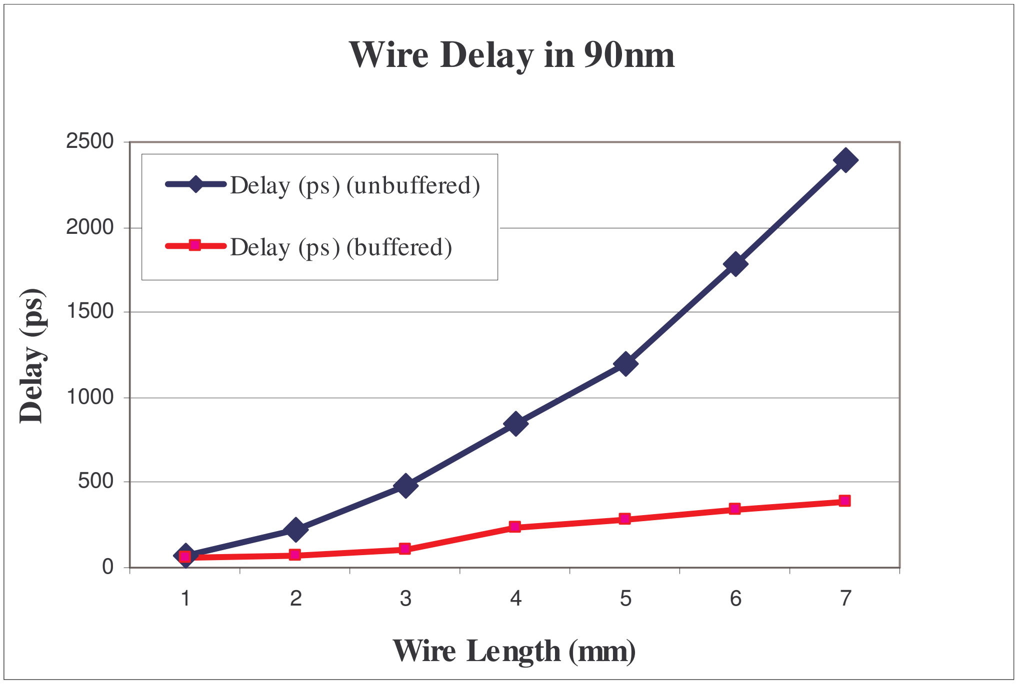

Figure 3.4: Wire delay in 90 nm process technology

Though the quadratic delay was reduced to linear but still for long interconnects, the signal propagation delay exceeds one clock cycle. If we consider the data channel as a pipe, and if the rate of filling in the data pipe in comparison to the rate of extracting data out are different, then that gives rise to loss of data along the interconnect, as there is no memory element to store data. One approach called the "Wave-Pipelining" was proposed in [22] but it too is implementation dependent on the rate of flow of data and it has been shown to work perfectly when the sender and receiver frequencies match. In order to validate this, a detailed experimental study was carried using the set-up as shown in Figure 3.5. The waveform in Figure 3.6 clearly proves that the absence of storage elements along the data channel contributed to the loss of data. Even though the sender data arrives at the receiver, but since it is not synchronized with the clock, it is not correctly captured as shown by the DataReceiver signal.

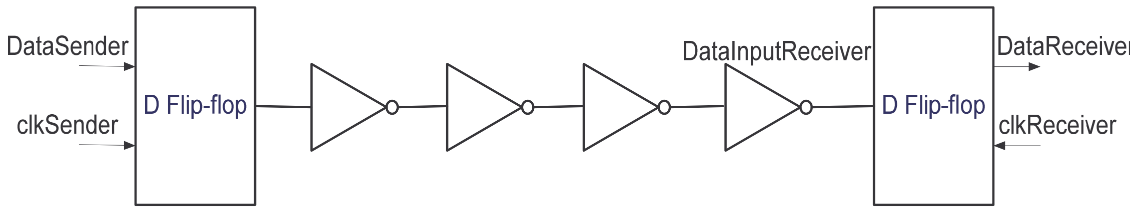

Figure 3.5: Experimental setup for Buffer insertion

The viable solution includes data-pipelining by having storage elements along the path. But, only having memory along the data-path does not help as the data flow rate is sporadic in nature and dependent on both the sender and receiver modules. If the storage elements are triggered either by the sender or the receiver clock then unless the rate of filling the memory balances the rate of clearing it, there would be either overflow or underflow issues. Another important challenge in this respect is metastability. If the transition at the clock input and that at the data input of a flip-flop violates the set-up and hold times of the flip-flop, the registered value is metastable and such a state is hazardous. In a metastable state there is ambiguity in the amount of time the valid logic signal takes to change its state. Before the stable state is reached, if the input is registered, the captured value is indeterminate in nature. Thus, a synchronizer is required to protect against the effects of metastability.

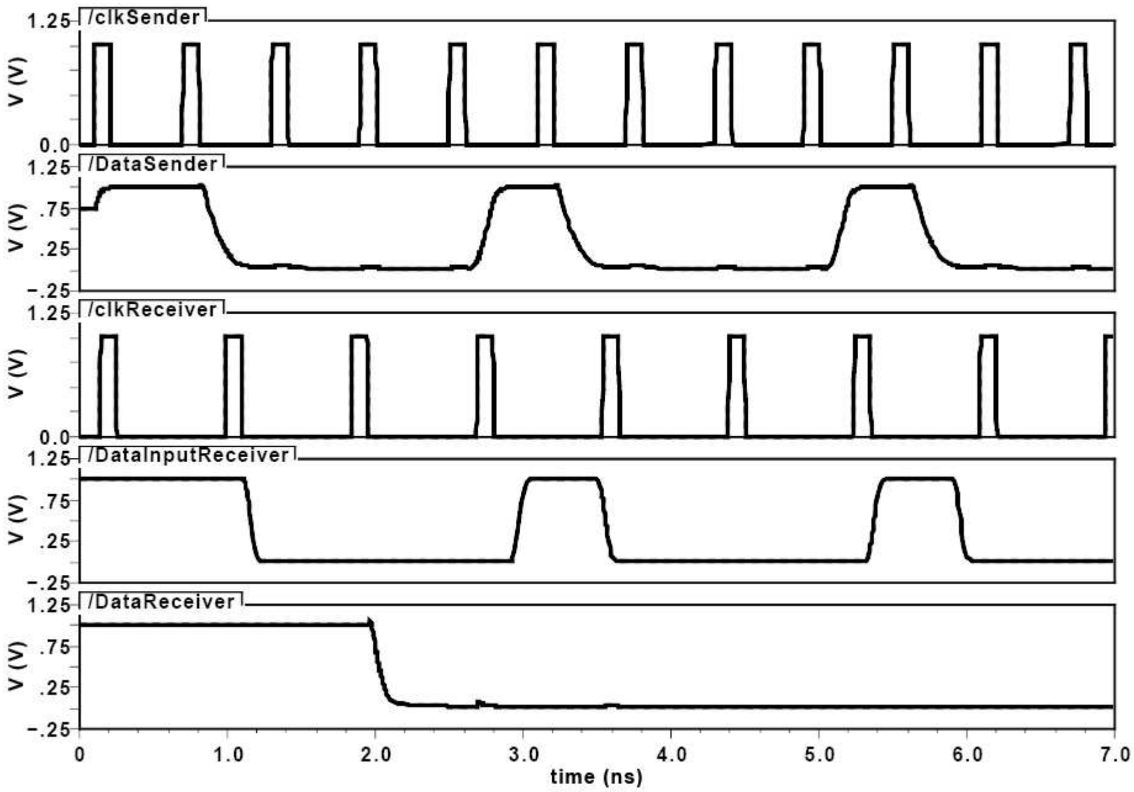

Figure 3.6: Waveform showing the limitation of Buffer Insertion

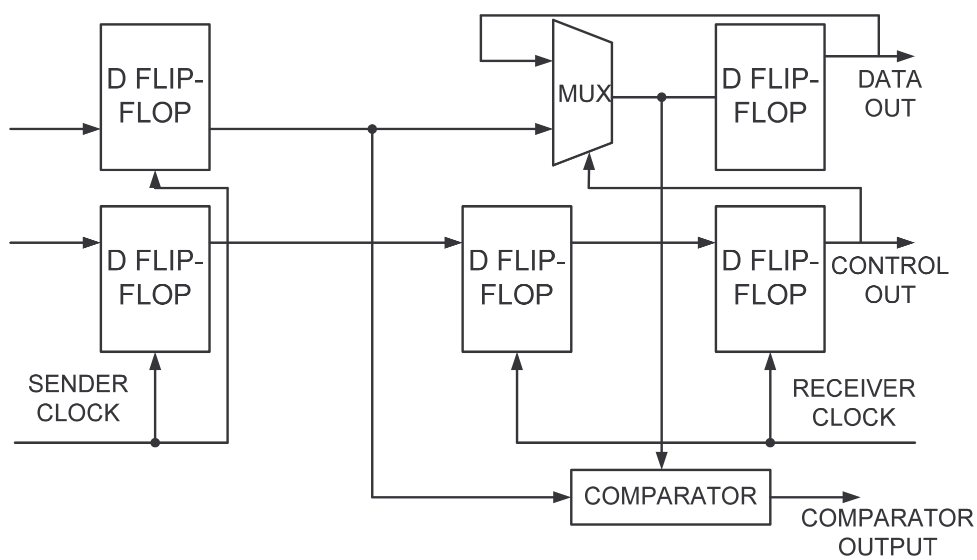

Figure 3.7: Synchronizer using double flip-flop

First the basic double flip-flop synchronizer was considered. The schematic diagram of the scheme is presented in Figure 3.7. The scheme is simple and effective in communicating between multiple clock domains, as the flip-flop in the sender is controlled by the sender clock and the one at the receiver by the receiver clock. Also, the storage is one extra flip-flop. So, if the rate of writing is greater by a factor of two or more than the rate of reading data from the storage elements, then that results in overflow. There is no feedback mechanism built in, so as to control the dynamic variation of the frequency between the sender and the receiver. An illustration in Figure 3.8 presents the shortcoming of the double flip-flop synchronization scheme. The signal from the data input channel at the receiver and the actual data in the latch of the receiver were sent to a comparator. The signal 'ComparatorOutput' shows the output of the comparator. The synchronizer is responsible for generating the control signal and that has a definite latency. Thus, if the data arrives at the receiver during this time, it is not properly registered. As shown in Figure 3.8 there is significant disparity between the sender's and receiver's data as shown by DataInput and DataOutput respectively. On the other hand, if it is delayed by a clock cycle so that it is registered properly, then the throughput suffers.

