Wave Pipelining for Application-Specific Networks-on-Chips

Jiang Xu
Dept. of Electrical Engineering
Princeton University
Jiangxu@Princeton.edu

Wayne Wolf
Dept. of Electrical Engineering
Princeton University
Wolf@Princeton.edu

ABSTRACT
This paper presents methods for optimizing application-specific networks-on-chips (NoCs). We show that wave pipelining provides more energy efficient data transport than non-wave pipelined communication. We observe 52% energy saving, 60% transistor area saving, and 1.7 times speedup by using wave pipelining in simulation. Wave pipelining is particularly well suited to networks-on-chips because the network's structured interconnection provides better delay control. Our analysis shows how designers can tune their network to the requirements of the application by choosing a design point along area/performance or area/energy curves.

Categories and Subject Descriptors
B.7.1 [**Types and Design Styles**]: Advanced Technology.

General Terms
Design, Measurement

Keywords
Networks-on-chip (NoC), wave pipelining, system-on-chip (SoC), coupling capacitance, interconnection

1. INTRODUCTION
This paper describes how wave pipelining can be used to design application-specific networks-on-chips (NoCs). Networks-on-chips have been proposed to enable locally synchronous/globally asynchronous design of systems-on-chips (SoCs). SoCs built with networks-on-chips are structured as multiprocessors with the processing elements connected by an on-chip network, such as an Ethernet-style network.

The International Technology Roadmap for Semiconductors 2001 edition (Table 1) [9] describes the expected path of VLSI technology development and points out some potential roadblocks. First, due to smaller feature sizes and higher clock frequency, global interconnections will have longer delays and become performance bottleneck, and data will need multiple cycles to cross a chip [4]. Second, with the decreased power supply and taller and closer wires, crosstalk is becoming an important problem on interconnections [11]. Third, while smaller feature sizes enable more than one billion transistors on a single chip, defects make yield targets more difficult to meet. All these challenges are related with on-chip global interconnections, which form an on-chip network.

Table 1. Data from ITRS 2001

Year	2003	2004	2005	2006	2007	2010
Process (nm)	107	90	80	70	65	45
Clock (MHz)	3088	3990	5173	5631	6739	11511
Chip size (mm^2)	572	572	572	572	572	572
Power supply (v)	1.0	1	0.9	0.9	0.7	0.6
Transistor (M)	810	1020	1286	1620	2041	4081

Networks-on-chips are particularly attractive for SoCs because the network design can be optimized to fit the requirements of the application. Circuit, logic, and protocols of the NoC can be customized to meet the performance/power/area requirements posed by the system-on-chip. Wave pipelining, which is a pipelining without latches, has been known for quite some time to provide high-performance communication, but the design of wave-pipelined logic is particularly challenging. But this design technique is well suited to networks-on-chips because the layout and logic structure provided by these networks simplifies analysis and allows us to more accurately control delay variations. As a result, we believe that wave pipelining is particularly well suited to NoCs.

We present a new analysis of wave pipelining that evaluates the tradeoffs between area, performance, and energy in the design of a communication system. We show that wave pipelining is more energy efficient than non-wave pipelined communication. We also provide a new metric, average transmission time, that compares the communication rates of wave pipelined and non-wave pipelined communication links. This analysis allows NoC designers to choose appropriate area/performance and area/energy tradeoffs in the physical layer of the communication link. This methodology allows designers to customize the design of the physical layer to the SoC requirements.

In this paper, we will discuss the use of wave pipelining in application-specific NoC design. In the next section we introduce the previous work on NoC and wave pipelining. In Section 3 we

CASES 2002, October 8-11, 2002, Grenoble, France.

study wave pipelining in NoC in detail, present our analysis, and validate it by circuit simulation.

2. PREVIOUS WORK

Networks-on-chips have become an active research area in the past several years with contributions from several groups. Some system level researchers describe it as communications between processes [16]; some architecture researchers view it as the links between functional units [7] [12]; some physical level researchers understand it as a collection of physical interconnections [4].

The different views come with the layered concept of networks and the layered research areas. Several researchers propose layered models for NoC design. Sgroi et al [17] suggest using the OSI 7-layer model, and Benini et al [2] suggest a 3-layer model. The division of the NoC into layers depends on not only the various NoC functionalities but also the existing design and research areas. A properly layered model should help people to understand NoC as well as facilitate the design using existing knowledge. NoC design practices are still in their early stages. Some researches use networks in on-chip multiprocessors [15]. Others are related with on-chip buses [1][5][13].

