

A Novel Clock Distribution and Dynamic De-skewing Methodology

Arjun Kapoor[‡] Nikhil Jayakumar* Sunil P Khatri*

* Department of EE, Texas A&M University, College Station TX 77843.

[‡] Department of ECE, University of Colorado, Boulder, CO 80309.

Abstract

In present day VLSI ICs, intra-die processing variations are becoming harder to control, resulting in a large skew in the clock signals at the end of the clock distribution network. In this paper, we describe a novel buffered H-tree technique to distribute the clock signal and to de-skew a clock network. The clock shielding wires (which are connected to GND in normal operation) are, in de-skewing mode, used to selectively return the clock signal for de-skewing, and for serial communication with the clock distribution sites for skew adjustment. Our forward and return clock networks are buffered, with identically sized and co-located wires and buffers. This results in both these networks exhibiting identical delay characteristics in the presence of intra-die process variations. Unlike existing approaches, our method utilizes a *single* phase detection circuit, and can achieve a very low maximum chip-level clock skew. This skew value is *not* dependent on the resolution of the phase detector. Further, our technique can be applied dynamically, either at boot time or periodically during the operation of the IC, as necessary. Additionally, our buffered H-tree enables us to implement efficient clock gating by allowing the user to turn off clocks in the distribution network itself, thus disabling entire sections of the clock network. We demonstrate the utility of our technique on a 6-level H-tree clock distribution network. In a clock distribution network which is initially skewed by up to 300ps, our technique can de-skew signals to within 4ps of each other. We show that the total wiring area of our clock distribution and de-skewing methodology is about 35% higher than a traditional H-tree (which does not have a de-skewing functionality), while the active logic area overhead is about 25%. The power consumption of our network is 5% lower than that of a traditional H-tree network with no de-skewing functionality.

1 Introduction

In synchronous 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 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 in a VLSI chip has always been an important and difficult problem. In recent times this task has become even more challenging. This is primarily due to increasing

die sizes and also due to increasing clock frequencies. The difficulty is compounded by more acute intra-die processing variations in recent times. Clock skew can significantly deteriorate the performance of a high-speed IC [1]. Methods to design balanced clock distribution networks have been reported [2], but these methods result in zero skew only when there are no process or temperature 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 clock distribution network along with a companion 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. Finally, our technique can be easily modified to allow de-skewing to be performed during factory test.

The remainder of this paper is organized as follows: Section 2 discusses some previous work in this area. In Section 3 we describe our new method of constructing and de-skewing the clock tree network. In Section 4 we present experimental results comparing our method with a traditional H-tree. Conclusions are drawn in Section 5.

2 Previous Work

Traditionally, there are several methods to distribute clock signals. The simplest is the star topology [3]. In this method, a central clock signal is driven to n points using n separate wires. These wires are identically sized, ensuring a zero skew clock distribution network. A popular clock distribution methodology is the *H-tree network* [4]. 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 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 [5], the authors describe an analytical model and closed-form expression for on-chip clock skew based on device and interconnect variations.

In [6] the authors 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 network with n levels [6]. In the Mesh de-skew structure, the authors use phase detectors between each pair of leaf 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 $(2n + 1)D$. Another disadvantage of this method is that the required number of phase detectors grows exponentially with the number of levels in the H-tree.

Another clock de-skewing methodology [7] 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. 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 schemes of [6] which achieve a maximum chip-level skew of $(2n + 1)D$, our scheme has a maximum chip-level skew of D . A hierarchical clock de-skewing methodology would have a maximum chip-level skew of mD (assuming m phase detectors and skew tuning circuits). Further, our approach utilizes exactly one phase detector unlike [6, 5].

3 Our Approach

Our clock de-skewing methodology uses an appropriately delayed reference clock signal. We match the delay (using tune-able capacitor banks) of each leaf node of the clock tree to this reference 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 Figure 1. The forward and return networks have identical wire sizes. Additionally, 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). 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).

In Figure 1, signal wires A and B , along with the return clock wire, are held low during normal circuit operation, and act as clock shields. The signal wire A is used for clocking the serial control logic 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 co-located. This results in the intra-die processing variations being highly correlated in both networks. Hence if the smallest delay increment in the tune-able buffer is D , the largest skew between any two leaf nodes of the clock tree network is equal to $D/2$. This is because the delay of the forward clock signal is half the round trip delay. This is in contrast to the other de-skewing techniques [6, 5], which require a larger number of phase detectors, and cannot de-skew the clock network to such a fine tolerance.

With our approach, the forward clock distribution network has tri-state buffers. These can be used to perform clock gating [4]. 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).

3.1 Network Topology

In our approach, we construct an H-tree *forward* clock distribution network with an identical, co-located 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-state ability 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.

In terms of signal quality our buffered network is superior since slew-rates at the end-point of any segment are extremely high and uniform across the distribution tree. This is because each segment is driven by a dedicated tri-stateable inverter. The delays of the forward and return networks are made identical by co-locating 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.

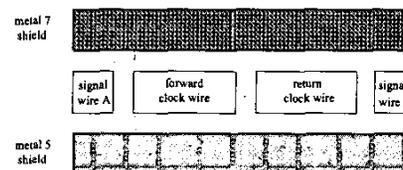


Figure 1: Cross section of clock, return clock and serial signaling wires

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 (Figure 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 skew adjustment mode, it can be noted from Figure 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 tune-able capacitance banks that are capable of adding incremental amounts of delay to the forward and return path. Figure 3 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.