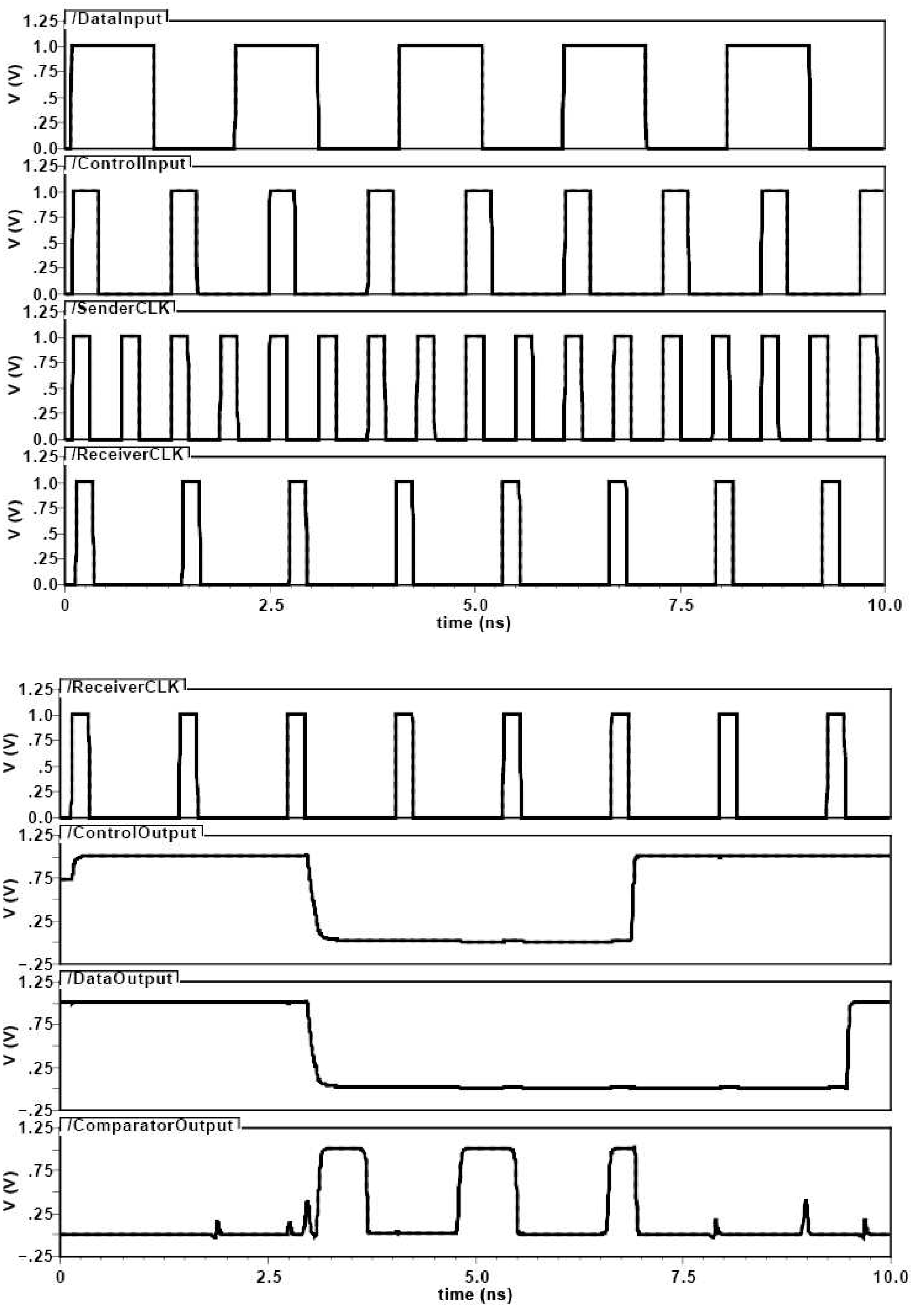

Figure 3.8: Double Flip-flop synchronizer waveforms

The above mentioned deficiencies of buffer insertion and double flip-flop methods are addressed in this work by distributing the FIFO buffers all the way from the source to the destination. Figure 3.9 shows our proposed structure of a distributed FIFO driven by pipelined clocks. This scheme allows the data to travel with its associated clock pulse. Propagating the clock at the same rate as the data results in reduced clock loading since each Flip Flop is driven by a "local" clock that matches the delay of the data path. There are several configurations that need to be addressed. The one that interests us the most is one in which the IPs run at different frequencies and we thus have to design an interface to synchronize the clocks. The communication channel can be such that the local signals to read from and write to the FIFO are generated in conjunction with both the sending and the receiving modules' request to either send or receive.

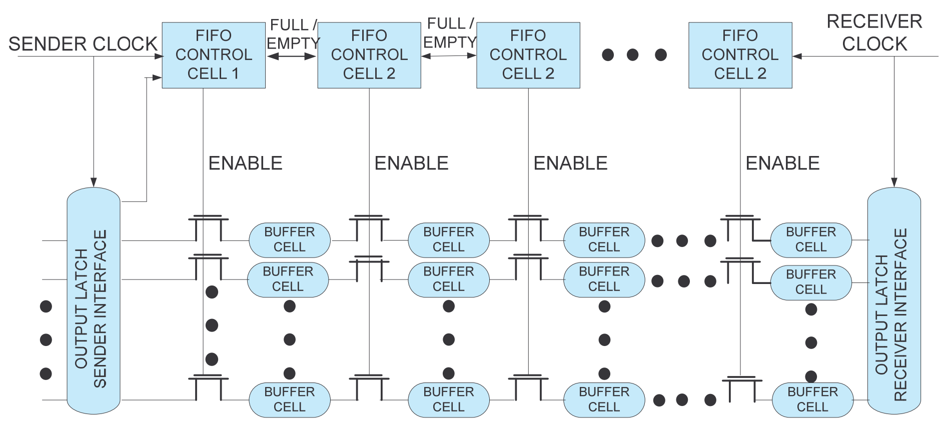

Figure 3.9: Schematic representation of the Inter-switch communication link

The circuit in Figure 3.9 illustrates the pipelining scheme, where each of the network switches would incorporate the FIFO control cell and the buffer cell. In order to test the efficiency of the above mentioned scheme, we compare the proposed scheme with an existing synchronization proposed in [21]. When considering architectures where single cycle data transfer is possible (i.e. the wire length is significantly small), intermediate control and data storage are not required. The memory at the sender and receiver interface would directly be communicating. On the contrary, when the wire length increases beyond a threshold such that single cycle communication is no longer possible, suitable number of intermediate stages would be introduced so that the data-pipelining takes place in one cycle.

3.3 Synchronizer for the NoC.

For seamless integration of the different clock domains in the NoC, we present our proposed synchronizer and compare its performance to that of an existing synchronizer [21] to perform a comparative study in terms of performance and power. They are listed below.

3.3.1 The Chelcea-Nowick (C-N) Interfaces

The work of Chelcea and Nowick [21] discusses a number of low-latency mixed timing FIFO designs that interface system on chip modules running at different frequencies. The work of Chelcea and Nowick has a wealth of designs to choose from but of interest to our research is the synchronous to synchronous interface (abbreviated as C-N henceforth). The Chelcea-Nowick synchronous interfaces require detectors in order to compute the current state of the FIFO (full or empty). The full and empty detectors shown in Figure 3.10 monitor and report the status of the FIFO cell. The delay in the synchronizer can cause overflow and underflow. Hence, the full/empty signals are included to monitor the imminent full and empty states. The output of the full detector is passed to the *put* interface, while that of the empty detector is passed to the *get* interface. The put and get controllers filter data-operation requests to the FIFO. These detectors and controllers stall the data transfer unless it is safe to do so. A full FIFO cell cannot be written to by the sending module, but can be read from by the receiving module. An empty FIFO cell cannot be read from, but can be written to by the sending module. These detectors ensure that FIFO cell accesses occur only when valid operations can be performed. This scheme also has external controllers for conditionally passing requests for data operations to cell arrays. Each FIFO has two interfaces: a *put* interface (for the sender) and a *get* interface (for the receiver). These external controllers are the *put* and *get* controller modules. All the modules, along with their associated input and output signals are shown in the block diagram of Figure 3.10.

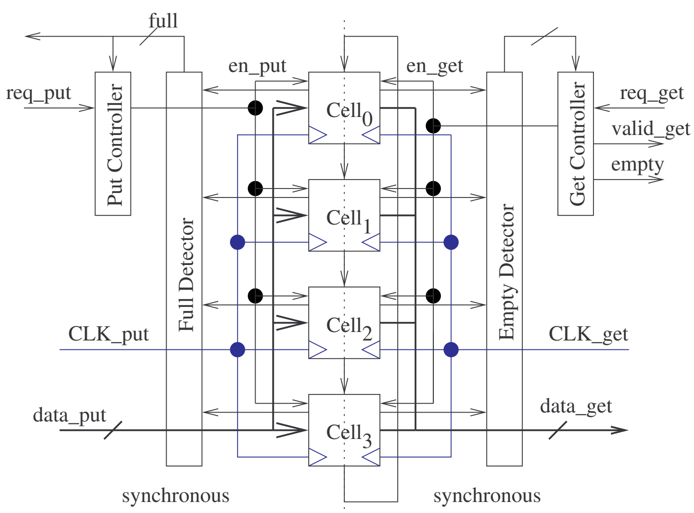

Figure 3.10: Block Diagram of the C-N synchronous-synchronous Interfaces

The synchronous *put* interface [Figure 3.10] is controlled by CLK_put. There are two inputs: one controls requests and the other serves as the bus for data items. The *full* output is only asserted when the FIFO is full, otherwise, it is de-asserted. The synchronous *get* interface is controlled by CLK_get and a control input req_get. Data is placed on output bus, and *empty* is asserted only when the FIFO is empty. The circuit level implementation of the block is shown in Figure 3.11. The *f_i* and *e_i* outputs correspond to the individual cell's full and empty signals.

