Wave-Pipelined On-Chip Global Interconnect

Lizheng Zhang
ECE Department,
University of Wisconsin, Madison
lizhengz@cae.wisc.edu

Yuhen Hu
ECE Department,
University of Wisconsin, Madison
hu@engr.wisc.edu

Charlie, Chung-Ping Chen
EE Department,
National Taiwan University
cchen@cc.ee.ntu.edu.tw

***Abstract*— A novel wave-pipelined global interconnect system is developed for reliable, high throughput, on-chip data communication. We argue that because there is only a single signal propagation path and a single type of 1-input gate(inverter), a wave-pipelined interconnect will have less stringent timing constraints than a wave-pipelined combinational logic block. A phase-lock loop based clock and data recovery unit architecture, adopted from off-chip high speed digital serial link, is designed for on-chip application so as to minimize power and area cost. Preliminary Monte Carlo simulation indicated that the wave-pipelined global interconnect architecture potentially can offer 18% higher throughput than a flip-flop pipelined global interconnect architecture at about the same level of reliability. While delivering data through long interconnect at the same bit rate, the wave-pipelined architecture consumes less power and requires less chip real estate.**

I. Introduction

Continuing shrinkage of feature size and ever-increasing SoC (system on chip) design complexity present great challenges to design of high-reliability, high throughput on-chip interconnection circuitry [1]. In an ideal global on-chip interconnection system, large amount of information should be transported across the chip with negligible latency and error. However, design uncertainties in future SoC rise sharply due to coupling noises, process variations, and power delivery fluctuations, among many other factors. The cost of maintaining nearly error-free transmission further compromises the attainable throughput rate. Hence there is an urgent need to explore novel global interconnection system architecture that enables the designers to break-through the performance-reliability envelop.

Repeater insertion is the classical approach to reduce the wire propagation delay [2]–[5]. By properly inserting buffers on the wire, the overall wire delay grows only linearly with respect to the wire length, rather than quadratically. However, as the desired clock operating frequency continue to rise, repeater insertion alone is insufficient to meet the ever increasing performance demands.

The state-of-the-art solution to further enhance the throughput of the interconnect is to insert flip-flops to break a long wire into shorter pipelined stages [6]–[10]. Shorter wire segments means higher clock frequency can be supported, and hence higher throughput rate.

However, due to the overhead of flip-flop insertion, the speed-up of flip flop pipelining(FFP) is not linear. To illustrate, let's assume that a well-buffered wire has a propagation delay $T(\ell)$ that is proportional to the wire length ℓ. Therefore, we would assume $T(\ell/m) = T(\ell)/m$. If we insert $m-1$ flip-flops to break the wire into m pipelined segments, then the total propagation delay of each wire segment will be $\tau_{prop} + T(\ell)/m$ where τ_{prop} is the propagation delay of the flip-flop. Additionally, there is a setup time τ_{setup} constraint for every flip flop. Hence, the clock frequency of the FFP architecture will be

$$\frac{1}{\tau_{prop} + \tau_{setup} + T(\ell)/m} < m \cdot \frac{1}{T(\ell)} \tag{1}$$

which is lower than the m-fold (linear) speed-up one would have hoped.

In this paper, we demonstrate the feasibility of using wave-pipelining (WP) as a novel global interconnect architecture. Wave-pipelining is a circuit technique that has been developed since 1960s ([11]–[14]). Specifically, it advocates the application of a new input signal to a combinational logic block *before* the previous input reaches its intended destination storage elements. As such, multiple waveforms corresponding to successive evaluations co-exist concurrently within the same combinational logic block, mimicking multiple computation *waves*. Hence, the name of *wave-pipelining*. By overlapping computation of successive signals, higher computation throughput can be accomplished.

A major challenge in a wave-pipelined design is to ensure the signal integrity. In a conventional logic block, there are multiple paths and multiple-input gates between inputs and outputs. Furthermore, to safeguard for processing parameter variations, additional timing margin must be reserved. All of these impose some structural and physical constraints which are often too stringent to be practical.

