

A Wave-Pipelined On-chip Interconnect Structure for Networks-on-Chips

Jiang Xu and Wayne Wolf
Dept. of ELE, Princeton University
jiangxu@princeton.edu, wolf@princeton.edu

Abstract

The paper describes a structured communication link design technique, wave-pipelined interconnect, for networks-on-chip. We achieved 3.45GHz and 55.2Gbps throughput on a 10mm 16bit interconnection in a 0.25um technology. It uses 0.079mm² of area, and it only needs 18.8pJ to transmit one bit. We reduce 79% crosstalk delay by using two techniques -- interleaved lines and misaligned repeaters. This paper shows the various techniques we used to save power and area and achieve high performance in a relative old technology in detail. Wave-pipelined interconnect design is relatively easy, but many features of it give a large and flexible design space for high-performance chips.

1. Introduction

This paper introduces an on-chip interconnect structure using wave pipeline for networks-on-chip (NoC). By using this structure, we achieved 3.45GHz on a 10mm bidirectional on-chip interconnect in a 0.25um aluminum technology, and it gives a 55.2Gbps throughput on a 16bit interconnect. It uses very small area and saves power. The design is easy. It supports asynchronous communication and globally asynchronous and locally synchronous scenarios.

The number of usable transistor is increasing exponentially to help designers to fulfill the growing market demand. High performance and multifunction chips, for example, system-on-chips (SoC) will work at very high frequency (about 10 GHz) and have huge amount of transistors (more than one billion) and large area (about 572mm²). These chips will use globally asynchronous and locally synchronous clock schemes [2]. Global interconnects are needed to communicate among different clock zones. They have to be high performance in many cases, for example, the interconnect between a processor core and embedded memories. Another design trend is reusing IP (Intellectual Property) cores. To accelerate large designs, designers try to reuse large number of IP cores on a single chip. We, among many

other researchers, suggest connecting IP cores by NoC. NoC can be reused and further reduce the time-to-market [1]. Global interconnects are important elements of NoC, and they connect IP cores together. Structured interconnect is the fundamental of NoC design. Wave-pipelined interconnect is a technique to support structured interconnect.

While transistors become smaller and faster, the International Technology Roadmap for Semiconductor (ITRS) [3] predicts that global interconnects will become slower and relative larger, even after repeaters are properly inserted. On one hand, due to smaller feature sizes, the area of interconnects' cross sections decreases, and the resistance per unit length increases, which increases delay. On the other hand, global interconnects become taller and relatively closer, and crosstalk becomes a serious problem. Lower supply voltages make communication through global interconnects more sensitive to crosstalk. Slow global interconnects limit the communications between IP cores, which will run at GHz level.

In this paper, we show a new technique for on-chip interconnects using wave pipeline, which attacks the performance, crosstalk, and power issues in global interconnects. This technique also gives designers more flexibility on NoC design. In the following section, we talk about some related previous works. We will detail our technique in section 3. Section 4 shows the simulation results and analysis. A design methodology for better using this technique in NoC design is described in section 5. A brief conclusion is given in section 6.

2. Previous work

In the near future, data and control signals will need multiple clock cycles to cross a chip using relatively slow global interconnects [4]. This fact will force designers to pipeline data on global interconnects to achieve higher throughput under a large delay. An optimistic view is many data are quite large, and only the first set of bits will suffer the large delay. As with other pipelines, the communication performance of wave-pipelined interconnects are related with the delay, the pipeline

throughput, and the data size. We show the relationship in another paper [5].

Our study shows that wave pipeline is a favorable choice for communications on global interconnects. Wave pipeline is brought up 34 years ago [6]. In wave pipeline, data are pipelined in circuits without using latches, because wires and transistors not only transmit data and compute but also store them for a period of time. Instead of using latches, wave pipeline stores data on wires and transistors on the fly. Usually wave pipeline design is very hard, because different paths from one input to one output of a circuit often have different delays, and all the delays must be balanced [7]. But the benefit is significant. The delay overhead of latches can be saved. Increasing number of practices is brought up in recent years [8] [9] [10]. RAMBUS memory is a successful design using wave pipeline.

