Flexible and Reconfigurable Mismatch-Tolerant Serial Clock Distribution Networks

Atanu Chattopadhyay, *Member, IEEE*, and Zeljko Zilic, *Senior Member, IEEE*

***Abstract*—We present a clock distribution network that emphasizes flexibility and layout independence. It suits a variety of applications, clock domain shapes and sizes using a modular, standard cell-based design approach that mitigates the effect of intra-die temperature and process variances. We route the clock line serially, using an averaging technique to eliminate skew between clock regions in a domain. Routing clock lines serially allows optimal wire usage for clock networks by eliminating the redundant wires required to match path delays. Our clock network provides control over regional clock skews, can be used in beneficial skew applications and facilitates silicon-debug. Serial clocking permits the use of routing switches in the clock network and allows post-silicon resizing and reshaping of clock domains. Defective sections of the clock network can be bypassed, providing post silicon repair capability. The system uses a closed-loop synchronization phase to combine the clock skew reduction of an actively synchronized clock network with an open-loop operating phase that minimizes power consumption like passive clock networks. Our clock network provides significant flexibility for application-specific integrated circuit, system-on-chip, and field-programmable gate-array designs, exhibiting good operating characteristics everywhere in the design envelope. Our silicon implementation achieves a maximum edge-to-edge uncertainty of 80 ps for regional clocks, which is roughly equal to the cycle-to-cycle jitter of the on-chip clock source.**

***Index Terms*—Clocks, reconfigurable architectures, synchronization, tunable circuits and devices.**

I. Introduction

With the decreased cost and increased availability of silicon area, integrated circuits (ICs) have become significantly more complex in recent years. Fixed tree-based methodologies used to minimize clock skew limit the potential versatility of an IC by reducing the flexibility of the clock distribution network (CDN). Conversely, field-programmable gate arrays (FPGAs) allow flexible logic and clock domains, and avoid the upfront design cost of application specific designs, but their imperfect clock networks must allow for more significant variation in clock signal arrival times, which is unexploited computation time.

Manuscript received February 14, 2010; revised August 09, 2010 and December 10, 2010; accepted December 18, 2010. Date of publication January 28, 2011; date of current version February 17, 2012.

A. Chattopadhyay was with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 2A7, Canada. He is now with PMC-Sierra, Montreal, QC H3R 3L5, Canada (e-mail: atanu.chattopadhyay@mail.mcgill.ca).

Z. Zilic is with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 2A7, Canada (e-mail: zeljko.zilic@mcgill.ca).

Digital Object Identifier 10.1109/TVLSI.2011.2104982

The clock distribution problem is complicated by the complexity of modern systems, decreased power supply voltages, larger die sizes, and higher clock frequencies [1]. Traditionally, passive forms of clock skew reduction are used to balance clock trees with a combination of matching wire length and adjusting clock buffer delays [2]. While the performance of automated clock layout tools has improved significantly, clock buffer mismatches and in-die process variations make it difficult to maintain tight skew tolerance [3], [4]. The outcome is a fixed tree generated using user-specified parameterization that is hard to modify, often requiring complete reimplementation for every iteration of the design. Floorplanning and circuit layout add additional constraints to the problem and introduce asymmetry to the clock routing. Mismatch and process variation in silicon cannot be accounted for in advance and will cause skew to accumulate through the clock network [5].

This text describes a flexible clock distribution network using serial interconnections that can be used in application-specific integrated circuits (ASICs), FPGAs, or microprocessors while including an online clock synchronization mechanism to maintain the clock skew performance required by modern ICs. Our clock network allows small micro-regions to be connected to and disconnected from one-of-many clock domains post-silicon, adding significant flexibility to the clock network. This approach can provide additional versatility for FPGA designs. In ASICs, our clock network will permit a single device to be reconfigured dynamically for multiple tasks or in multiple distinct operating modes.

This paper is organized as follows. Section II describes existing approaches to clock network design. Section III describes our serial approach. Section IV describes a single clock network. Section V describes the design and implementation of our reconfigurable clock network. Section VI describes the circuits used in our proof-of-concept design and Section VII presents the simulated and experimental performance of circuits and clock networks developed using our approach.

II. Clock Networks

Modern devices contain multiple clock domains that each must be routed appropriately to achieve the required clock arrival times for every register in the domain. In the presence of process variation for devices, environmental changes and electrical fluctuations, it is necessary to allow a safety margin to maintain correct circuit behaviour; a 10% of the clock period tolerance is common [6], [7]. Timing violations that occur at the edge of this range are often difficult to detect and reproduce, as they may be device-dependent and intermittent. Where designers once focused primarily on device delay, interconnect delay is becoming increasingly important in clock networks

[8]. In deep-submicrometer technologies, wire delay does not shrink as quickly as device delay since modern wires are thinner and taller, and have higher resistance and capacitance. These wires are particularly susceptible to simultaneous switching noise (SSN) increasing uncertainty in the location and slope of the clock edges.

A. Clock Networks

1) Symmetric Clock Tree: Symmetric clock trees are among the most widely studied clock distribution topologies. The clock source is the root, the signal path to the first branching point is known as the trunk, the intermediate paths are branches and the clock destinations, usually registers, are leaves. We will refer to any end point of a clock network as a *tap* or *leaf*. The number of branches traversed in a clock route represents the number of levels in a clock tree [3]. In a binary or Y-tree, each branch point splits into two branches with the same size and shape, but possibly different orientation. In an H-tree, the clock signal is split into two, twice per level. Wire length is constant for every segment in an H-tree level and is halved for each subsequent level. H-structures use Manhattan routing since every wire is either vertical or horizontal. An X-tree structure [9] replaces horizontal and vertical wires with diagonal ones saving interconnect length, but are not always realizable. Examples of these three symmetric tree structures are shown in Fig. 1. A serial clock network is shown in Fig. 1(D) for comparison. If we assume a unit length distance between adjacent leaves, the interconnect length of a serial network is reduced by about one third for networks with 256 or more taps when compared to H-trees [10]. For a regularly-spaced structure with N-leaves there are multiple possible leaf orderings with an optimal wire length of $N-1$; N if we include the routing of the reverse clock to the clock trunk.

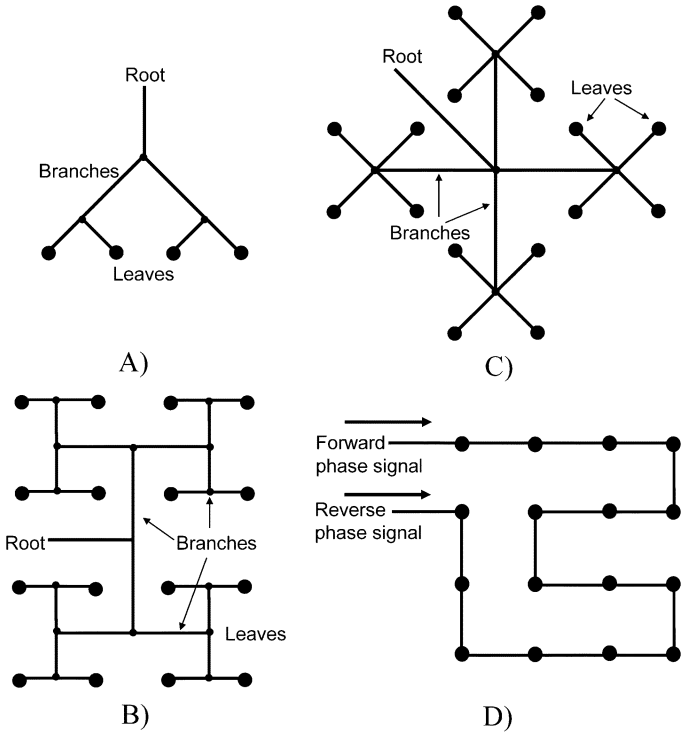

Fig. 1. Clock network topologies. (A) Two-level Y-Tree. (B) Two-level H-Tree. (C) Two-level X-Tree. (D) Serial clock network.

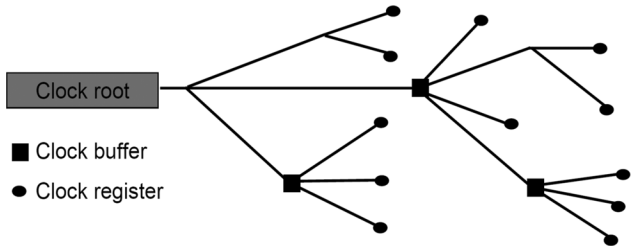

Fig. 2. Asymmetric buffered clock tree.

While small trees may be completely *passive* with no buffering, the fan-out is usually too large for this approach to work. In practice, clock buffers may need to be placed symmetrically at arbitrary intervals creating an *active* tree. To maximize signal integrity, impedance needs to be matched at every branching point. For a tree that has b branches per level, one common approach is to design the output impedance at each branch point to be a factor of b larger than at the input, resulting in a configuration known as a tapered tree [11].

2) Asymmetric Buffered Clock Trees: An asymmetric buffered clock tree is the most commonly used clock network in integrated circuits and ASICs [3] (see Fig. 2). Automated clock tree generation tools use the location and timing requirements of each leaf to create a suitable tree. The timing information contains the acceptable range of clock arrival times taking into consideration a leaf's interdependence with adjacent registers and is also known as a skew schedule. The delay is adjusted by varying the wire length, the number of buffers, the buffer size and the fan-out at each branch point to satisfy the skew schedule. Since this method is based on dependencies between registers, even small changes to the design can have a major impact on the required clock network.

3) Clock Mesh: Mismatch in buffer and interconnect delay will create skew between points in a clock network that are ideally matched [12]. This clock skew will accumulate as signals propagate from source-to-leaf. To counteract this skew, it is possible to shunt the clock buffer outputs at a given level using links, creating a clock mesh [13]. These shunts delay clock signals that arrive too soon and advance signals that are late. Clock meshes are common in high performance microprocessors. This approach increases power consumption due to temporary short circuits through the shunt wires and the added capacitive load seen by the clock drivers. There is also a routing cost due to the shunt wires that increases for larger clock networks. Due to the transport delay of the shunt signals, clock skew cannot be completely eliminated using this approach. Link insertion [14]–[16] is a related technique where critical sections of a clock tree are shunted, realizing many of the benefits of a clock mesh.