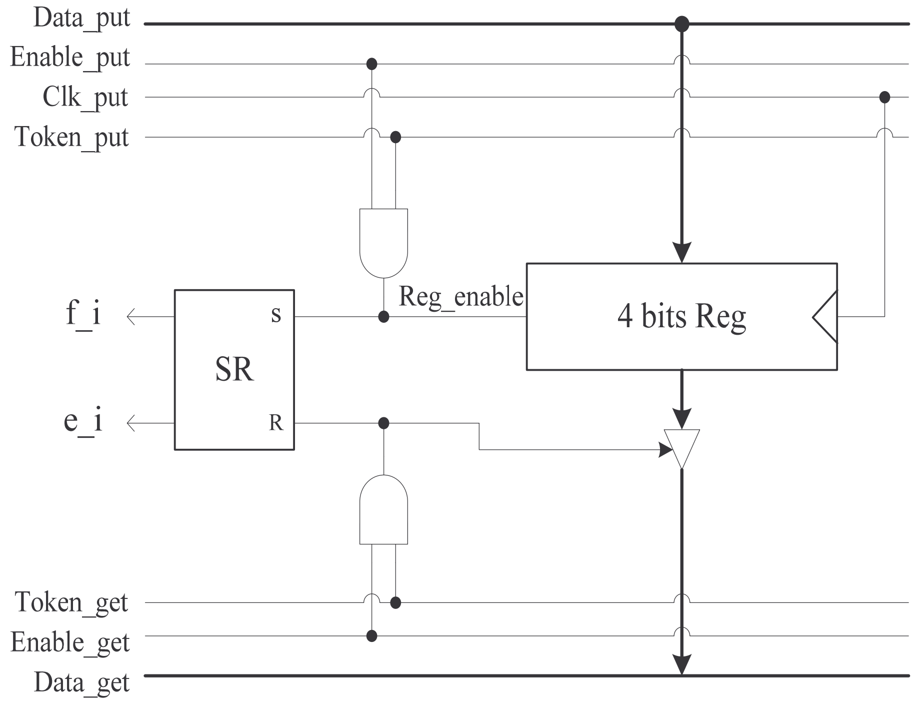

Figure 3.11: The circuit diagram for C-N synchronous-synchronous interface

The generated control signal is in turn fed to the data latches. The circuit level details for one bit register are shown in Figure 3.12. The complete/entire implementation was done for 32 bit wide data bus.

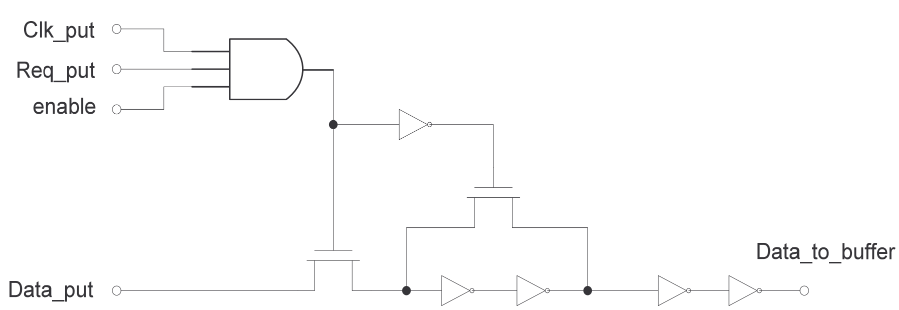

Figure 3.12: One bit Register and its associated control circuitry.

3.3.2 The Proposed Circuit

Here, we present the newly proposed interface for crossing clock domains. The interface uses self-timed control circuitry to generate local clocks and allow communicating modules operating at independent and arbitrary frequencies to exchange data [25]. The communicating modules can be close to each other or far apart but the operation principle remains the same. The control circuitry depends on both $clock_1$ (sender clock) and $clock_2$ (receiver clock) to trigger generation of the local clocks to enable the FIFO cells of the data path to shift data along the communicating channel. Figure 3.13 shows a block diagram of the scheme. The diagram in Figure 3.13 is a subset of the one shown in Figure 3.9. The Sender and the receiver interfaces in the figure can either be a switch or an IP.

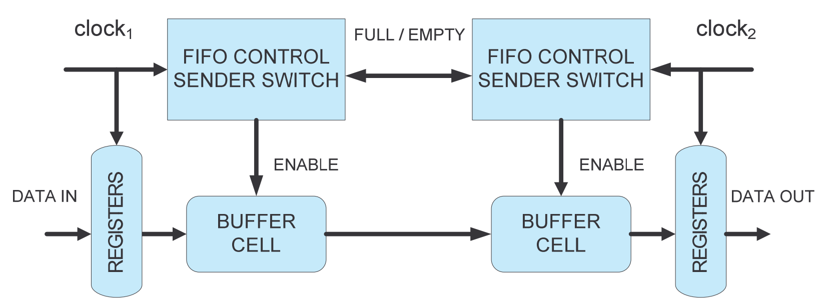

Figure 3.13: The schematic diagram of the of the Synchronization Scheme

This scheme allows either of the sending or the receiving module to initiate a request for data transfer. The empty and full signals play a major role in the synchronization of data transfer when $clock_1$ and $clock_2$ are independent clocks running at arbitrary frequencies.

Logic 0 on the *empty* signal does not permit for the generation of the enable signal allowing data stored at the buffer cell to remain there. The request, if initiated by the sender remains queued and when *empty* changes to logic 1 the enable signal gets generated. The change of the *empty* signal from logic 0 to logic 1 for this scenario resulted due to the arrival of $clock_2$. This depicts a situation in which $clock_2$ arrives after $clock_1$, implying that $clock_2$ is slower. In the event that $clock_1$ is slower than $clock_2$, the *empty* signal would be at logic 1 before the sender's request ($clock_1$) arrives. The arrival of $clock_1$ will ensure that an enable signal is generated after some delay allowing data to stabilize on the data bus. Note that the empty signal's status of logic 0 indicates a *full* status. The above description of clock activity represents cases (i) $clock_1 > clock_2$ and (ii) $clock_1 < clock_2$. The clock synchronization events are better understood by studying the circuit level diagram shown in Figure 3.14.

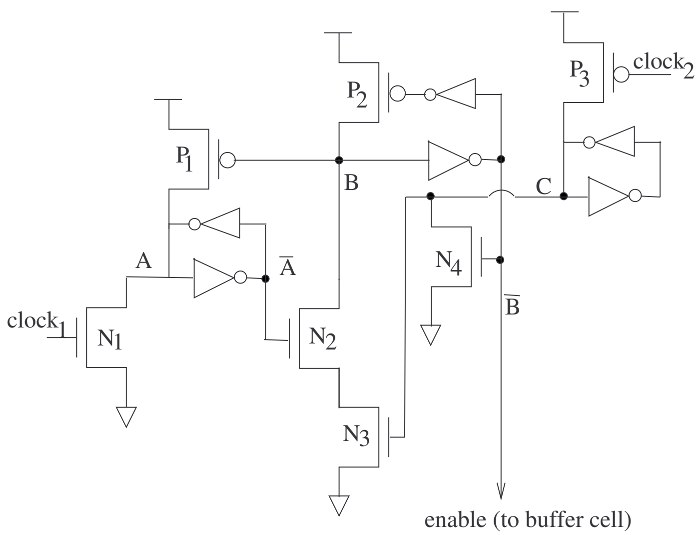

Figure 3.14: Circuit level representation of the FIFO control circuit

The full/empty signal in the block diagram of Figure 3.9 is represented on the schematic by the signal of node C. The sender's request is stored in the cross-coupled inverters with nodes A and A_bar allowing transistor N_2's gate to be at logic 1. When the full/empty signal transitions from logic 0 to logic 1 the 'enable' signal gets generated. In the event that the receiving module operates with a faster clock than that of the sending module the full/empty signal is retained at node C. When $clock_1$ arrives, the enable signal gets generated. This operation ensures that events can be triggered either at the sending module or the receiving module. It is this mechanism that permits $clock_1$ and $clock_2$ to be either of equal frequency, or of arbitrary frequencies. The enable signals constitute the local clocks and enable for data propagation from one buffer cell to the next. The principal advantage of this scheme is that it can be used to interface totally arbitrary, independent clock domains. It is also demonstrated that this method outperforms some of the existing clock synchronization schemes in terms of power and latency in a NoC [27].

This interface circuitry has been incorporated in the design of the NoC switches so that there is seamless transfer of data across the different clock regions of the SoC. When considering the global scenario, it has been assumed that the different clock domains would have their independent clock sources. This eliminates the need for the use of PLLs (Phase Locked Loops) running on a parent clock, thus reducing the length of the global clock wires and saving power.

3.4 Performance Evaluation

The previous section presented two FIFO interface mechanisms to handle communication between modules operating at different clocks with arbitrary frequencies. We considered a system with 64 embedded cores and mapped that onto the regular MESH and Folded Torus-based NoCs. The network topologies of the same are briefly discussed below.

3.4.1 MESH

A Mesh based architecture called CLICHÉ (Chip Level Integration of Communicating Heterogeneous Elements) is proposed in [29]. This architecture consists of *mxn* mesh of intelligent switches interconnecting IP's placed along with each switch. Except for the switches on the edges, every switch is directly connected to four of the neighboring switches. The ones on the edges are directly connected to three of its neighbors and the ones on the corners are directly connected to two such neighbors. This is illustrated in the Figure 3.15(a).

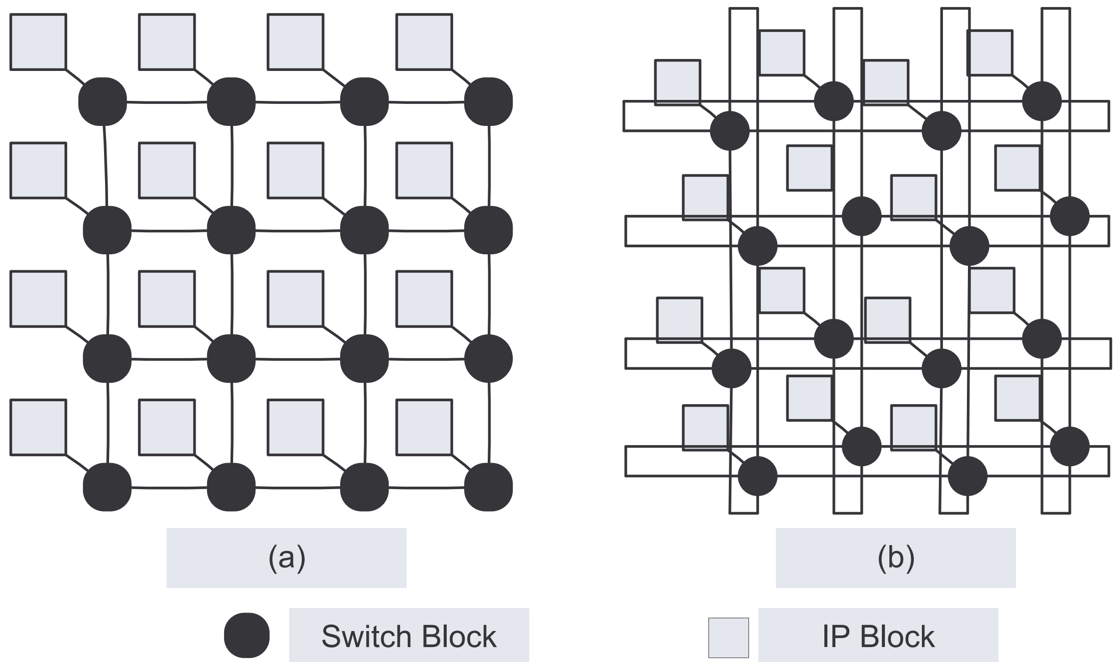

Figure 3.15: The (a) Mesh (b) Folded Torus

3.4.2 FOLDED-TORUS

A 2-D Torus was proposed in [30]. It is very similar to the mesh architecture and here the switches on the edges are connected to the switches on the opposite edge by wrap-around channels. In some cases, this reduces the communication hops across switches. However, in this case these wrap around channels tend to be very long and hence cause huge delays. As an alternative the modified Folded-Torus (FT) architecture shown in Figure 3.15(b) is suggested which folds the 2-D Torus structure so that all the wire lengths become same. Thus the long wrap-around wires are avoided in the Folded-Torus architecture.

