Hindawi Publishing Corporation
VLSI Design
Volume 2007, Article ID 95402, 12 pages
doi:10.1155/2007/95402

Research Article

Variation-Tolerant and Low-Power Source-Synchronous Multicycle On-Chip Interconnect Scheme

Maged Ghoneima,[1] Yehea Ismail,[2] Muhammad Khellah,[3] and Vivek De[3]

[1] *VLSI Design Group, NVIDIA Corporation, Santa Clara, CA 95050, USA*
[2] *Electrical Enginering and Computer Science Department (EECS), Northwestern University, Evanston, IL 60208-3118, USA*
[3] *Circuit Research Laboratories, Intel Corporation, Hillsboro, OR 97124, USA*

Received 6 November 2006; Revised 27 February 2007; Accepted 16 March 2007

Recommended by Davide Bertozzi

A variation-tolerant low-power source-synchronous multicycle (SSMC) interconnect scheme is proposed. This scheme is scalable and suitable for transferring data across different clock domains such as those in "many-core" SoCs and in 3D-ICs. SSMC replaces intermediate flip-flops by a source-synchronous synchronization scheme. Removing the intermediate flip-flops in the SSMC scheme enables better averaging of delay variations across the whole interconnect, which reduces bit-rate degradation due to within-die WID process variations. Monte Carlo circuit simulations show that SSMC eliminates 90% of the variation-induced performance degradation in a 6-cycle 9 mm-long 16-bit conventional bus. The proposed multicycle bus scheme also leads to significant energy savings due to eliminating the power-hungry flip-flops and efficiently designing the source synchronization overhead. Moreover, eliminating intermediate flip-flops avoids the timing overhead of the setup time, the flip-flop delay, and the single-cycle clock jitter. This delay slack can then be translated into further energy savings by downsizing the repeaters. The significant delay jitter due to capacitive coupling has been addressed and solutions are put forward to alleviate it. Circuit simulations in a 65-nm process environment indicate that energy savings up to 20% are achievable for a 6-cycle 9 mm long 16-bit bus.

1. INTRODUCTION

Today's system-on-ship (SoC) devices are targeting complex applications, where there is a need for a significant amount of computing power and data transfer. This implies that the number of on-chip modules will increase, and so will the number of on-chip buses connecting these modules. With the continuous scaling of technology, increased die area and faster clock speeds, the delay and power dissipation of on-chip buses are becoming one of the main bottlenecks in current high-performance SoC design [1].

As technology scales and die sizes increase, gate delay improves, but on the contrary, interconnect delay is increasing. This is mainly due to the increase in interconnect length, and due to increasing coupling capacitance which is increasing to dominate the overall interconnect capacitance [2]. Previously, the maximum clock frequency was determined by the longest interconnect, as data was expected to propagate along the entire interconnect length in a single clock cycle.

In order to decouple the maximum system clock frequency from the interconnect delay, latency-insensitive interconnect schemes, such as the multicycle interconnect scheme, were proposed [3–6]. In this multicycle scheme, shown in Figure 1, interconnects are partitioned into segments bounded by flip-flops. The maximum segment length is chosen to satisfy the desired clock speed of the design. Thus, the maximum clock frequency is bounded by the longest segment, and not the whole interconnect. Thus, in this conventional multicycle (CMC) interconnect scheme, the notion of interconnect delay is transformed into interconnect latency, which is measured in clock cycles.

In the CMC scheme, flip-flops are inserted between the interconnect segments to ensure that the interconnect latency is independent of the clock period. This is important to ensure correct system functionality during low-frequency boundary-scan testing [7]. However, these flip-flops are power-hungry cells that dissipate about 20% of the overall bus power dissipation [8]. Moreover, these flip-flops insert a

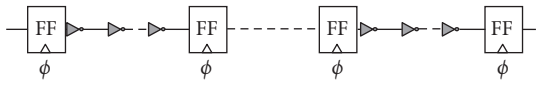

Figure 1: Conventional multicycle interconnect scheme.

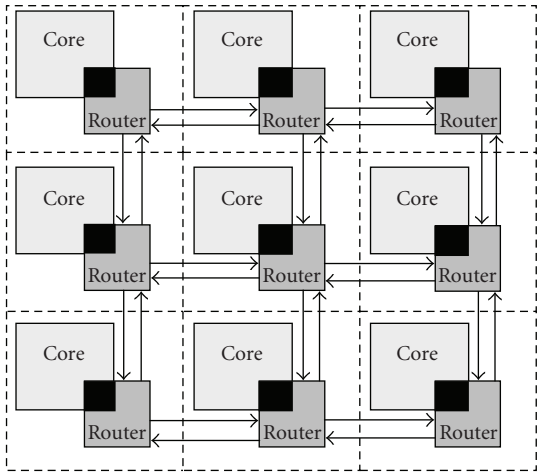

Figure 2: Many-core platform with a network-on-chip (NoC) interconnect fabric.

delay overhead for *each* clock cycle, which leads to upsized repeaters and a higher energy dissipation. The overall clock period T has to encompass all the delay components, such that

$$T \geq t_d + t_{\text{flop}} + t_{\text{setup}} + t_{\text{cycle-jitter}}, \quad (1)$$

where t_d is the interconnect delay (including repeaters), t_{flop} is the flip-flop delay, t_{setup} is the setup delay, and $t_{\text{cycle-jitter}}$ is the clock single-cycle jitter. Moreover, intermediate flip-flops disable cycle-sharing, which makes the CMC scheme very sensitive to process variations.

Thus, an interconnect scheme without intermediate flip-flops is required to reduce the power dissipation and variation sensitivity. At the same time, a certain synchronization scheme is still required to maintain the fixed interconnect latency, and keep it independent of the clock period. In this paper, we propose a source-synchronous multicycle (SSMC) bus scheme with a period-independent latency. The SSMC scheme eliminates the energy dissipation of the intermediate flip-flops and increases the interconnect variation tolerance. The SSMC also appears to be very suitable for *serial links* used for intercore communication in the many-core platform [9], which contains multiple cores, each in a different mesochronous clock domain, as shown in Figure 2. It can be considered a globally asynchronous locally synchronous (GALS) interconnect scheme [10], which solves synchronization issues when crossing different clock domain boundaries. It will also be useful for interdie communication in 3D die stacks [11, 12], shown in Figure 3, because of the limitations imposed on where flip-flops can be added to the vertical through-silicon vias (TSV).

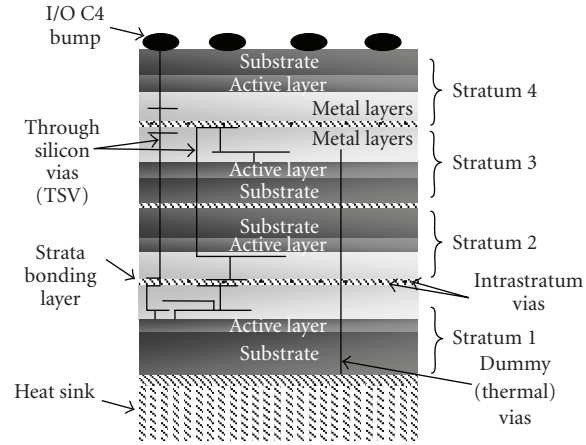

Figure 3: 3D die stack.

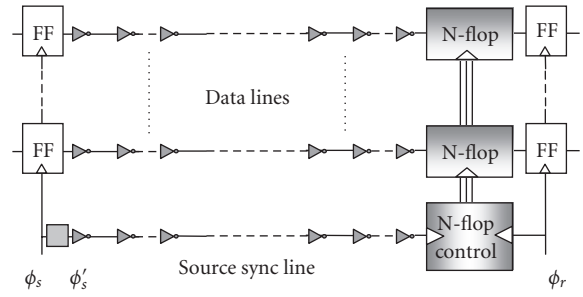

Figure 4: Proposed SSMC scheme.

