Wave-Pipelining the Global Interconnect to Reduce the Associated Delays

Jabulani Nyathi, Ray Robert Rydberg III and José G. Delgado-Frias
Washington State University
School of EECS
Pullman, Washington, USA
jabu@eecs.wsu.edu

Abstract—**The majority of digital circuits/systems primarily use synchronous clocking methodology. With clock distribution networks dissipating ever more power and the wire delays expected to become dominant, there has been increased activity to provide alternative solutions. This paper explores some potential methods for reducing global interconnect delays and improving throughput between communicating modules. Analysis of the classical repeater insertion is performed and a wave-pipelined repeater insertion scheme that addresses some shortfalls of the classical repeater insertion is proposed. An extension of the wave-pipelined repeater insertion scheme is presented and results show that its data retention capability offers reliable communication between any number of computing elements. The design of the communication channel is based on the assumption that the computing elements employ synchronous clocking while the communication channels are driven by locally generated clocks. Locally generating clocks along the communication channel avoids the clock distribution complexities and offers an ability to stop and start data transfer along the channel without the need for elaborate clock gating circuitry. Furthermore, no additional clock cycles are required to flush the pipe in the event of stalls. The circuitry that generates local clocks increases area and power, but shows significant performance advantages, particularly in providing a seamless interface between communicating modules running at different clock frequencies. Simulation results of the distributed FIFO communication channel in a modest 180 nm technology show locally generated clocks running at 2.22GHz with the memory buffers placed 2 mm apart. The handshaking interchange between neighboring control circuits shows an average delay of 285.35 ps.**

I. INTRODUCTION

Global on-chip interconnects are increasingly becoming a limiting performance factor in highly integrated systems such as system-on-chip (SoC). Some of the reasons leading to global interconnects being a limiting factor in system performance include: power supply drop variations, process variations, single clock synchronization [1], large wires with unpredictable delays [2], and interconnect power dissipation [3]. These limitations also imply that it would be difficult to achieve correct functional and reliable operations while maintaining low energy consumption within the interacting modules or components of SoC [3]. Proposed solutions to alleviate these limitations include: separating the computation problem from the communication problem and introducing networks on chips [3], having communicating components operate asynchronously, while the computational blocks operate synchronously based on locally generated clocks [4] and optimizing repeater insertion for global interconnect [5].

This work is based on the premise that global wires spanning a significant fraction of the chip will impose signal delays that exceed the clock period [1]. Additionally, synchronizing operations on components running at different clock speeds is becoming more difficult due to clock skew and distribution [3]. With wire delays that exceed the clock period, it becomes apparent that the interacting components in a SoC design will perform computations much faster than the results could be transferred between components (a process that now requires multiple clock cycles). This assertion can further be substantiated by the fact that transistor switching speeds are much faster than wire delays. It is stated in [1] that as the transistor switching speeds improve, the wire delays increase.

Much work has been done at the architectural, system and circuit levels to present some solutions to the various on chip communication issues. Some examples include: bus splitting, router based communication architectures and system level techniques such as communication based power management and adaptive supply voltage links [6]. In [3]

1-4244-0173-9/06/$20.00 ©2006 IEEE.

the interacting components are viewed as a micronetwork with communication taking place among the components. This view allows for the replacement of the global wiring on a chip with a general purpose interconnection network. There are also circuit level techniques that have been explored to deal with the wire delay issue. The most widely used being the classical two inverter configuration with intermediate repeaters used for longer transmission distances. This paper explores the feasibility of applying the wave-pipelining technique to the classical repeater insertion approach and further examines the elimination of the global clock signals, replacing them with local clocks.

II. Reducing Interconnect Delays

There has been a number of approaches that have been employed to deal with the interconnect delays and the classical repeater insertion is the widely used approach. In the following sub-sections, a brief summary of some the work on repeater insertion is presented and the scheme that improves the channel's throughput introduced.