We show global interconnect is yet another right place to use wave pipeline in our previous paper [5]. The relatively simple and regular circuit of interconnect makes wave pipeline design difficulties much easier. The benefit by using this advanced technique is great. If a global interconnect is pipelined using latches, those latches must be at least as large as repeaters on the interconnect, because they need not only to store data but also to drive the next interconnect section, which needs very large inverters. In our case, these inverters are 20~50 times larger than the smallest inverters. Large inverter is slow and power-hungry. Needless to say, latches on global interconnect will introduce a lot of delay, consume a lot of power, and use more area. Wave pipeline does not use latches, so it has less delay and uses much less power and area, comparing to traditional methods. Wave pipelined interconnect also gives designers a new method and larger space to tradeoff between performance, area, and power. In the previous work, we show wave-pipelined interconnect itself without the senders and receivers. In this work, we show other features of a complete wave-pipelined interconnect design.

3. Design description

Our design achieves 3.45GHz on every single bit line in an old but widely used 0.25um technology, and we expect it to achieve much higher frequencies in new technologies. The maximum chip size is about 300mm² at 0.25um technology and 572mm² from 107nm to 65nm technologies [3]. So we choose a moderate length, 10mm, for our wave-pipelined interconnect design. The design uses TSMC 0.25um technology and is simulated using Cadence Spectre. We use metal-3 in a 5 metal layer process. The delay-optimized 3.2um-pitch global interconnects are 1.2um wide based upon the research by Kahng [11], and we verified part of his results.

There are three components in our design (Figure 1). Data and clock are fed into the same drivers (or senders), and then they are pumped into clock and data lines. At receiver end, rectified data signals are caught by flip-flops driven by amplified clocks. In following, we describe each component in detail.

3.1. Clock and data lines

The bidirectional wave-pipelined interconnect is 16 bits wide. It could be easily changed to any other width, for example 128 bits, as long as a clock signal is transmitted along with a set of data lines. For the technology we use, one clock is along with every eight data lines. The proportion is determined by the number of receivers a single clock signal can drive, and it may change in different technologies. In the 16-bit wave-pipelined interconnect, there are two lines for two reversed clock signals embedded with data lines on each direction. The clocks are used by receivers to latch data. By sending clocks along with data, our design avoids using high-frequency phase locked loops (PLL) or coding schemes to regenerate timing information from data. A direct result of this is that several clock cycles are saved. Clock lines use the same layout as data lines. This eases matching delays between data and clock lines in design and fabrication. Also the environment will have equal effects on both data and clock lines. These characters are highly desirable in most designs. Clock circuit is the critical path of our design, because with the same signal strength, a clock drives about 40 times more load than a data signal. Sending two reversed clocks avoid the delay overheads for generating a reversed clock and amplifying clock signals to match the load.

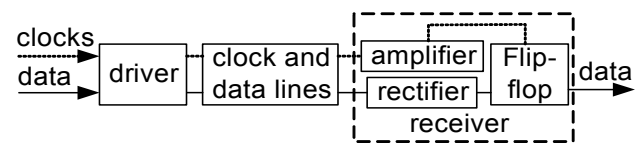


Figure 1. Wave-pipelined interconnect structure for one direction

Crosstalk is a major issue in global interconnect design. It causes delay variation among interconnects. We observed as much as 140ps delay variation on the 10mm interconnect between the fastest and slowest data signals. Crosstalk also distorts signals and so limits the pipeline frequency. Coupling capacitance and inductance among nearby lines create crosstalk. Coupling capacitance makes signals in nearby lines to accelerate each other if they change in the same direction and hold back each other if they change in different directions. Coupling inductance always has an opposite effect than coupling capacitance.

While coupling capacitance reduces with the square of distance, coupling inductance reduces linearly with the distance. In our design, coupling capacitance is dominant. But we expect more coupling inductance effects in next several technology generations.