The rest of the paper is organized as follows. Section 2 presents the proposed SSMC bus scheme in detail, where system constraints are derived and discussed. Section 3 provides the circuit simulation setup and results. Finally, Section 4 provides the conclusion.

2. ON-CHIP SOURCE-SYNCHRONOUS MULTICYCLE INTERCONNECT SCHEME

In this section, the architecture of the proposed source-synchronous multicycle (SSMC) bus architecture and each of its components will be explained. The operation of this scheme will be illustrated, and system design constraints will also be derived and discussed. It should be noted that even though the concept of the proposed architecture is very similar to that used for off-chip communication, many differences exist due to the difference in nature between off-chip and on-chip interconnects. On-chip interconnects are primarily RC dominated and strongly capacitively coupled as opposed to the LC and widely spaced off-chip interconnects.

2.1. SSMC architecture

A schematic of the proposed SSMC scheme is shown in Figure 4. The data lines of the proposed SSMC scheme are exactly the same as that of a conventional CMC scheme, except that the intermediate flip-flops are replaced by a synchronizing element, an N-flop, at the receiver end of the line. The

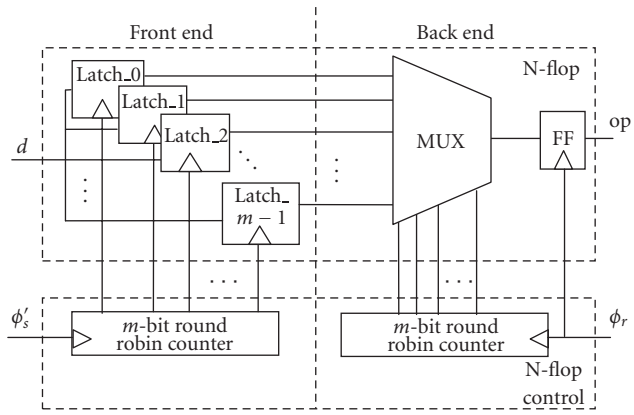

Figure 5: Schematic of an N-flop and the N-flop control.

N-flop is an element that synchronizes the receiver with the transmitter, such that it provides a fixed latency (n-cycles) irrespective of the cycle period T. The N-flop synchronizes the received data using timing information extracted from both the receiver and transmitter clock signals.

Transmitter clock timing information can be delivered to the N-flop in one of two ways: (1) send the transmitter clock signal on a separate line alongside the data, or (2) extract transmitter clock timing information from the received data (clock recovery). The main disadvantage of the first method is the high energy dissipation of sending the clock signal along a dedicated line. However, using some modifications, explained in Section 2.7, the energy dissipation of the source synchronizing line can be significantly reduced. Moreover, as the energy dissipation of this line will be amortized over the number of data bus lines, the energy overhead of this extra line will be considerably small. On the contrary, the second method requires a clock recovery circuit for each bus line, which is a large area and power overhead. Thus, the first option of adding a single separate line to the bus carrying the transmitter clock signal was implemented in the SSMC described in this paper, as shown in Figure 4.

2.2. N-flop

The N-flop is basically a circuit that synchronizes the data sent from a certain clock domain (ϕ_s) with the clock of another clock domain (ϕ_r). In this paper, the two clock domains are considered mesochronous, that is, they have the same frequency f but with a possible phase difference. The N-flop also ensures that data enters the datapath of the second (receiver) domain after a fixed latency of n-cycles, irrespective of the clock frequency.

The schematic of the N-flop and its control logic is shown in Figure 5. Each of the N-flop and its control consists of two main parts. The front part of the N-flop control is a round robin counter fed by the source synchronizing signal ϕ'_s. Only one bit of the m output bits of the counter is "1" during any given cycle, and each line is "1" once every m cycles. Thus, the output of the counters is "one hot" and increments as follows: $0 : \text{“}100\cdots0\text{”} \rightarrow 1 : \text{“}010\cdots0\text{”} \rightarrow 2 : \text{“}001\cdots0\text{”} \rightarrow \cdots m-1 : \text{“}000\cdots1\text{.”}$ The front end of the N-flop is a circular FIFO buffer that has m parallel latches; each of which is activated by one of the counter's output bits. As the front end counter increments, the FIFO latches are individually and sequentially activated. Thus, only one latch is active at any given time and each latch is active once every m cycles.

The back end of the N-flop is an m-to-1 multiplexer, whose inputs are the front end latch outputs as shown in Figure 5. The multiplexer is controlled by another round robin counter fed by the receiver clock signal ϕ_r. The incrementing counter causes the multiplexer to sequentially output its n inputs. An initial offset Δ exists between the values of the front end and back end counters to avoid premature release of the data before the preset latency of n-cycles.

The construction of an N-flop is very similar to the *mesochronous FIFO synchronizer* used in source-synchronous off-chip communication [13]. The N-flop can also be viewed as the general case of a flip-flop that outputs data after a latency of n cycles. Note that when $m = 1$, each of the front end and back end of the N-flop becomes a single latch, whose control (clock) signals are out-of-phase. Note also that the critical path of the N-flop is the same as that of a flip-flop. Moreover, as only one latch is active at any time in each of the N-flop front end and back end, the active energy dissipation of the N-flop is comparable to that of a conventional flip-flop, but its leakage is $m - 1$ times that of a conventional flip-flop.

2.3. SSMC operation

In order to illustrate the operation of the SSMC scheme, let us assume that $n = m = 4$, and that the front end and back end counters are initially reset to $0 : \text{“}1000\text{”}$ and $1 : \text{“}0100\text{,”}$ respectively. This difference in initial counter values is to avoid premature release of data before the required latency in the case when the data propagation delay across the interconnect is less than the clock cycle. This will be apparent from the context of the second example in this subsection.

The maximum clock frequency $f_{\max} = 1/T_{\min}$ is set such that data propagates along the interconnect, goes through the N-flop, and is set up at the output flip-flop in n-clock cycles, that is,

$$nT > t_d + t_{\text{latch}} + t_{\text{mux}} + t_{\text{setup}} + t_{\text{long-jitter}}, \tag{2}$$

where T is the cycle time, t_d is the propagation delay along the data line, t_{latch} is the propagation delay of the N-flop front end, t_{mux} is the propagation delay of N-flop back end (multiplexer), t_{setup} is the setup time of the output flop, $t_{\text{long-jitter}}$ is the n-cycle long-term clock jitter. The source sync signal path is also designed such that the source sync signal edge arrives early enough to increment the front end counter before the arrival of its corresponding data edge.

At maximum frequency operation $f = f_{\max}$, the operation, shown in Figure 6, is as follows: at the first clock edge ($t = 0$), the first source sync signal edge and the first data edge are launched. Meanwhile, the front end counter is still at $0 : \text{“}1000\text{”}$ and latch_0 is active, while the other FIFO

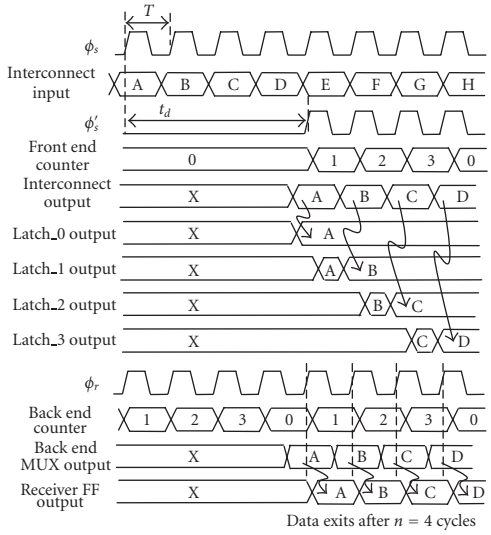

Figure 6: Example of SSMC operation at $f = f_{\max}$.