In this paper, we show that, because of the fundamental differences between logic computation and interconnect, timing constraints on a wave-pipelined interconnect architecture can be much relaxed compared to those on a wave-pipelined combinational logic block.

Nonetheless, the nonlinearity of the inserted buffers as well as the adverse effects of thermal noise, cross-talk noise, clock skews and jitters, and process parameter variations, the waveform passing through a wave-pipelined global interconnect may be severely deformed. For this, we propose to incorporate a phase-lock-loop (PLL) clock-and-data recovery (CDR) receiver circuitry to capture and synchronize with the incoming waveform. CDR has been used extensively for

off-chip high speed digital communication subsystems([15]–[17]). To design a PLL CDR for on-chip application, we have sought trade-offs between performance degradation, and power-area consumption. We further conducted preliminary Monte Carlo simulation to compare the reliability, throughput, power and area consumption of a wave-pipelined global interconnect against a baseline FFP architecture that uses identical processing parameters and design methods.

The rest of this paper is organized as follows: In section II, we discuss the timing constraints on WP global interconnect design. In section III, we present the receiver design of a WP global interconnect. In sections IV and V, we experimentally compare the performance, noise immunity, and power consumption between FFP and WP and conclusions are given in section VI.

II. Timing Constraints in Wave Pipelining

A. Timing Constraints in WP Computation

Shown in the left part of Figure 1 is the classical setup of a wave pipelined logic computation: a wave pipelined logic block is sandwiched by input and output flip flops, FF_I and FF_O which are clocked by clock with cycle time of T_{CLK}. Because of the complexity of the logic computation, signals coming out of FF_I can propagate through different paths to reach FF_O. There will be two paths of special importance: the longest path and the shortest path whose delay will be T_{max} and T_{min} respectively. Usually, $T_{max} > T_{min} > T_{CLK}$ and output data can arrive at the FF_O at any time between T_{max} and T_{min} [13].

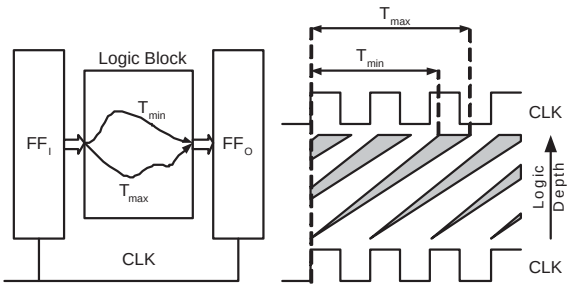

Fig. 1. Timing Constraints of Wave Pipelined Computation

The first timing constraint then comes from the requirement to make sure FF_O can register a valid data at every clock. The number of wave pipelining stages N between FF_I and FF_O have to satisfy:

$$\frac{T_{max}}{T_{CLK}} \leq N < \frac{T_{min}}{T_{CLK}} + 1 \qquad (2)$$

Clearly it is not always possible to find such a stage number N since $T_{max} > T_{min} > T_{CLK}$. In other words, it is not possible to wave pipeline all logic computation system.

The second time constraint, coming from the internal logic gates of the logic block:

> *The next earliest possible wave should NOT arrive at a gate input until the latest possible wave has propagated through.*

It is very hard to satisfy this constraints with the existence of gates with multiple inputs in the logic block: All inputs of a gate have to be synchronized with each other to get correct output. Even if a very good delay analysis and delay balance have been performed on these logic gates, data dependent delay can still fail the system by violating this second constraint.

B. Time Constraints in WP Interconnect

The typical hardware setup and I/O waveforms for wave-pipelined interconnection are shown in Figure 2. Basically Wave pipelining uses the same hardware as uniformly buffer inserted wire to achieve higher performance.