Wave pipelining theory is developed by L. Cotton in 1969 [6]. He calls it "Maximum rate pipelining" because signals are directly pipelined in circuits without using latches. Wave pipelining roots from the fact that on a chip each gate is a small storage element and sometimes on a circuit board connection length is much larger than signal wave length. So signal can be stored in gates or PCB connections during computation or transmission. Since the wave pipelining was born, a lot of practices have been done [8][14][18]. One successful example is the RAMBUS memory, which store signals on PCB connections.

3. WAVE PIPELING

We learned that pipelining will be used to increase NoC throughput when long delays are inevitable. RAW architecture from MIT uses pipeline in a multiprocessor network [15]. In RAW architecture each jump from one router to another is a pipeline stage. Here we try to bring pipelining idea into a lower level, and we find wave pipelining can directly pipeline data onto global interconnections.

Although wave pipelining is simple, the design is difficult because of multi-path problem. Usually multiple data paths exist between inputs and outputs in a circuit. Wave pipelining needs a small delay difference among all the paths to avoid signal racing. Moreover multiple data paths also exist from inputs to some internal nodes, and these paths must also be balanced. Large and complex circuits make balancing multi-path delay difficult.

3.1 Wave Pipelining in NoC

On-chip global interconnections have very simple circuits. They are chains of wires and inverters, which are also called buffers or repeaters. Furthermore, there is only one data path on each interconnection and identical interconnections in a parallel connection. Such simplicity makes wave-pipelining design easy. However global interconnections have large coupling capacitance, which makes the interconnection delay to vary from time to time. Coupling capacitance makes adjacent interconnections affect each other. When the signals on two adjacent interconnections change in the same direction, the interconnection delay is the shortest, called best-case delay. When the signals change in opposite directions, the delay is the longest, called worst-case delay. This delay variation limits the wave pipelining frequency. We can see this effect in the following proof.

Our model is an interconnection with an input buffer and an output buffer. The two buffers send and receive data to and from the interconnection with the same frequency $1/T_c$, and there is a phase difference T_d. The single edge skew of the clock is $\pm\Delta$. The output buffer has a setup time T_s and a hold time T_h. The longest and shortest delays are D_{max} and D_{min} respectively. Following the steps in [3], we can show the effect of the interconnection delay variation $D_{max} - D_{min}$ is

$$T_c > (D_{max} - D_{min}) + 2\Delta + T_s + T_h \quad (1)$$

So the pipelining frequency is limited by the interconnection delay variation.

Usually in interconnections, the D_{max} is the worst-case delay, in which the coupling capacitances are doubled, and the D_{min} is the best-case delay, in which the coupling capacitances are ignored. Because an interconnection has two adjacent wires, one is on left and the other is on right, the factors become 4 and 0 times of the coupling capacitances. However, when we look into detailed signal schemes, the variation is only $(D_{max} - D_{min})/2$. The reason is: when a signal edge runs in the worst case, the very next signal edge cannot run in the best case. This fact reduces the coupling capacitance effect in wave pipelining. So in practice, (1) becomes

$$T_c > (D_{max} - D_{min})/2 + 2\Delta + T_s + T_h \quad (1')$$

3.2 Analysis

This section analyzes the tradeoffs between area, performance, and power in wave-pipelined NoCs. We start by defining some variables:

T --- Average transmission time is the time between an input buffer begins to send the first bit and an output buffer begins to receive the last bit in a transaction.

d --- Delay is the time that a step signal needs to pass an interconnection.

t --- Pipeline delay is the minimum interval between sending two reverse step signals.

n --- The average number of bit in a transmission.

E --- The energy required to transmit one bit.

The throughput is n/T. In non-wave-pipelining method, called traditional method, the average transmission time T_t is (2), where the subscript t stands for traditional method.

$$T_t = n * d_t \quad (2)$$

In wave pipelining, the average transmission time T_w is (3), where the subscript w stands for wave pipelining.

$$T_w = (n - 1) * t + d_w \quad (3)$$

When $T_w < T_t$, wave pipelining provides a smaller average transmission time than the traditional method, and the following inequality holds.

$$n > (d_w - t) / (d_t - t) \quad \text{when} \quad d_t > t \quad (4)$$

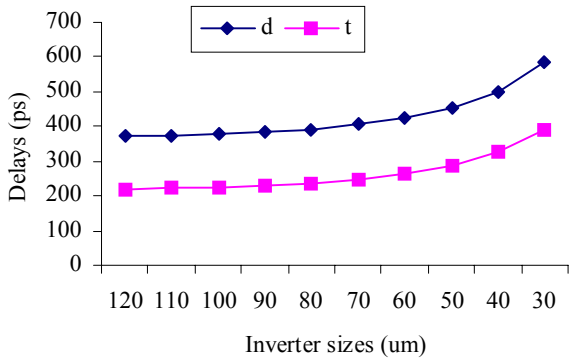
We can also calculate the threshold N to compare wave-pipelined and non-wave-pipelined communication. When the average

number of bit is equal to N, wave pipelining has the same average transmission time as the traditional method.