4) Resonant Clocking: Resonant clocking is a newer approach for clock network design that creates an LC-tank to allow the clock signal to oscillate at a natural frequency [17]. In an ideal configuration, the energy consumption would be zero, but due to the resistance of the metal lines in the network, the configuration is lossy. The *LC*-tank is formed using the capacitance of the clock leaves and adding an on-chip inductor. By removing all the clock buffers, resonant clock networks have

demonstrated IC core power savings of 20%–35% compared to conventional clock networks [18], [19]. Resonant clock networks generate sinusoidal clocks which can momentarily induce short-circuits because their long rise and fall times allow both pull-up and pull-down transistors to simultaneously conduct. Sinusoidal clocks are also more susceptible to clock jitter when compared to clocks with sharper edges. Resonant clocking requires modified registers and latches that are compatible with sinusoidal clocks [20], [21].

5) Standing and Travelling Wave Networks: A standing wave clock network superimposes a sinusoidal forward phase clock with a reverse phase clock produced by reflecting the forward clock off of a ground termination at the opposite end of the conductor [22]. This approach results in fixed phase clock signals at every tap with varying amplitude [23], depending on the location of the tap.

Travelling clock networks often use a rotary clock ring configuration [24], [25] with a pair of conductors separated by cross-coupled inverters to regenerate signals and maintain oscillation. The frequency and amplitude of the clock signal is constant for every tap, but the phase will vary depending on the tap's location, increasing the complexity of synchronous circuits implemented using this technology.

6) Hybrid Structures: Modern clock networks often mix clocking strategies, pairing one global clock routing technique to another local routing technique [26]. A common pairing would be a global H-tree followed by a local mesh [27], minimizing the clock skew in the network and the power consumption of the mesh. Typically, the global distribution can be constructed using a mesh or tree, and the local network can use a tree, mesh or fishbone structure [13]. A fishbone structure, otherwise known as a center clock tree, has a clock trunk with leaves arbitrarily attached orthogonally to the trunk wherever a clock is needed [28]. FPGAs typically use multiple independent clock trees for global clock distribution and fishbone structures locally so in this case, flexibility comes at the expense of skew tolerance.

7) Serial Clock Networks: Serial clock networks have been used less commonly since there is inherent skew between clock taps using this technique. Here, serial clock networks refer solely to clock networks that use *averaging* to position local clocks between two reference clocks travelling in opposite directions, first proposed by Grover *et al.* [29]. Their scheme uses a three-wire method with a raw clock line and the reduced frequency forward and return phase signal lines. The pulse interval between the forward and reverse phase signals is found at each tap by delaying the forward signal to align it with the reverse one. The raw clock is then delayed by half of this pulse interval to create the local clock. The reference clock lines may exhibit different signal behavior than the raw clock line due to differences in geometry and frequency. The system cannot employ clock buffers, limiting the total clock load, the distance between taps and the total number of taps.

Work in [30] applies a similar averaging approach using a two-wire method. The authors of [31] use analog multipliers at each tap to achieve the same result, but require sinusoidal clocks and produce local clocks that do not swing rail-to-rail. Because of the transmission line nature of their clock network, their method does not permit the use of buffers or logic within the clock network, making the system rigid. The authors of [31] use the term "bidirectional signalling" to describe the transport of signals propagating in opposite directions along two wires. We will reserve use of the term for signals propagating in different directions on the *same* conductor (time-multiplexed).

B. Dynamic Skew Compensation Techniques

To overcome variability, some clock networks use an active clock skew reduction technique that can cope with large die sizes and high-speed clocks [32]. Active clock skew reduction techniques require additional power and area compared to their unsynchronized counterparts. Typically, programmable clock buffers use run-time tuning to eliminate skew in selected regions of a clock network [33]. These methods will reduce the total skew but cannot eliminate it completely since there is no perfect approach to clock synchronization.

Die temperature can also cause significant delay variation in a clock network and changes over time depending on local switching activity and ambient conditions. Localized temperature spikes, known as *hotspots*, can severely impact the skew in a clock network [34]. Temperature and process variance can result in delay changes of over 50% for sub-65 nm devices [35]. The authors of [36] have developed a clock network with self-adjusting delay buffers to handle temperature variation, but it can only cope with inter-die temperature fluctuations and not with localized ones on a single die. Traditional system-level dynamic temperature management techniques are limited by the accuracy of their temperature sensors [37].

Some schemes simply employ skew reduction techniques on existing tree distributions [38], [39], but can require extra wiring and power. Kapoor, Jayakumar and Khatri's approach [1] employs skew compensation at every leaf using a distribution tree and a co-located feedback tree. Some schemes perform root-to-leaf skew compensation at the root or leaf for each local area on the IC [2], [4]. Without adequate matching, the feedback lines and synchronization hardware required for these techniques can introduce error to the skew compensation approach [38].

C. Clock Power

Increased die size and device density of integrated circuits have led to a marked increase in the energy consumption of deep submicrometer designs. The clock signal typically represents the largest load on an IC and oscillates at the highest frequency. As a result, the clock usually consumes the largest portion of on-chip power. Clock power can range from 30%–50% [40], [41] in standard high-performance integrated circuits, and up to 70% [11] in some specialized devices like certain FPGAs. There are three broad sources of power consumption in a device contributing to clock power P_{clk}: short-circuit power P_{sc}, leakage power P_{leak}, and dynamic switching power P_{dyn}.

Short-circuit power is consumed when power and ground supplies are temporarily short-circuited during signal transitions of a gate. Short circuit power is proportional to $V_{\mathrm{dd}} - |V_{\mathrm{tn}}| - |V_{\mathrm{tp}}|$, the clock frequency, and the rise and fall times of the input signals. V_{dd} is the supply voltage, and V_{tn} and V_{tp} are the threshold voltages of the pull-down and pull-up devices, respectively. Since modern technologies rely

on decreased supply voltage, short circuit power is decreasing as $V_{\rm dd}$ approaches $|V_{\rm tn}| + |V_{\rm tp}|$.

Leakage power is an unavoidable phenomenon caused by the imperfect nature of a CMOS switch. For modern technologies, there are two dominant sources of leakage that are typically considered. Sub-threshold leakage power is consumed exclusively when transistors are in an "OFF" state and is increasing for newer technologies due to a decrease in threshold voltage and an increase in operating temperature. Gate leakage is a concern for transistors that are both "ON" and "OFF" and is caused by a reduction in the thickness of the gate oxide that prevents the transistor gate from acting like a perfect insulator [42].

Compared to typical logic blocks, clock power is more affected by dynamic power than leakage power since clock networks have the highest switching activity of any subsystem in a clock domain [43]. Charge is sourced when the output capacitance toggles from 0-to-1 and sunk when it transitions from 1-to-0, so power is only consumed once every two transitions. Dynamic power consumption is

$$P_{\rm dyn} = \alpha \cdot C_L \cdot V_{\rm dd}^2 \cdot f \tag{1}$$

where α is the switching activity of the system between 0 and 1, C_L is the total capacitance being switched, $V_{\rm dd}$ is the supply voltage and f is the operating frequency. For clock networks, the C_L term includes the clock driver input, the clock interconnect, and the clock load capacitances [44]. Clock power due to interconnect is increasing as wire length, capacitance and resistance increase. The switching activity represents the average number of charge/discharge cycles per clock period and is 1 for clock networks.

III. Dual Reference Signal Serial Clock Networks

Our clock network operates in two distinct phases, a closed-loop synchronization phase where the forward phase clock is aligned to the midpoint of the forward and reverse phase signals for each clock destination and an open loop operating mode where the reverse phase signal, the phase detector, up/down clock delay controller, and idle delay lines are disabled to save power. To implement a single clock design using our serial approach, the clock domain is divided into n smaller regions, each of which is connected to a tap. All the taps are connected serially in a "*thread*" of the required shape and size with the global clock connected to both the head and tail of the thread, as shown in Fig. 3 for a four-tap thread. Since the reverse signal is disabled at run-time and since all other taps are idle while a tap is being synchronized, a single conductor is sufficient to transport both forward and reverse reference signals through mutually exclusive portions of the thread. The signal flow in the thread is controlled using the 2:2 switches in Fig. 4. *Forward CLK* and *Reverse CLK* represent the serial clock distribution lines between taps and *CLK A* and *CLK B* represent the forward and reverse phase reference clock signals entering a tap. The switch operates in one of three modes. First, *Reverse CLK* can be routed to *Forward CLK* while a previous tap is being synchronized. Second, *Forward CLK* and *Reverse CLK* can be routed to *CLK A* and *CLK B* when the current tap is being synchronized. Third, the *Forward CLK* can be routed to *Reverse CLK* and fed to *CLK A*

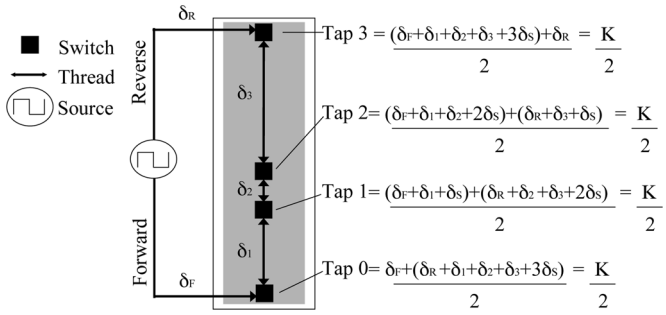

Fig. 3. Four-tap averaging clock thread.

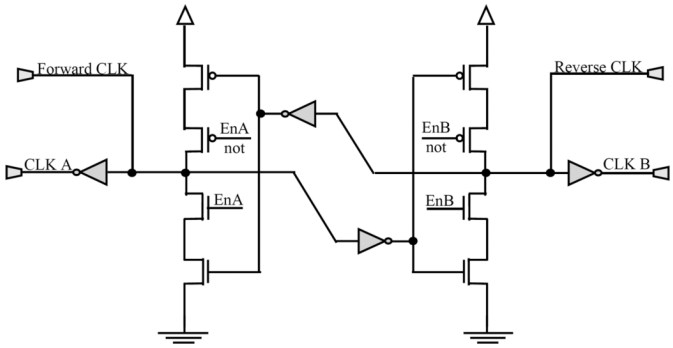

Fig. 4. 2:2 switch to choose forward/reverse phase signals.

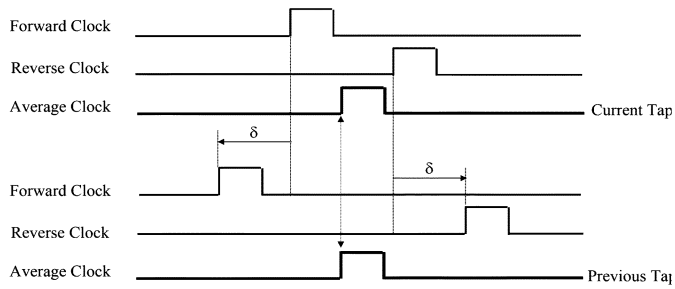

Fig. 5. Underlying concept behind averaging.

after the current tap has been synchronized after the current tap has been synchronized and during operation. Taps are synchronized in order from *Tap 0* to *Tap 3* (or n).