We use interleaved lines and misaligned repeaters to reduce crosstalk delay variation from 140ps to 29ps (Figure 2). We interleave the data and clock lines for two directions, up-link and down-link. Interleaving increases the distance among lines for each direction. In many cases, up-link and down-link are not active at the same time, and then lines from the idle direction serve as grounded shields for the lines from the active direction. When both directions are active at the same time, misaligned repeaters [11] reduce crosstalk delay among nearby lines. We assign repeaters from different directions about 2.5mm from each other, so the reversed signal changing activities on the two sides of a repeater will cancel each other's effects on nearby lines. For example in Figure 2, we assume the signal on line A0 is 0→1 in region 0 and the signal on line B0 is also 0→1 in region 0. Then in region 0, line A0 accelerates line B0. However, in region 1, the signal on line A0 is 1→0 and the signal on line B0 is still 0→1. Then in region 1, line A0 holds back line B0. In total, line A0 has two opposite effects on line B0, and they cancel each other.

3.2. Sender and receiver

At receiver end, data signals are rectified to sharpen the edges and smooth out noises before they can be used. Then rectified data are caught by flip-flops in receivers. It is difficult to design a flip-flop working at 3.45GHz in the 0.25um technology. Instead, we use a pair of flip-flops (Figure 3), which work at different clock edges. While one flip-flop catches a datum at rising clock edges, the other flip-flop catches the next datum at falling clock edges.

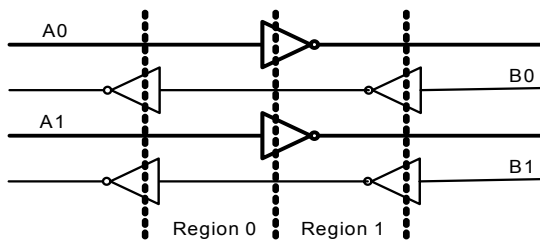


Figure 2. Interleaved lines

In a flip-flop, the feed-back loop prevents data from dissipating, but it also increases the response time to incoming data. To increase the sensitivity and reduce respond time of a flip-flop, we simplify the first stage to a single inverter. And to prevent data from being lost

during a long idle period, we also send one more clock edge to transfer the datum stored in the first stage of a flip-flop to its second stage, which a feed-back loop can keep it as long as needed. The sender is a chain of inverters, which amplify and pump low-strength signals into clock and data lines. The interfaces of wave-pipelined interconnect ease the connection between it and other parts of a chip. The input of a sender can be driven by a minimum size inverter. The output data can be store in a serial buffer and be picked up using a local clock.

3.3. Signaling

The signaling in our design is simple. Except for data lines, there are only clock lines. Clocks also embedded control information. They mark the beginning and the end of data. Clocks are sent only when there are data sent, and there is no warm-up time for clock. After the last data, one more clock edge is sent to transfer datum in the first stage of the flip-flops to its second stage. Up-link and down-link frequencies are not necessary the same. In many designs, there are different performance requirements for up-links and down-links.

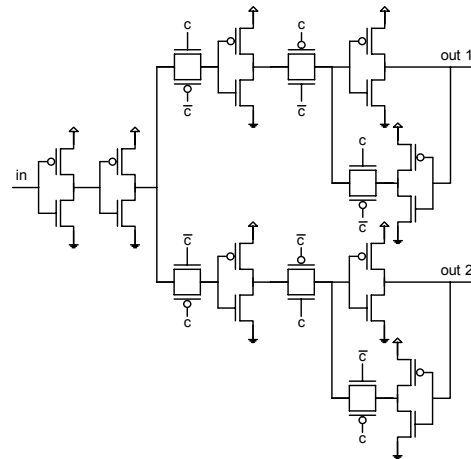


Figure 3. A receiver with a pair of flip-flops

Clocks consume a lot of power during an active period, and our design reduces its power in two ways. First, clock frequency is the half of the pipeline frequency. Second, the clocks are sent along with data, and clocks are idle when no data need to be sent. Much power is saved by reducing frequency and active period.

4. Simulation results and analysis

We simulated the whole design in Cadence Spectre. The design work stably at 3.45GHz on every data line, and the total throughput is 55.2Gbps. The area is about

0.079mm². It needs 18.8pJ to transmit one bit, including the clock overhead.