3.4.3 Experimental Setup

It is assumed that the NoC-switch blocks operate with different clock frequencies. Consequently the multiple clock domain crossing needs to be accounted for while considering inter-switch communication. The experimental set up is depicted in Figure 3.9. The two communicating switch blocks are running with different clocks $clock_1$ and $clock_2$. The inter-switch wire lengths depend on the architecture under consideration. For the MESH-based NoC this inter-switch wire length turned out to be 3 mm and for Folded Torus it was 6 mm. Both the receiver and sender's clock signals are involved in the generation of the synchronization signals at the interface. The bi-directional control signal between the interface circuitry represents the empty/full signal. Simulations were done in 90 nm technology node and for both the C-N interface and the distributed FIFO based interfaces and different clock frequencies were used.

Table 3.1 Performance comparison of the C-N and the Proposed scheme based on FIFO interfaces

Architecture	Sender (GHz)	Receiver (GHz)	Latency (ps)		Energy Dissipation (pJ)	
			C-N Interface	FIFO Interface	C-N Interface	1. FIFO Interface
MESH	1.00	1.00	1950	332	2.80	1.42
	1.66	0.66	1940	340	4.58	1.28
	0.66	1.66	1940	300	5.56	1.60
Folded Torus	1.00	1.00	2019	480	5.31	2.08
	1.66	0.66	2009	475	6.27	1.37
	0.66	1.66	2012	468	9.72	2.33

Table 3.1 shows the latency and energy values for the C-N interface and the distributed FIFO interface. In all categories the proposed synchronization scheme based on FIFO interface out-performs the other interface. For various relationships between the sender and the receiver clocks, the latency of the former interface shows around 80% improvement over that of the C-N FIFO interface. The energy values in Table 3.1 show that the 'C-N synchronous to synchronous FIFO interface' to dissipate significantly more

energy than the proposed FIFO interface for both the MESH and Folded-Torus based NoC architectures.

Next the scheme was extended to irregular architectures like Butterfly-Fat-Tree (BFT) . In a 64 IP based BFT architecture, the first two levels of the tree have wire length of 2.5mm and 5mm respectively. Thus signal propagation can take place in a single clock cycle, when communication is restricted to these levels. But, at the third level, the wire length grows to 10mm in length and signal propagation can no longer be done in one clock cycle. The schematic diagram of a 16 IP BFT is shown in Figure 3.16.

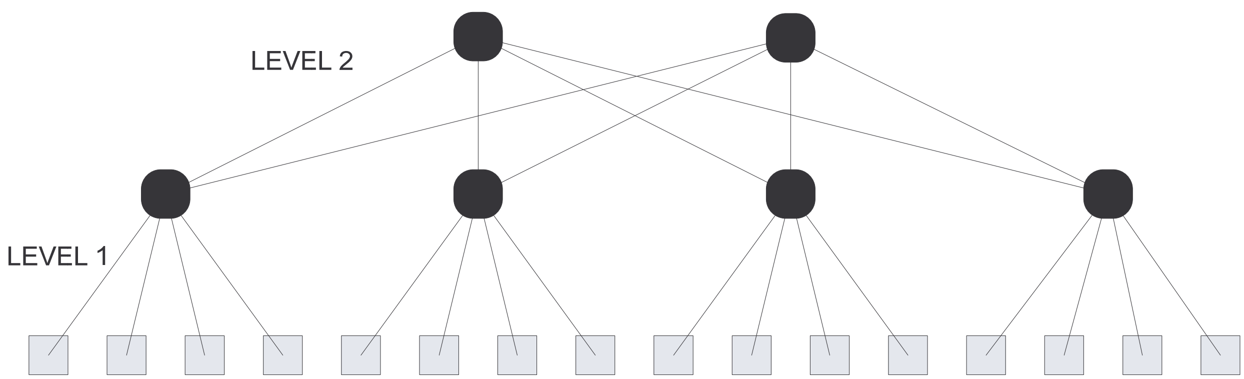

Figure 3.16: 16 IP based BFT architecture

In order to handle this case, two stages were considered. In between two switches, an intermediate stage was introduced, thus dividing the entire wire segment into two smaller segments of 5mm each. The effect of pipelining is shown in Figure 3.17. The 'Enable_1' and 'Enable_2' are the enable signals from the first and the second controller. The data ripples through the two stages along with the clock. The major limitation of this scheme is governed by the time-delay between the issue of the sender clock and generation of the enable signal. Essentially, it involves four inverter delays. Assuming the receiver is always ready to accept data, the sender clock cannot be issued unless the enable signal goes down (logic 0) again. So, neither of the interfaces can go faster than the limit set by this delay. Thus, it can be concluded that data can travel no faster than the controller interface. So, if either of the sending or the receiving modules is running faster, a suitable feedback mechanism needs to be incorporated to restrict the rate of transfer. In Table 3.2, the energy dissipation and latency associated with the two stage synchronizer are presented. The latency reported is the total latency incurred for data transfer from sender to the receiver.

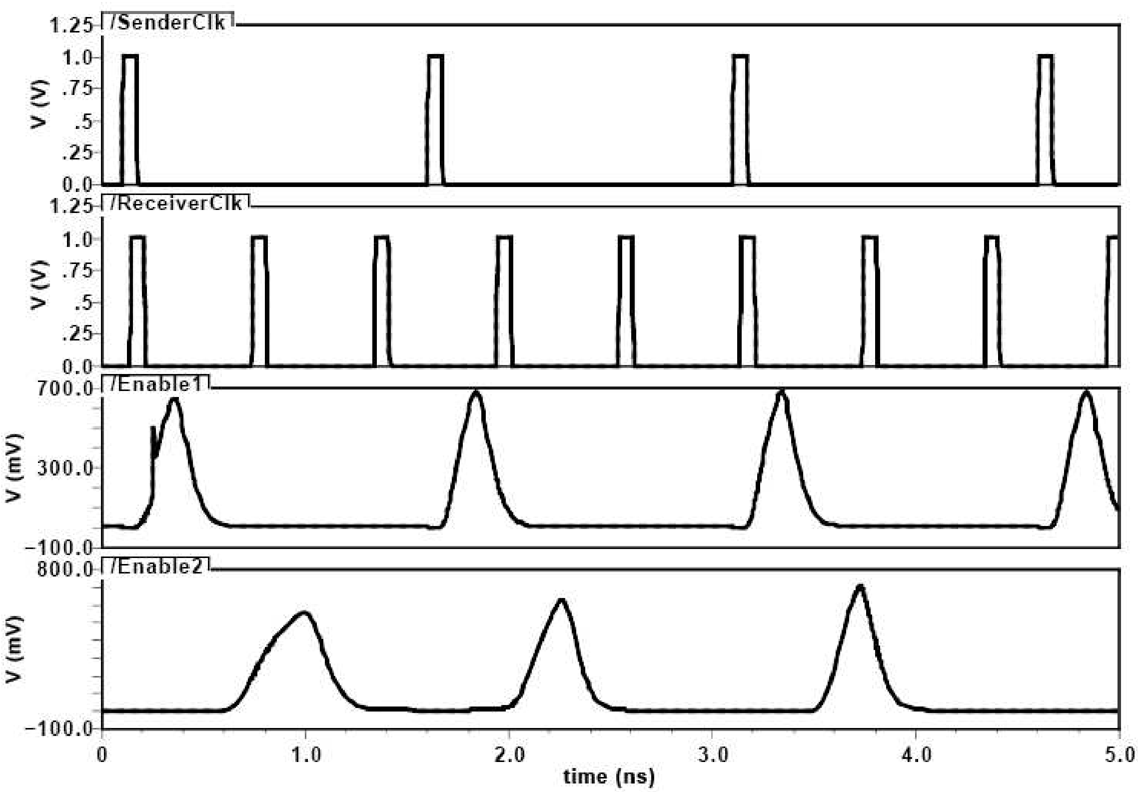

Figure 3.17: Timing diagram of the controller signals for BFT architecture

Table 3.2: Performance of the Synchronization based FIFO interfaces for BFT

Architecture	Sender Clock (GHz)	Receiver Clock (GHz)	Latency (ps)	Energy Dissipation (pJ)
BFT	1.00	1.00	894	3.36
	1.66	0.66	828	3.006
	0.66	1.66	818	3.279

3.5 Conclusion

In this chapter, design of an efficient multiple clock domain synchronizer to be used in NoC communication fabrics was presented. The proposed synchronization scheme is more efficient compared to the N-C schemes with regard to the NoC architectures considered here. This synchronizer is incorporated in the network switches to handle communication among signals crossing clock boundaries. The penalty in terms of energy of such synchronizers is quite small, and in addition it provides the opportunity to optimize energy dissipated by the global clock signals. By having such a synchronizer, we can easily interface arbitrary and independent clock domains in NoC architectures.

CHAPTER 4

4.1 Total Power Model

In this chapter our aim is to study the energy dissipation profile of NoC in the presence of multiple clock domains. The energy dissipation due to communication in a NoC is dependent on the number of clock domains, mutual interaction between them as well as their spatial distribution along the whole chip. In a NoC-based system the communication energy can be optimized by selecting an appropriate number of different clock regions and their relative placement. The synchronizer whose design was elaborated in chapter 3 will be utilized to establish communication between differing clock domains. The primary motive is in computing the average communication energy. In this chapter, we will discuss the methodology adopted for computing the communication energy, and its variation depending on multiple factors of clock distribution.

In a NoC the total communication energy will depend on energy consumed by the inter-switch links, switches, clock network and the synchronization circuitry. Wormhole routing [5] is assumed where the data transport mechanism is such that the packet is divided into fixed length flow control units or flits. The total communication energy in a NoC with a single clock domain can be modeled as

$$E_{total\_single} = n * E_{link} + m * E_{switch} + E_{clock}$$

where n is the number of inter-switch flits and m is the number of intra-switch flits. E_{link} denotes the inter-switch link energy, E_{switch} represents the switch energy, E_{clock} represents the clock energy. The clock distribution network of a 64 IP Mesh based system is shown in Figure 4.1. H-Tree clock distribution was assumed. In the multiple clock domain SoCs, the synchronization interface is required only when different IPs are involved in communicating with signals crossing clock boundaries. In the case of multiple clock domains, the total energy would be given as

$$E_{total\_multi} = n * E_{link} + m * E_{switch} + E_{clock} + E_{synch}$$

where E_{sync} is that due to the synchronization circuitry. The synchronization energy includes the overhead due to the synchronization circuit, when messages cross the clock domains. In single clock domain, as the different components are running at the same frequency, we do not require the synchronizers.