Assume that $K + \delta_s$ is the propagation delay over a clock thread's entire length and δ_s is the delay of a single switch (see Fig. 3). If the propagation delay of the forward phase clock to a tap is δ_+ and the delay of the reverse phase clock is δ_-, once the reference clocks are averaged, the rising edges of all the local clocks will all occur at a fixed time:

$$\frac{\delta_- + \delta_+}{2} = \frac{(K - \delta_+) + \delta_+}{2} = \frac{K}{2}. \tag{2}$$

For two adjacent taps, the resulting averaged local clocks are shown in Fig. 5. The signals are treated as pulses for presentation purposes only.

Our serial clock network has three major advantages over traditional clock networks. First, its synchronization phase compensates for skew in clock networks and is resilient to interconnect and device variance due to manufacturing and temperature variation since the devices and interconnect requiring

matching are co-located. This is a capability that static clock networks do not have. Second, our clock network is inherently more energy efficient than most existing clock networks with post-silicon clock synchronization since it performs clock alignment using low power variable delay inverters and does not use phase-locked loops (PLLs) or delay-locked loops (DLLs). Our serial clock network would require more power than traditional unsynchronized clock trees, but disabling the clock alignment circuitry at run-time minimizes the clock power overhead. Third, serial inter-connections between clock taps allow routing to be introduced to the clock network to reshape clock domains on the fly using the built-in clock alignment circuitry to correct for differences in clock skew. Our approach permits frequency scaling of specific clock domains and localized throttling to relax instantaneous energy consumption, mitigating wide temperature swings. When combined with voltage scaling, this approach can also have a significant impact on chip power.

IV. Single Clock Serial Network

A. Implementation

By using a single or collocated delay lines to transport the reference signals required for averaging, our approach is tolerant to process, temperature and power supply variations of the distributed buffers in a clock network, which is a significant source of clock skew [45]. The non-local mismatch sources are limited to the adjacent clock drivers between taps. A single wire serial clock thread inherently requires less wire length than a similar H-tree based network. The wire length savings does not take into account the control overhead required to program the synchronization circuitry, but this overhead is common for all clock synchronization systems that are programmable at run-time and correct clock skew at each leaf. We can easily incorporate clock gating at each leaf without disrupting the loading of the clock drivers or increasing wiring cost [46]. Using an online closed-loop synchronization approach, our clock network has all the benefits of active skew compensation techniques without the power consumption typical of these techniques.

1) Configuration Variants: The average of the reference clocks at each tap is taken by delaying the forward phase synchronization signal to align with the reverse one through two identically set delay lines. The placement and architecture of these delay lines affect the area required, the matching between delay lines, the system's susceptibility to process variation and the usability of the system. Four such variants are explored for an n-tap single clock structure: one using $2n$ delay lines, one with $n+1$ delay lines, one with n delay lines, and one with $2n$ delay lines using unidirectional conductors for the forward and reverse phase clock signals. The first three variants use a single bidirectional conductor between taps and must be synchronized sequentially and the fourth variant uses a two-wire configuration that can synchronize all taps simultaneously.

a) $2n$ Delay Line Configuration: The first approach for averaging uses a pair of delay lines at each tap to perform clock alignment (see Fig. 6). This method can use a single conductor because the forward and reverse phase clock paths are mutually exclusive to any given tap. Each tap consists of two delay lines, one clock thread switch (see Fig. 4), and one phase detector. During operation, half of the delay lines are idle. An up/down controller uses the phase detector outputs to control the delay setting of each tap and direct the forward and reverse phase clocks between taps. Each tap is aware of whether or not it is synchronizing so the delay setting and up/down signals can be broadcast on a common bus. The accuracy of the average is largely dependent on the matching between the two in-tap delay lines, which is expected to be good considering their proximity.

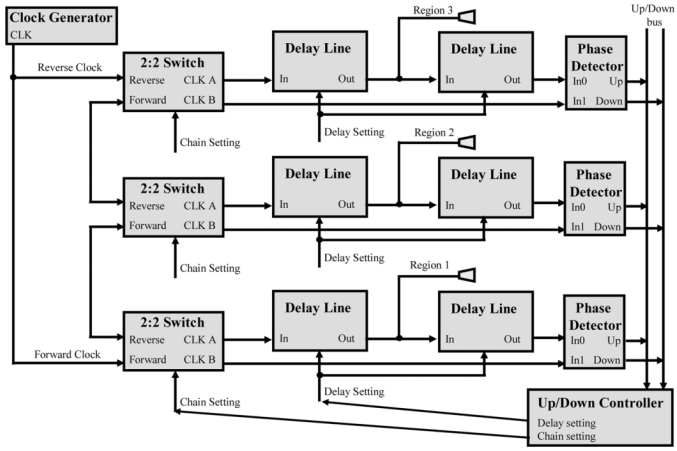

Fig. 6. $2n$ delay line configuration.

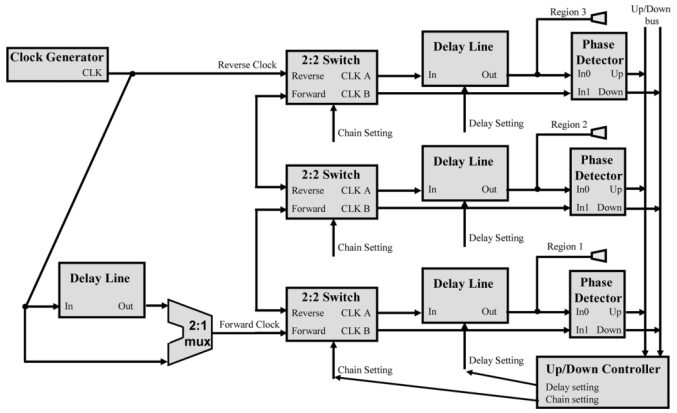

Fig. 7. $n+1$ delay line configuration.

b) $n+1$ Delay Line Configuration: It is possible to share a single delay line at the clock source when synchronizing every tap, requiring only one dedicated delay line at each tap (see Fig. 7). The source delay line is disabled at run-time. Any path inequalities at the clock source affects all taps equally, so it does not alter the clock skew between taps. The source and the tap delay lines must each be programmed separately, slowing down synchronization time. This configuration is also subject to greater mismatch between delay lines than the previous approach.

c) n Delay Line Configuration: The third configuration eliminates the need to match delay lines by reusing a single delay line at each tap to perform the required averaging. The delay line is modified to prevent signal races by converting the 50% duty cycle reference clock to a pulse for the clock synchronization phase. The circuitry is designed to operate

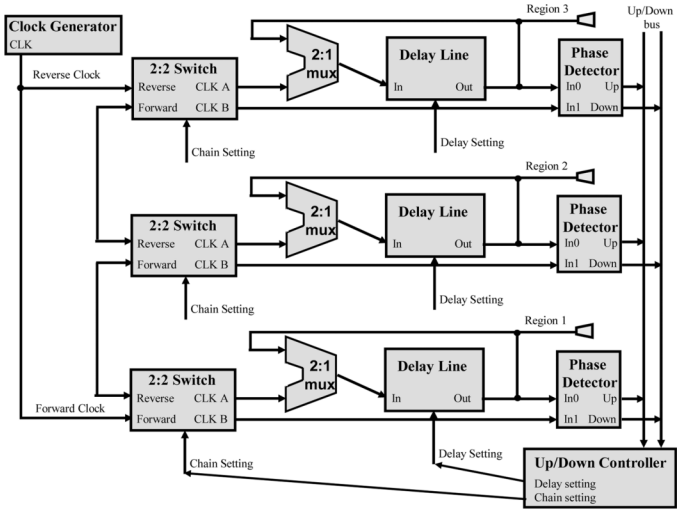

Fig. 8. n delay line configuration.

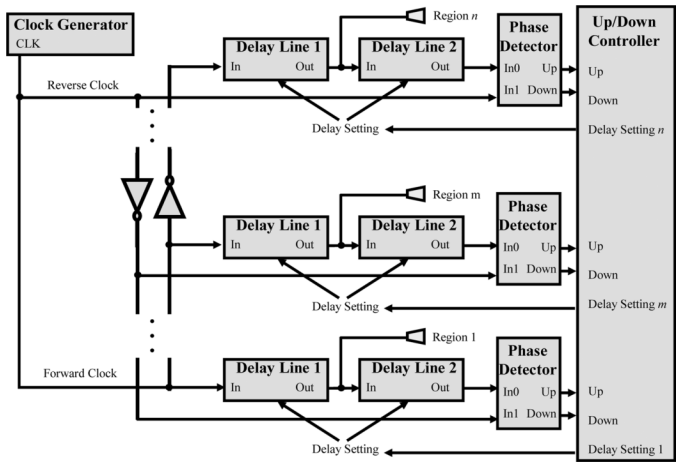

Fig. 9. Hotspot tolerant configuration.

autonomously and asynchronously using signal transitions as cues. Since the rising edges are used for synchronization, the system only modifies the falling edge keeping the rising edge path identical for synchronization and operation (see Fig. 8).

d) Hotspot Tolerant Configuration: Once synchronized, temperature hotspots can appear, shift, or disappear on an IC. Changes in temperature over time require a resynchronization of the entire clock thread to react to the change in operating condition. If pausing the complete system is not desirable, our serial clock network will need to use a two-wire approach with dual delay lines per tap. Since the online synchronization is not necessarily sequential, the up/down controller will need dedicated control signals for each tap or each tap will need to be individually addressed over a common bus. Online skew correction is effective when all taps were previously synchronized and only require fine adjustment around their initial synchronization point. Fig. 9 shows a hotspot tolerant architecture using dedicated control lines. By controlling when delay setting changes are applied to the local clocks, glitches and shortened clock pulses which could potentially cause timing errors can be eliminated [47]. Since this configuration requires more hardware resources than the single wire configurations, it is intended to only be used when the additional functionality it provides is required by the system.

