

Dynamically De-Skewable Clock Distribution Methodology

Arjun Kapoor, Nikhil Jayakumar, and Sunil P. Khatri

Abstract—In a typical clock distribution scheme, a central clock signal is distributed to several sites on the integrated circuit (IC). Local regenerators at these sites buffer the clock signal for the logic in regions close to the regenerator. Minimizing the skew between the clocks at these regeneration sites is critical. In recent times, this is becoming harder due to increasing intra-die processing variations. In this paper, we describe a novel technique to distribute a clock signal from a central location to several sites on a VLSI IC. Our technique uses a buffered H-tree and includes circuitry to dynamically remove any skew that may result due to intra-die processing variations. While existing approaches to deskewing a clock tree have utilized several phase detection circuits (number of phase detectors dependent on the number of clock regenerators), our method requires only one phase detector. Also, in our approach, the resolution of the phase detector is inconsequential unlike existing techniques. Our deskewing technique can be applied dynamically, either at boot time or periodically during the operation of the IC. Using a six-level H-tree clock distribution network with process variations deliberately included, we demonstrate that our technique can reduce skews as high as 300 ps down to just 3 ps. We compare our clock tree with traditional buffered and unbuffered H-tree networks.

Index Terms—Clocks, CMOS, integrated circuits, synchronization, very large scale integration, .

I. INTRODUCTION

IN SYNCHRONOUS integrated circuit (IC) design, it is critical to ensure that the clock signals at different points on the die are in phase. Incorrect circuit operation can occur if the clock signals at the different clock distribution end-points are not in phase.

In a typical synchronous design, a central buffered clock is distributed to several *clock distribution sites* which are located uniformly across the die. The raw clock signal at each of these sites is then buffered using *clock regenerators*. Clock regenerators are designed to drive the clock signal for the circuitry in a local region. Depending on the total clock load in that region, an appropriately sized clock regenerator is utilized. A typical IC will have several clock regenerators for a designer to select from. One of the clock regenerators' signals is considered as a reference signal for the die, and is phase locked to the external clock, thus ensuring that the external clock and all internal clock signals are in phase (all internal clock signals are in phase if the

clock distribution network as well as all clock regenerators are designed to have zero skew).

Designing the clock distribution network has always been a critical part of the design of a VLSI chip. The design of the clock distribution network has become more challenging in recent times due to increasing die sizes and also due to increasing clock frequencies. Intra-die processing variations have compounded the difficulties involved in the design of a clock distribution network. Clock skew can significantly deteriorate the performance of a high-speed IC [1]. There are two kinds of clock skews [2], static skews due to capacitive load and clock path length mismatches, and dynamic skews due to process, voltage, and temperature variations. Methods to design balanced clock distribution networks have been reported [3], but these methods result in zero skew only when there are no process, temperature, or supply voltage variations. Recently, there has been serious attention devoted to de-skewing a clock distribution network, since intra-die process variations make it impossible to distribute a low-skew clock signal without skew reduction circuitry.

In this paper, we describe a novel skew reduction technique that requires *exactly one* phase detector, located at the source node of the clock tree. Our technique can de-skew a clock distribution network to a much finer tolerance than existing techniques. Clock shielding signals (which are connected to GND during normal operation) are, in de-skewing mode, utilized to carry the return clock signal as well as to perform serial signaling to program the skew control logic at different clock distribution sites. Our technique can be used to perform dynamic de-skewing. It can be invoked at boot time, or periodically during the operation of the chip. This could be useful if large temperature variations on the die during normal operation warrant an invocation of the de-skewing approach. Our technique can also be easily modified to allow de-skewing to be performed during factory test.

The remainder of this paper is organized as follows. Section II discusses some previous work in this area. In Section III, we describe our new method of constructing and de-skewing the clock tree network. In Section IV, we present experimental results comparing our method with a traditional H-tree. Conclusions are drawn in Section V.

An extended abstract of this work can be found in [4]. However, there are key differences in the construction of the clock network. These are discussed in greater detail in Section III.

II. PREVIOUS WORK

Traditionally, there are several methods to distribute clock signals. The simplest is the star topology [5]. In this method, a central clock signal is driven to n points using n separate wires.

Manuscript received February 14, 2007; revised July 12, 2007. Published August 20, 2008 (projected).

A. Kapoor is with Sandisk Corporation, Milpitas, CA 95035 USA.

N. Jayakumar is with the Texas Instruments Inc., Dallas, TX 75243 USA (e-mail: nikhil@ece.tamu.edu).

S. P. Khatri is with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: sunilkhatri@tamu.edu).

Digital Object Identifier 10.1109/TVLSI.2008.2000729

These wires are identically sized, ensuring a zero skew clock distribution network. A popular clock distribution methodology is the *H-tree network* [6]. In this style of clock distribution, the network appears much like a “recursive” letter H. The number of branches in such a network is referred to as the *level* of the network. A k level H-tree network has 2^k endpoints or *leaf nodes*.

In a typical H-tree clock distribution network, the signal is not locally buffered before the endpoints. This is done to keep the clock skew at the endpoints to a minimum, with the rationale that with an unbuffered network, we would only have to deal with variations in metal width, thickness, and interlayer dielectric (ILD) thickness. With a locally buffered distribution network, intra-die processing variations in the transistors could cause significant additional clock skew at the endpoints.

With increasing die-sizes, and the large intra-die process variations, it is becoming increasingly hard to distribute a clock reliably. In [7], Zarkesh-Ha *et al.* describe an analytical model and closed-form expression for on-chip clock skew based on device and interconnect variations.

Rajaram *et al.* [8] propose adding crosslinks statically during clock tree construction to help reduce skew and skew variation. While this technique does help reduce skew variation (at the expense of increased capacitive load), there is no dynamic compensation for clock skew due to effects such as temperature and supply voltage variations.

In [9], Tam *et al.* describe a dynamic de-skewing scheme applied to a H-tree network. They use capacitively controlled delay lines to phase-lock clock signals at the leaves of a H-tree with a reference clock signal. In [2], Maxim uses the same kind of distribution as [9], but proposes using analog phase interpolation instead of capacitively controlled delay lines for the de-skewing. In both [9] and [2], the reference clock signal is assumed to be distributed alongside the core clock in such a manner that its skew is very low. Tam *et al.* [9] report that their de-skew mechanism helped reduce clock skew from 110 to 28 ps. The overall skew of [9] and [2] is dependent on the skew of the reference clock signal, mismatches in the feedback clock paths and the guardband of the phase detectors used for each feedback clock signal. Thus, the worst-case skew of such a scheme would be $S_{\text{ref}} + D$, where S_{ref} is the skew of the reference clock signal and D is the guard-band (tolerance) of the phase detectors. In contrast, in our scheme, while we do generate a reference clock signal, we do not have to distribute it through a low skew network. Also, we use a single phase detector which eliminates any skew due to phase detector mismatches. As a consequence, the maximum skew in our scheme is much lower.