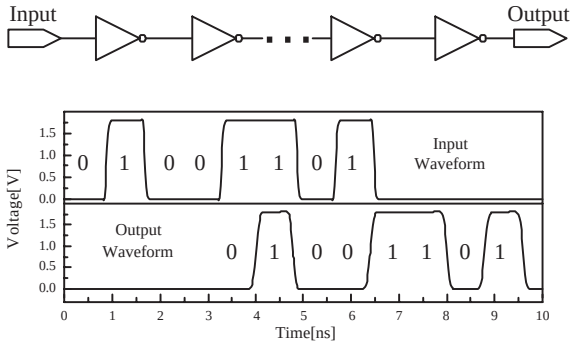

Fig. 2. Wave pipelined Interconnection

The first fundamental difference between wave pipelined computation and interconnect is that the only gate type inside a wave-pipelined interconnect is the inverter which has only one input. This simplification greatly alleviate the effort to satisfy the second timing constraint stated above since no delay balance is needed in internal nodes.

The second fundamental difference between wave pipelined computation and interconnect is that only one possible signal propagation path exists between signal source and destination in interconnect, as shown in Figure 2. So intrinsically, $T_{max} \equiv T_{min}$ and it is then always possible to find a pipelining stage number N to satisfy the Equation (2) for any given clock cycle time T_{CLK}.

C. Noise Effects and Solution

As discussed above, without considering noise effects, the classical arrangement of wave pipelining can be directly applied to interconnect without time constraint violation. But with the inevitable existence of all kinds of noise effects such as process variation, clock skews and jitters, supply voltage fluctuation and thermal uncertainty, classical setup of wave pipelining will potentially fail since these noise effects will cause a difference between the values of T_{max} and T_{min} and violate Equation (2).

The solution to fight against these noise effects is to have a PLL-Based CDR receiver to recover the data from received waveform, as discussed in the following section.

III. Wave Pipelined Interconnect

The interconnect system is treated as a digital communication system in which the wave pipelined interconnect forms the channel connecting the transmitter and receiver. Transmitter and receiver in the WP interconnect system are assumed to be clocked with the same rate, and the bit rate transferred between the transmitter and receiver is the same as the common clock rate.

The transmitter is simply a flip flop and the channel is no more than a uniformly repeater-inserted wire, so the main challenge of the WP interconnect system design comes from the receiver which will recover the data from the received waveform.

Although PLL-based CDR can successfully recover the data sequence from the received data waveform D_{in}(Figure 3), but the generated data sequence D_S only synchronizes to a specific *sampling clock*, CLK_S which may be significantly skewed and jittered from the desired *receiver clock* CLK_R. A FIFO retimer is used then to synchronize the bit stream from CLK_S clock domain to the CLK_R clock domain [18], forming the final output data D_{out}.

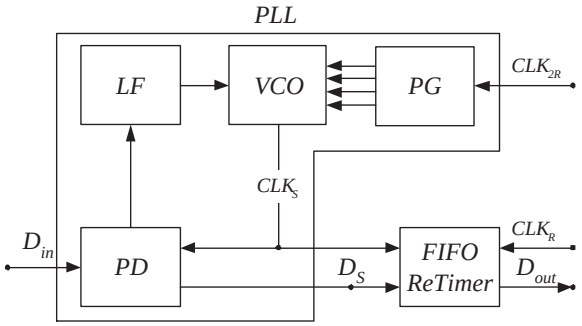

Fig. 3. Receiver of Wave Pipelined Interconnect

A. On-Chip CDR

The PLL-based CDR method has been successfully applied into off-chip communication where it is called *serial link* [16], [17]. But in the context of the on-chip global interconnect, requirements for design are significantly different.

The primary requirement to use PLL for on-chip CDR is that it has to be designed in a way with extremely low cost in area and power consumption. A complicated and sophisticated CDR is affordable in the serial link since there are usually only one or a few lines to be synchronized. But in the case of global on-chip interconnect, to have large CDR logic for every signaling wire is practically impossible since the global on-chip interconnect is usually bus-based and tens of wires are synchronized at the same time. Furthermore, the implementation for on-chip CDR had better be all digital to integrate it easily into a digital system.