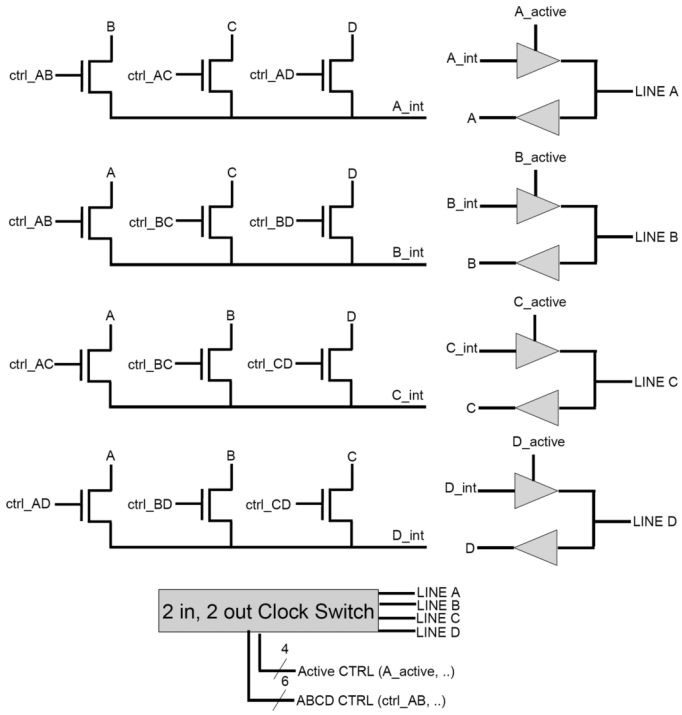

Fig. 10. Four-port clock routing switch.

V. Reconfigurable Networks Based on Averaging

A. Multiple Clock Architectures

Multiple clock designs with independent clock domains are common for designing large modular ASICs and SoCs [48]. Current clock networks are primarily optimized for a single configuration or contain limited amounts of reconfigurability that comes with increased clock skew. In contrast, our serial clock network allows clock domains to be reshaped post-silicon at predefined switch points without introducing any additional clock skew. The configuration of the switch points will determine the extent to which the clock network can be altered. Clock networks can be modified to correct for some manufacturing defects, including bypassing certain clock lines and clock buffers. The approach is compatible with irregularly-shaped distribution areas and simplifies the floorplanning of an integrated circuit.

The most versatile configuration is a serial clock mesh network where each clock tap can connect to one of many clock domains through routing switches. For additional flexibility, a hierarchical approach to clock routing can be used with local and regional switches. These routing switches are designed with equal internal propagation delays between ports. A four-port (two input, two output) clock routing switch is shown in Fig. 10. This switch can be designed with an arbitrary number of ports and differs from the clock thread switch where only two of the ports can be used as inputs. A fully reconfigurable and reprogrammable hierarchical clock network with 96-taps is shown in Fig. 11. The large lightly shaded squares represent local clock regions, the darker shaded squares represent switch points in

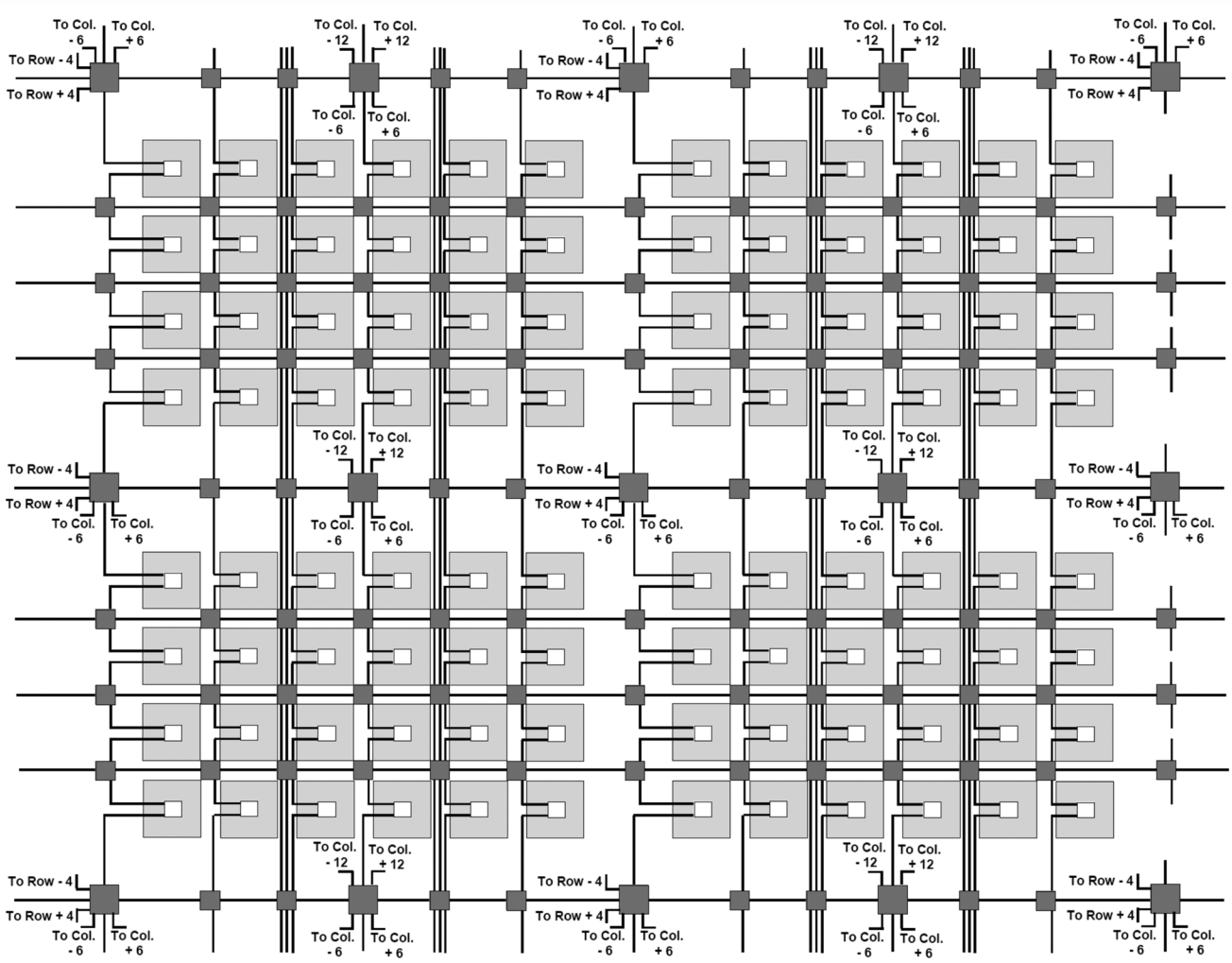

Fig. 11. Mesh architecture with 96 taps incorporating express paths.

the network and the white squares represent each local tap. To reroute clocks during operation, it is necessary to synchronize each modified thread to eliminate skew in the clock domain. Predicted or cached delay settings can be used to configure the clock threads, resulting in fast setup times for the clock network at the expense of a looser skew bound.

B. Implementation Methodology

Our averaging system assumes a hybrid approach, using our technique for global distribution and a mesh, tree, or fishbone locally. Clock routing points can be inserted between the serial links, as required. The number of switches used will depend on the required versatility of the clock network. Fully reconfigurable structures, such as the one shown in Fig. 11 provides many benefits in either an FPGA or an ASIC, including post-silicon correction of some device and interconnect defects. Studies have shown that up to 88% of defective devices could be salvaged with small changes to the circuit's critical paths [49]. Work in [50] shows that increasing the number of global regions in an FPGA can result in significant power savings for the device by allowing fine-grained clock management. Since our technique makes every region a global one, it could exploit this trend.

C. Versatility of a Programmable Clock Mesh Network

A variety of fully programmable clock networks can be created using different combinations of taps and switches. The number of ports within each switch, the number of switches present in the mesh and the number of taps between switches can all be modified to tailor the clock network to a particular application. Fig. 12 shows a 15-tap solution containing a combination of 4- and 8-port switches that allow up to 2 or 4 simultaneous connections, respectively. The taps are divided into three clock domains, A, B, and C. This figure shows the how the taps get assigned to each of the single clock threads. Each clock tap could also be arbitrarily assigned to another clock domain without modifying the hardware by reprogramming the clock switch settings and resynchronizing the clock threads. The network is shown with one tap on each vertical edge between switches, but the taps can be placed anywhere. For an arbitrary $m \times n$ mesh network with c domains, given p_i ports per switch and each switch connected to the clock generator with d_i ports, the number of configurations is

$$\text{configurations (network)} = \left[\frac{1}{2}\left(\sum_{i=1}^{m\cdot n} p_i - d_i\right)\right]^2. \quad (3)$$

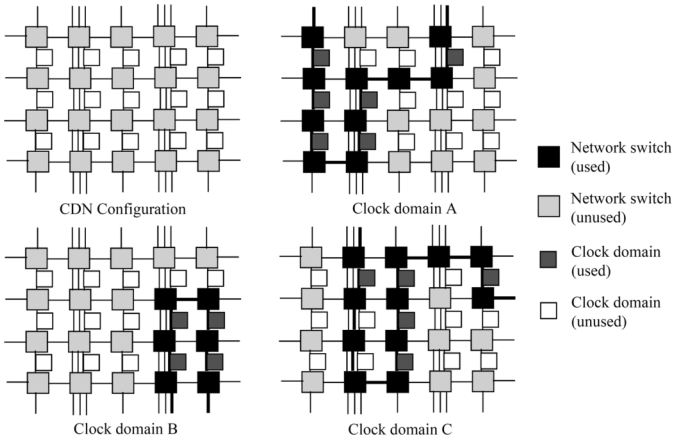

Fig. 12. Fully programmable clocking architecture.

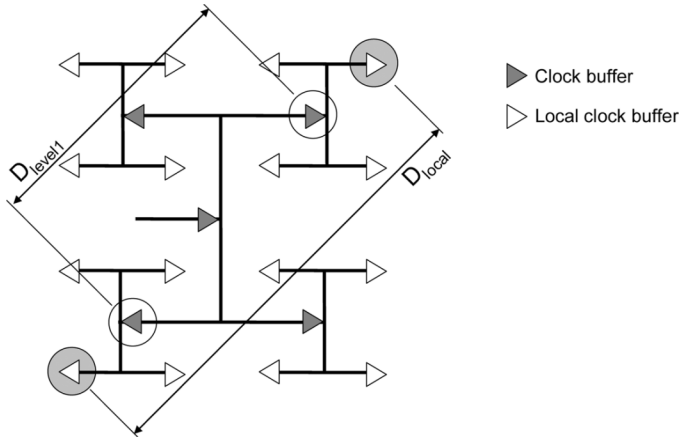

Fig. 13. Separation of clock buffers in a tree network.