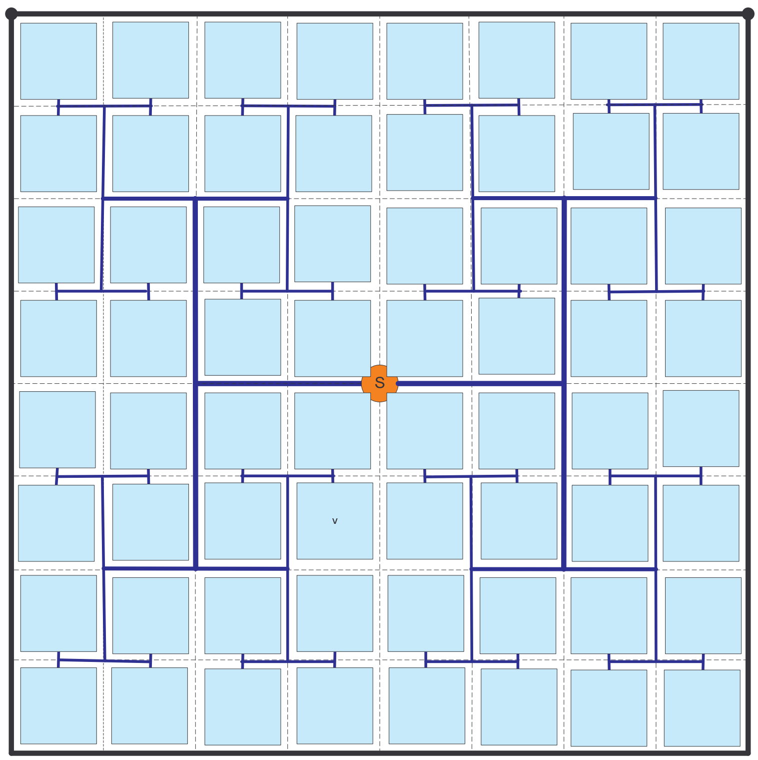

Figure 4.1: The clock distribution network for Single Clock Domain

In Figure 4.2, a multiple clock domain based system with 64 IP blocks is shown, which was considered for the simulation. We mapped this system onto a mesh based NoC. Here the clock network was routed in the intermediate metal layers as the individual clock trees are much smaller when compared with the single clock domain. Also system size with 64 IP blocks was selected to reflect the state of the art emerging SoCs. Intel has demonstrated an 80-core processor arranged in an 8x10 regular grid built on fundamental NoC concepts [19].

4.2 The Clock Network Design

The power dissipation by the clock circuitry is a major share of the total power dissipation. In order to estimate this, the clock generator circuitry and the entire clock distribution circuitry were designed using the H-Tree. For the single clock domain case, a single large H-tree was designed. The main trunk and the higher levels of the clock tree were routed in the topmost metal layers, and the lower levels of the tree were routed using the intermediate metal layers. This was done with regard to the RC delay associated with the entire wire segment. For the single clock domain, the wire length is much larger compared to that in the multiple clock domain scenario. One of the main reasons for adopting the multiple clock domains was the wire delay (resulting in high skew) and the high power dissipation associated with single domain clock network. For systems having multiple clock domains, the entire area was partitioned into smaller regions and the clock tree for each such region was designed as shown in Figure 4.2. Since, different clock domains operate at different frequencies the clock generator oscillators were customized to generate frequencies in the range 0.5-2.0 GHz. The notion of selecting this range of frequencies was done in accordance to the clock frequency associated with the 90nm technology node. (90nm process technology was used throughout the design).

A die size of 20mmX20mm was assumed and accordingly the lengths of the branches of the H-Tree were calculated. Buffer insertion was performed suitably as discussed in Chapter 3 [24].

4.3 Experimental Setup

A set of experiments were carried out for the single clock domain and then the number of clock domains was varied. For the single clock domain, the range of frequencies was varied from 0.66 GHz to 1.67 GHz. This range of frequencies was selected keeping the clock frequency at 90 nm process technology in mind. The clock frequency in any technology node can be denoted in terms of fan-out-of 4 (FO4) delay. The FO4 delay is defined by the delay incurred when a single inverter drives four of its kind sized identically, as shown in Figure 4.2. According to International Technology Roadmap for Semiconductors (ITRS) [25], the clock frequency for a particular technology node can be considered to be equal to 15 FO4. Following this the clock frequency at the 90 nm technology node turns out to be 1.67 GHz.

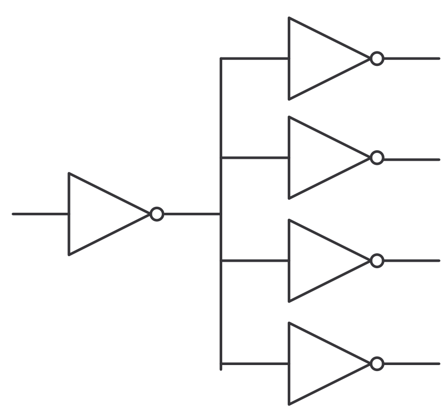

Figure 4.2: Fanout of four

The average energy dissipation with different clock frequencies was noted and is shown in Figure 4.4. The network parameter 'injection load' is measured as the number of flits injected by each IP in unit time. The injection load of the whole network was kept fixed at 0.4. When the whole system was run at the highest frequency, i.e. at 1.67 GHz energy dissipation was maximum but it also corresponds to the highest performance in speed. This frequency is less than the maximum frequency of 2.0 GHz for multiple clock domain case because beyond 1.667 GHz, the skew gets significantly large and is comparable to the clock period.

Figure 4.3: The H-Tree clock network for 8 clock domains.

On the contrary, when the operation frequency was reduced to 0.66 GHz, the energy dissipation reduced significantly and so did the performance. Driving the chip slowly cannot be tolerated and at the same time, the demand is in saving power. Both the objectives are accomplished if the individual switches are clustered into different clock domains depending on the operating speeds of their corresponding IPs. This situation requires synchronizing when communicating across the clock domains along the network of switches.

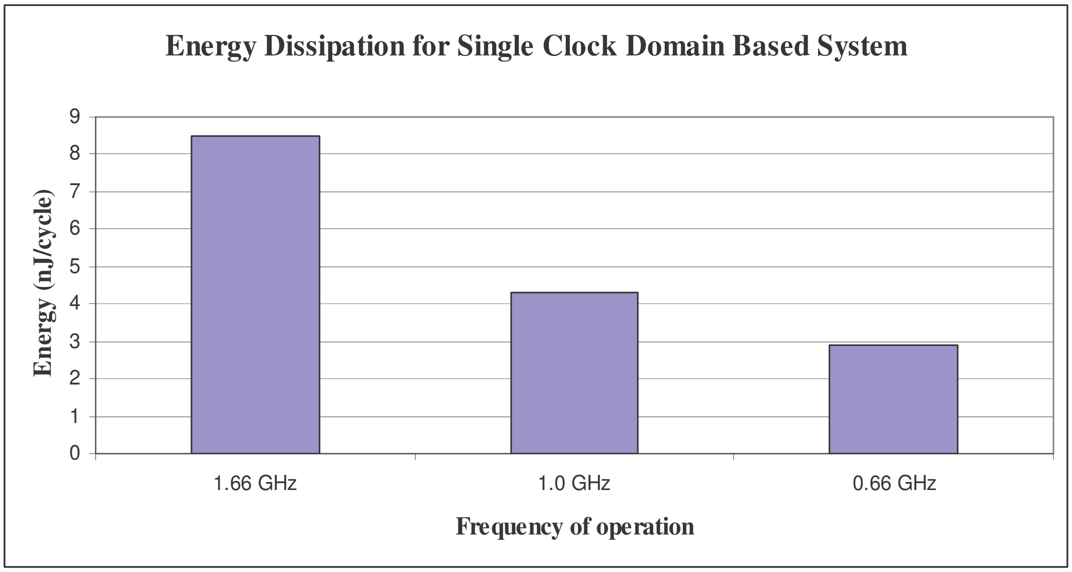

Figure 4.4: Energy dissipation in Single Clock Domain Based NoC

Next, we discuss the multiple clock domain scenarios. The number of clock domains was varied from 4 to 16. At the same time the frequency of operation of the individual domains was randomly selected in the range of 0.66 GHz to 1.6 GHz. In order to note the trend in variation of energy dissipation and optimal number of clock domains for minimum power, the frequency of operation of each domain was arbitrarily assigned. It is observed that with 4 clock domains, the energy dissipation is highest among all the possibilities considered (for multiple clock domain case), which reduces gradually as the number of clock domains is increased from 4 to 10. When the number of different clock domains is increased to 16, an increase in the energy dissipation is noticed. This is shown in Figure 4.5, for a particular injection load. These characteristics are consistent with the fact that as the number of clock domains increases from 4 to 10, the individual clock networks and the corresponding buffers are becoming smaller. Even though the synchronization energy increases, substantial savings arise from smaller clock networks. If the number of clock domains increases beyond a certain limit, then the synchronization energy starts to dominate. This is evident in the case of 16 clock domains. In that case even though the clock energy is reduced, the total communication energy starts to increase. This can be attributed to the rise in the synchronization energy as with the increase in the number of clock domains, the amount of communication across clock domains greatly increases.