In [10], Dike *et al.* discuss two methods to de-skew a clock distribution network: the H-tree de-skewing structure and the Mesh de-skewing approach. In the H-tree de-skewing structure, the de-skewing is done in a hierarchical fashion using phase detectors that are located on the domain boundaries of each leg of the H-tree. Each phase detector reduces the skew between its measurement points within a certain guard-band D . Hence, the skew at each leg of the clock tree is kept within this guard-band. The problem with this approach is that it is possible for the skew between two neighboring leaves to be as high as $(2n+1)D$ for a H-tree with n hierarchical levels. In the Mesh de-skew structure [10], Dike *et al.* use phase detectors between each pair of leaf

TABLE I
SUMMARY OF SKEW TOLERANCES OF VARIOUS DE-SKEWING APPROACHES

Approach	Worst case skew
[9], [2]	$S_{\text{ref}} + D$
[10] H-tree	$(2n+1)D$
[10] Mesh	kD
Ours	d

nodes of the H-tree. This ensures that the clock skew between neighboring leaves is within one guard-band D . However, the maximum skew across a chip (from a leaf at one corner of the chip to the leaf at the opposite corner of the chip) can be as high kD , where k is the number of clock domains between two leaves. The value of k would be 5 for a square die with 16 clock domains, while for a square die with 64 clock domains, the value of k would be 13. Another disadvantage of this method is that the required number of phase detectors grow with the number of levels in the H-tree. With 16 clock domains, this method requires 24 phase detectors, while 64 clock domains would require 112 phase detectors. Dike *et al.* [10] report that their Mesh de-skewing technique reduced global clock skew from 23 to 13 ps and local clock skew from 11 to 3 ps. However, the worst case skew could be much higher depending on the the value of the number of clock domains between two leaves, k ($k = 13$ for a square die with 64 clock domains) and the guardband of the phase detector, D (typical value of $D = 3\text{--}12$ ps).

Another clock de-skewing methodology [11] utilizes a similar idea, equalizing the delay between two spines of the clock distribution network, using signals from a single node from each domain of a representative pair of clock domains. A reduction in skew from 60 to 15 ps is reported in [11]. With increasing die-sizes and intra-die processing variations, such an approach is likely to be inadequate to de-skew clock distribution networks for future designs, unless the clock domains are made smaller and/or more leaf nodes are sampled in the process.

Unlike the two schemes presented in [10] which achieve a maximum chip-level skew of $(2n+1)D$ and kD , the maximum chip-level skew in our scheme is equal to smallest delay increment possible (using tuneable capacitor banks at the leaf nodes of the clock tree)— d . In our case $d = 3$ ps. Using the de-skewing technique, we can reduce the clock skew of a network from 300 ps down to 3 ps. Table I compares the worst case skew of previous approaches and our approach.

III. OUR APPROACH

Our clock de-skewing methodology uses an appropriately delayed reference clock signal (generated by delaying the clock source signal). We match the delay (using tuneable capacitor banks) of each leaf node of the clock tree to this reference signal. This matching of the delay is indicated by a simple phase detector. If we assume that the phase detector has a resolution/guardband of D ps, the phase detector trips (signals that the delays match) when the returned clock signal of a leaf node is within D ps of the reference signal. Note that unlike [9] and [2] that distribute a global reference clock signal in a low skew manner, in our approach, the reference clock signal remains at the center of the clock tree. Since we match the signals from each node of the clock tree to the same reference signal, at the

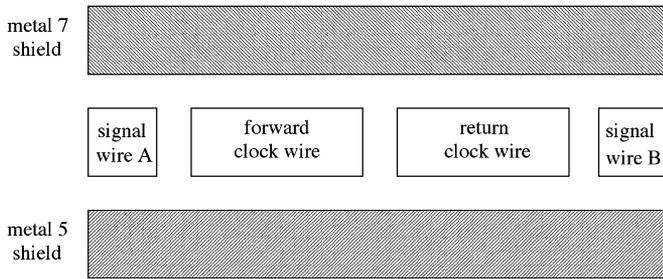


Fig. 1. Cross section of clock, return clock, and serial signaling wires.

end of the de-skewing process, the clock signals returned from all the leaf nodes of the clock tree are brought to within D ps of the reference clock signal. As a result, the minimum phase resolution of the phase detector is not of consequence. Since we use only one phase detector (located at the center of the clock tree), we also need a mechanism for the clock signal to return to its source. We ensure that this return path encounters the same electrical environment as the forward path. This is done by routing the return path alongside the forward path as shown in Fig. 1. The forward and return networks have identical wire sizes and are buffered using tri-stated inverters. The tri-stated inverters along the return path are identical to those in the forward clock path, and are located at exactly the same locations as in the forward clock path. This allows us to balance the forward and reverse path delays. The tri-state functionality is not used in the forward network, but only used in the return network in de-skewing mode of operation (to ensure that at a given time, a *single path* is returned to the phase detector for de-skewing).

Clock trees in current VLSI ICs are buffered due to the fact that the IC dies are growing larger. However, the addition of these buffers worsens the skew problem and hence the use of these clock buffers is usually minimized. Since we de-skew the signal at all clock distribution sites, we can utilize a buffered distribution network (the de-skewing functionality erases the skew that is introduced due to intra-die process variations that affect the delay of the tri-stateable inverters).

Fig. 1 shows a cross-section of our clock tree network. The wires labeled *A* and *B* are signal wires. These signal wires are used during the de-skewing operation to control the tri-stateable inverters and the tune-able capacitor banks. The signal wires *A* and *B*, along with the return clock wire, are held low during normal circuit operation, and act as clock shields. This shielding of the clock wires also helps reduce inductance of the clock wire since the return path (GND) is now right next to the clock wire. The signal wire *A* is used for clocking the serial control logic required during de-skewing operations, while wire *B* is used to transmit serial data to the controllers. Serial controllers (located at the tri-state buffer sites as well as the skew adjustment banks) manipulate the tri-stateable return drivers in the clock return path that is enabled during de-skewing, and also update the skew adjustment capacitors.