$$N = (d_w - t) / (d_t - t) \quad (5)$$

3.3 Simulation Results

We have validated our analysis by circuit simulations. We simulated a 10000µm global interconnection in a 0.25µm aluminum process. There are three parallel interconnections, and we study the middle one. The wires are modeled by π3 model, and coupling capacitances are explicitly modeled. We measure the delays at 10% and 90% points. The delay of the traditional method is the worst-case delay, and the delay of wave pipelining takes account of the delay variation. The input and output buffers use the same size inverters as those inserted into the interconnections. Based upon Kahang et al [10], when the pitch is 3.2µm, the fastest interconnection is 1.2µm wide and uses 5000µm-spaced inverters, in which PMOS transistors are 100µm wide.



Using Cadence Spectre, we simulated the circuit under different inverter sizes (Figure 1). When inverter size is 100µm, there is a 379ps delay to a signal pass the 10000µm interconnection. So in traditional methods, the maximum clock is 2.64GHz. However, if we use wave pipelining, the maximum clock is 4.42GHz, which is about 1.7 times of the traditional method.

By setting d_t = 379ps in (5), we compare other schemes with the traditional method using 100µm inverters, called target method. When inverter size reduces to 40µm, the wave pipelining will have the same average transmission time as the target method, if the average number of bit N is 3.28. The maximum clock is 3.08GHz, and the inverters only use 40% area of the target method.

We also measure the power efficiency of the interconnections. Without considering throughput, wave pipelining needs more power than the traditional method. But average energy consumption is the real criteria. In fact, wave pipelining consumes less energy on each bit than traditional method (Figure 2), and it consumes more power simply because of much higher throughput. While the target method needs 20.5pJ/bit, wave pipelining only needs 17.1pJ/bit, which is 83% of the former. When the size is 40µm, the average energy is 9.88pJ/bit, which is only 48% of the target method.

Because wavelength of on-chip signals is about 1~10 centimeters, which is comparable with global interconnection length, data store in inverters instead of wires in wave pipelining. By inserting more inverters, it is possible to increase the number of pipeline stage and reduce inverter sizes, while slightly increasing delays. Inserting more inverters can also reduce crosstalk noise by limiting its propagation along interconnections. To prove this

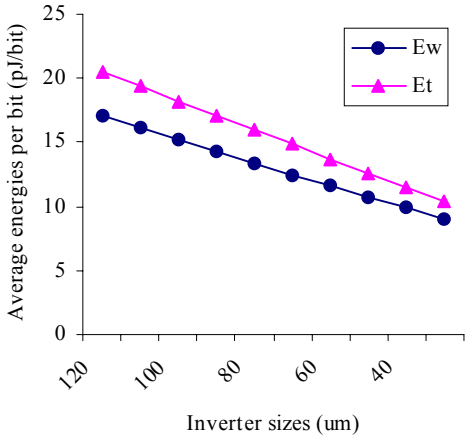


Figure 2. Average energy consumptions per bit

idea, we simulate the 10000µm interconnection with 4 inverters at a 2500µm interval (Table 2). In the results, when inverter size is 40µm, the maximum clock is increased to 3.55GHz, and the output signals have better shapes than two-inverter case. If the average number of bit reaches 7.31, using 30µm inverters will further reduce the energy and area. We also shrink the wire width to 1.0µm and observe some decreases of average energies and pipeline delays.

Table 2. Simulation results – 4 inverters

Inverter size (µm)	d (ps)	t (ps)	N (d_t = 379)	Average pipelining energy (pJ/bit)
50	556	254	2.42	16.8
40	605	282	3.33	14.9
30	688	330	7.31	13.0

From the simulation results, we make two observations. At one end, wave pipelining can increase throughput and energy efficiency, if using the same circuit as the traditional method. At the other end, wave pipelining can reduce area and energy consumption, while keeping the same throughput. In the middle, we can save area and energy while increasing throughput. So wave pipelining gives NoC design a widely choose among area, power, and performance. Such flexibility is very helpful to application-specific NoC designs.

3.4 Design Methodology

We summarize our design methodology for wave-pipelined application-specific NoCs as follows:

1. Choose the (even) number of wave pipelining buffers to be used in the communication link.
2. For several buffer sizes in a feasible range:

a. Determine circuit parameters t, d_t, d_w, E_t, and E_w using circuit simulation.

b. Compute N, the breakeven transmission length.

3. Select N based on the principal objective function, which may be either performance or energy. Based on N, determine the required buffer sizes.