4.1. Delay, throughput, and delay variation

We measured the 50% delay from the driver inputs (Table 1). Up-link and down-link have different delays because they have slightly different structures, but if necessary, they could be designed similarly. Signals fed into drivers are 290ps wide including 20ps skews on front and back ends.

Table 1. Delay and pipeline stages

| Up-link/down-link | Before flip-flop | After flip-flop |
|---------------------|------------------|-----------------|
| 50% delay (ps) | 793/910 | 1167/1290 |
| Pipeline stages | 2.7/3.1 | 4.0/4.4 |
| Delay variation(ps) | ≤29 | -- |

The throughput on each data line is 3.45Gbps. For up-link, data need 793ps to reach a flip-flop, and there are 2.7 pipeline stages. Because wave pipeline does not use latches to clocking data, so there could be a fragment of pipeline stage. Flip-flops use another 374ps or 1.3 pipeline stage to catch and store data. The delay variation is less than 29ps, compared to 140ps without using interleaving and misalignment.

4.2. Power

When active, the maximum power consumption of the 55.2Gbps up-link is 1039mW, and the energy efficiency is 18.8pJ/bit including clock overhead (Table 2). When idle, the leakage power is 0.43mW. A typical 10mm non-pipelined global interconnect working at 2.64GHz uses 20.5pJ/bit without considering receiver and clock [5]. Wave-pipelined interconnect works 1.3 times faster and are more energy efficient.

In the total power, clocks use 125mW or 12%. This power will be wasted if no data is sending. We send clocks along with data, and instead of 125mW, only 0.34mW leakage power is consumed when no data are sent. Repeaters use 34% of the total power, and drivers use 55% of the total, because large size inverters are used to drive long interconnects.

4.3. Area

Based upon MOSIS lambda rules, the area of drivers including clock circuit is about 0.067mm², and the area of receivers including clock circuit is about 0.012mm². Compare to 2.7mm² of ARM9E-S or 2.0mm² of PowerPC405 [12] [13], the area is very small. The interconnect uses about 1.13mm² of metal. The area will

be even less in newer technologies. Because wave-pipelined interconnect has a small area, it can be used to give data-hungry IP cores enough communication bandwidth.

4.4. Analysis

Wave-pipelined interconnect can be designed at various length and bit width other than 10mm and 16 bits. It fits well into the globally asynchronous and locally synchronous scenarios when more than one billion transistors are on a single chip. Wave-pipelined interconnect can be used to communicate between asynchronous clock zones. Because sender clock signals are transmitted along with data, receivers do not need to use local clock to catch data. After data are caught, they can be synchronized to local clock by storing them in buffers. The buffers are written using sender clocks and read using local clock. The high working frequencies of wave-pipelined interconnect easily match clocks of IP cores. Because it can run at the same clock as local IP cores, there is no need to generate an extra clock for wave-pipelined interconnect, and this also simplifies the timing issues on the interfaces.

Table 2. Maximum power and energy efficiency

| | Power (mW) | Energy efficiency (pJ/bit) |
|-----------------------|------------|----------------------------|
| Data | | |
| Driver | 571 | 10.3 |
| Receiver | 30 | 0.543 |
| Interconnect repeater | 313 | 5.67 |
| Data total | 914 | 16.6 |
| Clock | | |
| Driver | 72 | 1.30 |
| Amplifier | 14 | 0.254 |
| Interconnect repeater | 39 | 0.707 |
| Clock total | 125 | 2.26 |
| Total | 1039 | 18.8 |

Wave-pipelined interconnect can also be extended to buses, where multiple IP cores share the high bandwidth global interconnects. There will be different wave pipeline stages between different IP cores, but the design is still simple because there is no need to balance the delay among each pair of IP cores as designing a pipelined bus using latches.