The delay of the forward and return networks are identical since their wires and tri-stateable inverters are identical and colocated. This results in the intra-die processing variations being highly correlated in both networks. Additionally, identical tuneable capacitor banks are placed in both the forward and

return paths at each leaf node of the clock tree. The capacitor banks are connected such that the same value of the capacitor is switched-in, in both the forward and return capacitor banks. All this ensures that the delay of the forward network is at all times equal to half the round-trip delay. If the smallest delay increment in the tuneable buffer is d , the largest skew between any two leaf nodes of the clock tree network is equal to $d/2$. Hence, in our approach, the tolerance to which two signals can be de-skewed is simply equal to the smallest delay increment that can be achieved with the tune-able capacitor bank and independent of the resolution of the phase detector (unlike previous approaches to clock de-skewing [7], [10]).

With our approach, the forward clock distribution network has tri-state buffers. These can be used to perform clock gating [6]. Since the forward network is buffered, clock gating will not result in any skew variations at the buffered H-tree leaves. This would result in additional power reductions than traditional clock gating (since portions of the buffered H-tree are also disabled).

A. Network Topology

In our approach, we construct an H-tree *forward* clock distribution network with an identical, colocated clock *return* network. The return network wire acts as a shielding wire during normal operation (at which time it is tied to ground). The fact that we de-skew the signal at all leaf nodes of the network allows us to utilize a buffered distribution network, since any skew introduced due to the buffers will be negated once the nodes are de-skewed. We ensure that each buffered segment of both networks has identical load characteristics. This is beneficial since each forward or return segment can be driven by identically sized tri-stateable inverters to locally buffer the signal. For the return path, the buffering inverters used are tri-stateable. This is a requirement since while de-skewing a particular leaf, we need to turn on *only* the return path from that leaf and turn off all other return paths to prevent drive contention. To balance the forward and return path delays, we use identical tri-stateable inverters in the forward path as well, although the tri-stateability of the forward path tri-stateable buffers is not utilized. We observe that the majority of the delay in either path is due to wiring, hence small differences in the loads due to devices in the forward versus return paths are not of consequence. The delays of the forward and return networks are made identical by colocating their (identically sized) wires and their (identical) tri-stated inverters, resulting in very tightly correlated delays in both networks even in the presence of intra-die processing variations.

Shielding wires are commonly used in present day clock distribution networks. In our approach, we use the shielding wires for de-skew control and also to return the clock signal in skew adjustment mode. The serial control signals (which control the de-skewing capacitor banks at each endpoint of the clock distribution tree as well as the tri-state inverters on the return network) are routed alongside the forward and return clock path (see Fig. 1).

A control line and the return clock line are placed on either side of the forward clock line. During normal operation these lines are connected to GND to act as shields for the clock. In

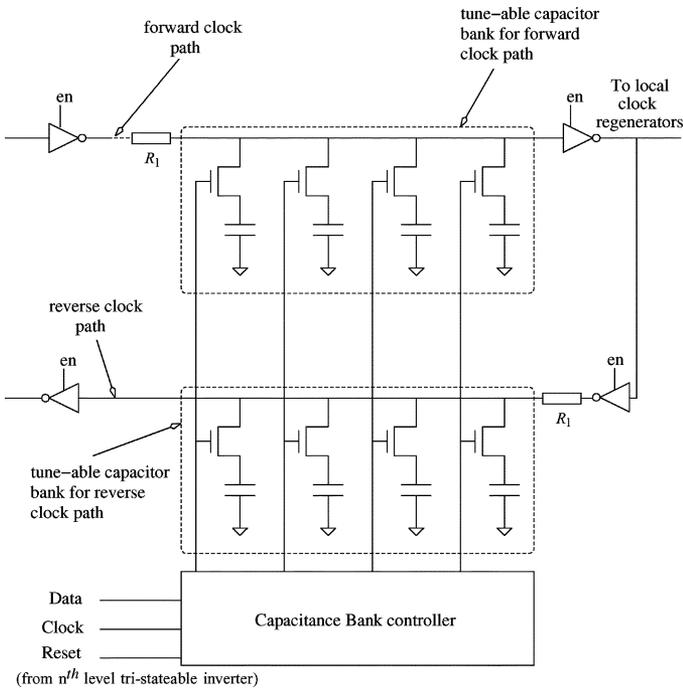


Fig. 2. Schematic showing a 4-bit tuneable capacitor bank at leaf nodes.

skew adjustment mode, it can be noted from Fig. 1 that both the forward and return path wire would have equal parasitic capacitances due to their neighbors. This ensures that the delays of the two paths are identical.

At each leaf node of the network, there is a pair of tuneable capacitance banks that are capable of adding incremental amounts of delay to the forward and return path. Fig. 2 describes a 4-bit capacitor bank (although our implementation uses 7-bit banks). We use an appropriately delayed reference clock signal and match the delay of each leaf node of the clock tree (using the capacitor banks mentioned above) to this reference signal, thus utilizing just one phase detector. This delay matching operation is performed sequentially for each leaf node.

B. Design of the Network

The H-tree network by itself is a zero-skew balanced network (if process and temperature variations are not considered). A traditional H-tree is designed assuming that the clock driver at the center of the H-tree is large enough to drive the entire clock tree. Wire widths and driver sizes are fixed to make sure that the clock signal can drive the local clock regenerators at the leaves of the H-tree for the required frequency of operation, with a sufficiently high slew-rate. The optimal (with respect to having a high slew-rate clock signal and also in terms of reducing the clock distribution delay) wire sizing methodology dictates that we utilize wider wires near the center of the H-tree, and narrower wires as we get closer to the leaves. For a buffered H-tree construction, there is no need to size wires this way. The wires can be of uniform width throughout.

In our previous work [4], we followed an opposite sizing trend from the traditional unbuffered H-tree. We used narrower wires near the center of the H-tree and wider wires near the leaves. This was done so that the resistance-capacitance (RC) loads

seen by all the buffers were the same. In this work, we have eliminated this requirement. We use the same wire width for all segments of the H-tree. This is one of the key differences between this paper and our previous work [4]. This change results in much lower area overheads compared to [4].

Another difference between this paper and [4] is that we compare our H-tree network with the network in [4], a buffered H-tree network and an unbuffered H-tree network. In [4], comparisons were made with an unbuffered H-tree only.

The structure of our buffered H-tree clock distribution network is shown in Fig. 3. The numbers on each of the branches of the H-tree indicate the level of that branch. The logic at the leaf of each H-tree (capacitance banks and capacitance bank controller) is shown in Fig. 2. Also, each branch of the H-tree has an address decoder (shown in Fig. 4) to control the tri-stateable buffers on the branch. Details on the function and operation of this decoder is discussed in Section III-C.