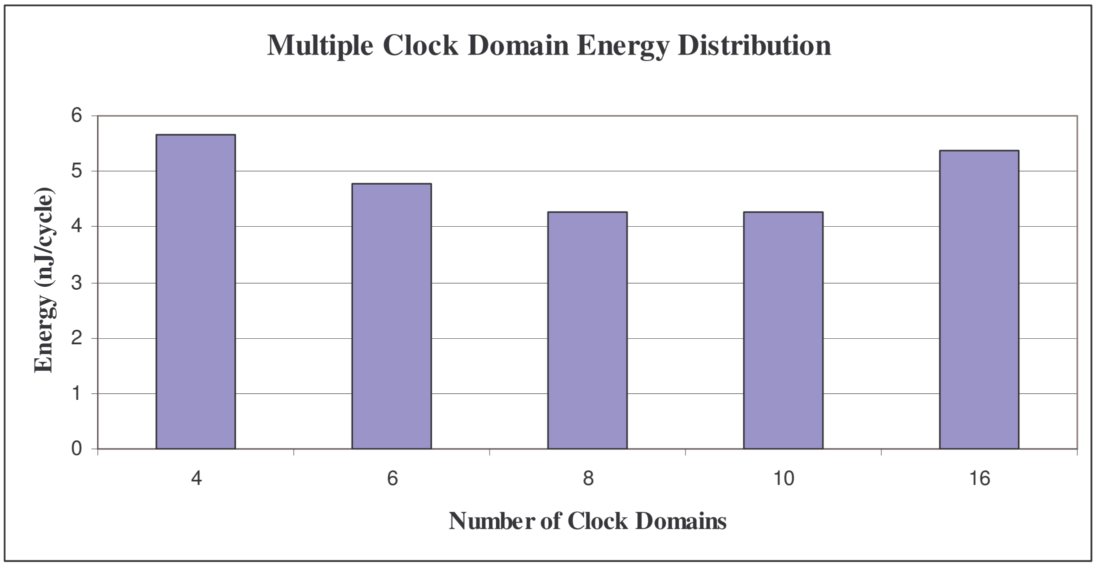

Figure 4.5: Energy dissipation in Multiple Clock Domain Based NoC

It was also observed that the skew associated with the single clock domain was much larger compared to that of the multiple clock domain scenario. This phenomenon is easily explained by the fact that the wire lengths associated with the single clock domain is far longer than that of the multiple clock domains. Among the different multiple clock domain cases, the skew is proportional to the size of the clock network for that respective domain. The maximum skew values associated with the different partitioning schemes are listed in Table 4.1.

Table 4.1: Maximum Skew associated with the different clock regions

Number of Clock Domains	Maximum Skew (ps)
1	438
4	339
16	134

4.4 Performance Evaluation with varying Injection Loads

In NoC energy dissipation varies with injection load as shown in Figure 4.6(a). It shows an initially increase followed by a saturating characteristic when the injection load reaches the throughput level. Beyond saturation, no additional messages can be injected into the system and hence no additional energy is dissipated.

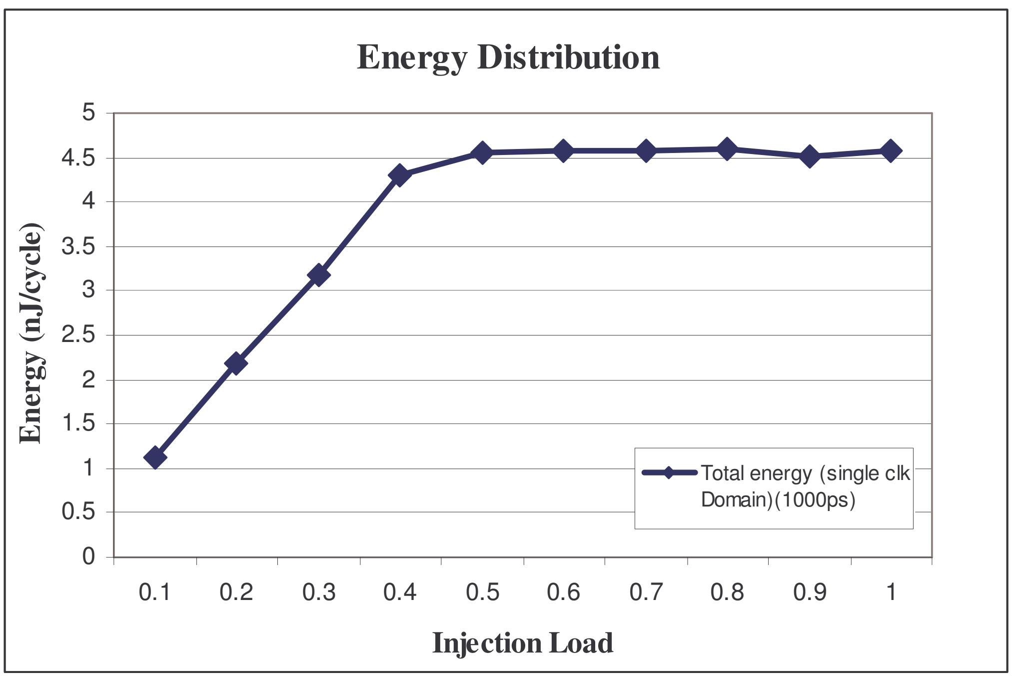

Figure 4.6(a): The Energy Dissipation for uniform traffic in Single Clock Domain case

For complete study of the energy dissipation profile, the injection load of the network was varied and energy consumption for both the single and multiple clock scenarios was noted. The trend saturates from injection load of 0.5 onwards and the energy dissipation trends are consistent throughout. Figure 4.6(b) shows the energy dissipation profile for all the different schemes with varying injection load in the network. We observe that among all the cases, the maximum energy is consumed by the single clock domain running at its maximum frequency and the minimum energy is dissipated when the single clock domain runs at its slowest speed. In neither of the cases, we are able to extract the best performance out of the whole system both in terms of power and speed. On the contrary, multiple clock domains allow each of the individual modules to run at the optimal frequencies, thus saving power and not degrading the performance.

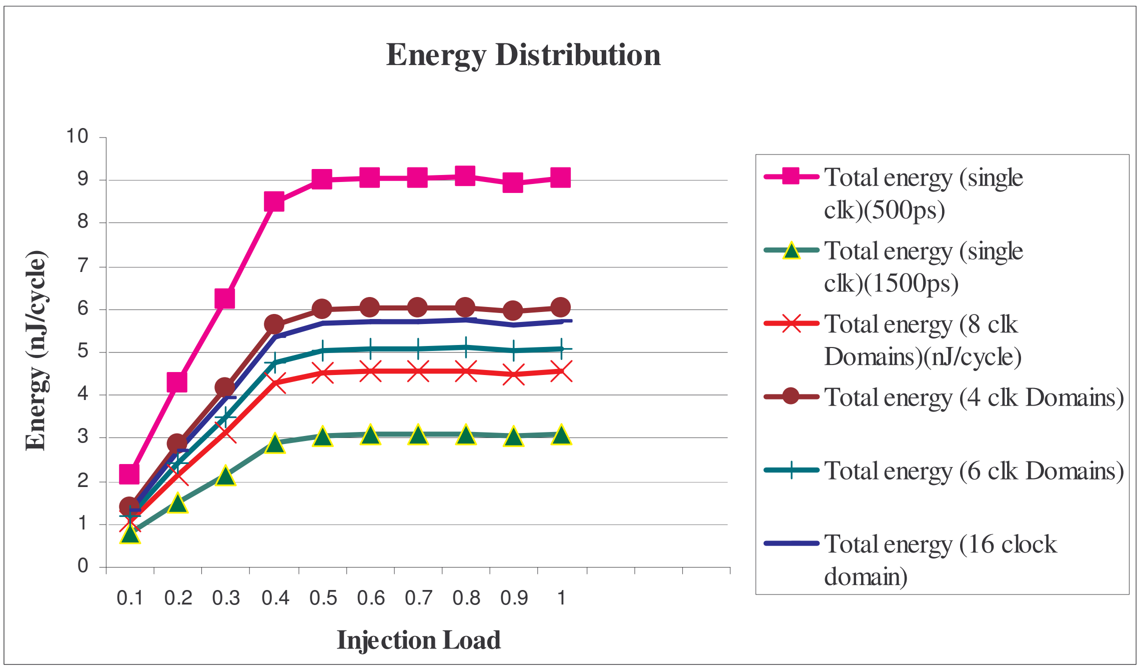

Figure 4.6(b): The energy distribution for the different clock domain cases

When calculating the energy for the multiple clock domain case, the synchronization energy was added whenever there was data transfer across the clock domains. Among the various other factors that can influence the energy dissipation, the spatial distribution of the domains plays a significant role. In the next section, spatial distribution is addressed in detail.

4.5 Performance Evaluation with varying Spatial Distribution of Clock Domains

Now, we explore the variation of the total energy dissipation when the spatial distribution of the clock domains is changed. An example to illustrate the situation is the case of 8 clock domains as depicted in Figure 4.7. Figure 4.7(a) shows an arrangement of the NoC switch blocks based on area. The area of each domain was chosen arbitrarily keeping the symmetry of the H-Tree and the unequal distribution of the number of IPs operating with the same frequency. This depicts a more realistic case as practical NoCs would not be very symmetrical in terms of the clock domain and its frequency of operation. Figure 4.7(b) shows another different placement of the same 8 domains in a bid to demonstrate that there is interdependence between clock partitioning, the position of different clock domains on the die and the area. This interdependence has a direct effect on energy dissipation. The number of times a data packet crosses the clock domains, the synchronization energy gets added to it. Thus the minimum energy case corresponds to minimum number of clock domain crossings and also it depends on the operating frequencies of the two domains. The data routing algorithm is also responsible for lower energy. If there are two options, for routing data packets from one switch to another, the one corresponding to lower synchronization energy should be considered. We had considered a fair and generic case where we assumed that there is equal probability of every switch to communicate with another. In reality, the clock distribution entirely depends on how different applications are mapped onto different parts of the NoC and the probabilities of individual domains communicating. By shuffling the clock domains spatially, different configurations can be created as shown in Figure 4.7. Figures 4.8 (a) and (b) show two such configurations for 4 clock domains.

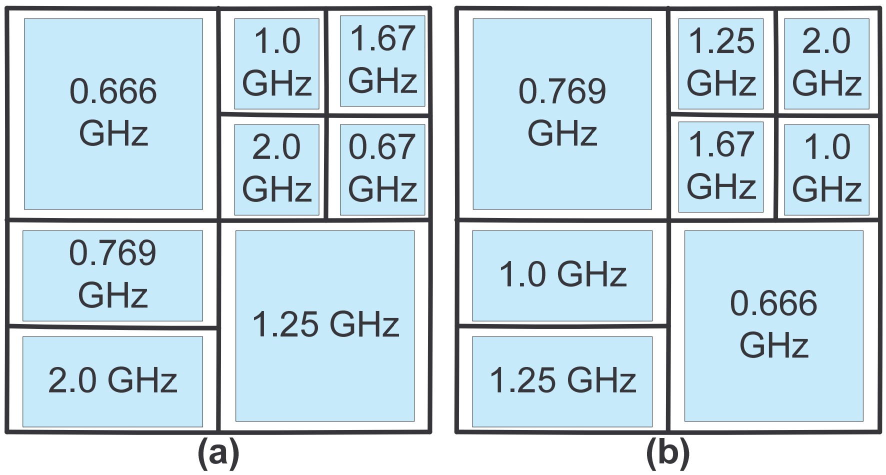

Figure 4.7: Illustration of two different configurations of the 8 clock domain case.