Trade-off in noise immunity have to be made to achieve those simplicities. The good news is that, on-chip interconnects usually suffer much less noise than off-chip interconnects so that to reduce the noise immunity to some extent will not seriously impact the overall system's reliability.

B. Phase Resolution

The phase of CLK_S is dynamically adjusted by PLL in stepwise. The biggest step size is then defined as the phase resolution of the PLL, $\Delta\phi$. Phase resolution is critical for a PLL since it decides the noise immunity of the PLL-based CDR. The smaller the $\Delta\phi$, the higher the phase resolution, the better the noise immunity but the more complex the implementation.

In digital PLL, CLK_S could have a total of N_ϕ possible phase value distributed over 0° to 360°. The best phase resolution for PLL is then obtained when all these N_ϕ phases are evenly spaced:

$$\Delta\phi = \frac{360^\circ}{N_\phi} \tag{3}$$

The number of possible phases in digital PLL is usually a power of 2. It is impossible to have a two-phase PLL because the feedback loop is not stable. So a four-phase PLL is selected in the receiver implementation. These four phases used are evenly spaced which implies the phase resolution is 90°.

C. Phase Lock Loop

PLL implemented here is based on phase selection logic. Four clocks, with the same frequency as CLK_R but 90° phase difference among them, are generated by a digital counter from a clock CLK_{2R} whose frequency is two times as that of the receiver clock. This counter is called phase generator(PG).

Among these four clocks from PG, CLK_S is then selected to sample the oncoming data. The selection unit, conventionally called voltage controlled oscillator(VCO), is actually another digital counter assisted by a 4-1 multiplexor.

Since the data sampling happens at the rising edge of CLK_S, the best noise immunity is obtained if the oncoming data transition is aligned with the falling edge of CLK_S. So the phase selection in VCO is based on the goal to achieve the best alignment of the falling edge of CLK_S and oncoming data transitions.

Due to the channel noises, the position of the oncoming data transition will dynamically change so that the phase selection has to be dynamically adjusted to maintain the alignment required by the noise immunity. Alexander phase detector(PD) is the component used for this dynamic phase adjustment. Based on the current relative position between the data transition and the falling edge of CLK_S, PD makes the decision to increase or decrease the phase of CLK_S in order to maintain good alignment.

During operation, PD will possibly generate the phase adjustment signal every clock cycle but each adjustment signal will take three more clock cycles to be effective. To make PLL stable, three out of four phase adjustment signals are filtered out by the loop filter(LF), a 4-state counter.

D. FIFO Retimer

There is no phase agreement between PLL and CLK_R, CLK_S generated by PLL will be significantly jittered and skewed from CLK_R. But since CLK_S and CLK_R are guaranteed to have the same frequency, the phase difference between them is bounded on the upper side: 180° in the worst case.

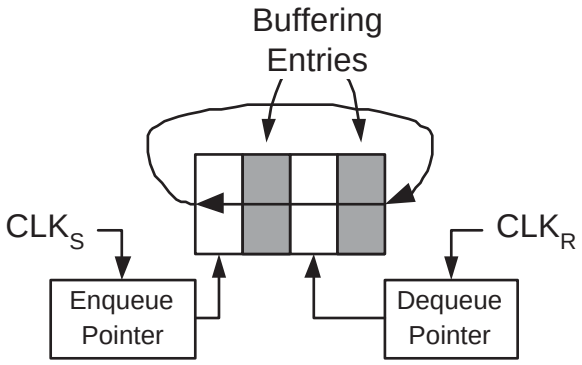

Fig. 4. FIFO Retimer For Wave Pipelined Interconnect

The FIFO queue used for clock domain transformation is a cyclic array in which CLK_S and CLK_R control *enqueue* and *dequeue* operation respectively. The minimum FIFO size is decided by the maximum phase difference between CLK_S and CLK_R. (Figure 4) Because the phase difference between CLK_S and CLK_R will not exceed $180°$, one entry between enqueue and dequeue points is enough to buffer off the phase difference between CLK_S and CLK_R.