The lengths and widths of the wires used in our network are as shown in the Table II. The wire lengths are derived for a six-level H-tree network covering a $20\text{ mm} \times 20\text{ mm}$ die using the equations from [12]. The target frequency of operation is 1 GHz. The tri-stateable inverters of the return network are collocated with the inverters of the forward network. The arrangement of the clock, signaling, and return wires is as shown in Fig. 1. All sizes were finalized after extensive simulations using SPICE [13]. The capacitance extraction was performed using SPACE3D [14], using a strawman $0.1\ \mu$ technology file [15]. The clock wires were assumed to run on the METAL6 layer with METAL5 and METAL7 shield wires, as shown in Fig. 1.

The drawback with buffering the clock signal at each stage is that process and temperature variations at each of the inverters can now cause greater skew at the leaves of the H-tree. In an unbuffered H-tree the clock skew is primarily caused by the process variations in the interconnect. In a buffered clock tree, the inverters (buffers) contribute additional skew, and hence a *de-skewing network is required as well*. The phase detector used in our design is located at the center of the clock tree. It consists of a simple latch using two cross coupled inverters and devices as shown in Fig. 5. The operation of this detector, the output of which affects the tuneable bank of capacitors, is described in Section III-C.

In our method, the maximum chip-level skew is half the minimum incremental delay offered by the capacitance bank, in addition to the delay due to variations of the capacitance value. Since these capacitors are implemented as gate capacitances, the variation would be equal to the t_{ox} variation for the process (1.2% in our case, based on [7]). A tuneable 7-bit bank of capacitors is located at each of the leaves of the clock network as shown in Fig. 2. Each capacitor can be switched in as required. The capacitors used are binary weighted to facilitate the precise control of delay (within the resolution that the smallest capacitance allows) as well as provide the ability to de-skew clock networks with wide range of skew values (this range increases exponentially with number of capacitors in the bank). A tuneable bank of capacitors is provided in the return path too, as shown in Fig. 2. Both the forward and return capacitor banks are controlled by the same serial controller to ensure that the delays on both paths are always equal.

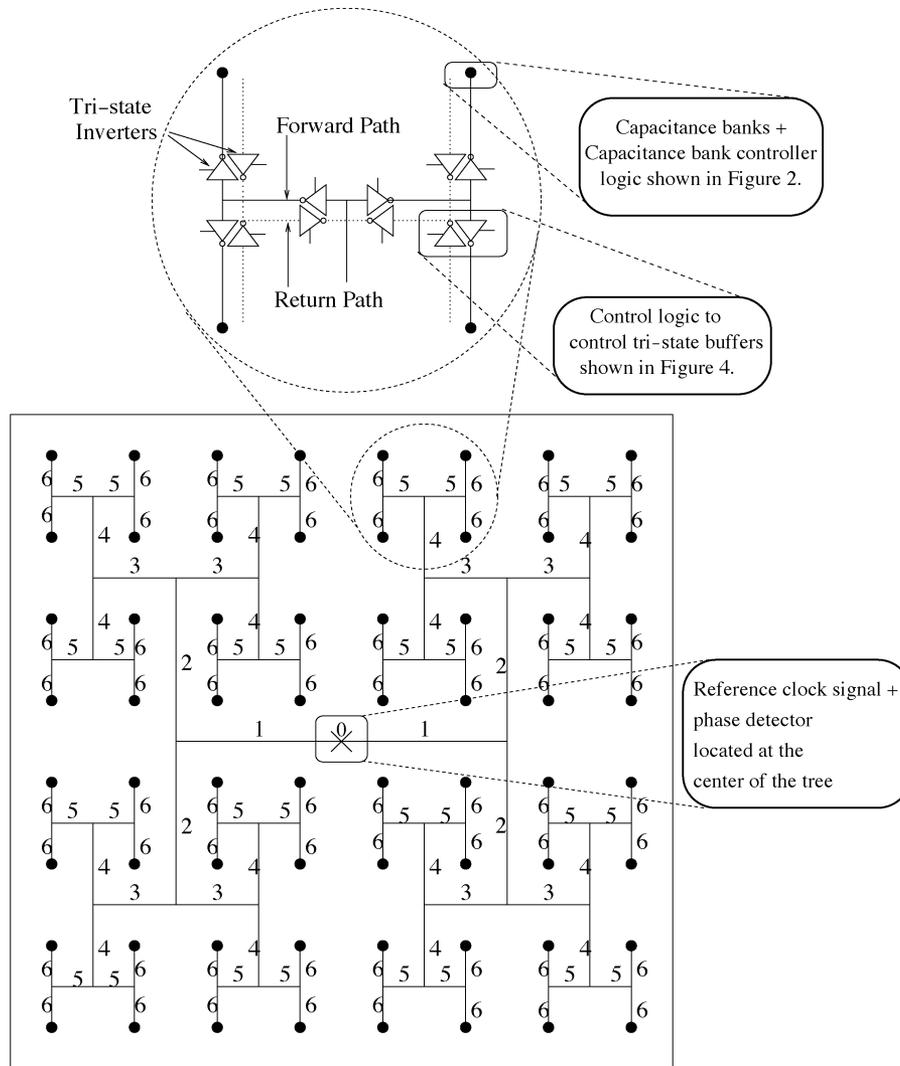


Fig. 3. Buffered H-tree clock distribution network with 6 levels.

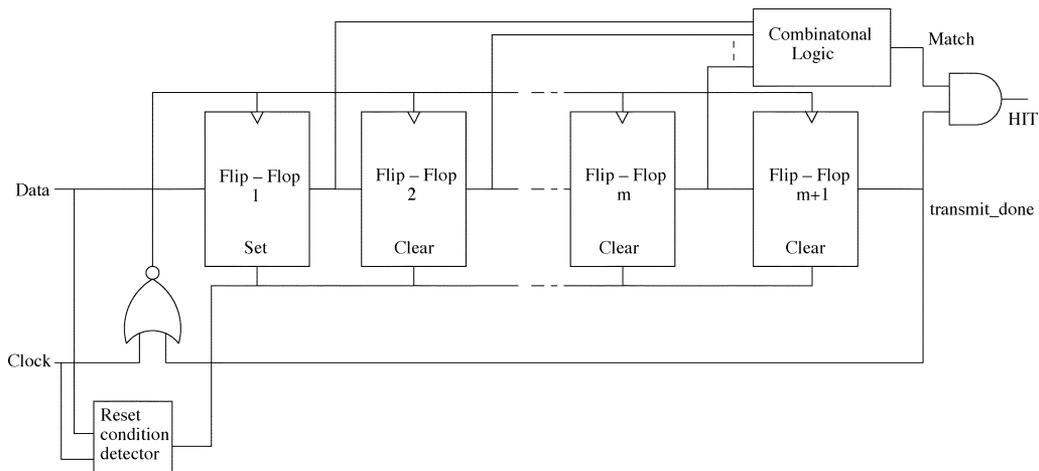


Fig. 4. *M*-bit address decoder.