A. The classical Repeater Insertion

It is now an accepted fact that transistors scale faster than the wires used to transmit signals within a chip, hence wires dominate circuits/system delay. Buffer insertion has been a widely used method for reducing the dominant wire delays. Existing literature shows that considerable work on repeater insertion at the algorithmic, as well as circuit, levels has been done [7], [8]. The published work presents approaches for optimizing current or voltage drivers and determining optimal distances at which buffers can be inserted. Power dissipation and optimization for various interconnect lengths have been reported in [9]. In the nano-meter regime, repeater insertion will not cope with increased disturbances caused by parasitic effects. To this end, Kaul and Sylvester have proposed a transition aware global signaling scheme and report clock rates of 1.5 GHz in 130 nm technology for 8 mm long wires [10]. It is instructive to consider pipelining unrelated data waves as signals propagate from the sender module to the consumer module. There are several ways this could be done and one approach is the wire pipelining scheme [11]. This paper proposes and simulates some of the schemes that could be used to enable for data pipelining along the channel. In order to provide a sense of the delays being considered device and interconnect delays along with parasitic capacitance values at the 180 nm technology node are presented first. The simulations show a loadless (only intrinsic capacitance constitute the load) unit sized inverter switching at approximately 24 ps. Adding a 3 μm by 3 μm wire results in a 42% degradation in switching speed. Table I shows how fast the inverter switching speed degrades with increasing wire length. This demonstrates nothing new since these are established facts however this reality calls for an intervention.

TABLE I. Wire Parasitics and Delays

Structure (@ 180 nm)	Capacitance (fF)	Propagation Delay (ps)
Unit Sized Inverter	Loadless	24
Inverter pair with 1 mm $metal_1$ wire	123.9	228.5
Inverter pair with 2 mm $metal_1$ wire	247.1	392.8
Inverter pair with 4 mm $metal_1$ wire	493.5	711.5
Inverter pair with 8 mm $metal_1$ wire	986.3	1338.8

The structure measured is representative of the buffer insertion scheme widely used to reduce skew and manage delays on long wires. The pair of inverters is appropriately sized and intended to break the RC delay of the wire. Here the inverter pair has been used for the express purpose of demonstrating the dominance of wire delays. A transistor whose diffusion area has similar dimensions to those of a metal level 1 piece of wire switches faster than the wire as evidenced by the loadless inverter propagation delay compared to that of an inverter pair driving 1 mm of wire. The classical inverter insertion leads to a reduction of the wire delays however the communicating modules could remain faster than the channel restricting data transfer rates. In addition the classical inverter insertion lacks memory and cannot tolerate changes resulting from temperature effects or voltage fluctuations.

B. Wave Pipelining Classical Repeater

The buffer insertion approach can easily support wave-pipelining. The wave-pipelining [12], technique has no intermediate latches for temporary synchronization of data at intermediate nodes, but sustains multiple unrelated data waves propagating along the channel as shown in Fig. 1. The fact that there are unrelated data waves implies that there is a possibility of data overrun especially when a wave's path has shorter and longer (critical) paths. $Wave_1$'s critical path can easily be overtaken by $wave_2$'s shortest path. The buffer insertion represented in Fig 1 has equal paths and thus data overrun can occur only if the valid frequency regions of operation are violated. The data can therefore propagate through the different stages of the channel at the same rate and no delay balancing is required. Long wires with delays much larger than the transistor switching speed and which require multiple clock cycles to transmit a single data wave along the channel between communicating modules can employ the wave-pipelining approach. The clock suffers from the same effects as the data being transmitted and would require an elaborate clock managing scheme. In this exercise, the assumption is that the clock distribution scheme is established. Another assumption made is that the communicating modules are driven by the same clock but to accommodate clock skew the figure shows the clocks labeled as CLK_1 and CLK_2. This simply denotes the differing arrival times of the same clock.