D. Process Variance

In a clock network, the process variance can cause an alteration of the expected delay through devices and interconnect. The amount of deviation can be different for ideally identical devices on a given die, so this variance can have a significant effect on clock skew in a clock network. Process variation results in two kinds of mismatch in an integrated circuit. Inter-die mismatch affects all devices on a die equally and does not alter the matching of components on a single die. Intra-die mismatch occurs between different devices on the same chip and has been modeled by Pelgrom *et al.*'s relation for variance due to parameter (P) deviation [51]

$$\sigma^2(\Delta P) = \frac{A_p^2}{WL} + S_p^2 D^2. \tag{4}$$

W and L represent transistor width and length, respectively, and D represents the distance between devices. The S_P^2 term models the distance variance that increases with distance and the A_P^2 term models the discrete variance that always exists between distinct devices. The larger the distance between devices, the greater the potential mismatch, Fig. 13. Smaller transistors result in a discretization of dopant levels and lead to greater fluctuation in threshold voltage and propagation delay. Minimizing process variance requires using sufficiently large transistors to decrease discrete variance and locating devices as close together as possible. Centroid layouts where the geometric centers of devices are minimized can thus be used to decrease device mismatch. Equation (4) can be extended to the following layouts:

$$\sigma^2_{\text{centroid}}(\Delta P) = \frac{A_p^2}{WL} + \frac{S_p^2 D_x^2 D_y^2}{12 D_w^2} \tag{5}$$

where D_x and D_y are the horizontal and vertical distances between devices and D_w is the wafer diameter [52].

Even though process gradients are never perfect planes, co-locating clock buffers will result in better matching. In a serial network, process variance only affects performance when it occurs between forward and reverse direction clock drivers located between adjacent leaves. This is a much smaller distance compared to the dispersed clock buffers typical of other CDNs.

E. Temperature Variation

Small changes in ambient temperature can have a significant effect on transistor performance [53]. Our hotspot tolerant configuration will account for temperature and power supply fluctuations during synchronization better than tree structures since devices that require matching are placed within the same environment [34]. If conditions change during operation, the clock network can be resynchronized to account for this variation.

F. Clock Jitter

Jitter is a random variation of the phase of a clock signal and can change quickly from cycle-to-cycle or slowly over many clock cycles. Traditionally, cycle-to-cycle jitter is the most difficult to predict and potentially the most harmful. The primary source of jitter is power supply noise caused by switching activity [54]. Clock trees are a source of jitter since clock buffers simultaneously switch large capacitances that can negatively affect the power signals.

Jitter can accumulate through each buffering level. Since the noise level will vary depending on the switching activity within a particular region, the jitter across the IC will also vary. As jitter accumulates through many stages, the total jitter will be bounded since jitter sources are Gaussian [55]. Our serial distribution will have less impact on the power distribution system since clock drivers are small and distributed. Analog circuits traditionally used to synchronize clocks are significantly more sensitive to noise than the digital components used in our system [56].

G. Clock Skew

Device and interconnect variance is leading to an ever-increasing amount of fixed uncertainty [57]. Our system is resistant to inter-die mismatch due to online skew compensation. Our serial clock network also corrects for the majority of intra-die variation in the global clock network. Table I summarizes the differences between our serial clock networks and buffered and unbuffered clock trees.

VI. Circuit Implementations

An averaging clock network can contain an arbitrary number of nodes and can be laid out automatically, manually or using

TABLE I
COMPARISON OF SERIAL CLOCK NETWORKS USING AVERAGING WITH BUFFERED AND UNBUFFERED TREES

	Unbuffered clock tree	Buffered clock tree	1-wire serial network	2-wire serial network
Multiple clocks	Requires multiple trees	Requires multiple trees	Re-routable at runtime	Re-routable at runtime
Clock skew	Must be known at design time	Must be known at design time	Corrected in silicon based on averaging	Corrected in silicon based on averaging
Clock leaves (#)	Limited by source driver	Unlimited	Unlimited	Unlimited
Wire length	Long. Requires additional wire to match trace length.	Long. Requires additional wire to traces in H-tree.	Short (excluding control overhead).	Long. 33% longer than comparable H-tree.
Process variation effects	Affected by every corresponding wire segment in a clock domain. Wires are spread over device.	Affected by every corresponding wire segment and clock driver in a clock domain. Wires and devices are spread over device.	Limited to variation between adjacent serial clock buffers.	Limited to variation between adjacent serial clock buffers and co-located inter-leaf wire segments.
Tolerance to temperature variation	No	No	Clock network can be paused and resynchronized to account for changes in interconnect or driver delay.	Clock network can be resynchronized on-the-fly to account for changes in interconnect or driver delay.
Reconfigurable	No	No	Yes	Yes
Power	Not well suited for multiple clock domains with frequency scaling.	Not well suited for multiple clock domains with frequency scaling. Clock gating is possible.	Well suited to low power techniques including multiple clock domains, clock gating and frequency scaling. Power required for control hardware at synch time (static logic at run-time).	Well suited to low power techniques including multiple clock domains, clock gating and frequency scaling. Power required for control hardware at synch time (static logic at run-time).

standard cells. The required components are delay lines, routing circuitry, phase detectors and a synchronization controller, which can be implemented in hardware as a finite state machine, or software. Our serial clock network is designed to be implementation-independent. We discuss here a proof-of-concept implementation in a mature technology to show the average case performance.

The circuitry is entirely digital, small, and easy to port into different technologies. Digital circuits are also much less sensitive to matching than analog ones [58]. All of the circuits are designed and laid out using TSMC's 0.18-μm standard process. Our implementation of the delay lines and phase detectors are discussed in the following sections.

A. Delay Lines

The skew bound of our system is directly related to the minimum delay increment, or resolution between delay line settings. Large delay increments, typical of other digital delay lines, would impact the quality of the clock synchronization. Using analog delay lines could hurt the practicality, size and power consumption of our multi-tap, multi-delay line system [59]. Our delay line achieves equal duty cycles for input and output clocks and uses a coarse/fine grain adjustment system. The coarse delay is achieved using the ladder structure in Fig. 14. Depending on the total fine grain delay needed, we serially connect together an even number of fine grain variable delay inverters shown in Fig. 15. The design scales well and allows for longer delays and lower frequencies whenever necessary. The delay line used for our experimental networks is shown in Fig. 16 and contains one coarse grain cell from Fig. 14 and four fine grain cells from Fig. 15.

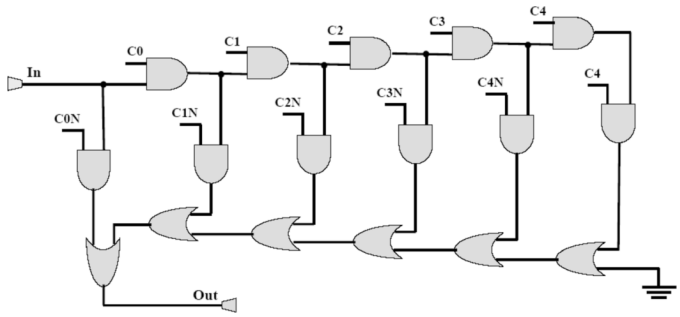

Fig. 14. Scalable coarse grain delay line ladder structure.

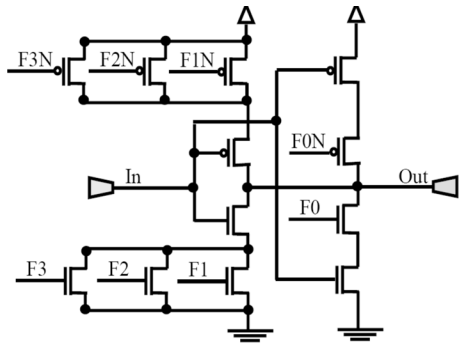

Fig. 15. Fine grain variable delay inverter.

Allowing each fine delay inverter to be programmed individually increases the total number of delay settings and the highest potential resolution, but also requires the most overhead. The number of fine delay inverter groups that can be uniquely programmed (g) and the number of control lines per inverter (c) sets the achievable resolution of the delay line

$$\text{resolution}_{\text{ideal}} = \frac{\text{coarse\_delay}}{2^{c \cdot g}}. \tag{9}$$

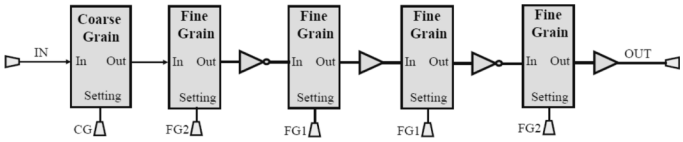

Fig. 16. Heterogeneous delay line.

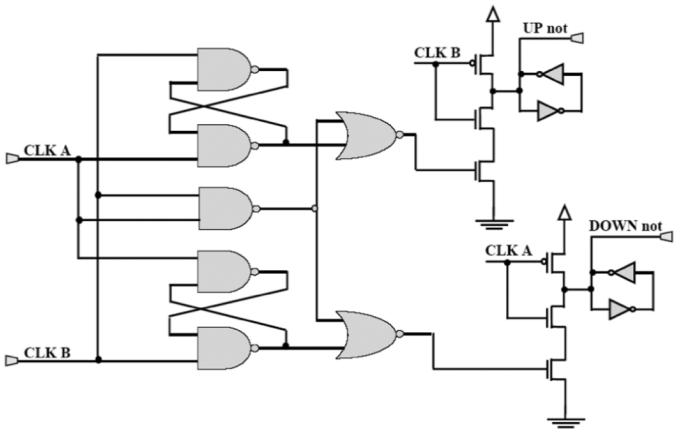

Fig. 17. Fixed tolerance phase detector.

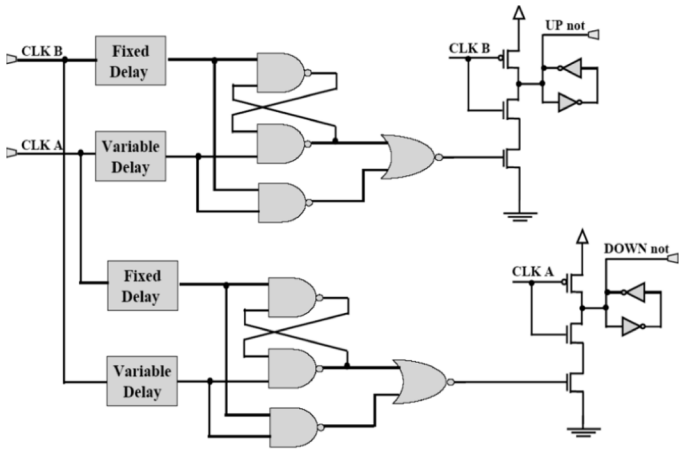

Fig. 18. Variable-tolerance phase detector.