A MOSFET's capacitance is expected to be lower at high frequency in the inversion region (since the device goes into deeper depletion instead of inversion at high frequencies). We performed simulations using both pMOS and nMOS gate ca-

pacitances and verified that the capacitance remained approximately the same at both the deskewing frequency of 100 MHz and at the operating frequency of 1 GHz. Hence, in our case, the capacitance change due to frequency is not an issue.

TABLE II
WIRE SIZES IN MICROMETERS

Level	Un-buffered H-tree		Buffered H-tree		Our Previous [4]		Our clock tree	
	Length	Width	Length	Width	Length	Width	Length	Width
1	5000	50	5000	1.5	5000	1.5	5000	1.5
2	5000	20	5000	1.5	5000	1.5	5000	1.5
3	2500	6	2500	1.5	2500	3	2500	1.5
4	2500	3	2500	1.5	2500	3	2500	1.5
5	1250	1.5	1250	1.5	1250	6	1250	1.5
6	1250	1.5	1250	1.5	1250	6	1250	1.5

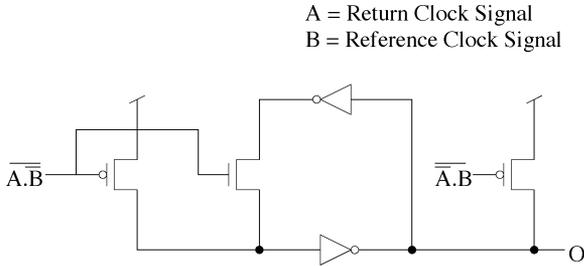


Fig. 5. Schematic of the phase detector.

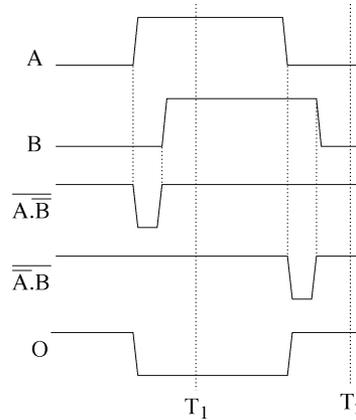


Fig. 6. Waveforms showing operation of the phase detector.

Note that in order to reduce the size of the individual capacitors that are required in any bank, we introduce a resistance R_1 in the last tree segment, as shown in Fig. 2. The value of R_1 is chosen such that the slew rate of the last segment is not appreciably changed, while the incremental delay adjustment due to the smallest capacitance is increased as desired. This resistance is present in both the forward and return paths (to maintain a balanced delay between these paths) of each leaf segment in the clock tree. Since this value of R_1 multiplies with the capacitance in the capacitance bank to give a larger RC delay, a larger value of R_1 would result in a coarser de-skewing ability, but a reduced area overhead due to the reduced size of the capacitors. For example, if we consider the output resistance of the driver that drives the capacitance bank is 100Ω , then the value of capacitance required in the capacitance bank for a 3 ps delay is 30 fF. If we increase this resistance to $1 \text{ K}\Omega$ by adding the resistance R_1 , the value of the capacitance required to get the same 3 ps delay drops by a factor of $10\times$ to 3 fF. R_1 is implemented using the diffusion layer, which has a high sheet resistivity (approximately 50 to $75 \Omega/\square$ in current technologies). In our simulations, we chose a resistance of $1 \text{ K}\Omega$. This resistance can be implemented with approximately $20 \square$ s of diffusion.

C. Operation of the Network

1) *De-Skewing Mechanism*: In normal operation, the signaling and return path lines are connected to GND and act as shields. When the clock network is being de-skewed, it is switched to the de-skewing mode. In this mode, the signaling and return path lines are not used as shield lines anymore. Also the frequency of the clock is reduced. The reason for this is twofold. The first is that a slower clock is required for the phase detector to compute its output. The second reason is to minimize cross-talk between the forward and return clock signals. To minimize cross-talk, we have ensure that when the clock signal returns on the return line, the forward clock line is

kept stable, such that the crosstalk encountered by the return signal is of type 2C as opposed to 3C [16]. This can be done by ensuring that half the time period of the clock is greater than the round trip delay of the clock signal, which ensures that when a forward clock transition occurs anywhere along the forward path, the return signal is static. In our case, the round-trip delay is about 2 ns, which means that the operational frequency during de-skewing operations should be less than 250 MHz. Also, we verified that based on the total load on the serial signaling lines A and B, these wires can switch at a rate of 200 MHz. We conservatively choose the clock frequency during de-skewing operations to be 100 MHz. Since the wires in question are routed on METAL6, and are quite wide, the problem of cross-talk is not serious to start with.

In de-skewing mode, each of the leaf nodes of the network are de-skewed one at a time. The signals from all leaf nodes are matched to the same reference clock signal. The reference clock is designed to lag the slowest return signal (without the tuning capacitors switched in). The waveforms corresponding to the situation where the reference clock lags the return signal are shown in Fig. 6. In Fig. 6, waveform A refers to the return signal and waveform B refers to the reference clock signal. Waveform O shows the output of the phase detector. When the signal B lags the signal A, the output of the phase detector is logic-0 at time T_1 and logic-1 at time T_2 . This condition (phase detector output logic-0 at time T_1 and logic-1 at time T_2) indicates that the reference clock signal lags the return signal. Capacitances from the capacitor bank at the corresponding leaf node are switched in until this condition is violated. This is repeated for each of the return signals. The logic required to test this condition consists of a delayed clock generator, whose input is the signal A, and

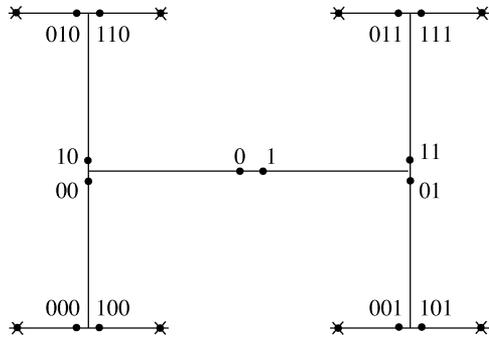


Fig. 7. Address assignment for return path tri-stateable inverters.

whose edges transition at T_1 and T_2 (ideally, T_1 and T_2 occur at the mid-points of the high and low phases of A , respectively). Let the output of the delayed clock generator be denoted as A^* . If, at the rising edge of A^* , O is low and at the falling edge of A^* , O is high, we successively switch in a larger capacitance value until this logical condition fails. This operation is repeated for each of the return signals.

