

MULTIPLE CLOCK DOMAIN SYNCHRONIZATION FOR NETWORK ON CHIP ARCHITECTURES

Jabulani Nyathi, Souradip Sarkar, and Partha Pande

Washington State University
Pullman, Washington-99163

ABSTRACT

The Network-on-Chip (NoC) is emerging as a revolutionary methodology in 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 paper 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.

I. INTRODUCTION

With shrinking geometries, global interconnects are becoming the principal performance bottleneck for high-performance Systems-on-Chip (SoCs) [1]. The Network-on-Chip (NoC) model [2] is emerging as a revolutionary methodology in solving the performance limitations arising out of long interconnects. It is a widely accepted fact that several clock cycles are required for a global signal to travel from one end of a chip to the other. Consequently synchronization of future chips with a single clock source and negligible skew will be extremely difficult, if not impossible. As a result of the cross-chip signaling constraints, instead of trying to distribute synchronous clocks, it has been proposed that the whole SoC be divided into multiple functional islands with independent clocks. The inherent characteristics of NoCs allow this division into multiple clusters as dictated by the interconnect infrastructure. Given that these

clusters operate at different frequencies, communication among them gives rise to new challenges in terms of data integrity, synchronization and energy dissipation.

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 FIFO buffers either at the input or output and we propose to re-use these buffers to manage multiple clock domain synchronization. Section II presents the general architectures of two FIFO interfaces considered in this work. In Section III performance evaluation of these two schemes when applied to common NoC architectures in terms of latency and energy dissipation are presented. Section IV presents some concluding remarks.

II. NOC CLOCK SYNCHRONIZATION

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 and also the SoC often must work on external clocks or data obtained from an external clock. We aim 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. One of several possible synchronization schemes was presented by Hataminian and Cash [3]. 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.

Due to the regular structure of the NoCs, shown in Figure 1, it is proposed that the Hataminian and Cash solution can be easily extended to these. In [4], 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 NoC. 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 to the regular structure of the NoCs. The principal limitation of 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 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.

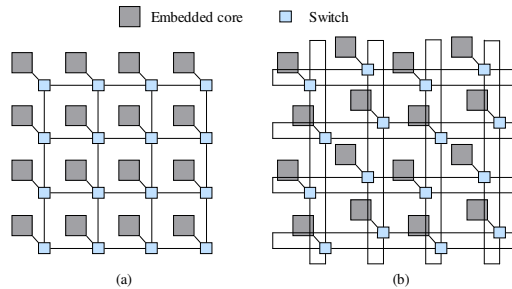


Figure 1: (a) Mesh; (b) Folded-Torus based NoCs

There are different circuit-level design methodologies to address interfacing signals crossing multiple clock boundaries. In this paper we present a new FIFO interfacing scheme that uses the asynchronous symmetric pulse protocol (asPP) approach abbreviated GasP henceforth in accordance with the definition given in [5]. The interfacing control circuitry's performance is compared to that presented by Chelcea and Nowick [6]. Key metrics of energy dissipation and latency are evaluated.

Chakraborty and Greenstreet [7] present self-timed interfaces for crossing clock domains and address various clock relationships between two communicating modules. Chelcea and Nowick [6] present robust interfaces for mixed timing systems.

Our scheme is capable of providing an extensive array of interfaces except we limit our study to a scenario in which all the communicating modules are synchronous even though they might be clocked by clock signals with different frequencies.

A. The Chelcea-Nowick (C-N) Interfaces

The work of Chelcea and Nowick [6] 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. The Chelcea-Nowick synchronous interfaces referred to as the C-N scheme from here on, require detectors in order to compute the current state of the FIFO (full or empty). The full and empty detectors shown in Figure 2 monitor and report the status of the FIFO cell. 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. The C-N scheme also has external controllers for conditionally passing requests for data operations to cell arrays. These external controllers are the put and get modules. All the modules, along with their associated input and output signals are shown in the block diagram of Figure 2.

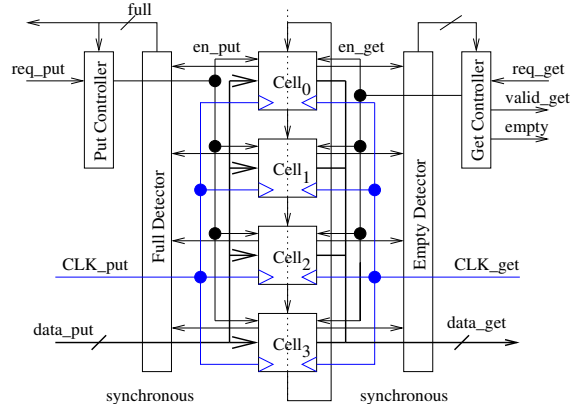


Figure 2: Block diagram of the C-N synchronous-synchronous interface (courtesy of [C-N])

Further details on the basic functions of each interface module are not repeated in this study and an interested reader is referred to [6]. Logic level details that can be inferred from a close examination of the block diagram of Figure 2 indicate considerable logic depth. Essentially the C-

N interface will incur significant delays due to the number of logic stages the majority of the signals have to traverse.