3.5 Reducing Delay Variation in Wave Pipelining

In our technology, coupling capacitance is only 22% of the total capacitance. But this share is expected to increase fast in the next few years. For non-pipelining interconnections, designers try to find ways to reduce coupling capacitance so that the worst-case delay can be reduced. In wave pipelining, we are interested in reducing not only the coupling capacitance but also the delay variation. Increasing the space between interconnections is the simplest way to accomplish this goal. We also could insert grounded wires between interconnections to shield them. But these methods will either increase areas or reduce wire widths. Kahang et al [10] introduces a method called repeater offset, which places repeaters at different points on adjacent interconnections instead of placing them next to each other. When a piece of interconnection accelerates its neighbors through coupling capacitances, the next piece of interconnection, which is after an inverter, will decelerate its neighbors. So if the inverters are placed right, this method can reduce both the worst-case delay and the delay variation.

3.6 NoC Timing with Wave Pipelining

NoC will use a locally synchronous and globally asynchronous timing scheme. Usually the send buffer and receive buffer work at the same frequency but different phases, and the buffers synchronize data to local clocks. In wave-pipelined NoC, data packets can bear a header to synchronize a send buffer with a receive buffer. The header only needs several bits, and not every packet needs a header. Headers can be sent in a fix interval, which depends on the stability of local clocks. Because wave pipelining can work even the send buffer and receive buffer have a phase difference, it works more efficient.

4. CONCLUSIONS

NoC design is a promising solution for large system-on-a-chip designs in the near future. Wave pipelining amplifies the merits of NoC design. It can save 60% transistor area and 52% energy, while increasing NoC performance. It gives application-specific NoC designs a much wider and efficient design space. Moreover, it can be efficiently used in a locally synchronous and globally asynchronous NoC.

5. ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation.

6. REFERENCES

[1] AMBA http://www.arm.com/arm/AMBA.

[2] Luca Benini, Giovanni De Micheli, “Networks on chips: A new SoC paradigm”, Computer, 35(1): 70-78, 2002.

[3] Wayne P. Burleson, Maciej Ciesielski, Fabian Klass, Wentai Liu, "Wave-pipelining: A tutorial and research survey", IEEE Trans. VLSI systems, 6(3): 464-474, 1998.

[4] Jason Cong, “An interconnect-centric design flow for nanometer technologies”, Proceedings of the IEEE, 89(4): 505-528, 2001.

[5] CoreConnect Bus. http://www-3.ibm.com/chips/techlib /techlib.nsf/productfamilies/CoreConnect_Bus_Architecture.

[6] L. Cotton, “Maximum rate pipelined systems”, Proceedings of AFIPS Spring Joint Comput. Conf., 1969.

[7] William J. Dally, Brian Towles, “Route packets, not wires: on-chip interconnection networks”, Proceedings of the 38th Design Automation Conference, 2001.

[8] O. Hauck, A. Katoch, S.A. Huss, “VLSI system design using asynchronous wave pipelines: a 0.35um CMOS 1.5 GHz elliptic curve public key cryptosystem chip”, ASYNC 2000, 188-197, 2000.

[9] Sematech, International Technology Roadmap for Semiconductors, 2001 Edition.

[10] Andrew B. Kahng, Sudhakar Muddu, Egino Sarto, Rahul Sharma, “Interconnect Tuning Strategies for High-Performance ICs”, Proceedings of DATE ‘98.

[11] M. Kuhlmann, S.S. Sapatnekar, “Exact and efficient crosstalk estimation”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 20(7): 858-866, 2001.

[12] S. Kumar, A. Jantsch, Juha-Pekka Soininen, M. Forsell, M. Millberg, J. Öberg, K. Tiensyrjä, A. Hemani, “A network on chip architecture and design methodology”, Proceedings of IEEE Computer Society Annual Symposium on VLSI, April 2002.

[13] K. Lahiri, A. Raghunathan, S. Dey, “Evaluation of the traffic-performance characteristics of system-on-chip communication architectures”, Proceedings of the IEEE International Conference on VLSI Design, 29-35, 2001.

[14] Byoung-Hoon Lim, Jin-Ku Kang, “A self-timed wave pipelined adder using data align method”, AP-ASIC 2000, 77-80, 2000.

[15] RAW architecture. http://www.cag.lcs.mit.edu/raw/documents.

[16] A. Sangiovanni-Vincentelli, M. Sgroi, L. Lavagno, “Formal models for communication-based design”, Proceedings of CONCUR 2000, 29-41, 2000.

[17] M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey, A. Sangiovanni-Vincentelli, “Addressing the system-on-a-chip interconnect woes through communication-based design”, Proceedings of the 38th Design Automation Conference, 2001.

[18] D. Wong, G. De Micheli, M. Flynn, “Designing high performance digital circuits using wave pipelining: Algorithms and practical experiences”, IEEE Trans. Computer-Aided Design, 12(1): 25-46, Jan. 1993.