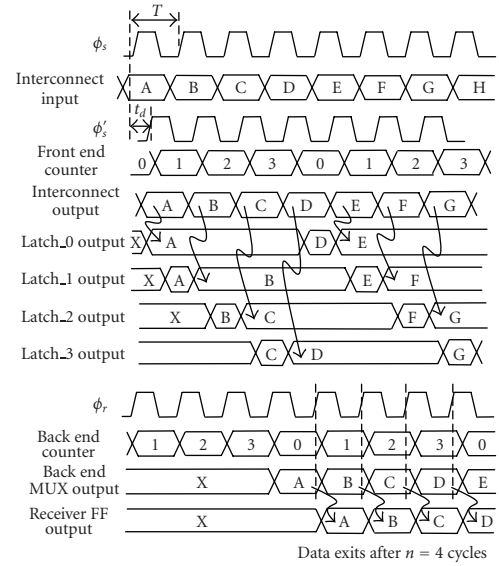

Figure 7: Example of SSMC operation when $t_d < T$.

latches are inactive. The first receiver clock edge will also occur and the back end counter will increment from 1 : "0100" to 2 : "0010," such that the multiplexer is reading the output of latch_2. At $t = T$, the launched data and sync signal edges will have traveled further along the interconnect, but have not reached their destination yet. Also, the next receiver edge will come and increment the back end counter to 3 : "0001." The data and sync signal edge will not reach until after $3T$, meanwhile the back end counter will increment to 0"1000" and then 1 : "0100." Once, the first sync signal edge arrives at the receiver ($4T > t > 3T$), the front end counter will increment from 0 : "1000" to 1 : "0100" and the data will be latched into latch_1. And as the back end counter is also at 1 : "0100," then the multiplexer will read the data which was just latched into latch_1, deliver it to setup at the FF input. Thus, after $n = 4$ clock cycles, the output flip-flop will launch the received data into the receiver datapath.

The other extreme is when operation is at a very low frequency ($T \gg t_d$), which occurs during boundary-scan testing. In this scenario, shown in Figure 7, the first source sync signal edge and the first data edge are launched at the first clock edge ($t = 0$). Meanwhile, the front end counter is still at 0"1000" and latch_0 is active, while the rest is inactive. The first receiver clock edge will also occur and the back end counter will increment from 1"0100" to 2 : "0010." Because $t_d \ll T$, the source sync signal edge and data will arrive before the next clock edge, which causes the front end counter to increment from 0 : "1000" to 1 : "0100," and the received data will get latched to latch_1. Note that if both front end and back end counters were initialized to the same value, the data arriving at the receiver would have been immediately latched to the receiver output, that is, after a latency of 1 cycle and not after the required latency of $n = 4$ cycles. Thus, an initial difference Δ between the front end and back end counters must exist to avoid premature release of data and ensure proper operation at low frequencies. This initial difference Δ will be quantified in (5) and (6).

The data latched in latch_1 will remain there for three more receiver clock edges, such that the back end counters increment from 2 : "0010" to 3 : "0001" to 0 : "1000" to 1 : "0100." At such point, the multiplexer reads the data in latch_1 and delivers it to setup at the flip-flop output. At the following receiver clock edge, $n = 4$ clock cycles have passed, and the first received data is launched into the receiver datapath.

2.4. SSMC advantages

One of the main advantages of the SSMC scheme is that it provides a variation-tolerant characteristic. In the CMC scheme, the delay of some of the interconnect segments may increase due to within-die (WID) parameter variations, which leads to a reduction in the maximum clock frequency of the whole bus. However, SSMC eliminates intermediate flip-flops, and hence, removes timing boundaries throughout an interconnect. Thus, SSMC enables better averaging of delay variations across the whole interconnect, which reduces bit-rate degradation due to WID process variations.

SSMC is also scalable and suitable for data transmission across clock domain boundaries. Another advantage of SSMC compared to a conventional MCB is that it replaces its power-hungry synchronizing flip-flops by a more energy efficient synchronizing scheme. The intermediate flip-flops in CMC are replaced by an N-flop per data line. An N-flop has almost the same active energy dissipation but $n - 1$ times the leakage of a single flip-flop. A shared source sync line and an N-flop control block are also required in the SSMC, but their energy dissipation is amortized over the number of data bus lines. Significant energy dissipation is expected, as shown from the simulation results in Section 3. Note that

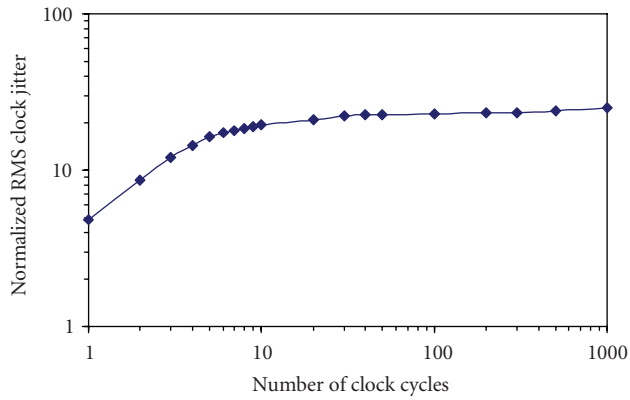

Figure 8: Normalized long-term jitter versus the number of clock cycles.

these energy savings are expected to increase as the number of bus cycles increases, because more flip-flops will be eliminated.

The SSMC scheme also reduces the overhead per cycle due to clock jitter. The dependency of jitter on the number of clock cycles [14, 15] is presented in Figure 8. Single-cycle jitter is defined as the clock edge variation in consecutive clock cycles. Long-term jitter is the accumulated clock edge variation across multiple cycles. For a small number of clock cycles, jitter primarily results from voltage controlled oscillator (VCO) variation, where jitter is proportional to the square root of the number of cycles [14]. For a large number of clock cycles, the phase-locked loop (PLL) phase detector and loop filter sense the VCO error and adjust the VCO input accordingly, leading to a saturation of the long-term jitter. This indicates that the long-term jitter per cycle ($t_{\text{long-jitter}}/n$) monotonically decreases as the number of cycles increases. Hence, the clock jitter overhead per cycle in the SSMC scheme is less than that in the CMC scheme ($t_{\text{long-jitter}}/n < t_{\text{cycle-jitter}}$), and is further reduced as the number of cycle increases, as shown by (1) and (2). Thus, this extra timing slack can be used to further reduce the repeater sizes and energy dissipation.

2.5. System constraints

Several system constraints have been identified to ensure correct system functionality.

2.5.1. Critical path

The maximum clock frequency $f_{\max}$ ensures that data propagates along the interconnect, goes through the N-flop, and is set up at the output flip-flop in n-clock cycles

$$nT > t_d + t_{\text{latch}} + t_{\text{mux}} + t_{\text{setup}} + t_{\text{jitter}}, \tag{3}$$

where T is the cycle time, t_d is the propagation delay along the data line, t_{latch} is the propagation delay of the N-flop front end, t_{mux} is the propagation delay of N-flop back end (multiplexer), t_{setup} is the setup time of the output flop, t_{jitter} is the n-cycle long-term clock jitter.