However, it is not possible to know which clock, CLK_S or CLK_R, has the leading phase, so the one-entry buffering is needed on both sides of enqueue and dequeue point. So the minimum size of the FIFO is 4.

E. Noise Immunity of Receiver

Because the FIFO retimer is already designed for the worst case, the noise immunity of the overall receiver is then decided by the PLL.

PLL will finally get into a steady state, *locking state*, where it *locks* the oncoming data transition to the falling edge of the CLK_S within the *lock region* around the falling edge of the CLK_S as shown in Figure 5. The half size of the lock region, T_{diff}, is decided by the phase resolution of the PLL.

$$T_{diff} = \frac{\Delta\phi}{360^\circ} T_{CLK} = \frac{T_{CLK}}{N_\phi} \quad (4)$$

where T_{CLK} is the clock cycle time. For a four-phase PLL, $T_{diff} = 0.25T_{CLK}$.

So if the point of time zero($t = 0$) is set at the falling edge of the sampling clock and PLL is in its locking state, the time of current data transition, t_{cur} will satisfy:

$$-T_{diff} \leq t_{cur} \leq T_{diff} \quad (5)$$

Assuming the next data transition is expected to come in one clock cycle, but the real time duration of the current data bit is W_d, the real arrival time of the next data transition will be bounded as follows:

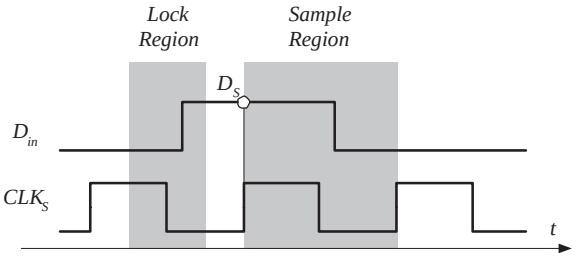

Fig. 5. PLL Timing

$$W_d - T_{diff} \leq t_{nxt} \leq W_d + T_{diff} \quad (6)$$

To make sure that the current data bit be correctly sampled by the rising edge of the current clock cycle, t_{nxt} has to be in the *sample region* shown in Figure 5:

$$0.5T_{CLK} \leq t_{nxt} \leq 1.5T_{CLK} \quad (7)$$

which requires the waveform distortion rate,

$$DR = \frac{|\, W_d - T_{CLK} \,|}{T_{CLK}} \leq 0.5 - \frac{T_{diff}}{T_{CLK}} \quad (8)$$

With the four-phase PLL discussed above, $T_{diff} = 0.25T_{CLK}$ so that the the maximum tolerable distortion rate is 25%.

IV. Noise and Throughput Experimental Analysis

Noise from different sources over a global interconnect may contribute to the demise of signal integrity. These sources include thermal fluctuation, process variations, crosstalk effect and supply voltage uncertainty etc. To evaluate the cumulative effects of these multiple noise sources, we designed a simple test circuits using generic $0.18\mu m$ technology parameter values and simulated using Monte Carlo method. Specifically,

- Temperature fluctuation is set to $50°C$ (Select the worse case between $27°C$ and $77°C$).
- Supply voltage uncertainty is set to be 10% and uniformly distributed.
- Process variation is also assumed as a uniform distribution within the limits provided by technology files.
- Crosstalk is modeled by draw parallel aggressor wires around the victim wire and random signals are passing through those aggressor wires.

A. Noise in Flip Flop Pipelining

A single FFP wire stage is shown in Figure 6 which includes a wire with length of l driven by an inverter driver with size of x and terminated by another inverter acceptor with size of z. Flip flops used to pipeline the interconnect have the same size as the acceptor.