Suppose the minimum phase resolution of the phase detector is C . Since we de-skew all leaf nodes so that they are C time units earlier than the reference signal, the value of C is immaterial. Thus, we do not need a complex phase detector which minimizes the value of C (unlike the situation in [7] and [10]). The guard-band within which we can de-skew the network is therefore solely a function of the smallest capacitance in the tuneable capacitor bank and *not determined by the accuracy of the phase detector utilized*. It must also be noted that the tuneable capacitor bank on the return path is operated in tandem with the capacitor bank on the forward path, as shown in Fig. 2. This ensures that the forward and return delays are always balanced.

2) Serial Controller Operations: We use the serial communication lines (A and B in Fig. 1) to communicate with the tri-state controllers. Signal A transmits the serial clock, while B is a data signal. In a serial communication sequence, we first assert a reset signal (which is derived from A and B at the controllers), followed by a 6-bit address (in general, an n bit address) and the 7-bit data (in general, we transmit as many bits as we have capacitors in the capacitor bank). The transmission of each address bit enables a unique tri-stateable buffer at each level, such that when the last address bit is transmitted, a unique clock return path is established. This also enables the corresponding capacitance bank to read in the 7-bit data that follows, loading this 7-bit value to enable the corresponding capacitors. For example, if the 7-bit data is 1000101, then the first, fifth, and seventh capacitors are connected to the leaf node. After this point, if a reset signal is reasserted, the vector which enables the capacitors is *not* reset. However, after a reset operation, all return path tri-stateable buffers are tri-stated (although all capacitor bank controllers hold their current state).

a) Addressing Mechanism: The address of each tri-stateable inverter is determined in the following manner. Let a “1” value correspond to the directions “up” and “right,” and let a “0” value correspond to the “down” and “left” directions. Consider the three-level buffered H-tree shown in Fig. 7. In this

buffered H-tree, the bullets refer to a tri-stateable buffer. This addressing scheme, when applied to this buffered H-tree, yields the addresses shown against each of the tri-stateable buffers of Fig. 7.

Note that an address of 1111...1 would be the tri-stateable inverter in the top right-hand corner of the network, while the address 0000...0 would be the bottom left tri-stateable inverter. Also note that with the application of the i th address bit, a unique tri-stateable buffer at level i in the buffered H-tree becomes enabled. The application of all n (six in our case) address bits enables a unique path from the source node to a specific leaf, establishing the corresponding clock return path. Also, the tri-stateable buffers at level i in the buffered H-tree respond only to the first i bits of the address, and disregard the remaining bits, since their address length is i .

We next discuss the implementation of an i -bit serial address decoder which we use for tri-stateable buffers at level i of the buffered H-tree.

b) m -bit Serial Address Decoder: We use the two serial communication lines (lines A and B) to serially address the tri-stateable inverter. Signal A is used to clock the decoder circuits at the tri-stateable inverters, and signal B is used to transmit serial address (6-bit) and data (7-bit) information. The general structure for an m -bit serial address decoder is shown in Fig. 4. The “Data” terminal is connected to the B signal, while the “Clock” signal is connected to the A signal. Upon reset, the output of flip-flop 1 (FF1) is set. After m serial bits are transmitted, the clocking of the flip-flops is stopped. At this point, if the m loaded bits match the address of this controller, the “Match” line is asserted, which in turn asserts “HIT”. If the address of the controller is 1001, then the combinational logic block of Fig. 4 implements the function $FF1\overline{FF2}FF3FF4$. Hence, the HIT signal is asserted when the first m address bits match the assigned address of the serial controller. At this point, only a reset signal can deassert the HIT signal. The reset signal of each decoder is locally generated, by toggling the data line during a *high* clock phase (typically the data line is updated during the low clock phase). The circuit to generate this reset signal is not shown in Fig. 4.

The enabling of a particular return path is carried out by enabling the appropriate return path tri-stateable inverters, in increasing level order, as the address bits are transmitted serially. The address decoders for different tri-stateable inverters will vary in length. The length will be equal to the level at which that inverter resides.

c) Capacitor Bank Controller: Fig. 8 describes the logic for the capacitor bank controller for a particular leaf node. The logic is very similar to that of the serial address decoder of Fig. 4. The HIT signal of this controller is connected to the HIT signal of the n^{th} level tri-stateable inverter’s address decoder for that leaf node, while its “Data” signal is connected to wire B. The assertion of HIT (indicating that this leaf node has been selected for de-skewing), creates a pulse on “reset”. Again, after seven serial data bits have been read by the capacitor bank controller, this controller no longer responds to serial data that is transmitted on wire B. Recall that the serial data bits are transmitted on wire B after the n address bits have been transmitted. The seven data bits that are read in now enable the appropriate

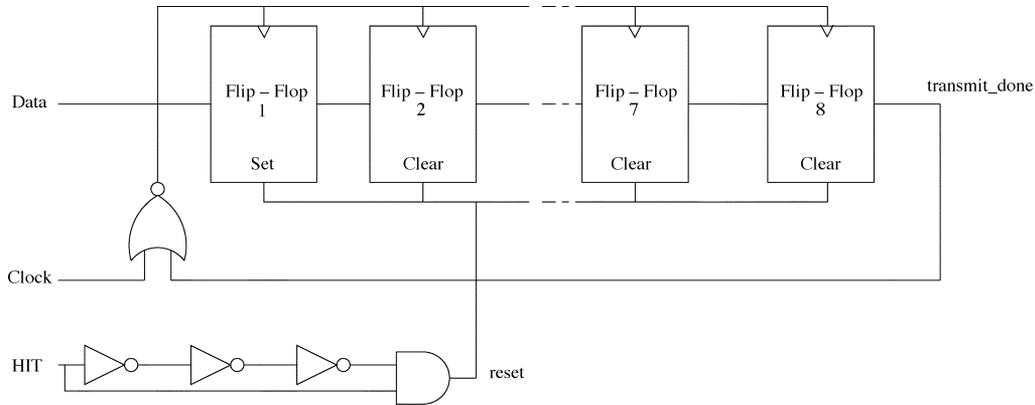


Fig. 8. Capacitor bank controller logic.