2.5.2. Avoiding FIFO buffer overrun

In an n-cycle bus, data takes n-cycles to be transferred from the source to the receiver datapath. During the data transfer, the data spends some time on the line and the rest is stored in the N-flop front end (FIFO buffer). In case $T \gg T_{\min}$, the data stays stored in the FIFO buffer most of the time. And in that case, if the FIFO buffer capacity is not large enough, buffer overrun may occur, and data temporarily stored in the buffer may be overwritten before being delivered to the output flip-flop. Thus, if the N-flop FIFO buffer has a depth of m-bits, buffer overrun can be avoided if the buffer can accommodate at least n bits, that is,

$$m \geq n. \tag{4}$$

As explained earlier, an initial offset Δ exists between the front end and back end counters to avoid premature release of the data. And to ensure that data is delivered into the receiver datapath after a latency of exactly n-cycles, then

$$n = m - \Delta + 1. \tag{5}$$

Combining (4) and (5) shows that

$$\Delta \geq 1. \tag{6}$$

However, increasing the depth of the FIFO m leads to more devices and more devices loading, and hence, a higher energy dissipation for the N-flop. Thus, for a low-power design, the FIFO depth should be kept at its minimum $m = n$, and hence a counter offset of $\Delta = 1$.

2.5.3. Reducing intersymbol interference (ISI)

To ensure correct sampling, the bit period has to be large enough to accommodate any data jitter and to account for data attenuation, that is, has an appropriately open "eye," as shown in Figure 9. Thus, given a certain cycle time T, the data line and its repeaters have to be designed such that

$$T \geq \Delta t_d + t_{\text{margin}}, \tag{7}$$

where Δt_d is the data jitter and t_{margin} is an extra time margin to account for signal attenuation (and rise time) at the end of any interconnect segment and ensure full-scale switching.

2.5.4. Ensuring correct sampling

In order to ensure correct sampling, the source sync signal has to arrive at the front end N-flop counter and increment it before the data arrives at the N-flop front end input. In order for this condition to be satisfied in the presence of a certain data jitter $\Delta t_d = t_{d,\max} - t_{d,\min}$ and a certain source sync signal jitter $\Delta t_c = t_{c,\max} - t_{c,\min}$, the worst-case increment of the front end counter has to occur before the earliest arrival of the data, as shown in Figure 10, that is,

$$t_{c,\max} + t_{\text{counter}} \leq t_{d,\min}, \tag{8}$$

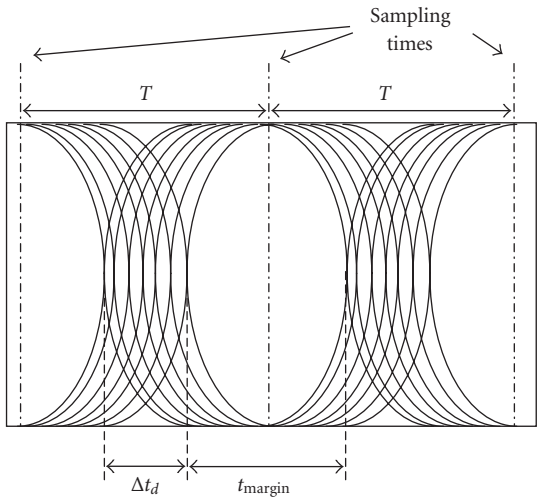

Figure 9: Eye diagram for the data signal at the end of the interconnect.

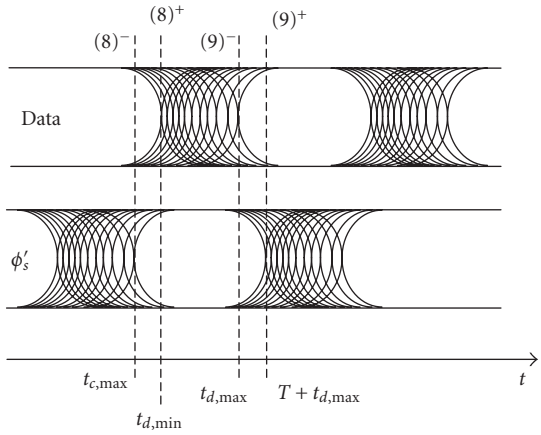

Figure 10: Illustrating the system constraints for correct sampling (8)-(9).

where t_{counter} is the counter delay. Also, the worst-case data edge has to arrive early enough before the following front end counter increment, that is,

$$t_{d,\max} \le T + t_{c,\min} + t_{\text{counter}} - t_{\text{cycle-jitter}}. \tag{9}$$

Rearranging (8) and (9) yields

$$T \ge \Delta t_d + \Delta t_c + t_{\text{cycle-jitter}}, \tag{10}$$

$$\Delta t_c + t_{\text{counter}} \le t_{d,\min} - t_{c,\min} \le T - \Delta t_d + t_{\text{counter}} - t_{\text{cycle-jitter}}. \tag{11}$$

2.5.5. Summary

The system design constraints can be summarized by the following set of equations:

$$n = m - \Delta + 1, \quad \Delta \ge 1. \tag{12}$$

$$\Delta t_c + t_{\text{counter}} \le t_{d,\min} - t_{c,\min} \le T_{\min} - \Delta t_d + t_{\text{counter}} - t_{\text{cycle-jitter}}. \tag{13}$$

$$T \ge \max\left\{\Delta t_d + \Delta t_c + t_{\text{cycle-jitter}}, \Delta t_d + t_{\text{margin}}, \frac{t_{d,\max} + t_{\text{latch}} + t_{\text{mux}} + t_{\text{setup}} + t_{\text{long-jitter}}}{n}\right\}. \tag{14}$$

2.6. Reducing the jitter of the data and the source synchronizing signals

One of the main differences between SSMC and its similar off-chip communication scheme is the nature of the interconnect. On-chip interconnects have smaller cross-sections and narrower spacing as compared to off-chip interconnects. Thus, on-chip interconnects tend to be RC dominated while off-chip interconnects are LC dominated. Hence, on-chip interconnects tend to have higher dispersion and attenuation compared to off-chip interconnects, which leads to a relatively poor pulse response. Also, the delay of on-chip interconnects has a quadratic dependence on length as opposed to the linear relationship for off-chip interconnects. Thus, repeater insertion [16] is required to improve the pulse response and delay-length relationship of on-chip interconnects. Moreover, due to narrower line spacing, on-chip interconnects are more capacitively coupled than off-chip interconnects. Hence, different switching scenarios have different delays, which cause large signal jitter (Δt_d) and degrade the overall bus performance as shown in (14).

The delay of a repeated interconnect can be expressed [16, 17] as

$$t_d = 0.38 r_t c_t \frac{l^2}{k} + 0.69\left(k \cdot R_u C_u + \frac{R_u}{h} c_t l + r_t l C_u h\right), \tag{15}$$

where R_u and C_u are the output resistance and input capacitance of a unit repeater. l is the total interconnect length, k are the number of interconnect sections (i.e., number of repeaters), and h is the repeater sizing multiplier. r_t is the per-unit-length interconnect resistance and c_t is the per-unit-length equivalent interconnect capacitance, which can be expressed as in [18]

$$c_t = c_g + \Sigma \text{MCF}_i c_c, \tag{16}$$

where c_g and c_c are the physical vertical and lateral (coupling) capacitance components, respectively. MCF_i is the Miller coupling coefficient between the line of interest and line i. The MCF indicates the contribution of the coupling capacitance C_c to the overall equivalent capacitance C_t. In the literature, the commonly used values of MCF are zero for similarly switching coupled lines (1) when only one of the two coupled lines switches, and (2) when the coupled lines oppositely switch. Thus, different switching scenarios lead to MCF

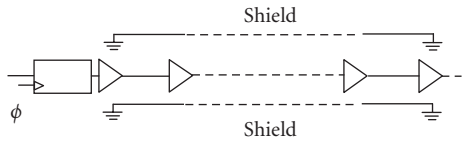

Figure 11: Interconnect shielding technique [19].