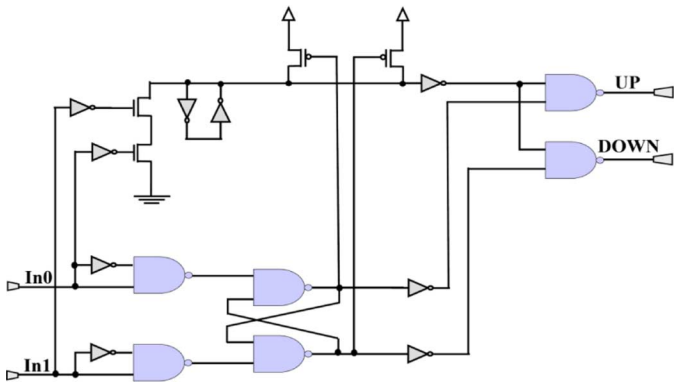

Fig. 19. Modified phase detector for shared delay line systems.

B. Phase Detectors

Our phase detector designs are non-traditional since the target application is one that will have a finite resolution due to the digital delay line so the goal is not to eliminate metastable signals. Instead, the system performs two simultaneous comparisons on slightly skewed versions of the inputs to guarantee that one of the two comparisons will resolve and produce a useful result, differentiating our design from other phase detectors presented in literature [60], [61].

1) Fixed Tolerance Phase Detector: Our phase detector samples the input clocks and retains the result for roughly one-third of a clock cycle. Along with *UP* and *DOWN* detection, it also senses a *LOCKED* condition when the clocks are within a defined skew bound. Like most detectors, the design in Fig. 17 uses a cross-coupled NAND gate latch structure to perform the signal comparison. Two interconnected latches are used in a method similar to [62]. In our case, we modify one latch to be more sensitive to one of the inputs so that if the clock edge inputs are very close and one of the latches enters a metastable state, the other latch will drive the system to a *LOCKED* state.

2) Variable-Tolerance Phase Detector: We have also developed a phase-detector with a variable "locked" width. Our variable-tolerance phase detector delays each of its two input signals using two parallel delay lines. The first is a digitally-controlled delay line (DCDL) and is used to adjust the width of the *LOCKED* region. The second delay line is fixed to the minimum latency of the DCDL, Fig. 18. The cross-coupled NAND gates used for each latch are designed with no input preference. Adjustability is useful since there is little benefit to synchronizing clocks beyond the required skew bound for a circuit.

3) Phase Detector for Shared Delay Line Implementations: The phase detector for the shared delay line taps must handle a forward phase clock with a shortened duty cycle due to the feedback loop used to prevent signal races. There is guaranteed to be a time when both phase detector inputs are low. This both input zero state is used to trigger a detection for the Fig. 19 phase detector. If there is more than one instance where both inputs are zero per clock period, preference is given to the one immediately following the high-to-low transition of the reverse phase clock signal.

C. Controller Notes

Once synchronized, our serial clock network may undergo environmental changes that require resynchronization. We suggest three possibilities for resynchronization: periodic, on-demand, or polled. A periodic resynchronization can be triggered after a user-set period of time. Periodic resynchronization will be fast since it only requires small adjustments to the previous delay setting. The second is an on-demand approach that requires the inclusion of a dummy tap at the end of the clock thread. This tap will always have access to both forward and reverse phase clock signals to monitor alignment. The forward phase clock passes through the complete thread and is affected by all the local environmental and electrical fluctuations in the clock domain. If the phase detector detects a change in alignment, a thread-wide resynchronization can be triggered. A third approach would be to poll each tap sequentially and resynchronize them as required. This method requires the use of the dual conductor hotspot tolerant configuration.

VII. System Performance

A. Controller Models

Our clock networks were modelled using extracted layout simulation data and suitable controllers were implemented in VHDL. These models are designed using a generic approach, but require specific configurations to demonstrate their operation. Controllers for two specific configurations were designed: a 4-tap single clock configuration and a 15-tap three-clock domain reconfigurable configuration, Fig. 12. The results show that the controller/circuitry interface is simple and versatile. When synthesized to an FPGA, the 15-tap reconfigurable controller requires 447 LUTs, 127 registers, and 2048 memory bits in an Altera Stratix II device. The total synchronization time for all taps assuming a 500 MHz controller clock is less than 60 μs.

B. Performance of Delay Lines

The maximum delay of our fine grain line exceeds that of our coarse grain delay line by over 20%, creating robustness and a tolerance to process variability as demonstrated in [62]. The coarse grain delay increment is 188 ps for the schematic version of the circuits and 160 ps for the extracted ones. A six coarse cell delay line was chosen for a maximum clock period of 1920 ps, or a minimum clock frequency of 521 MHz. Additional coarse settings can be added as needed to be used with arbitrarily long clock periods. We chose to have four variable fine delay inverters divided into two uniquely-controlled groups. Since the all zero setting is not allowed for the fine grain inverter, the maximum possible resolution given this configuration is 0.84 ps. Non-linearity in the delay line settings results in a resolution of 5.85 ps for the extracted model of the $2n$ delay line and 5.72 ps for the n delay line configuration. Clock frequencies between 500 MHz and 2 GHz were selected to represent typical clock ranges.

C. Simulation Results for Single Clock System

At synchronization, the skew bound is equal to twice the sum of the phase detector error and the maximum delay increment. Since one of the variable delay elements are removed at run-time, the effective skew is half this total. The expected skew of the system is 7.35 ps, but in practice, it is larger due to duty cycle changes that occur through the fine delay line. Using an $n + 1$ architecture and a fixed tolerance phase detector, the skew bound of an 8-tap single clock averaging clock network is roughly 10 ps for an extracted layout design.

Comparable solutions offer similar or worse levels of skew reduction, sub-10 ps in a 90-nm technology [64], 70 ps in a 0.18-μm technology [2] and 15 ps in a 0.25-μm technology [65]. [66] demonstrates a skew reduction scheme capable of reducing skew to within 10% of the clock period, versus under 4% here for the same feature size. The 0.10 μm design in [1] ideally achieves a 3 ps skew bound in simulation by modifying a traditional H-tree network and using a duplicate co-located return path to synchronize clock leaves. However, it is susceptible to intra-die variation and requires wire overhead for the return signals.

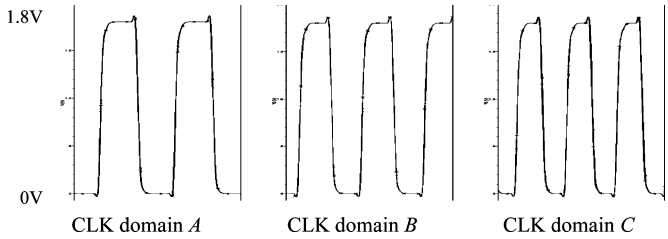

Fig. 20. Clocks in a three-clock domain reconfigurable network.

Simulations show that for a typical configuration, the power consumption of an 8-tap $n + 1$ clock distribution circuit at its maximum frequency is 33.2 mW during the phase alignment cycle and 18.0 mW (2.2 mW per tap) at run-time. In an older 2-μm technology, [67] demonstrates 0.21 mW power consumption per de-skew tap for a 56 ps skew bound. Analog solutions such as those using PLLs typically consume hundreds of mW [39]. Tests on a laid out and extracted 4-tap single clock domain n delay line clock network show an overall skew bound of 12 ps and power consumption of 2.5 mW per tap at 1 GHz.

Each $2n$ delay line tap requires 6000 μm^2, each $n + 1$ tap requires 3750 μm^2 and each shared delay line tap requires 5100 μm^2 in our 0.18-μm technology. This includes fixed tolerance phase detectors which require 500 and 540 μm^2 for the designs in Figs. 17 and 18, respectively. A variable-tolerance phase detector would require 1300 μm^2. The variable-tolerance phase detector can be set to one of 15 skew tolerance settings between 0 and 253 ps.

D. Simulation Results for Reconfigurable System

We simulated the extracted version of the three-clock domain reconfigurable network in Fig. 12 with the 15 taps numbered in column-wise fashion. Taps 1, 2, 3, 5, 6, and 10 are connected to clock domain A with a frequency of 1.11 GHz. Taps 11, 12, 14, and 15 are connected to clock domain B with a frequency of 1.33 GHz. Taps 4, 7, 8, 9, and 13 are connected to clock domain C with a frequency of 1.66 GHz. Fig. 20 shows the synchronized regional clocks produced by our clock network. The total skew from the first-to-last edge is 5.5 ps for domain A, 4.7 ps for domain B and 3.9 ps for domain C. No other comparable clock networks allow the ability to reconfigure clock domains post-silicon as we do here. The total power consumed is 62.82 mW or 4.188 mW per tap. Roughly half of the power is consumed by the routing switches of the reconfiguration circuitry.

E. Measured Results for 6-Tap Fabricated System

We have created a prototype test chip with two independent clock domains divided between six clock regions. The chip was designed to show that reconfiguring clock domains post-silicon is possible and that an averaging approach is effective in eliminating skew in a serial clock network. The 1.5 mm^2 test chip is shown in Fig. 21 and has been designed with a $2n$ delay line structure and two coarse grain delay blocks to permit clock frequencies down to 260 MHz. For a single clock domain, Fig. 22 shows the synchronized waveforms for six 500 MHz regional clocks. The clock network can reconfigure the six local taps into either a single clock domain or a pair of independent ones. The test chip has two dedicated clock inputs (*CLKA* and *CLKB*)

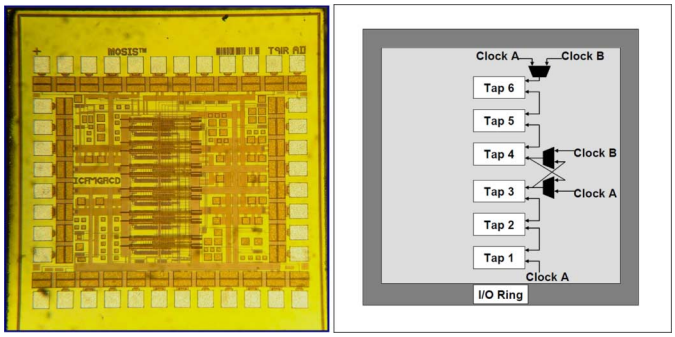

Fig. 21. Photograph and schematic of the six tap test IC.