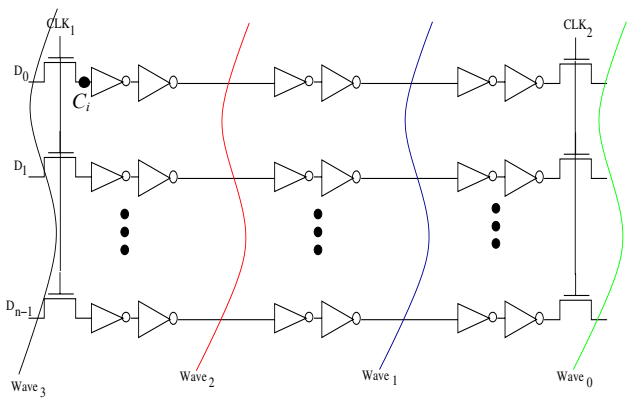

Figure 1. Classic inverter insertion with Wave Pipelining

The clock's (CLK_1) arrival times are such that there is at a minimum two inverter delays between successive pulses. This allows the current data to advance by a safe distance in time before the new data wave is admitted into the pipe. If this time interval is violated with the clock arriving earlier than two inverter delays there would be data overrun. A slower clock than this results in fewer data waves within the channel or a single wave depending on how slow the clock is. The clock pulse is determined by the delay required to charge or discharge the node labeled C_i on Fig. 1. In this experiment three inverter pairs were used implying that the maximum number of unrelated data waves that can be sustained within such a channel is three. Table 2 provides the clock periods required to sustain three, two or one wave within the channel.

TABLE II. DISTANCE IN TIME BETWEEN SUCCESSIVE DATA WAVES

Range of Clock Period (ps)	Number of Waves	Wire Length Between Buffers
250—350	3	1 mm
420—800	2	1 mm
1070 and Slower	1	1 mm

The values of Table II reflect delays associated with drivers being placed 1 mm apart. This is not optimized spacing and is intended as proof of concept. Traces of simulations of the wave-pipelined buffer insertion scheme of Fig 1 appear in Fig. 2. In addition to being able to break the RC delay of long interconnects the scheme supports multiple data waves and thus improves throughput. The latency becomes a function of the number of buffers along the channel. The data transfer rate once the pipe is full is such that a data item is output every cycle. Given the fact that the inverter devices can switch as fast as the devices of the modules it is feasible to have the channel transfer data at the same rate as the modules produce/consume the data. The traces show an instance when the unrelated data waves within the channel for the first data line are: $wave_0 = 0$, $wave_2 = 1$ and $wave_3 = 0$.

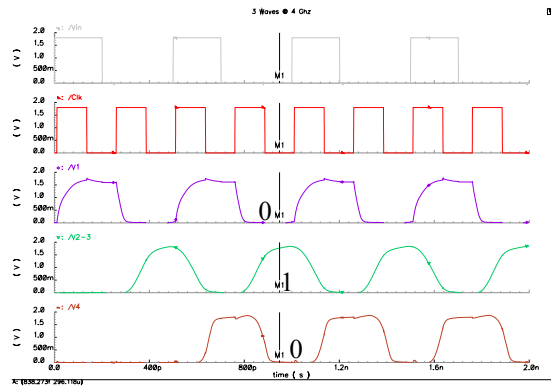

Figure 2. Simulation results of Wave Pipelined scheme

Both the classical inverter pair and the modified wave-pipelined scheme lack memory. To avoid data overrun, the component producing data has to operate at the rate the signals can propagate through the inverter stages when the classical inverter insertion scheme is used. The computing components, if far apart, have to perform operations at slower speeds to allow for slow global transfers through the global wires. The repeaters are inserted to break the wire delays. The wave-pipelined approach offers improved throughput but must be operated at appropriate time intervals. Another issue of concern with the wave-pipelined scheme is that if there are hazards of any sort (e.g. some stage having a delay that differs from that calculated due to environmental or process variations) data overrun would occur. This would occur if the delay falls outside the valid clock period range. There is also no method to correlate each wave in the channel with its associated clock pulse. The intermediate nodes are not observable, requiring many clock cycles to pinpoint a problem in the event of failure. In both schemes inter-wire capacitive coupling can be reduced by staggering each data line further reducing the clock cycle time.