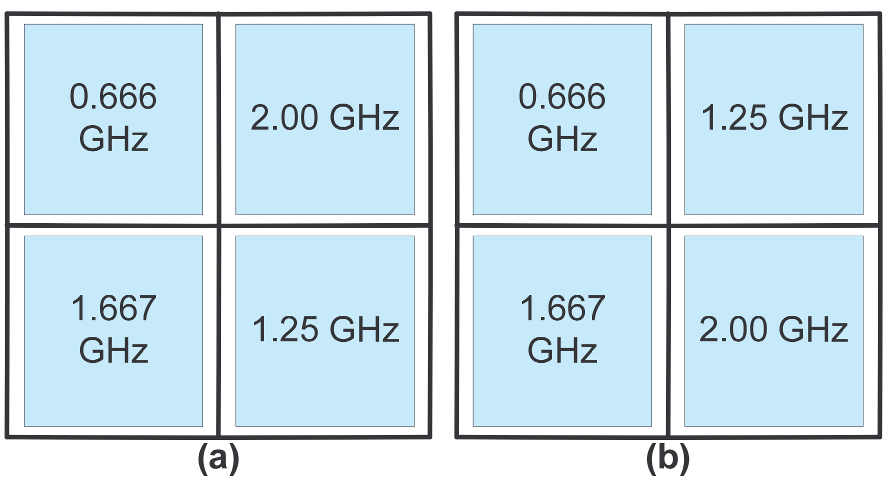

Figure 4.8: Illustration of two different configurations of the 4 clock domain case

Figures 4.9 and 4.10 show the energy dissipation for the context of 8 and 4 clock domains respectively when the configurations were varied. The important point to note here is that when the clock domains with different areas are shuffled that leads to noticeable changes in the energy dissipation. The number of switches, inter-switch links and also the length of the clock network depends on the area of a particular clock domain. There are many possible ways to rearrange the 8 clock domains and 5 such possible configurations are presented in this study shown in Figure 4.9. It shows an energy variation of 14% as the blocks are shuffled. This shows that when the different clock domains are not uniformly distributed, shuffling them results in energy dissipation changes. Contrary to this, in the 4-clk domain case, it is evident that all the clock domains occupy equal physical area of the chip and hence there is no significant change in the energy dissipation due to shuffling of the domains. This meager change is attributed to the small share of the synchronization energy compared to the switch, link and clock energies. Thus, it can be concluded that energy dissipation also depends on the placement of the individual domains within the NoC.

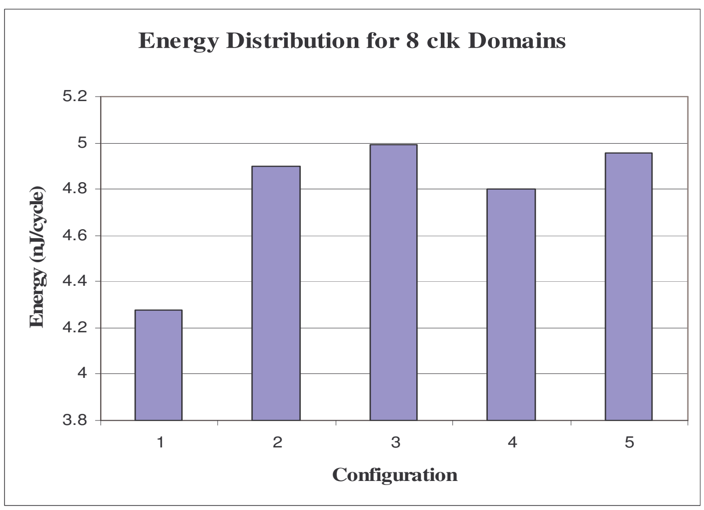

Figure 4.9: Energy distribution for five different configurations of 8 clock domains

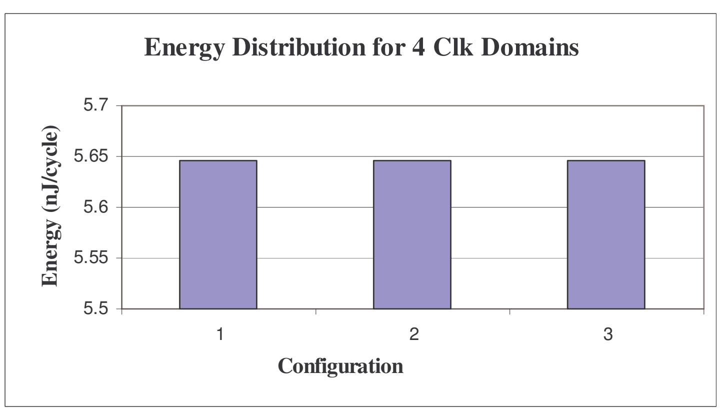

Figure 4.10: Energy distribution for five different configurations of 4 clock domains

It was noted from the experiments that the energy depends on a number of factors, like, spatial distribution of the domains, their physical area and mutual interactions. It is shown that if the physical areas covered by different clock regions on a NoC are not uniform then shuffling their placement gives rise to change in energy dissipation. Thus, not only the number of clock domains, but the physical placement of the respective modules is greatly responsible for the energy dissipation.

4.6 Conclusion

The communication energy of the entire system is significantly dependent on the clock network supporting the underlying NoC. Efficient design of the clock net in turn provides the necessary savings in power. Instead of distributing a single clock across the entire communication infrastructure, if we divide the communication fabric into multiple clock domains and have locally generated clocks drive them; we achieve significant savings in the total energy dissipation. Also, having smaller clock network means that we can use the intermediate metal layers for routing the clock trees and hence perform faster communication. Another important observation from the placement perspective is that proper spatial arrangement of the clock domains across the chip can reduce the power significantly.

CHAPTER 5

5.1 CONCLUSION

High level of integration and increasing clock speeds are the driving forces in the modern VLSI industry. In order to achieve this vision, the on-chip communication plays a great role. The complexity of the communication is increasing at a great rate. With increasing die sizes and shrinking device dimensions, the global interconnects are no longer able to transfer data in a single clock cycle. Thus, the concept of NoC is getting a great boost. The advancement of this novel technique depends on how easily this novel methodology can accommodate the different functional modules into its infrastructure. As the different modules operate with different frequencies, synchronization and efficient communication is the key challenge.

The major contribution of this work is the implementation of a novel synchronization scheme and doing a comparative study of the same with an existing synchronization scheme tailored for NoC. We have shown that communicating NoC switch blocks running at the same or different arbitrary frequencies can be managed by the proposed distributed FIFO scheme. The proposed distributed FIFO interface circuitry is simple yet effective, reducing energy dissipation significantly. Overall it has been shown that instead of depending on the architectural regularity of NoC architectures for clock synchronization, the NoC switch blocks should be designed in such a way that they can handle communication among modules operating in different clock domains.

From the systems level perspective, the energy dissipation profile of a multiple clock domain NoC was investigated. This energy depends on a number of factors, like, spatial distribution of the domains, their physical area and mutual interactions. Today's massive multi-core chips generally consist of multiple clock domains. Consequently it is imperative to quantify the effects of communicating signals crossing clock domains on the performance of NoC fabrics. The communication among differing clock domains can be achieved in a NoC by modifying the FIFO buffers in the switch blocks. But the spatial distribution, total number of clock domains and the physical area of these domains have significant impact on the energy dissipation of the NoC. We have demonstrated how the number of clock domains and their placement impact the energy dissipation of a mesh-based NoC. It is shown that if the physical areas covered by different clock regions on a NoC are not uniform then shuffling their placement gives rise to change in energy dissipation.

5.2 Future Work

The research in the direction of NoCs can be extended not just to reducing the clock power and interconnect delay in NoCs but to any VLSI system. Successful work in this endeavor will lead to further advancement of technology in the area of low power design. Some of the directions for future research include:

5.2.1 Locally Generated Clock and Hybrid Clock Networks

Generating local clocks and showing that this is a viable solution might lead to the re-introduction of clock system design. From the clock distribution perspective, various other distribution schemes like the clock grid need to be evaluated. For large heterogeneous systems with multiple clock domains, having smaller clock grids at the leaf nodes of a large H-Trees is a potential area that needs to be explored. A few illustrations of the same are shown in Figure 5.1. Also, new placement guidelines for handling such hybrid clock networks have not received due attention.

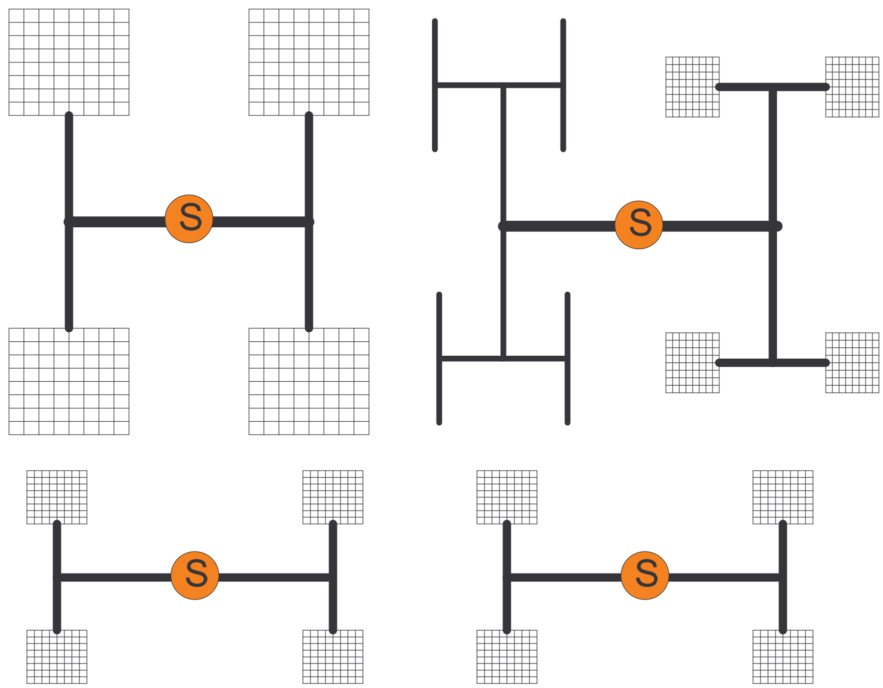

Figure 5.1: Multiple Clock Domain Hybrid distribution scheme

5.2.2 Comparison with other Synchronization Schemes

The comparison of the proposed synchronizer has been done only with a single existing prototype [21]. For a complete study, the other synchronization schemes like that proposed in [20] and [19] needs to be evaluated and tested.