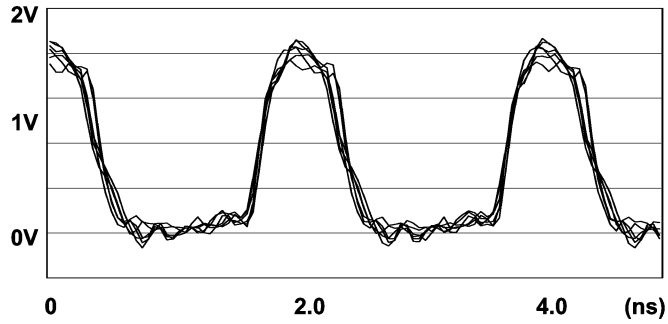

Fig. 22. Synchronized measured regional clock signals.

TABLE II
INPUT CLOCK ASSIGNMENTS FOR TEST CHIP

Single Domain		Dual Upper Domain		Dual Lower Domain	
Forward CLK	CLKA	Forward CLK	CLKB	Forward CLK	CLKA
Reverse CLK	CLKB	Reverse CLK	CLKB	Reverse CLK	CLKA

TABLE III
OUTPUT SIGNAL ASSIGNMENTS FOR TEST CHIP

Port	Upper synching	Lower synching	Run-time
1	Forward tap 1	Forward tap 4	Averaged tap 1
2	Reverse tap 1	Reverse tap 4	Averaged tap 4
3	Forward tap 2	Forward tap 5	Averaged tap 2
4	Reverse tap 2	Reverse tap 5	Averaged tap 5
5	Forward tap 3	Forward tap 6	Averaged tap 3
6	Reverse tap 3	Reverse tap 6	Averaged tap 6
7	Feedback CLKA	Feedback CLKA	Feedback CLKA
8	Feedback CLKB	Feedback CLKB	Feedback CLKB

and eight clock output ports including the two input clock feedback outputs on Ports 7 and 8. Forward, reverse, and averaged clock output signals are required for each tap, so the chip is designed to operate in three output modes: operation, synchronizing upper half and synchronizing lower half. The input clock uses are shown in Table II and the output clock configurations are shown in Table III.

To synchronize a clock domain, our test environment observes the forward and reverse reference clocks off-chip using an oscilloscope and we manually adjust the on-chip delay settings for each local tap using and 8-bit data/address bus to program the configuration registers. The results represent measurements taken from a single chip, although the behavior of all devices tested was similar with small variations in delay line settings required to achieve clock alignment. For a given clock edge, the maximum time variation between regional rising edges is less than 80 ps. This value includes any remaining clock skew present after synchronization and the long and short-term jitter of each of the local clocks. This is a nearly optimal result since the measured maximum cycle-to-cycle jitter of any reference or synchronized regional clock in our system is 79 ps. Care must be taken while aligning each tap to ensure that the reference clock edges read by the phase detector are as close as possible to their statistical average. One solution is to perform multiple tests and use the majority decision. The power consumption of the test chip is not meaningful due to the significant power consumed by the I/O ring. Further, mismatched input and output pin delays result in a duty cycle shift that is reflected in the signal waveforms.

The spacing between clock taps is roughly 200 μm, on average. Due to the daisy-chained nature of a serial network, the skew between taps without any compensation is significantly higher than a comparable clock tree and equivalent to a much larger clock network. By reducing the clock uncertainty to the order of the clock source jitter, we effectively eliminate skew. However, it is important to note that the chip cannot prove that the process variance would always be accounted for since the distances between adjacent drivers in our test chip is not significant enough to produce mismatch on the order of the jitter present in our test chip.

VIII. CONCLUSION

Today, most clock network designs require precise information on the exact clock load for each branch, the placement of each tap on the die and the location of the clock root. With modular and IP-centric design strategies, circuit components may come from many distinct sources and there may be little knowledge concerning internal circuit characteristics available to integrators. Our single and multiple clock reconfigurable clock networks are designed to be oblivious to clock domain shapes and sizes, using an implementation-independent standard cell approach. Using a dual reference signal averaging technique in the clock network allows designers to delay some of the critical clock tuning requirements to facilitate the design flow and speed up time-to-market of designs. It allows circuit blocks to be moved around conveniently and resized easily with a simple change in the number and location of taps. It is a new type of serial clock distribution that exceeds significantly the capabilities of all previous ones.

Using programmable repeaters allow us to redirect clocks post-silicon at certain predefined switch points, making the network reconfigurable. Clock networks can be modified to correct for manufacturing defects, including bypassing selected clock lines and buffers. The operating clock frequency can be changed to fit an IC's many possible target applications. Our system is compatible with irregularly shaped distribution areas and simplifies the IC floorplanning. By using delay lines instead of PLLs and allowing a single synchronization controller to be used for

all taps, the required circuitry is small and unobtrusive and versatile.

The network provides multi-point active skew compensation to correct device mismatch and process variance—enabling our design to have all the benefits of other active clock skew reduction methods. However, since this synchronization circuitry can be disabled at run-time, the system can operate in an open loop to save power—a typical benefit of traditional clock trees. By using selective resynchronization approaches, it is possible to also overcome changes to intra-die thermal gradients and hotspots.

In future, we will use this network with NoCs [68] and novel FPGA designs [69].

ACKNOWLEDGMENT

The authors would like to thank CMC Microsystems for providing access to design and manufacturing resources.

REFERENCES

[1] A. Kapoor, N. Jayakumar, and S. P. Khatri, "A novel clock distribution and dynamic de-skewing methodology," in *Proc. ICCAD*, 2004, pp. 626–631.

[2] H. Lee, H. Q. Nguyen, and D. W. Potter, "Design self-synchronized clock distribution networks in an SoC ASIC using DLL with remote clock feedback," in *Proc. ASICSOC*, 2000, pp. 248–252.

[3] E. G. Friedman, "Clock distribution networks in synchronous digital integrated circuits," *Proc. IEEE*, vol. 89, no. 5, pp. 665–692, May 2001.

[4] S. Tam, S. Rusu, U. Nagarji Desai, R. Kim, J. Zhang, and I. Young, "Clock generation and distribution for the first IA-64 microprocessor," *IEEE J. Solid-State Circuits*, vol. 35, no. 11, pp. 1545–1552, Nov. 2000.

[5] S. Zanella, A. Nardi, A. Neviani, M. Quarantelli, S. Saxena, and C. Guardiani, "Analysis of the impact of process variations on clock skew," *IEEE Trans. Semicond. Manuf.*, vol. 13, no. 4, pp. 401–407, Nov. 2000.

[6] M. Omana, D. Rossi, and C. Metra, "Fast and low-cost clock deskew buffer," in *Proc. DFT*, 2004, pp. 202–210.

[7] V. Varghese, T. Chen, and P. Young, "Stability analysis of active clock deskewing systems using a control theoretic approach," in *Proc. ACC*, 2005, vol. 3, pp. 1758–1763.

[8] D. Matzke, "Will physical scalability sabotage performance gains?," *Computer*, vol. 30, no. 9, pp. 37–39, Sep. 1997.

[9] M. A. Karami, A. Afzali-Kusha, R. Faraji-Dana, and M. Rostami, "Quantitative comparison of optical and electrical H, X, and Y clock distribution networks," in *Proc. ISVLSI*, 2007, pp. 488–489.

[10] A. Chattopadhyay, "Dual reference signal post-silicon reconfigurable clock distribution networks," Ph.D. dissertation, Dept. Elect. Comput. Eng., , McGill Univ., Montreal, QC, Canada, 2009.

[11] M. A. El-Moursy and E. G. Friedman, "Exponentially tapered H-tree clock distribution networks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 8, pp. 971–975, Aug. 2005.

[12] A. Iyer and D. Marculescu, "Power efficiency of voltage scaling in multiple clock multiple voltage cores," in *Proc. ICCAD*, 2002, pp. 379–386.

[13] C. Yeh, G. Wilke, H. Chen, S. Reddy, H. Nguyen, T. Miyoshi, W. Walker, and R. Murgai, "Clock distribution architectures: A comparative study," in *Proc. ISQED*, 2006, pp. 85–91.

[14] W.-C. D. Lam, J. Jam, C.-K. Koh, V. Balakrishnan, and Y. Chen, "Statistical based link insertion for robust clock network design," in *Proc. ICCAD*, 2005, pp. 588–591.

[15] R. Saeidi and N. Masoumi, "Clock skew reduction by link-region technique," in *Proc. MWSCAS*, 2006, pp. 213–216.

[16] G. Venkataraman, N. Jayakumar, J. Hu, P. Li, S. Khatri, A. Rajaram, P. McGuinness, and C. Alpert, "Practical techniques to reduce skew and its variations in buffered clock networks," in *Proc. ICCAD*, 2005, pp. 592–596.

[17] A. J. Drake, K. J. Nowka, T. Y. Nguyen, J. L. Burns, and R. B. Brown, "Resonant clocking using distributed parasitic capacitance," *IEEE J. Solid-State Circuits*, vol. 39, no. 9, pp. 1520–1528, Sep. 2004.

[18] B. Mesgarzadeh, M. Hansson, and A. Alvandpour, "Low-power bufferless resonant clock distribution networks," in *Proc. MWSCAS*, 2007, pp. 960–963.

[19] Cyclos Semiconductor Inc., "Elizabeth resonant-clocked ARM926EJ-S (information brochure)," 2008 [Online]. Available: http://www.cyclos-semi.com/NewFiles/ElizLitFront.pdf

[20] V. S. Sathe, J. C. Kao, and M. C. Papaefthymiou, "Resonant-clock latch-based design," *IEEE J. Solid-State Circuits*, vol. 43, no. 4, pp. 864–873, Apr. 2008.

[21] M. Hansson, B. Mesgarzadeh, and A. Alvandpour, "1.56 GHz on-chip resonant clocking in 130 nm CMOS," in *Proc. CICC*, 2006, pp. 241–244.

[22] S. C. Chan, K. L. Shepard, and P. J. Restle, "Design of resonant global clock distributions," in *Proc. ICCD*, 2003, pp. 248–253.

[23] V. H. Cordero and S. P. Khatri, "Clock distribution scheme using coplanar transmission lines," in *Proc. DATE*, 2008, pp. 985–990.

[24] G. Venkataraman, J. Hu, and F. Liu, "Integrated placement and skew optimization for rotary clocking," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 2, pp. 149–158, Feb. 2007.