B. New GasP Based Interfaces

In this subsection we present a newly proposed interface for crossing clock domains. The interface uses self-timed control circuitry to generate local clocks and allow communicating modules operating at different arbitrary frequencies to exchange data [8]. 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 $clock_1$ and $clock_2$ 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 shows a block diagram of the GasP based scheme. There are fewer control signals involved in the generation of the enable signals than those of Figure 2.

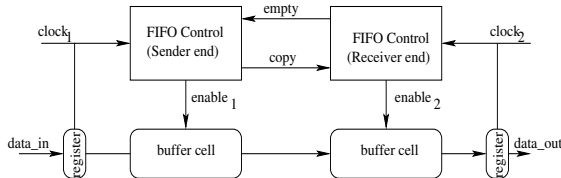


Figure 3: Block diagram of the GasP based Scheme

It must be noted that this scheme allows either the sending or receiving module to initiate a request for data transfer. The empty and copy signals play a major role in the synchronization of data transfer when $clock_1$ and $clock_2$ are independent clocks with 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 albeit after some delay allowing data to be stable 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 4.

The full/empty signal in the block diagram of Figure 3 is represented on the schematic by the signal of node C. The sender's request is stored in the keeper circuitry 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. Note that 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 start at either 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.

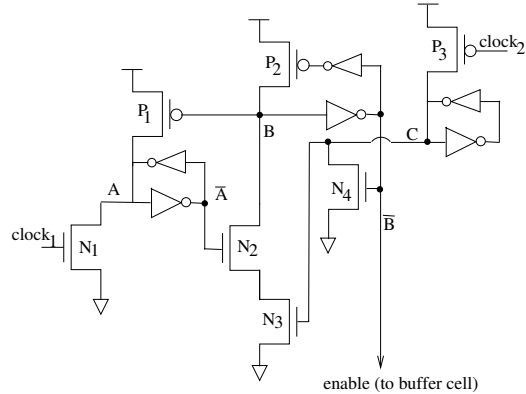


Figure 4: Circuit level representation of FIFO control circuitry

Latency for both interfaces is considered as the time it takes to queue the data on FIFO cells to the time the data is read. Latency and energy dissipation are evaluated for the various clock scenarios such as both modules operating at the same frequency and one module having a slower clock than that of the other module. Performance evaluation based on latency and energy metrics is presented in the next section.

III. 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 MESH and Folded Torus-based NoCs. 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 5. 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 and the GasP based FIFO interfaces different clock frequencies were used.

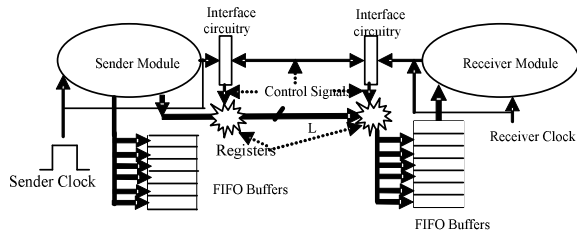


Figure 5: The experimental set up

Table 1 shows the latency and energy values for the C-N and the GasP based FIFO interfaces. In all categories the GasP based FIFO interface outperforms the C-N FIFO interface. For various relationships between the sender and the receiver

clocks the latency of the GasP based FIFO interface shows around 80% improvement over that of the C-N FIFO interface. The energy values shown in Table 1 show the C-N synchronous to synchronous FIFO interface to dissipate significantly more energy than the GasP controlled FIFO interface for both the MESH and Folded-Torus based NoC architectures. This is a direct result of the simplicity and the reduced circuitry of our design.

IV. CONCLUDING REMARKS

A FIFO interface scheme that addresses the multiple clock domain synchronization issue in a NoC platform has been presented. It has been shown in this study that communicating NoC switch blocks running at the same or arbitrary frequencies can be managed by the proposed FIFO interface. The proposed FIFO interface circuitry is simple yet effective, reducing energy dissipation significantly. At a minimum the C-N FIFO interface is shown to dissipate 1.97 and 1.78 times more energy than the newly proposed FIFO interface for the MESH and Folded Torus architectures respectively with much higher latency. Overall it has been shown in this paper that instead of depending on the architectural regularity of NoC architectures for clock synchronization, the NoC switch blocks can be designed in such a way that they can handle communication among modules operating in different clock domains.

Table 1 Performance comparison of the C-N and the GasP based FIFO interfaces

Architecture	Sender (GHz)	Receiver (GHz)	Latency (ps)		Energy Dissipation (pJ)	
			C-N Interface	GasP Interface	C-N Interface	GasP 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

References

- [1] L. P. Carloni and A. L. Sangiovanni-Vincentelli, "Coping with Latency in SoC Design," *IEEE Micro*, Oct. 2002, pp. 24-35.
- [2] L. Benini and G. De Micheli, "Networks on Chips: A New SoC Paradigm," *IEEE Computer*, Jan. 2002, pp. 70-78.
- [3] 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
- [4] 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
- [5] I. E. Sutherland and J. Ebergen, "Computers Without Clocks," *Scientific American*, Vol 287, No. 2, Aug. 2002, pp. 62-69.
- [6] 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.
- [7] 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.
- [8] I. Sutherland and S. Fairbanks, "GasP: A Minimal FIFO Control," *Proc. of ASYNC*, 2001, pp. 46-53.