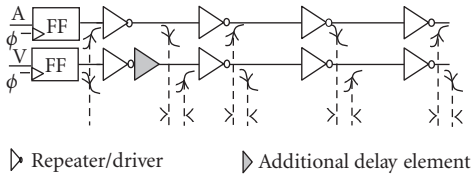

Figure 12: Alternate delayed line (ADL) bus scheme: delay element placed on alternate lines as proposed in [20, 21].

variation ΔMCF, interconnect capacitance variation Δc_t, and eventually a signal jitter Δt_d,

$$\begin{aligned}\Delta t_d &= \left(0.38 r_t \frac{l}{k} + 0.69 \frac{R_u}{h}\right) \cdot \Delta c_t l \\ &= \left(0.38 r_t \frac{l}{k} + 0.69 \frac{R_u}{h}\right) \cdot \Delta\text{MCF} \cdot c_c l.\end{aligned} \tag{17}$$

As shown in (17), the delay jitter can be reduced by reducing the value in parentheses. Thus, the higher ($k \uparrow$) number of upsized ($h \uparrow$) repeaters is required to reduce the signal jitter on-chip interconnects. This solution has the disadvantage of a higher repeater capacitance (hkC_u), which in turn increases the energy dissipation. Another disadvantage is that even if the value within parentheses is reduced, it does not go down to zero, and a jitter Δt_d will remain. This jitter is proportional to length, and in order to achieve a certain bit rate, a limit will be placed on the maximum bus length.

A more energy efficient solution that avoids this bus length limitation is to reduce the MCF variation ΔMCF. This can be performed by using a bus scheme that unifies the MCF for all switching scenarios. One of these scenarios is bus shielding [19], where V_{dd} or ground lines are inserted between the switching lines, as shown in Figure 11. This scheme has the advantage of transforming the MCF for all switching scenarios to 1, and hence eliminates data delay variation $\Delta\text{MCF} = 0$. The disadvantage of this scheme is that it doubles the required bus metal area. Thus, shielding will not be applied to the data lines in our SSMC scheme, but will only be applied to the extra source sync line, as it occurs only once. An area efficient alternative for the data lines will be the alternate delayed line (ADL) scheme [20, 21], which adds a delay element to alternate lines, as shown in Figure 12. This inserted delay separates the switching events of adjacent interconnects, and the MCF for all switching scenarios becomes approximately MCF $\approx$ 1, that is, $\Delta\text{MCF} \approx 0$. And in case of a bidirectional bus, the optimal repeater staggering [18] scheme, shown in Figure 13, can be used to achieve MCF $\approx$ 1 for all switching scenarios, and hence, $\Delta\text{MCF} \approx 0$. Another advantage of these three schemes is that they reduce

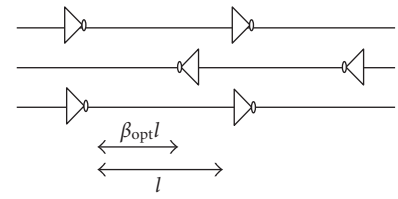

Figure 13: Staggered bidirectional bus [18].

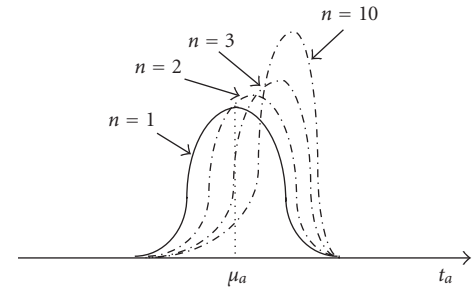

Figure 14: Worst-case arrival time distribution for a CMC.

the worst-case MCF from 2 to 1, which reduces the interconnect delay. This delay slack can be used to downsize repeaters and further reduce energy dissipation.

2.7. Variation tolerance of SSMC

The data arrival time t_a (with respect to the clock edge) at the end of any cycle in the conventional MCB, shown in Figure 1, can be written as

$$t_a = t_{\text{flop}} + t_d. \tag{18}$$

Due to WID process variations, the arrival time t_a will have a distribution with mean μ_a and standard deviation σ_a. In this conventional synchronous interconnect scheme, CMC, the clock cycle has to accommodate the worst-case arrival time $t_{a,w}$ in any bus cycle ($1 \le i \le n$),

$$t_{a,w} = \max\{t_{a,1}, t_{a,2}, \ldots, t_{a,n}\}. \tag{19}$$

The max operation in (19) narrows and negatively skews the worst-case arrival time distribution, as shown in Figure 14, that is,

$$\mu_{a,w} > \mu_a, \qquad \left(\frac{\sigma_{a,w}}{\mu_{a,w}}\right) < \left(\frac{\sigma_a}{\mu_a}\right). \tag{20}$$

This increase in the mean arrival time indicates that a multicycle bus will need a larger minimum cycle time, and hence a lower maximum clock frequency $f_{\max}$. This performance degradation is exacerbated as the number of cycles increases as shown in Figure 14. The main reason behind this degradation is the presence of the intermediate flip-flops, which led to the max operation in (19).

The variation in the maximum clock frequency in SSMC depends on the system design. If the system variables are such

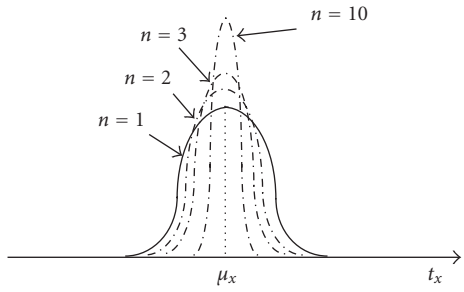

Figure 15: Worst-case arrival time distribution for a datapath delay constrained SSMC.

that the signal jitter on the data and source sync lines is negligible, then the minimum clock period is limited by the datapath delay as shown in (3). Assume that the SSMC scheme has n-cycles and has a length L. Assume also that the delay across the whole datapath is t_x, and that the delay across a single-cycle length L/n is t_s. Assume also that the delay across a single-cycle length L/n has an overall mean μ_s. Assuming that these process variations are Gaussian random variables, then the mean of the delay along the whole interconnect will be

$$\mu_x = E[t_x] = E[t_{s,1} + t_{s,2} + \cdots + t_{s,n}] = \sum_{1}^{n} E[t_{s,i}] = n\mu_s. \tag{21}$$

If all the random delay variables $t_{s,i}$ are uncorrelated, and if the standard deviation for the delay across a single-cycle length (L/n) is σ_{su}, then the standard deviation of the whole datapath delay t_x is

$$\begin{aligned}\sigma_x &= \sqrt{\mathrm{Var}[t_x]} = \sqrt{\mathrm{Var}[t_{s,1} + t_{s,2} + \cdots + t_{s,n}]} \\ &= \sqrt{\sum_{1}^{n} \mathrm{Var}[t_{s,i}]} = \sigma_{su}\sqrt{n}.\end{aligned} \tag{22}$$

As this overall delay represents the delay for n-cycles, then the delay per cycle μ_x/n remains at μ_s *showing no mean performance degradation* with a distribution that narrows as n increases, as shown in Figure 15,

$$\frac{\sigma_x}{\mu_x} = \frac{1}{\sqrt{n}}\frac{\sigma_{su}}{\mu_s}. \tag{23}$$

However, if the variations are correlated with a standard deviation of σ_{sc} for the delay across a single-cycle length (L/n), then the standard deviation of the whole datapath is

$$\begin{aligned}\sigma_x = \sqrt{\mathrm{Var}[t_x]} = \sqrt{\mathrm{Var}[t_{s,1} + t_{s,2} + \cdots + t_{s,n}]} < n\sigma_{sc}, \\ \frac{\sigma_x}{\mu_x} < \frac{\sigma_{sc}}{\mu_s}.\end{aligned} \tag{24}$$

This overall delay variation decreases (i.e., better variation averaging occurs) as the total bus length increases and becomes much greater that the correlation distance, which is typically in the range of $1 \sim 3$ mm in current technologies [1]. Thus, in the presence of correlated and/or uncorrelated process variations, *a datapath delay constrained SSMC outperforms a CMC* as it preserves its mean maximum clock frequency, while narrowing its maximum frequency distribution.