[25] Z. Yu and X. Liu, "Low-power rotary clock array design," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 15, no. 1, pp. 5–12, Jan. 2007.

[26] A. L. Sobczyk, A. W. Luczyk, and W. A. Pleskacz, "Power dissipation in basic global clock distribution networks," in *Proc. DDECS*, 2007, pp. 1–4.

[27] G. Tosik, L. M. S. Gallego, and Z. Lisik, "Different approaches for clock skew analysis in present and future synchronous IC's," in *Proc. EUROCON*, 2007, pp. 1227–1232.

[28] S. Lin and C. K. Wong, "Process-variation-tolerant clock skew minimization," in *Proc. ICCAD*, 1994, pp. 284–288.

[29] W. D. Grover, J. Brown, T. Friesen, and S. Marsh, "All-digital multipoint adaptive delay compensation circuit for low skew clock distribution," *Electron. Lett.*, vol. 31, no. 23, pp. 1996–1998, Nov. 9, 1995.

[30] T. Knight and H. M. Wu, "A method for skew-free distribution of digital signals using matched variable delay lines," in *Proc. ISVLSI*, 1993, pp. 19–20.

[31] V. Prodanov and M. Banu, "GHz serial passive clock distribution in VLSI using bidirectional signaling," in *Proc. CICC*, 2006, pp. 285–288.

[32] S. Hassoun and C. J. Alpert, "Optimal path routing in single- and multiple-clock domain systems," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 22, no. 11, pp. 1580–1588, Nov. 2003.

[33] M. Nekili, Y. Savaria, and G. Bois, "Minimizing process-induced skew using delay tuning," in *Proc. ISCAS*, 2001, vol. 4, pp. 426–429.

[34] S. A. Tawfik and V. Kursun, "Low-power low-voltage hot-spot tolerant clocking with suppressed skew," in *Proc. ISCAS*, 2007, pp. 645–648.

[35] D. P. Dimitrov, "Deep-submicron MOS transistor matching: A case study," in *Proc. DDECS*, 2008, pp. 1–4.

[36] L. Jieyi, K. J. Chun, S. O. Memik, and Y. Ismail, "A self-adjusting clock tree architecture to cope with temperature variations," in *Proc. ICCAD*, 2007, pp. 75–82.

[37] K. Duraisami, P. Sithambaram, A. Sathanur, A. Macii, E. Macii, and M. Poncino, "Design exploration of a thermal management unit for dynamic control of temperature-induced clock skew," in *Proc. ISCAS*, 2007, pp. 1061–1064.

[38] M. Saint-Laurent, M. Swaminathan, and J. D. Meindl, "On the micro-architectural impact of clock distribution using multiple PLLs," in *Proc. ICCD*, 2001, pp. 214–220.

[39] T. Yamashita, T. Fujimoto, and K. Ishibashi, "A dynamic clock skew compensation circuit technique for low power clock distribution.," in *Proc. ICICDT*, 2005, pp. 7–10.

[40] G. Tosik, F. Abramowicz, Z. Lisik, and F. Gaffiot, "Clock skew analysis in optical clock distribution network," in *Proc. CADSM*, 2007, pp. 422–425.

[41] N. Nedovic and V. G. Oklobdzija, "Dual-edge triggered storage elements and clocking strategy for low-power systems," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 5, pp. 577–590, May 2005.

[42] R. M. Rao, R. B. Brown, K. J. Nowka, and J. L. Burns, "Analysis and mitigation of CMOS gate leakage," in *Proc. 5th Adv. Studies Conf.*, 2004, pp. 7–11.

[43] D. Duarte, N. Vijaykrishnan, M. J. Irwin, H.-S. Kim, and G. McFarland, "Impact of scaling on the effectiveness of dynamic power reduction schemes," in *Proc. ICCD*, 2002, pp. 382–387.

[44] P. Zarkesh-Ha and J. D. Meindl, "Optimum chip clock distribution networks," in *Proc. ICIT*, 1999, pp. 18–20.

[45] M. M. Gourary, S. G. Rusakov, S. L. Ulyanov, M. M. Zharov, K. K. Gullapalli, and B. J. Mulvaney, "Approximation approach for timing jitter characterization in circuit simulators," in *Proc. DATE*, 2003, pp. 156–161.

[46] W. Qi and S. Roy, "Power minimization by clock root gating," in *Proc. ASP-DAC*, 2003, pp. 249–254.
[47] A. Chattopadhyay and Z. Zilic, "GALDS: A complete framework for designing multiclock ASICs and SoCs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 6, pp. 641–654, Jun. 2005.
[48] R. Watn, T. Njolstad, F. Berntsen, and J. F. Lonnum, "Independent clocks for peripheral modules in system-on-chip design," in *Proc. SOC*, 2003, pp. 25–28.
[49] S. Sivaswamy and K. Bazargan, "Statistical generic and chip-specific skew assignment for improving timing yield of FPGAs," in *Proc. FPL*, 2007, pp. 429–434.
[50] J. Lamoureux and S. J. E. Wilton, "Clock-aware placement for FPGAs," in *Proc. FPL*, 2007, pp. 124–131.
[51] M. J. M. Pelgrom, A. C. J. Duinmaijer, and A. P. G. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, Oct. 1989.
[52] B. Linares-Barranco and T. Serrano-Gotarredona, "Cheap and easy systematic CMOS transistor mismatch characterization," in *Proc. ISCAS*, 1998, vol. 2, pp. 466–469.
[53] A. Chattopadhyay and Z. Zilic, "Design and operating characteristics of a reconfigurable clock distribution network," in *Proc. NEWCAS*, 2007, pp. 9–12.
[54] S. A. Bota, J. L. Rossello, C. de Benito, A. Keshavarzi, and J. Segura, "Impact of thermal gradients on clock skew and testing," *IEEE Des. Test Comput.*, vol. 23, no. 5, pp. 412–424, 2006.
[55] J. Jinwook, O. Franza, and W. Burleson, "Period jitter estimation in global clock trees," in *Proc. SPI*, 2008, pp. 1–4.
[56] P. Heydari, "Characterizing the effects of the PLL jitter due to substrate noise in discrete-time delta-sigma modulators," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 52, no. 6, pp. 1073–1085, Jun. 2005.
[57] A. Maxim and M. Gheorghe, "A novel physical based model of deep-submicron CMOS transistors mismatch for Monte Carlo SPICE simulation," in *Proc. ISCAS*, 2001, vol. 5, pp. 511–514.
[58] M. Nekili, Y. Savaria, and G. Bois, "Design of clock distribution networks in presence of process variations," in *Proc. GLSVLSI*, 1998, pp. 95–102.
[59] P. Joonbae, K. Yido, and K. Wonchan, "A semi-digital delay locked loop for clock skew minimization," in *Proc. VLSI*, 1999, pp. 584–588.
[60] J. Zhou, D. J. Kinniment, C. E. Dike, G. Russell, and A. V. Yakovlev, "On-chip measurement of deep metastability in synchronizers," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 550–557, Feb. 2008.
[61] M. Renaud and Y. Savaria, "A linear phase detector for arbitrary clock signals," in *Proc. ISCAS*, 2002, pp. IV-775–IV-778.
[62] S. Soliman, F. Yuan, and K. Raahemifar, "An overview of design techniques for CMOS phase detectors," in *Proc. ISCAS*, 2002, vol. 5, pp. V-457–V-460.
[63] P. Raha, S. Randall, R. Jennings, B. Helmick, A. Amerasekera, and B. Haroun, "A robust digital delay line architecture in a 0.13 μm CMOS technology node for reduced design and process sensitivities," in *Proc. ISQED*, 2002, pp. 148–153.
[64] B. Doyle, P. Mahoney, E. Fetzer, and S. Naffziger, "Clock distribution on a dual-core, multi-threaded Itanium family microprocessor," in *Proc. ICICDT*, 2005, pp. 1–6.
[65] G. Geannopoulos and X. Dai, "An adaptive digital deskewing circuit for clock distribution networks," in *Proc. ISSCC*, 1998, pp. 400–401.
[66] M. Omana, D. Rossi, and C. Metra, "Low cost scheme for on-line clock skew compensation," in *Proc. VTS*, 2005, pp. 90–95.
[67] H.-Y. Hsieh, W. Liu, M. Clements, and P. Franzon, "Self-calibrating clock distribution with scheduled skews," in *Proc. ISCAS*, 1998, pp. 470–473.
[68] S. Bourduas and Z. Zilic, "A hybrid mesh/ring interconnect for network-on-chip using hierarchical rings for global routing," in *Proc. ACM/IEEE Int. Symp. Netw.-on-Chip (NOCS)*, May 2007, pp. 195–204.
[69] Z. Zilic and Z. G. Vranesic, "Multiple valued logic in FPGAs," in *Proc. 36th Midw. Symp. Circuits Syst.*, 1993, pp. 1553–1556.

Atanu Chattopadhyay (M'10) received the Ph.D. degree in electrical and computer engineering from McGill University, Montreal, QC, Canada, in 2009 and the M.Eng. degree in electrical and computer engineering from McGill University, Montreal, QC, Canada, in 2003.

He is currently an ASIC Development Engineer with PMC-Sierra, Montreal, QC, Canada. Previously, he worked as an FPGA Development Engineer with MDA Corporation in Sainte-Anne-de-Bellevue, QC, Canada. He is the author of several papers in the areas of asynchronous structures, and clock management and distribution.

Zeljko Zilic (M'97–SM'07) received the Dipl. Ing. degree in computer engineering from the University of Zagreb, Zagreb, Croatia, in 1989, and the M.Sc. degree in computer and electrical engineering and Ph.D. degree from the University of Toronto, Toronto, ON, Canada, in 1993 and 1997, respectively.

He is currently an Associate Professor with McGill University, Montreal, QC, Canada. He is researching various aspects of system design, test, and verification. He has published over 200 research papers, holds four patents in the area of clock and power management and has coauthored three books on verification, validation, and test. From 1997 to 1998, he was a Member of Technical Staff at FPGA Division, Microelectronics Group of Lucent Technologies.

Dr. Zilic is a recipient of a Chercheur Strategique research chair from province of Quebec, the Myril B. Reed Best Paper Award from IEEE International Midwest Symposium on Circuits and Systems in 2001, a Best Paper Award from Design and Verification Conference, DVCon 2005, and several Honorary Mention Awards. For his teaching, the National Council of Deans of Engineering and Applied Science and Sandford Fleming Foundation awarded Dr. Zilic with Wighton Fellowship in 2006.