Each pipelined stage of the global interconnect is designed to be identical. The wire length of a single pipelined stage is dictated by the desired clock frequency. Given the length of the wire segment, the optimum sizes for driver and acceptor are

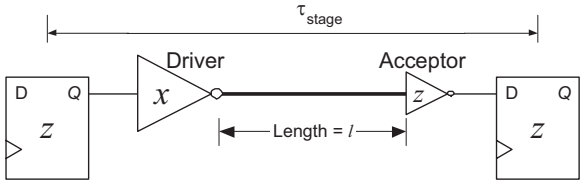

Fig. 6. FFP Wire Stage

subsequently optimized. In $0.18\mu m$ technology, if the stage wire length is $l = 1400\mu m$, the optimum sizes for the driver and the acceptor are found to be $x = 35\mu m$ and $z = 19\mu m$ respectively.

To ensure correct data transfer, the clock cycle T_{CLK} in FFP has to obey the relation:

$$T_{CLK} \quad \geq \quad \tau_{stage} + \Delta_{CLK} \tag{9}$$

where τ_{stage} is the total delay of the pipelining stage including the setup time of the flip flop and Δ_{CLK} is the clock cycle uncertainty because of skews and jitters.

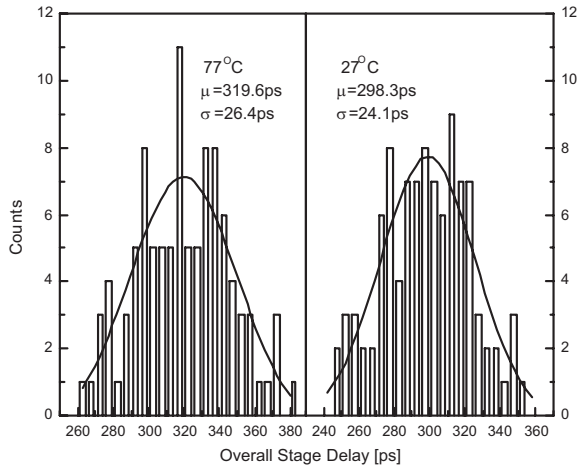

Fig. 7. Noise of FFP Stage

Figure 7 shows the Monte Carlo simulation results of 100 trials for the overall stage delay at different temperatures.

A bit error will happen if Equation (9) is violated. So the bite error rate(BER) will equal to the probability to have $\tau_{stage} + \Delta_{CLK} > T_{CLK}$. Assuming the clock uncertainty Δ_{CLK} is $51ps$ as reported in [19]. With the FFP setup described above, a bit error rate 10^{-15}(1 bit error per five days) is obtained if the clock frequency is $1.7GHz$. The reliable throughput can be improved if the wire length in one pipelining stage is reduced. But for the purpose of comparison, both FFP and WP are set to have the same wire length of $1400\mu m$ in one stage or segment.

B. Noise in Wave Pipelining

One wave pipelining segment is shown in Figure 8 where each segment has a driver inverter with size of s and a wire with length of l. For comparison purpose, the wire length is given the same as that in FFP as $l = 1400\mu m$ and the driver size is then optimized as $s = 31.3\mu m$.

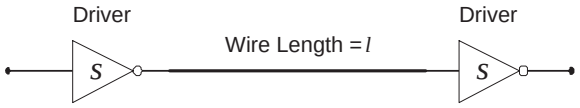

Fig. 8. WP Wire Segment

All segments in WP system are also uniform. Under simulation bit rate of 2GHz, the waveform distortion(DR) across wave-pipelined interconnect with different number of wire segments are evaluated through 100 Monte Carlo repetitions and the statistics of results is shown in Table I.

TABLE I
DR(%) OF WP WITH DIFFERENT LENGTH

Wire Length	$27°C$		$77°C$	
	μ	σ	μ	σ
4 segs	0.178	0.964	0.13	1.112
8 segs	0.35	1.12	0.552	1.272
12 segs	0.64	1.368	0.964	1.548
16 segs	0.904	1.324	1.362	1.516
20 segs	1.292	1.52	1.938	1.788

Although DR does increase when interconnect becomes longer, the value of DR, compared to the maximum tolerable value of the receiver, 25%, is still small even for a long wire with 20 segments(2.8cm).