5. Design methodology

Structured communication links are the fundamental of a design using NoC. Wave-pipelined interconnect is a

technique to support structured links. Wave-pipelined interconnect can be integrated into a design flow easily (Figure 4). After an architecture and a NoC for it are chosen, designer can identify the communication links and their throughputs and maximum tolerable delays. The links have long delays and high throughputs could be implemented using wave-pipelined interconnects. The maximum delays limit the maximum length of interconnects. Based upon the throughputs and delays, wave-pipelined interconnect can be designed using the method mentioned in our previous paper [5]. Designer can get the area and power estimations of the wave-pipelined interconnects and use them to estimate the performance, area, and power of the architecture.

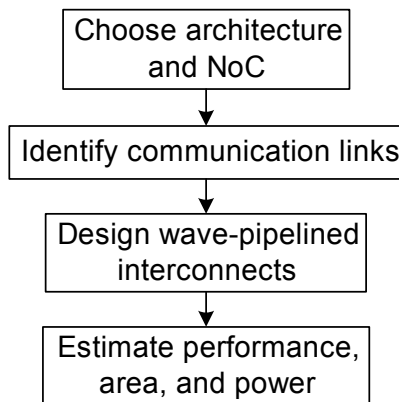


Figure 4. Design methodology

Wave-pipelined interconnect helps to overcome some difficulties of NoC design. Because wave-pipelined interconnect has a simple structure, it is easy to automate the design and even build a library of IP cores for wave-pipelined interconnects with various lengths and performance. Also wave-pipelined interconnect gives more room on performance, power, and area along with other useful features, and designs are more flexible by using them.

6. Conclusion

Supporting NoC design, wave-pipelined interconnect has a simple yet efficient structure. Wave-pipelined interconnect has a high performance while using less area and more energy efficient. It has several major advantages over pipelines using latches. It is compatible with the high clock frequencies of IP cores and able to give enough bandwidth to data-hungry IP cores. It fits well in globally asynchronous and locally synchronous clock schemes. Wave-pipelined interconnect design can be integrated into a design flow and shows designers a larger and more flexible design space.

7. Acknowledgement

This work was supported in part by New Jersey Commission on Science and Technology (NJCST).

8. References

- [1] William J. Dally, Brian Towles, "Route packets, not wires: on-chip interconnection networks", *Proceedings of the 38th Design Automation Conference*, 2001.
- [2] T. Meincke, A. Hemanj, etc, "Globally asynchronous locally synchronous architecture for large high-performance ASICs", *IEEE International Symposium on Circuits and Systems VLSI*, 1999.
- [3] Sematech, International Technology Roadmap for Semiconductors, 2001 Edition.
- [4] Jason Cong, "An interconnect-centric design flow for nanometer technologies", *Proceedings of the IEEE*, 89(4): 505-528, 2001.
- [5] Jiang Xu, Wayne Wolf, "Wave pipelining for application-specific networks-on-chip", *International Conference on Compilers, Architecture, and Synthesis for Embedded System*, Grenoble, 2002.
- [6] L. Cotton, "Maximum rate pipelined systems", *Proceedings of AFIPS Spring Joint Computer Conference*, 1969.
- [7] Wayne P. Burleson, Maciej Ciesielski, Fabian Klass, Wentai Liu, "Wave-pipelining: A tutorial and research survey", *IEEE Transactions on VLSI systems*, 6(3): 464-474, 1998.
- [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", *International Symposium on Asynchronous Circuits and Systems*, 188-197, 2000.
- [9] Byoung-Hoon Lim, Jin-Ku Kang, "A self-timed wave pipelined adder using data align method", *Asia-Pacific Conference on ASICs*, 77-80, 2000.
- [10] D. Wong, G. De Micheli, M. Flynn, "Designing high performance digital circuits using wave pipelining: Algorithms and practical experiences", *IEEE Transactions on CAD*, 12(1): 25-46, Jan. 1993.
- [11] Andrew B. Kahng, Sudhakar Muddu, Eginio Sarto, Rahul Sharma, "Interconnect Tuning Strategies for High-Performance ICs", *Design, Automation, and Test in Europe*, 1998.
- [12] "ARM9E Family Flyer", www.arm.com.
- [13] "PowerPC 405 Core White Paper", www.ibm.com.