If the communicating modules are running at different clock frequencies the presented methods require additional circuitry for synchronization. There might be need for multiple clock cycles for successful transfer of data implying that the sending component must accommodate this constraint to avoid data overrun. Another drawback is the need for an elaborate clock distribution scheme that is likely to contribute significantly to power dissipation.

It is proposed in this paper that the repeater insertion scheme be given memory. This effectively introduces a distributed first in first out (FIFO) buffering scheme that enables the communicating components to start and stop with minimal data loss risks and tolerate process and environmental parameter variations.

C. *The Classical Inverter Insertion with Memory*

To make less the problems outlined as persistent when using the wave-pipelined classical buffer insertion a shift register is implemented. The shift register takes the form of

a first in first out (FIFO) buffer and is distributed along the channel since it is intended to replace the buffer insertion method. The approach is given credence by the fact that the communicating modules will almost always be associated with storage buffers concentrated at the inputs and/or outputs. In this study these buffers are distributed along the communication channel to facilitate ease of communication. Fig 3 shows a schematic of a two stage FIFO with control signals $enable_1$ and $enable_2$. Ideally the control signals could be considered as clocks that shift data from one stage to the next. In this paper these signals are local clocks generated based on the sending module's clock in conjunction with the channel status as dictated by the receiving module's rate of data consumption. Generating local clocks will alleviate the problem of clock distribution and could potentially lead to ease of synchronization if the communicating components are operating at different frequencies.

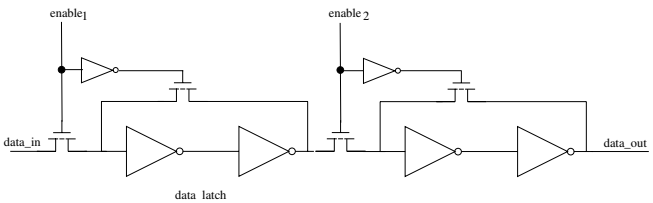

Figure 3. The classical inverter insertion with memory

III. GENARATING THE LOCAL CLOCK SIGNALS

The communicating modules are assumed to be far apart requiring global signals to communicate. Shifting data from one stage of Fig 3 to the next requires global clocking. Global signals require long wires. To break the RC delays of these wires buffer insertion has been widely accepted as the basic method. It has been shown in Section II that wave-pipelining can improve the throughput of the classical repeater insertion, but the scheme has some shortfalls. There are several approaches proposed to reduce the global interconnect delays. The globally asynchronous locally synchronous approach (GALS) [4] has gained prominence largely because the approach has appeal as a potential solution to the multiple clock domain synchronization issue of system on chip design. Interfacing the modules running at different clock frequencies has seen a rise in the number of proposed interfacing circuits. In this study the communicating modules are assumed to be synchronous and the communication channel controlled by local clocks ($enable_1$ and $enable_2$) in much the same way the GALS approach operates. The generation of the local clocks requires intricate control circuitry and to meet this design requirement the GasP [13] control circuit has been adopted albeit with some modifications. Fig 3 shows the FIFO control circuit proposed in [13] and is duplicated in this study to enable for ease of analysis.