With WP, clock skew between transmitter and receiver is not harmful. The clock jitter, as also reported in [19], will be 35ps which corresponds to 7% of DR. So probabilistically speaking, the chance for DR of a 20-segment WP wire(2.8cm) to go over the 25% threshold is less than 10^{-15} (1 bit error per 5 days).

So conclusively, to maintain the same noise immunity ($BER \leq 10^{-15}$) with comparable design setup, WP system will have 18% higher throughput than the FFP system.

V. AREA AND POWER CONSUMPTION

The total power consumption of the interconnect system, P_{total} can be broken into three parts:

$$P_{total} = P_{txm} + P_{wire} + P_{rcv} \tag{10}$$

where P_{txm} is transmitter power , $P_{wire}$is wire channel power and P_{rcv} is the receiver power.

So the overall power of a FFP system with N_{stage} pipelining stages is:

$$P_{total} = P_{txm} + N_{stage}P_{stage} + P_{rcv} \tag{11}$$

where P_{stage} is the power consumption of each stage. And that of a WP system with N_{seg} wire segments is:

$$P_{total} = P_{txm} + N_{seg}P_{seg} + P_{rcv} \tag{12}$$

where P_{seg} is the power consumption for each wire segment.

TABLE II

AREA AND POWER CONSUMPTION FOR FFP AND WP

	Wave Pipelining	DFF Pipelining
	Power Consumption	
P_{seg}	1.3mW	–
P_{rcv}	13.1mW	1.1mW
P_{stage}	–	3.9 mW
P_{txm}	1.1mW	1.1mW
	Area	
A_{seg}	$202\mu m^2$	–
A_{rcv}	$2631\mu m^2$	$138\mu m^2$
A_{stage}	–	$856\mu m^2$
A_{txm}	$138\mu m^2$	$138\mu m^2$

Random data bits at the rate of 2GHz have been used to test both FFP and WP systems. WP wire segments and FFP wire stages are designed similarly as those in the noise analysis.

Averaging over 50ns of simulation time, power consumption parameters for both systems are summarized in Table II. From this table, it is clear that the WP will be more power and area efficient than the FFP for a long wire interconnect. The critical length to make WP more power efficient is:

$$l^p_{crit} = \frac{P_{rcv,wp} - P_{rcv,ffp}}{P_{stage} - P_{seg}} \times l = 0.65cm \tag{13}$$

where $l = 1400\mu m$ is the wire length in one stage of FFP or one segment of WP. So, with the stated above interconnect design and with bit rate of 2GHz, WP will give better power efficiency if the wire is longer than 0.65cm.

Similarly, the critical length l^a_{crit} which makes either FFP or WP more area efficient is calculated from parameters shown in Table II :

$$l^a_{crit} = \frac{A_{rcv,wp} - A_{rcv,ffp}}{A_{stage} - A_{seg}} \times l = 0.56cm \tag{14}$$

where $A_{rcv,wp}$ and $A_{rcv,ffp}$ are receiver area for WP and FFP, A_{stage} and A_{seg} are area for wire stage in FFP and wire segment in WP. So WP will leads to more area efficiency if the wire length is longer than 0.56cm.

VI. CONCLUSIONS

A new method of using wave pipelining(WP) for high throughput synchronous global interconnect is proposed. The feasibility for WP interconnect is demonstrated. Using on-chip CDR as the synchronization method is presented and verified.

Compared to the state-of-the-art solution of flip flop pipelining(FFP), WP presents 18% better performance with the same robustness. In terms of power and area consumed for communication, FFP performs better in short interconnect while WP is better for long interconnect. In $0.18\mu m$ technology and 2GHz of bit rate, 0.65cm is the critical length which makes either FFP or WP more power efficient whereas the area efficiency switches at shorter length of 0.56cm.

Wave pipelinin is beneficial in many other ways. For example, the clock distribution effort, which is one major challenge for modern digital design, can be significantly alleviated because the clock skew is not harmful among blocks communicated through wave pipelined interconnect.