If, on the other hand, the SSMC system variables are such that the minimum cycle time is constrained by data jitter, as in (10), then the minimum cycle time does not depend on the overall interconnect delay, but rather on the minimum temporal separation between two consecutive data edges $D[t]$ and $D[t + T]$. As these two edges travel consecutively on the interconnect, the within-die process variations along the path are exactly the same for both data edges. Hence, the variation in the maximum clock frequency for this jitter-constrained SSMC interconnect scheme will be determined by the variation in data jitter rather than by interconnect delay due to WID variations. And if the techniques explained in Section 2.6 are employed, then data jitter variations can be significantly reduced, and hence, the *performance degradation can also be reduced*.

2.8. Reducing the energy penalty of the source synchronizing overhead

The overhead of the SSMC scheme is the area and energy dissipation of the N-flop control circuitry, and the extra line used to deliver the source sync signal. If the source clock is sent directly on the source sync line, a large energy overhead occurs. The reason is that the clock signal switches twice every cycle, whereas the data signals switch with a ~ 0.1 probability [8]. Thus, the source sync line will incur approximately 20 times the energy dissipation of a single data line. In order to reduce the energy overhead of this line, the source signal has to be altered to reduce its switching activity, while still keeping its required synchronizing edges. The first modification would be to send the clock signal over the source sync line after halving its frequency, as only one edge per cycle is required for synchronization. And in that case the regular flip-flops used in the front end round robin counter can be replaced by double-edge triggered flip-flops [22]. This first modification reduces the activity and energy dissipation of the source sync line by half. The next modification comes from the fact that the source sync is not required if data on the bus is not changing. In that case, a bus activity signal can be generated by XORing the input and output of the driving flip-flop for each bus line, and then ORing all these signals. This bus activity signal is then used to gate the source sync signal [23], and significantly reduce its activity factor and energy dissipation to a value closer to that of a data line. It should be noted that this overhead is shared by all bus lines. Hence, the energy overhead will be amortized over the number of bus lines.

2.9. Dealing with metastability

Whenever there are setup and hold time violations in any flip-flop, it enters a state where its output is unpredictable;

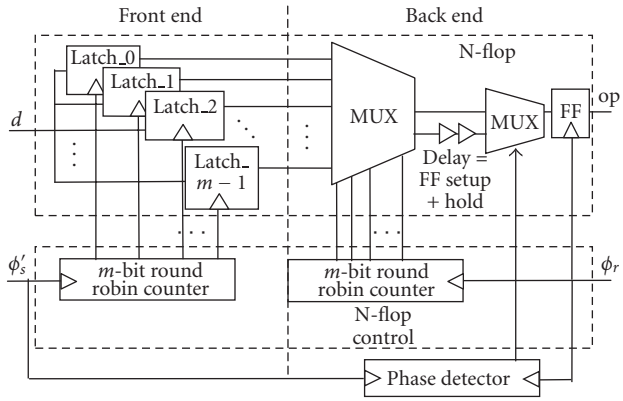

Figure 16: N-flop with metastability reduction overhead (enabled during reset only).

this state is known as metastable state (quasi-stable state); at the end of metastable state, the flip-flop settles down ambiguously to either "1" or "0." This whole process is known as metastability. The proposed scheme uses the same technique used in off-chip mesochronous links to reduce metastability [13]. When most chips are initially powered up, a power-on reset circuit delivers a global reset signal to the whole chip. During this initial reset period, the front and back end counters in the N-flop are reset to their initial values, which remain there until the reset signal is asserted low. Also, during this reset period, which is in the order of a 1 millisecond, metastability possibility is detected and accounted for using a phase detector, a delay element, and a multiplexer, as shown in Figure 16. Metastability occurs if the data at the back end flip-flop input changes during the critical keep out window around the flop's rising edge (setup + hold).

In order to avoid metastability in the SSMC scheme, two measures are usually taken. First, the back end flop is designed in order to reduce its keep out timing window by using high gain latches with high voltage transfer characteristic slopes in the metastable region. Second, if the data arriving at the back end flip-flop input can possibly transition during the back end's flop keep out window, the data is delayed to avoid transitioning in the keepout window and avoid metastability. This is usually implemented by comparing the phase of the received sender clock signal (sync ϕ_s') to the local receiver clock ϕ_r using a phase detector. If the detected phase difference is such that the data at the input of the back end flip-flop switches during this flop's critical keep out window, then the phase detector's latched output will be asserted to select a delayed version of the back end multiplexer output to be fed into the back end flop. Note that as the clock domains are mesochronous (both have constant phase), this phase detection process needs to be performed only. Thus, the phase detector circuit is disabled after the initial system reset period is over. Hence, the power dissipation penalty of the phase detector will have little impact on the reported power results during normal operation. The power dissipation penalty is comprised by the power leakage of the phase detector circuitry and the power dissipation overhead required to upsize interconnect repeaters to account for the extra (negligible) delay of the delay selection multiplexer shown in Figure 16.

2.10. SSMC implementation issues

This subsection will explain some implementation issues faced during the design of a global multicycle interconnect using both the CMC and SSMC schemes. These will be explained in the context of one of the most common design scenarios in microprocessors, GPUs and pipelined architectures, in which the system cycle time T, the interconnect latency n, and the source and destination locations are known a priori.

In the CMC scheme, n flip-flops linking $n - 1$ interconnect segments will need to be inserted between source and destination. The main constraints for CMC interconnect links are the locations of the intermediate flip-flops, which are constrained by positions with active area availability to insert a flip-flop and by the interconnect segment length L_{seg}. The interconnect segment lengths must have a propagation delay $t_{d,\max}$, that satisfies (1), that is,

$$t_d \leq T - t_{\text{flop}} - t_{\text{setup}} - t_{\text{cycle-jitter}}. \tag{25}$$

In most cases, (1) is not satisfied, and repeaters need to be inserted along the interconnect segments. The number and size of repeaters are determined by positions with repeater placement availability, and by the segment propagation delay t_d requirement set by (1).

Besides lower energy dissipation and better variation tolerance, the SSMC has three other main differences, compared to the CMC scheme, which are apparent during implementation. The first difference is the extra metal resources required by the SSMC scheme for the shielded sync signal. However, this penalty is reduced when the SSMC scheme is used to implement wide buses, as the shielded sync line area penalty will be shared by all bus lines. The second difference is that the SSMC scheme does not have any intermediate flip-flops. This implies the advantage of not needing any available active area for intermediate flip-flop placement, and hence, removes the constraints on the global interconnect route. However, it does mandate a larger area at the receiver end for the N-flop and its control circuitry. *The area of the N-flop and its control circuitry is almost the same as that of the n flip-flops in the CMC scheme*. Thus, the SSMC scheme approximately retains the same active area requirement for flip-flops, but instead of constraining the designer to distribute the flip-flops along the interconnect route as in CMC, it lumps it all at the receiver end. It should be noted that minor routing constraints still exist due to the presence of intermediate repeaters, but these constraints can be easily dealt with using back end placement and routing tools. So, in summary, SSMC removes the interconnect global routing constraints due to flip-flop placement, but requires a larger active area (approximately equal to the total CMC flip-flop area) at the receiver side. Removing the intermediate flip-flops in the SSMC scheme gives the advantage of avoiding the necessity of a low-skew global clock network to synchronize the intermediate flip-flops. The SSMC scheme only requires

local clock signals at the source and receiver ends of the interconnect, which reduces the energy dissipation, delay, and metal resource requirement for the low-skew global clock distribution network. It should be noted that in the context of this second difference, the SSMC scheme is suitable to implement serial links in network-on-chips and through-silicon vias (TSV) is 3D stacked chips. Both of these interconnect links cannot have flip-flops distributed along their lengths either due to the lack of a global clock network or due to the infeasibility of connecting active devices to it along its length.