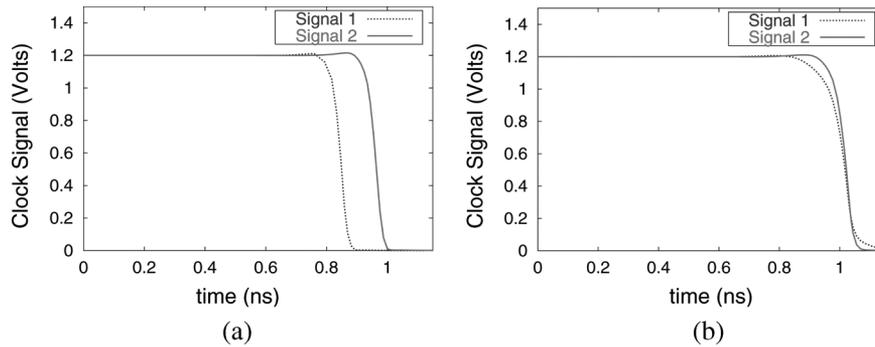


Fig. 9. Clock signals before and after de-skewing.

capacitors. For example, if the 7-bit data is 1000101, then the first, fifth, and seventh capacitors are connected to the leaf node.

Note that the derived reset signal for a *tri-stateable inverter* does not affect the capacitance bank controllers at all. As a consequence, the capacitance bank controllers retain their contents when the tri-stateable inverters are all disabled by the (serially derived) reset signal.

d) Overall Operation: Finally, we note that the serial-reset \rightarrow transmit-address \rightarrow transmit-data sequence must be followed while performing any serial controller operations. In response to the transmission of this sequence of information, the phase detection circuit is enabled, and checks if the corresponding leaf node needs to be slowed down. If so, the serial reset is asserted, the same leaf node is addressed, and a larger value of capacitance at the capacitor bank is selected (by transmitting a serial data value which corresponds to a larger number). When the leaf node has been de-skewed, a new leaf node is selected until all leaves have been de-skewed.

IV. EXPERIMENTAL RESULTS

Several experiments were performed to verify the utility of our new buffered H-tree network with dynamic clock de-skewing capability. We compared our clock tree network with our previous work [4] as well as a buffered H-tree network and an unbuffered H-tree network with no de-skewing. The wire sizes for our experiments are as described in Table II. We performed all simulations in SPICE3f5 [13], using the *bsim100* model cards [17]. The regenerators at each leaf were sized to drive a load of 6 pF, as per [7]. The smallest capacitor in our capacitor bank had a value of 3 fF. These capacitors are implemented as gate capacitances, using square devices

 TABLE III
POWER COMPARISONS

Clock Network	Power(mW)
Un-buffered H-tree	126.12
Buffered H-tree	85.22
Our Previous [4]	116.7
Our clock tree	94.94

with their source/drain connected to ground. Process variations based on values in [7] were introduced in each segment of the network. In particular, we changed the values of t_{ox} , μ , l_{eff} , and V_T as suggested in [7].

The plot in Fig. 9(a) shows two leaf node signals before de-skewing. The plot in Fig. 9(b) shows the two signals after de-skewing. While the skew between two return signals is 6 ps, the skew between the forward clock signals (which are used by the regenerators at the leaf nodes) is 3 ps. This is because the delay of the forward clock signal is half the round trip delay. The skew of the clock signals shown in Fig. 9(a) was 115 ps. Our deskewable network configurations were able to de-skew clock signals which were up to 300 ps out of phase from each other. This de-skewing range value can easily be increased if required, by using a capacitor bank with more than 7 bits.

We used a clock rate of 100 MHz for de-skewing operations. Note, however, that the results presented are from simulations performed at a clock frequency of 1 GHz (the actual operating frequency of the clock network). Our buffered H-tree consists of $2^6 = 64$ leaves, each requiring at most 2^7 serial-reset \rightarrow transmit-address \rightarrow transmit-data sequences. Each such sequence requires 13 clock cycles. Based on this, the maximum time for de-skewing the entire clock network is about 1 ms.

TABLE IV
AREA OVERHEADS

Category	Un-buffered H-tree (μ^2)	Buffered H-tree (μ^2)	Our Previous [4] (μ^2)	Our Clock tree (μ^2)	Overhead over Un-buffered H-tree	Overhead over Buffered H-tree	Overhead over [4]
Wiring	16.35×10^5	6.3×10^5	22.05×10^5	9.45×10^5	-42.2%	50.0%	57%
Central Clk Driver	480	–	–	–			
Regenerators	18432	18432	18432	18432			
TS inverters/buffers	–	608	4408	4408	24.56%	23.72%	0%
TS controllers	–	–	307	307			
Cap controllers	–	–	410	410			
Capacitors	–	–	4880	4880	–	–	0

A comparison of the power consumption of our clock network was performed against both an unbuffered and a buffered H-tree network. The size (W) of the unbuffered H-tree clock driver used was $3600 \mu\text{m}$ for the pMOS and $1200 \mu\text{m}$ for the nMOS device. The length of the devices was $0.1 \mu\text{m}$. The comparison of the power consumption of all the three clock networks is shown in Table III. The power consumption of the traditional network was 126.12 mW as compared to 94.94 mW for our design, a 24.7% improvement. When compared to the clock tree in our previous work [4], which had a power consumption of 116.7 mW , the clock tree presented in this manuscript has power consumption that is 19% lower. The power consumption of this clock tree is however (as expected) higher than that of a buffered H-tree network (85.22 mW) by 11.4%. Note that this is despite the fact that the *buffered and unbuffered H-trees we considered had no de-skewing mechanism*.

We also compared the area overheads of our buffered clock tree network over the buffered and unbuffered H-tree networks. Table IV describes the results of this experiment. All areas are in μ^2 . The first column describes the area component under consideration, while the second column describes the area of a traditional H-tree network. The third column describes the area of our buffered H-tree, while the fourth represents the overhead over the traditional method.

We studied the area overhead in three categories: wiring area, active logic area, and the area to implement capacitor banks. For the unbuffered H-tree and the buffered H-tree, the wiring area consisted of the area for the clock net as well as the area of the two shield wires on either side of the clock net. For the de-skewed clock trees, the total wiring area was calculated by adding the areas for the forward and return paths and the area of the two signaling wires. Our method exhibits a wiring area overhead of about 50% over a regular buffered H-tree and a savings of 42.2% in wiring area when compared to an unbuffered H-tree. When compared to the clock tree implemented in our previous work [4], the wiring area overhead is 57% lower. The total active logic area for our method is about 24%–25% larger when compared to the traditional unbuffered and buffered H-tree approaches. This component includes the areas of clock drivers, regenerators, tri-stateable inverters, tri-stateable inverter controllers and capacitor bank controllers. Finally, we report the area of our capacitors. Although the area overheads may appear large when compared to a H-tree with no deskewing circuitry, in relation to the die size assumed ($1 \text{ cm} \times 1 \text{ cm}$), the wiring area is only about 1% of the total die area while the active area is only 0.03% of the total die area.