REFERENCES

[1] S. I. Association, "International technology roadmap for semiconductors," Semiconductor Industry Association, Tech. Rep., 2001.

[2] H. Shah, P. Shin, B. Bell, M. Aldredge, N. Sopory, and J. Davis, "Repeater insertion and wire sizing optimization for throughput-centric vlsi global interconnects," *IEEE/ACM International Conference on Computer Aided Design(ICCAD 2002)*, pp. 280 –284, 2002.

[3] K. Banerjee and A. Mehrotra, "A power-optimal repeater insertion methodology for global interconnects in nanometer designs," *IEEE Transactions on Electron Devices*, vol. 49, no. 11, pp. 2001 –2007, Nov 2002.

[4] S. Srinivasaraghavan and W. Burleson, "Interconnect effort -a unification of repeater insertion and logical effort," *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, pp. 55 –61, Feb 2003.

[5] C. Chu and D. F. Wong, "Closed form solutions to simultaneous buffer insertion/sizing and wire sizing," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 6, no. 3, pp. 343–371, July 2001.

[6] S. Hassoun, C. J. Alpert, and M. Thiagarajan, "Optimal buffered routing path constructions for single and multiple clock domain systems," *Computer Aided Design, 2002. ICCAD 2002. IEEE/ACM International Conference on*, pp. 247 –253, Nov 2002.

[7] L. Scheffer, "Methodologies and tools for pipelined on-chip interconnect," *IEEE International Conference on Computer Design(ICCD)*, 2002.

[8] L. P. Carloni, K. L. McMillan, and A. L. Sangiovanni-Vincentelli, "Theory of latency-insensitive design," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 20, no. 9, pp. 1059 –1076, Sept 2001.

[9] P. Cocchini, "Concurrent flip-flop and repeater insertion for high performance integrated circuits," *IEEE/ACM International Conference on Computer Aided Design(ICCAD)*, pp. 268 –273, 2002.

[10] R. Lu, G. Zhong, C.-K. Koh, and K.-Y. Chao, "Flip-flop and repeater insertion for early interconnect planning," *Proceedings of Design, Automation and Test in Europe Conference and Exhibition, 2002*, pp. 690 –695, March 2002.

[11] S. Anderson, J. Earle, R. Goldschmidt, and D. Powers, "The ibm system/360 model 91 floating point execution unit," *IBM J. Res. Develop.*, Jan 1967.

[12] L. Cotten, "Maximum rate pipelined systems," *Proc. AFIPS Spring Joint Comput. Conf.*, 1969.

[13] W. P. Burleson, M. Ciesielski, F. Klass, and W. Liu, "Wave-pipelining: a tutorial and research survey," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 3, pp. 464–474, Sep 1998.

[14] O. Hauck and S. A. Huss, "Asynchronous wave pipelines for high throughput datapaths," *1998 IEEE International Conference on Electronics, Circuits and Systems*, vol. 1, pp. 283–286, 1998.

[15] B. Razavi, "Challenges in the design high-speed clock and data recovery circuits," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 94 –101, Aug 2002.

[16] J. Teifel and R. Manohar, "A high-speed clockless serial link transceiver," *Asynchronous Circuits and Systems, 2003. Proceedings. Ninth International Symposium on*, pp. 151 –161, May 2003.

[17] V. Stojanovic and M. Horowitz, "Modeling and analysis of high speed links," *Custom Integrated Circuits Conference*, Sept 2003.

[18] T. Chelcea and S. M. Nowick, "Robust interfaces for mixed-timing systems with application to latency-insensitive protocols," *Proceedings of the 38th conference on Design automation*, pp. 21–26, June 2001.

[19] N. A. Kurd, J. S. Barkatullah, R. O. Dizon, T. D. Fletcher, and P. D. Madland, "A multigigahertz clocking scheme for the pentium(r) 4 microprocessor," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 11, pp. 1647 –1653, Nov. 2001.