The third difference is that the intermediate interconnect segment length is only limited by its propagation delay, as shown in (1). However, the interconnect length between the source and the N-flop in the SSMC scheme is limited by both the interconnect propagation delay t_d and jitter on the data and sync lines $\Delta t_d + \Delta t_c$, as shown by (3) and (10). In the CMC scheme, if both the latency n and the total global interconnect length are increased such that their ratio is kept the same, the delay constraint (1) is still maintained. However, if the same occurs for the SSMC scheme, the delay constraint given by (3) will remain satisfied, but that given by (10) may be violated as the MCF-induced jitter Δt is length-dependent as shown in (17). In such a case, the interconnect can be segmented in one or more segments by inserting intermediate N-flops along the interconnect to reduce the data jitter, as shown in Figure 17. Note that the latency of each segment does not have to be equal, for example, n_1 does not necessarily have to be equal to n_2 in the example shown in Figure 19. This degree of freedom can be used to pick the best locations with available active area to implement the intermediate N-flops. To illustrate SSMC schemes with more than one segment, both the SSMC and CMC schemes were used to implement an n-cycle 4 GHz 16-bit bus in a 65- nm technology. The length of the bus and the required latency n were varied. Implementation details and normalized energy savings are shown in Table 1. Both implementations had almost the same active area requirement, however, the SSMC scheme required 13% more intermediate metal layer area to implement the shielded sync line. The results shown in Table 1 show that SSMC can still be used to implement very long buses if the interconnects are segmented by inserting intermediate N-flops to reduce MCF-induced data jitter and achieve the required bit rate. The results also show that the normalized energy reduction will be limited by that at the longest interconnect length that requires a single N-flop (i.e., $\%\Delta E_{\max} = 20\%$ which occurs for $L = 9$ mm in Table 1).

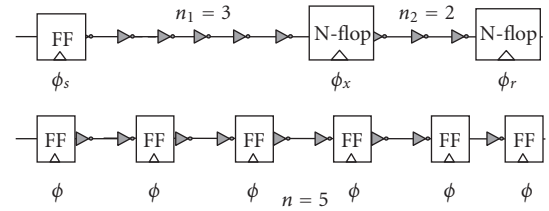

Figure 17: SSMC scheme for a long global multicycle $n = 5$ interconnects compared to its corresponding CMC scheme implementation.

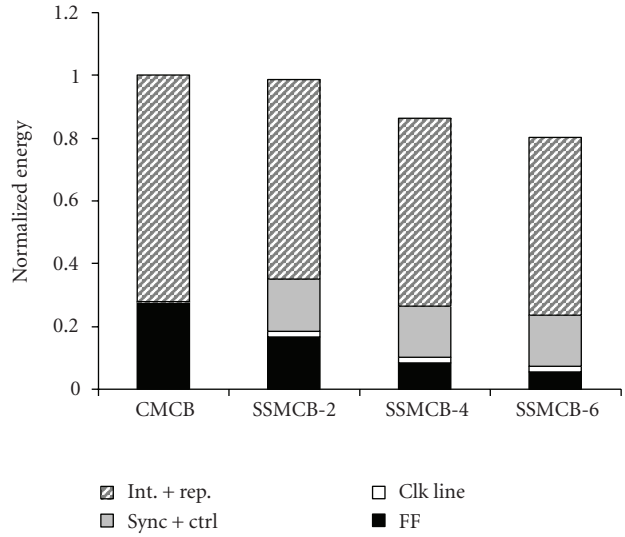

Figure 18: Energy comparison between the CMC and SSMC schemes for different numbers of cycles $n = 2, 4, 6$.

3. SIMULATION SETUP AND RESULTS

3.1. Simulation setup

Sixteen-bit conventional (CMC) and source-synchronous (SSMC) multicycle buses were designed in a 65-nm technology [24]. The flop-to-flop distance was set to 1500 μm and the target bit rate was 4 GHz. The interconnects were all at minimum width and minimum spacing ($w = s = 0.14\,\mu$m). In order to reduce signal jitter on the source sync line of SSMC scheme, the source sync line was fully shielded and placed in the center of the bus. Thus, the SSMC has a metal area overhead equivalent to three lines. The alternate delayed line (ADL) scheme [20, 21], discussed in Section 2, was applied to the data lines of the SSMC scheme to reduce data jitter Δt_d. But as the ADL scheme also reduces worst-case delay, it was applied to the bus lines of the conventional MCB for a fair comparison. Device sizes were optimized using a global optimizer in order to achieve the system constraints (12)–(14), the required bit rate, required signal slopes, and noise margins, while reducing energy dissipation. The energy dissipation overhead on the clock tree due to the flip-flops or the synchronizer was factored in our energy measurements for both schemes. This was all repeated for different numbers of bus cycles n. Both energy reduction techniques explained in Section 2.7 were implemented in the design of the SSMC bus.

Both buses were fed with random uniformly distributed datastreams with a 0.1 activity factor. The bus energy dissipation was recorded and divided by the number of bus lines. Random within-die (WID) variations were then applied to both of these buses. The sources of variations used were channel length, channel width, threshold voltage, metal thickness, metal width, line spacing, and ILD thickness.

Table 1: CMC and SSMC schemes comparison for implementing a 4- GHz 16-bit bus of latency n and length L. All energy values are normalized to the CMC energy dissipation at each (L, n).

L (mm)	n	CMC	SSMC		
		Number of flip-flops	Number of N-flops	Size of N-flops (m)	Normalized energy
3	2	2	1	2	0.98
6	4	4	1	4	0.84
9	6	6	1	6	0.80
12	8	8	2	4, 4	0.85
15	10	10	2	4, 6	0.82
18	12	12	2	6, 6	0.81

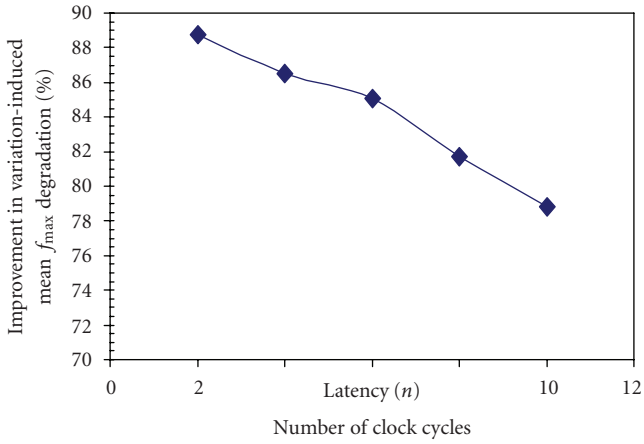

Figure 19: SSMC improvement in bit rate due to its better tolerance to process variations compared to CMC (Ref. bit rate= 4 GHz).

Although the sources of interconnect metal variation are mostly deterministic (e.g., metal thickness depends on metal density), the simulations are performed by assuming a random correlated variation model. Using these WID variations, a Monte Carlo circuit simulation is performed for both bus schemes resulting in a bit rate distribution.

3.2. Energy dissipation results

Table 2 shows the energy dissipation results obtained for 2, 4, and 6-cycle CMC and SSMC buses. All energy values for each n-cycle bus in Table 2 are normalized with respect to the respective total energy dissipation of the CMC bus. Even though SSMC did not achieve much energy reduction for a 2-cycle MCB, a significant 20% energy reduction was achieved as the number of cycles increased to 6. It should also be noted that the energy dissipation of the source sync line and the synchronizer is linear with the number of cycles, and hence their percentage contribution to the overall system remains almost constant as n increases, as shown in Table 2 and Figure 1. The energy reduction in the SSMC scheme increases as n increases, because of two main reasons: (1) the energy dissipation of the initial flip-flops becomes negligible as the bus gets longer; and (2) the timing overhead of the initial flip-flop and the synchronizer gets amortized better as n increases, which reduces the required size and number of repeaters as shown in Figure 18.