The scheme presented in [20] has two versions. The former uses a single stage FIFO consisting of three latches and a latch controller that generates the latching signal based on the clock inputs from the sender and the receiver ends as shown in Figure 5.2. The generation of the latching signal is dependent on the overlap period of the two clock signals. This scheme does not guarantee robustness in synchronization for all cases. The later version describes a method for handling arbitrary clock frequencies using rate multipliers. The design involves adjusting the clock jitter and skew by adjusting the speed of the self-resetting C element [26]. This might not enhance robustness with regard to introducing it in generic NoC switches.

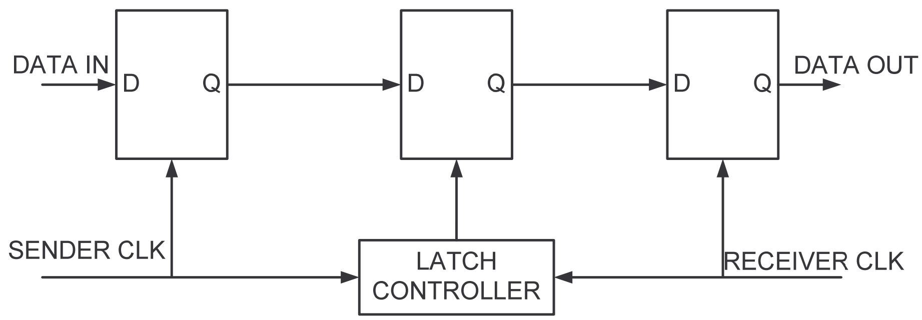

Figure 5.2: Schematic Diagram of single stage FIFO

The work in [19] uses a synchronizer with '*n*' latches. Though the number of latches being used at a time is programmable and hence the system design is quite robust, but structural complexity of the design contributes to greater power dissipation. The effect of having such a synchronizer in the network switches is best established by simulating it for varying injection loads.

5.2.3 Three Dimensional Network on Chips

Three dimensional NoCs have recently attracted the attention of the research community. In order to be able to exploit the maximum performance out of the third dimension, the design of the switches and high throughput synchronizers requires attention. The speed of communication in the third (vertical) dimension is not the same as in the horizontal plane and is usually a lot faster. The best performance under such conditions is obtained by stacking the different clock domains in the vertical planes. This layout requires lesser number of synchronizers, thus saving on the synchronizer power. One of the possible future directions would be to evaluate the performance of the of the distributed FIFO synchronizer investigated in this thesis for 3D NoC architectures.

The key achievements of this work are summarized below:

5.2.1 Clock Synchronization: We provide a method for synchronizing communicating components with arbitrary clocks, connected through a Network on chip. The cross-cutting approach is likely to lead to further increase in the number of IPs on a die. Network on Chip is emerging as a revolutionary methodology for integrating very high number of IP cores in a single chip. Having a low power clock distribution scheme supporting NoC design paradigm, will enhance wide adoption of this as a mainstream IC design methodology. Existing synchronization schemes either have a high number of handshake signals or require flip-flops to modify the frequency of the faster clock. The proposed scheme is likely to outperform other approaches since it presents signals that can be used in conjunction with the synchronous clocks to ensure proper data transfers and also has fewer handshaking signals.

5.2.2 Managing Clock Distribution Power: Pipelining the clock could potentially provide a good means of reducing the clock distribution network's power and combining this with distributing the FIFO to reduce wire delays could significantly reduce power associated with communication channels of the SoC. Today's systems are viewed as being communication bound instead of being computation bound and the proposed work could alleviate some of these problems, particularly energy dissipation.

5.3 Summary

The Network-on-Chip (NoC) design paradigm is viewed as an enabling solution for integrating exceedingly high number of computational and storage blocks in a single chip. In DSM VLSI design, it is a very challenging job to guarantee maximum achievable performance and yet be low power. By incorporating synchronizers, the discrepancy arising out of different operational speeds of different modules is resolved but it contributes to additional power consumption.

The major share of power consumption in today's complex systems is the clock, and the future of low power systems depend on how efficiently clock is generated and distributed across such complex systems.

REFERENCES

[1] R. Ho, K. W. Mai, M.A. Horowitz, "The Future of Wires", Proceedings of the IEEE*,* Vol. 89 Issue: 4, April 2001 pp. 490–504.

[2] L. Benini and G. De Micheli, "Networks on Chips: A New SoC Paradigm," *IEEE Computer,* Jan. 2002, pp. 70-78.

[3] L. P. Carloni and A. L. Sangiovani-Vincentelli, "Coping with Latency in SOC Design," *IEEE Micro*, Oct. '02, pp. 24-35.

[4] R. R. Rydberg III, J. Nyathi, J. G. Delgado-Frias, "A distributed FIFO scheme for on chip communication" Proceedings of IEEE International Conference on Circuits and Systems (ISCAS), 23-26 May, 2005, pp. 1851-1854.

[5] P. P. Pande, C. Grecu, M. Jones, A. Ivanov, R. Saleh, "Performance Evaluation and Design Trade-offs for Network on Chip Interconnect Architectures", IEEE Transactions on Computers, Vol. 54, No. 8, August 2005, pp. 1025-1040.

[6] S. Kumar, A. Jantsch, J. P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, A. Hemani, " A Network on Chip Architecture and Design Methodology," *Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI),* Pittsburgh, USA, 2002, pp. 117-124.

[7] E. Cota et al., "Power-Aware NoC Reuse on the Testing of Core-Based Systems," Proceedings of Int'l Test Conf. (ITC 03), vol. 1, IEEE CS Press, 2003, pp. 612-621.

[8] C. Grecu, P. Pande, B. Wang, A. Ivanov, R. Saleh, "Methodologies and Algorithms for Testing Switch-Based NoC Interconnects", Proceedings of 20th IEEE International Symposium on Defect and Fault Tolerance in VLSI systems, DFT 2005, Oct 3-5 2005, pp. 238-246.

[9] D. Bertozzi et al., "NoC Synthesis Flow for Customized Domain-Specific Multiprocessor System on Chip," IEEE Trans. Parallel and Distributed Systems, vol. 16, no. 2, Feb. 2005, pp. 113-129.

[10] M. Hataminian, and G. Cash, "A 70-MHz 8 bit x 8 bit parallel pipelined multiplier in 2.5 um CMOS", IEEE Journal on Solid-State Circuits, Vol. SC-21, pp. 505-513, Aug 1986.

[11] E. Nilsson, J. Oberg, "Reducing Power and Latency in 2-D Mesh NoCs using Globally Pseudochronous Locally Synchronous Clocking" Proceedings of International Conference Hardware/Software Co design and System Synthesis, 2004. CODES + ISSS 2004, pp. 176-181.

[12] R.Kol and R. Ginosar, "Adaptive synchronization for multi-synchronous systems," in *1998 IEEE Int. Conf. Computer Design (ICCD'98)*, Oct. 1998, pp. 188–189.

[13] K. Y.Yun and R. P. Donohue, "Pausible clocking: A first step toward heterogeneous systems," in *Proc. Int. Conf. Computer Design (ICCD'96)*, 1996, pp. 118–123.

[14] M. R. Greenstreet, "Implementing a STARI chip," in *Proc. Int. Conf. Computer Design (ICCD'95)*, pp. 38–43.

[15] C. L. Seitz, *System Timing, Introduction to VLSI Systems*. Reading, MA: Addison-Wesley, 1980, ch. 7.

[16] J. Seizovic, "Pipeline synchronization," in *Proc. IEEE Symp. Asynchronous Circuits and Systems (ASYNC'94)*, 1994, pp. 87–96.

[17] Y. Semiat and R. Ginosar, "Timing measurements of synchronization circuits," in *Proc. 9th IEEE Int. Symp. Asynchronous Circuits and Systems (ASYNC'03)*, 2003, pp. 68–77.

[18] R. Ginosar, "Fourteenways to fool your synchronizer," in *Proc. 9th IEEE Int. Symp. Asynchronous Circuits and Systems (ASYNC'03)*, 2003, pp. 89–97.

[19] J. Jex, C. Dike, and K. Self, "Fully asynchronous interface with programmable metastability settling time synchronizer," Patent 5 598 113, 28, 1997.

[20] A. Chakraborty and M. R. Greenstreet, "Efficient Self-Timed Interfaces for Crossing Clock Domains," 2003 IEEE Proceedings of the Ninth International Symposium on Asynchronous Circuits and Systems (ASYNC'03), May 12-15, 2003, pp. 68-78.

[21] T. Chelcea and S. M. Nowick, "Robust Interfaces for Mixed-Timing Systems," IEEE Transactions on Very Large Scale Integration Systems, Vol. 12, No. 8, Aug. 2004, pp. 857-873.

[22] J. Nyathi, R. R. Rydberg and J. G. Delgado-Frias, "Wave-Pipelining the Global Interconnect to Reduce the Associated Delays," *49th IEEE International MidWest Symposium on Circuits and Systems*, Puerto Rico, USA, August 6-9, 2006, pp. 208-212.

[23] R. Y. Chen, N. Vijaykrishnan, M. J. Irwin, "Clock Power Issues in System-on-a-Chip Designs" Proceedings IEEE Computer Society Workshop On Volume , Issue , 1999 Page(s):48 – 53.

[24] D. Hodges, H. Jackson, R. Saleh, "Analysis and Design of Digital Integrated Circuits in Deep Submicron Technology".

[25] ITRS 2005 Documents,
http://www.itrs.net/Links/2006Update/2006UpdateFinal.htm

[26] I. E. Sutherland and J. Ebergen, "Computers without Clocks," Scientific American, Vol 287, No. 2, Aug. 2002, pp. 62-69.

[27] J. Nyathi, S. Sarkar, P. P. Pande, “Multiple Clock Domain Synchronization for Network on Chip Architectures” Proceedings of IEEE International SoC Conference, SOCC 2007, 26th-29th September 2007.

[29] S. Kumar, A. Jantsch, J. P. Soininen, M. Forsell, M. Millberg, J. Oberg, K. Tiensyrja, A. Hemani, “A Network on Chip Architecture and Desgin Methodology”, Proceedings of IEEE Computer Society Annual Symposium on VLSI (ISVLSI), Pittsburgh, USA, 2002, pp. 117-124.

[30] W. J. Dally, B. Towles, “Route Packets, not Wires: On-Chip Interconnection Networks”, Proceedings of Design Automation Conference (DAC), Las Vegas, Nevada, USA, June 18-22, 2001, pp. 683-689.

Publications of Souradip Sarkar

1. Jabulani Nyathi, **Souradip Sarkar**, Partha Pratim Pande, "Multiple Clock Domain Synchronization for Network on Chip Architectures" Proceedings of IEEE International SoC Conference, SOCC 2007.
2. **Souradip Sarkar**, Partha Pande, Jabulani Nyathi, “Energy Dissipation Profile for Multiple Clock Domain Network on Chip” submitted at IEEE International Symposium on Circuits and Systems, ISCAS 2008.