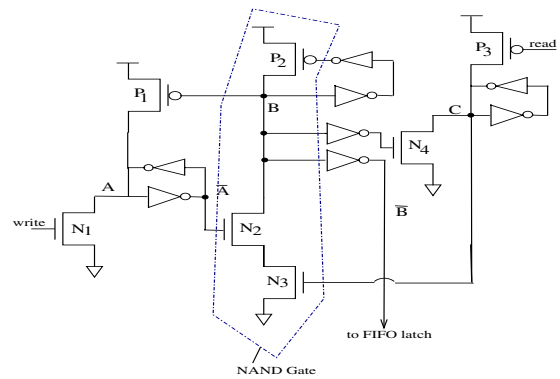

Figure 4. Minimal FIFO Control Circuit

Details of the circuit's operation can be found in [13] and are thus not repeated in this paper. The modifications performed on the original control circuit are motivated by the need to keep all the generated signals of interest synchronized. Node B in Fig. 4 shows the NAND gate having a fan-out of 3 inverters. These inverters are not identical since they drive widely differing loads. If the delay through the two inverters designed to reset the NAND gate is disturbed there is a possibility that the NAND output can be reset either too early or too late, affecting the duration of the enable pulse or status of node C. Admittedly appropriate sizing for the reset loop can be achieved based on the worst case delay path, however this leads to a requirement that the load capacitance of the enable signal remain fixed else any changes would see these signals getting out of synch. Granted, this is an asynchronous design, there should be no concerns about the nodes operating at different rates and affecting performance, but there is no built in control at this node to allow for asynchrony.

The NAND gate has been re-designed to have a fan-out of 1, leading to a case where the AND gate now has a fan-out of 3. There is still a potential to have the NAND output resetting early while the enable and status signals have not completed switching. This problem is minimized by designing for the worst case loading of the enable signals. The NAND with a fan-out of 3 shows variations in delay that range from 14 ps to 35 ps. This implies that once N_2 and N_3 are ON and pull Node B to logic 0, within 14 ps Node B can be reset back to logic 1. This might not be a long enough time to enable Node $\overline{B}$ to drive its load to the desired steady state values. A 35 ps delay on the other hand might be way too long causing new requests to be slower than average. The AND on the other hand remains within 1% of the base delay even in the event that the load increases.

The control circuits of Fig 4 if cascaded such that they abut would transfer data at a frequency of 3.5 GHz. This arrangement would impact power and area significantly. In order to maintain a fair comparison with the classical repeater insertion approach, save both area and power, the neighboring control circuits have been placed 2 mm apart.

With this arrangement data is transferred at a frequency of 2.22 GHz. Each control circuit sinks 26.36 µA per bit at peak transfer rates. Such a high value for current implies significant power dissipation and work is in progress to reduce the power.

Fig. 5 shows three local clock signals that result when the sending module issues a request to send once and the channel is ready to transfer data based on the receiving module's availability. The request pulse is generated based on the status of the channel and supports burst mode operation. The control circuits are placed 2 mm apart for this simulation and each enable signal drives 16-bits of the data bus. The generation of these local clocks eliminates the need for large clock networks and the control circuitry produces the pulses on demand and thus remains idle without employing complex clock gating schemes.

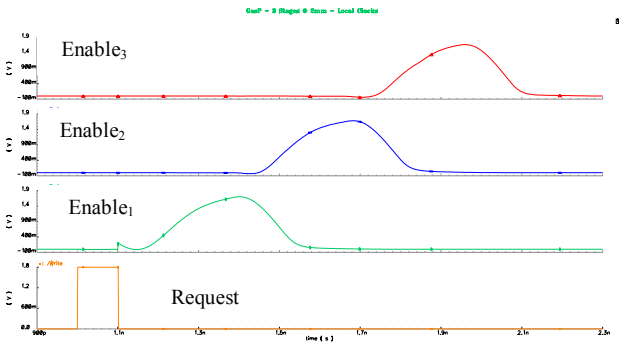

Figure 5. Local clocks propagating a single data item along the channel

Another aspect of the FIFO buffer being controlled is that it tolerates voltage changes well, retaining correct data even when the power supply voltage is dropped to 400 mV. The data being transmitted remains intact under these low voltages, but the transfer rate degrades significantly. The current experiments consider only a case in which the communicating modules operate at the same frequency, and aim to show the benefits of having local clocks replace the global signals. It is envisioned that multiple clock domains can easily be synchronized with the distributed FIFO communication scheme. This could be achieved by providing the buffer status to both the sender and the receiver. At any given time the communicating modules can determine if they can access the channel or not. If the channel has capacity the intending component can transfer data. On the other hand if the channel is full the sending component cannot access it. Only the receiver can retrieve data when ready. Nodes B and C (Fig 4) of the first and last buffer control circuits have signals that indicate the status of the channel i.e. whether it has capacity or not.