3.3. Variation results

Simulation results show that SSMC removes up to ∼ 90% of the maximum clock frequency degradation that occurred in CMC due to WID process variations, as shown in Figure 19. This improvement was reduced as the bus length increased, because the f_{max} degradation of CMC saturated faster than that of SSMC. Note that most of the delay variation is due to data jitter variation, which has a close-to-linear relationship with bus length. The data jitter variations induced by WID process variation can be further reduced by shielding data lines instead of using the ADL bus scheme but at the cost of doubling the required metal area.

4. CONCLUSION

Interconnect links are one of the main bottlenecks in improving the performance of state-of-the-art integrated circuits. Several aspects of recent DSM technologies limit the maximum bit rate that can be sent on the bus as well as degrade the signal integrity. Moreover, new system architectures and technologies such as the "many-core" platform and 3D-IC stacking lead to the presence of many clock domains.

In this paper, a low-power variation-tolerant source-synchronous multicycle (SSMC) interconnect scheme was proposed. This scheme is scalable and suitable for transferring data across different clock domains such as those in "many-core" SoCs and in 3D-ICs. SSMC replaces intermediate flip-flops by a source-synchronous synchronization scheme. Removing the intermediate flip-flops in the SSMC scheme enables better averaging of delay variations across the whole interconnect, which reduces bit rate degradation due to within-die (WID) process variations. Monte Carlo circuit simulations showed that SSMC eliminates ∼ 90% of the performance degradation in a 6-cycle 9 mm-long 16-bit conventional bus in the presence of WID process variations.

The proposed multicycle bus scheme also led to significant energy savings due to eliminating the power hungry flip-flops and efficiently designing the source synchronization overhead. Moreover, eliminating intermediate flip-flops avoided the timing overhead of the setup time, the flip-flop delay, and the single-cycle clock jitter. This delay slack was

Table 2: Energy simulation results. All energy values are normalized to the CMC energy dissipation at each n.

n	CMC			SSMC				
	FF	Int. + rep.	Total	FF	Source sync line	N-flop ctrl	Int. + rep.	Total
2	0.28	0.72	1.00	0.17	0.15	0.02	0.64	0.98
4	0.28	0.72	1.00	0.08	0.14	0.02	0.60	0.84
6	0.28	0.72	1.00	0.06	0.14	0.03	0.57	0.80

then translated into further energy savings by downsizing the repeaters. The significant delay jitter due to capacitive coupling was addressed, and solutions were put forward to alleviate it. Circuit simulations in a 65-nm process environment indicated that energy savings up to 20% are achievable for a 6-cycle 9 mm long 16-bit bus.

REFERENCES

[1] J. D. Meindl, J. A. Davis, P. Zarkesh-Ha, C. S. Patel, K. P. Martin, and P. A. Kohl, "Interconnect opportunities for gigascale integration," *IBM Journal of Research and Development*, vol. 46, no. 2-3, pp. 245–263, 2002.

[2] K. Rahmat, S. Nakagawa, S-Y. Oh, and J. Moll, "A scaling scheme for interconnects in deep-submicron processes," Tech. Rep. HPL-95-77, Hewlett Packard, Palo Alto, Calif, USA, 1995.

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

[4] L. Scheffer, "Methodologies and tools for pipelined on-chip interconnect," in *Proceedings of IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD '02)*, pp. 152–157, Freiburg, Germany, September 2002.

[5] R. Lu, G. Zhong, C.-K. Koh, and K.-Y. Chao, "Flip-flop and repeater insertion for early interconnect planning," in *Proceedings of the Design, Automation and Test in Europe Conference (DATE '02)*, pp. 690–695, Paris, France, March 2002.

[6] P. Cocchini, "Concurrent flip-flop and repeater insertion for high performance integrated circuits," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD '02)*, pp. 268–273, San Jose, Calif, USA, November 2002.

[7] "IEEE 1149.1-2001: Standard Test Access Port and Boundary-Scan Architecture," Institute of Electrical and Electronics Engineers.

[8] M. Khellah, M. Ghoneima, J. Tschanz, et al., "A skewed repeater bus architecture for on-chip energy reduction in microprocessors," in *Proceedings of IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD '05)*, pp. 253–257, San Jose, Calif, USA, October 2005.

[9] S. Borkar, P. Dubey, K. Kahn, et al., "Platform 2015: Intel Processor and Platform Evolution for the next Decade," 2005, *White Paper*, Intel Corporation.

[10] A. Edman and C. Svensson, "Timing closure through a globally synchronous, timing partitioned design methodology," in *Proceedings of the 41st Design Automation Conference (DAC '04)*, pp. 71–74, San Diego, Calif, USA, June 2004.

[11] B. Black, D. W. Nelson, C. Webb, and N. Samra, "3D processing technology and its impact on IA32 microprocessors," in *Proceedings of IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD '04)*, pp. 316–318, San Jose, Calif, USA, October 2004.

[12] S. Das, A. Chandrakasan, and R. Reif, "Three-dimensional integrated circuits: performance, design methodology, and CAD tools," in *Proceedings of IEEE Computer Society Annual Symposium on VLSI (VLSI '03)*, pp. 13–18, Tampa, Fla, USA, February 2003.

[13] W. J. Dally and J. W. Poulton, *Digital Systems Engineering*, Cambridge University Press, Cambridge, UK, 1998.

[14] J. A. McNeill, "Jitter in ring oscillators," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 6, pp. 870–879, 1997.

[15] M. Mansuri and C.-K. K. Yang, "Jitter optimization based on phase-locked loop design parameters," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 1375–1382, 2002.

[16] H. B. Bakoglu and J. D. Meindl, "Optimal interconnection circuits for VLSI," *IEEE Transactions on Electron Devices*, vol. 32, no. 5, pp. 903–909, 1985.

[17] T. Sakurai, "Closed-form expressions for interconnection delay, coupling, and crosstalk in VLSI's," *IEEE Transactions on Electron Devices*, vol. 40, no. 1, pp. 118–124, 1993.

[18] M. Ghoneima and Y. Ismail, "Optimum positioning of interleaved repeaters in bidirectional buses," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 3, pp. 461–469, 2005.

[19] A. B. Kang, S. Muddu, E. Sarto, and R. Sharma, "Interconnect tuning strategies for high-performance ICs," in *Proceedings of the Design, Automation and Test in Europe (DATE '98)*, pp. 471–478, Paris, France, February 1998.

[20] K. Nose and T. Sakurai, "Two schemes to reduce interconnect delay in bi-directional and uni-directional buses," in *Proceedings of IEEE Symposium on VLSI Circuits (VLSIC '01)*, pp. 193–194, Kyoto, Japan, June 2001.

[21] K. Hirose and H. Yasuura, "A bus delay reduction technique considering crosstalk," in *Proceedings of the Design, Automation and Test in Europe Conference (DATE '00)*, pp. 441–445, Paris, France, March 2000.

[22] M. Pedram, Q. Wu, and X. Wu, "A new design of double edge triggered flip-flops," in *Proceedings of the 3rd Conference of the Asia and South Pacific Design Automation Conference (ASP-DAC '98)*, pp. 417–421, Yokohama, Japan, February 1998.

[23] R. Bashirullah, W. Liu, R. Cavin, and D. Edwards, "A hybrid current/voltage mode on-chip signaling scheme with adaptive bandwidth capability," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 12, no. 8, pp. 876–880, 2004.

[24] Predictive Technology model, http://www.eas.asu.edu/~ptm/.