Note that the method in this paper as well as the method in [4] are able to dynamically de-skew a clock distribution network, while the traditional H-tree methods (buffered and unbuffered H-trees) compared in Tables III and IV are not capable of this.

V. CONCLUSION

In contemporary VLSI ICs, the intra-die processing variations result in a large skew in the clock signals at the leaves of the typical clock distribution network. In this paper, we describe a technique to distribute and de-skew a buffered H-tree network. By using the clock shielding wires to selectively return the clock signal from a particular leaf for de-skewing, our approach adds appropriate capacitances at the leaves to ensure that the clock signals at each leaf has the same phase. We apply our technique on a six-level buffered H-tree network, demonstrating the ability of our method to de-skew clock signals with up to 300 ps of initial skew to within 3 ps. The power consumption of our scheme is about 19% lower than our previous work [4], 24.7% lower than an unbuffered H-tree and 11.4% higher than a buffered H-tree with no de-skewing mechanism. The wiring area of our scheme is about 57% lower than the scheme in [4], 50% more than a traditional buffered H-tree distribution network and 42.2% smaller than a unbuffered H-tree. The active logic area overhead of our scheme is about 24%–25%. Note that the traditional H-tree distribution networks *did not have a de-skewing capability*.

Unlike existing approaches, our method utilizes a *single* phase detection circuit. It can be used at boot time or periodically during circuit operation. With a small modification, it can be used to de-skew a clock network during factory test. Finally, clock gating (for power reduction) can be easily integrated into our clock distribution methodology.

REFERENCES

- [1] E. G. Friedman, *Clock Distribution Networks in VLSI Circuits and Systems*. Piscataway, NJ: IEEE, 1995, pp. 1–36.
- [2] A. Maxim, "A 0.16–2.55-GHz CMOS active clock deskewing PLL using analog phase interpolation," *IEEE J. Solid-State Circuits*, vol. 40, no. 1, pp. 110–131, Jan. 2005.
- [3] P. J. Restle and A. Deutsch, "Designing the best clock distribution network," in *Proc. Symp. VLSI Circuits*, Honolulu, HI, Jun. 1998, pp. 2–5.
- [4] A. Kapoor, N. Jayakumar, and S. Khatri, "A novel clock distribution and dynamic de-skewing methodology," in *Proc. ICCAD*, Nov. 2004, pp. 626–631.
- [5] P. Ramanathan, A. J. Dupont, and K. G. Shin, "Clock distribution in general VLSI circuits," *IEEE Trans. Circuits Syst.*, vol. 41, no. 5, pp. 395–404, May 1994.
- [6] J. Rabaey, *Digital Integrated Circuits: A Design Perspective*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

- [7] P. Zarkesh-Ha, T. Mule, and J. D. Meindl, "Characterization and modelling of clock skew with process variation," in *Proc. IEEE Custom Integr. Circuits Conf.*, San Diego, CA, 1999, pp. 441–444.
- [8] A. Rajaram, J. Hu, and R. Mahapatra, "Reducing clock skew variability via crosslinks," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 25, no. 6, pp. 1176–1182, Jun. 2006.
- [9] S. Tam, S. Rusu, U. N. 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.
- [10] C. E. Dike, N. A. Kurd, P. Patra, and J. Barkatullah, "A design for digital, dynamic clock deskew," in *Proc. Symp. VLSI Circuits*, Jun. 2003, pp. 12–14.
- [11] G. Geannopoulos and X. Dai, "An adaptive digital deskewing circuit for clock distribution networks," in *Proc. 45th IEEE Int. Solid-State Circuits Conf.*, Feb. 1998, pp. 400–401.
- [12] H. Bakoglu, J. T. Walker, and J. D. Meindl, "A Symmetric clock distribution tree and optimized high speed interconnections for reduced clock skew in ULSI and WSI circuits," in *Proc. IEEE Int. Conf. Comput. Des.*, Oct. 1986, pp. 118–122.
- [13] L. Nagel, "SPICE: A computer program to simulate computer circuits," Univ. California, Berkeley, UCB/ERL Memo M520, May 1995.
- [14] TU Delft, Delft, The Netherlands, "Physical design modelling and verification project (SPACE project)," [Online]. Available: <http://cas.et.tudelft.nl/research/space/html>
- [15] S. Khatri, R. Brayton, and A. Sangiovanni-Vincentelli, *Cross-Talk Noise Immune VLSI Design using Regular Layout Fabrics*. Norwell, MA: Kluwer, 2001.
- [16] C. Duan, A. Tirumala, and S. Khatri, "Analysis and avoidance of cross-talk in on-chip buses," *Hot Interconnects 9*, Stanford, CA, Aug. 2001, pp. 133–138.
- [17] UC Berkeley, Berkeley, CA, "BSIM3 homepage," [Online]. Available: <http://www-device.eecs.berkeley.edu/~bsim3/intro.html>



Arjun Kapoor received his bachelor's degree in electronics engineering from the University of Mumbai, Mumbai, India, and the Master's degree in electrical engineering from the University of Colorado at Boulder, Boulder.

He is currently a Systems Engineer with SanDisk Corporation, Milpitas, CA. His research interests include the fields of VLSI design such as sub-threshold circuits and clock network design (where he has a publication) as well as embedded system design and architecture.



Nikhil Jayakumar received the Bachelor's degree in electrical and electronics engineering from the University of Madras, Madras, India, the Masters degree in electrical engineering from the University of Colorado at Boulder, Boulder, and the Doctoral degree in computer engineering from the Department of Electrical and Computer Engineering, Texas A&M University, College Station.

He is currently with Texas instruments, Inc., Dallas, TX. During his graduate and doctoral studies he has done research and published several papers

in many aspects of VLSI including formal verification, clock network design, routing, structured ASIC design, radiation-hard design, logic synthesis, LDPC decoder architectures, statistical timing, low leakage power design techniques, and sub-threshold circuit design.



Sunil P. Khatri received the B.Tech (EE) degree from IIT Kanpur, Kanpur, India, the M.S. (ECE) degree from the University of Texas, Austin, and the Ph.D. degree in electronic engineering computer science from the University of California, Berkeley.

For four years, he worked with Motorola, Inc., where he was a member of the design teams of the MC88110 and PowerPC 603 RISC microprocessors. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Texas A&M University, College Station. His research

interests include logic synthesis, and novel VLSI design approaches to address issues such as power, cross-talk, and cross-disciplinary applications of these topics. He has coauthored about 95 technical publications, 5 U.S. Patents, one book, and a book chapter. His research is supported by Intel Corporation, Nascetric, Inc., Lawrence Livermore National Laboratories, and the National Science Foundation.

Dr. Khatri was a recipient of two Best Paper Awards and two Best Paper Nominations.