IV. Concluding Remarks

This paper presents a wave-pipelining scheme for reducing global wire delays. Computing elements that need to communicate are assumed to be operating at the same frequency and the wave-pipelined classical repeater insertion is shown to have a significant increase in throughput. Giving the classical repeater insertion memory allows for a distributed FIFO communication channel that retains appropriate voltage levels even under uncertain conditions. Local clocks are generated and maintain a channel speed of 2.22 GHz with the neighboring clock generating circuitry placed 2 mm away. A 16-bit bus is driven by each enable signal (local clock) without any degradation in speed. The approach provides for the elimination of the global clocks, since each stage of the control circuitry becomes self-timed. The most important aspect of this scheme, is that the global interconnects can now allow data to propagate to the intended component at speeds limited only by the logic. Interconnects now have memory, permitting the communicating components to temporarily store the results on the data paths if necessary.

References

[1] L. P. Carloni and A. L. Sangiovani-Vincentelli, "Coping with Latency in SoC Design," *IEEE Micro*, October 2002, pp. 24-35.

[2] R. Seigmund and D. Muller, "Efficient Modeling and Synthesis of On-Chip Communication Protocols for Network-On-Chip Design," *Proceedings of the 2003 International Symposium on Circuits and Systems,* Vol 5, May 25-28, 2003, pp. 81-84.

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

[4] A. Iyer and D. Marculescu, "Power and Performance Evaluation of Globally Asynchronous Locally Synchronous Processors," *29th Annual International Symposium on Computer Architecture,* May 25-29, 2002, pp. 158-168.

[5] V. V. Deodhar and J. A. Davis, "Voltage Scaling and Repeater Insertion for High-Throughput Low-Power Interconnects," *Proceedings of the 2003 International Symposium on Circuits and Systems*, Vol 5, May 25-28, 2003, pp. 349-352.

[6] V. Raghunathan, M. B. Srivastava and R. K. Gupta, "A Survey of Techniques for Energy Efficient On-Chip Communication," *Proceedings of Design Automation Conference,* June 2-6, 2003, pp. 900-905.

[7] M. L. Mui, K. Banerjee and A. Mehrotra, "Global Interconnect Optimization Scheme for Nanometer Scale Dissipation," *IEEE Transactions on Electron Devices,* Vol. 51, February 2004, pp. 195-203.

[8] R. Bashirullah, W. Liu, and R. K. Cavin III, "Current-Mode Signaling in Deep Submicrometer Global Interconnects," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 11, No. 3, June 2003, pp. 406-417.

[9] 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, November 2002, pp. 2001-2007.

[10] H. Kaul and D. Sylvester, "Low-Power On-Chip Communication Based on Transition-Aware Global Signaling (TAGS)," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 12, No. 5, May 2004, 464-476.

[11] M. R. Casu and L. Macchiarulo, "On Chip Transparent Wire Pipelining," IEEE International Conference on Computer Design, October 11-13,2004, pp. 160-167.

[12] W. P. Burleson, M. Ciesielski, F. Klass and W. Liu, "Wave-Pipelining: A Tutorial and Research Survey," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol. 6, No. 3, pp. 464 - 474, September 1998.

[13] I. Sutherland and S. Fairbanks, "GasP: A Minimal FIFO Control," *Proc. of ASYNC*, 2001, pp. 46-53.