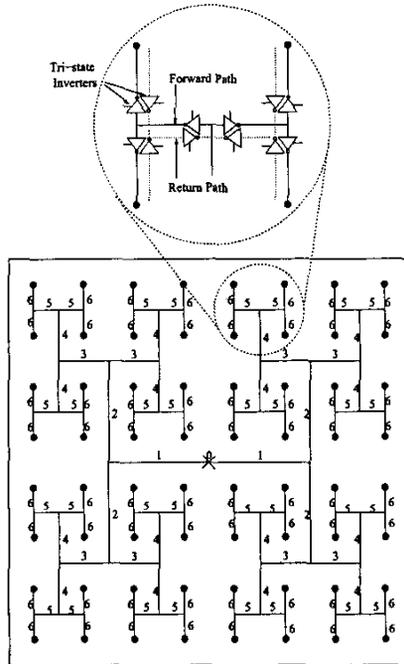


Figure 2: The buffered H-tree clock distribution network with 6 levels

3.2 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.

The structure of our buffered H-tree clock distribution network is shown in Figure 2. The numbers on each of the branches of the H-tree indicate the level of that branch. The tri-stateable inverters at each level are of the same size, made possible by the constraint that the capacitive load at the outputs of each of these inverters should

Level	Traditional H-tree		Our clock tree	
	Length	Width	Length	Width
1	5000	50	5000	1.5
2	5000	20	5000	1.5
3	2500	6	2500	3
4	2500	3	2500	3
5	1250	1.5	1250	6
6	1250	1.5	1250	6

Table 1: Wire sizes in microns

be identical. This choice allows for *uniform and high slew-rates* for all segments. It also allows us to utilize the same tri-stateable inverter everywhere in our buffered H-tree network. To achieve this, the wire widths follow an opposite trend from the traditional H-tree network, as shown in Table 1. The forward and return clock wires in our method are wider near the leaves and narrower near the center. With this choice of widths, we note that any wire segment (at any level of the buffered H-tree) has an identical RC delay, since the segment widths are inversely proportional to the segment lengths. Since each of the H-tree segments is now driven by inverters, the wire widths required are reduced tremendously compared to a traditional H-tree topology. This also helps minimize the device sizes of the inverters in each H-tree branch.

The lengths and widths of the wires used in our network are as shown in the Table 1. The wire lengths are derived for a 6-level H-tree network covering a $20mm \times 20mm$ die using the equations from [8]. The target frequency of operation is 1GHz. The tri-stateable inverters of the return network are co-located with the inverters of the forward network. The arrangement of the clock, signaling and return wires is as shown in Figure 1. The signaling wires do not have to be as wide as the clock and return wires since there is no constraint on these lines with regards to speed of operation. All sizes were finalized after extensive simulations using SPICE [9]. The capacitance extraction was performed using SPACE3D [10], using a strawman 0.1μ technology file [11]. The clock wires were assumed to run on the METAL6 layer with METAL5 and METAL7 shield wires, as shown in Figure 1.

This methodology of buffered H-tree construction is not typically used. 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 a traditional H-tree the clock skew is primarily caused by the process variations in the interconnect. In our method of clock distribution, the inverters 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 Figure 4. The operation of this detector, the output of which affects the tune-able bank of capacitors, is described in Section 3.3.

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 [5]). A tune-able 7-bit bank of capacitors is located at each of the leaves of the clock network as shown in Figure 3. 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 tune-able bank of capacitors is provided in the return path too, as shown in Figure 3. 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.

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 Figure 3. 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. A larger value of R_1 would result in a coarser de-skewing ability, but a reduced area overhead due to the capacitors. R_1 is implemented using the diffusion layer, which has a high sheet resistivity.

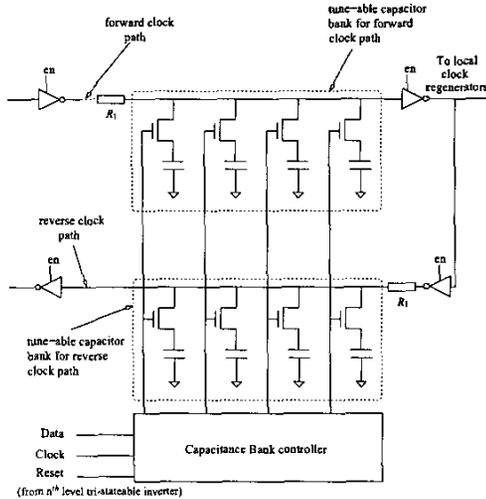


Figure 3: Schematic showing a 4-bit tune-able capacitor bank at leaf nodes

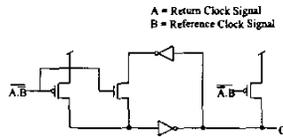


Figure 4: Schematic of the Phase Detector

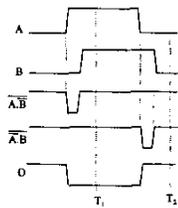


Figure 5: Waveforms showing Operation of the Phase Detector

3.3 Operation of the Network

3.3.1 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 ensured 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 [12]. 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 2ns, which means that the operational frequency during de-skewing operations should be less than 250MHz. 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 200MHz. We conservatively choose the clock frequency during de-skewing operations to be 100MHz. 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 Figure 5. In the Figure 5, waveform A refers to the return signal and waveform B refers to the reference clock signal. The 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 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 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 [6]). The guard band within which we can de-skew the network is therefore solely a function of the smallest capacitance in the tune-able capacitor bank and not determined by the accuracy of the phase detector utilized. It must also be noted that the tune-able capacitor bank on the return path is operated in tandem with the capacitor bank on the forward path, as shown in Figure 3. This ensures that the forward and return delays are always balanced.

3.3.2 Serial Controller Operations

We use the serial communication lines (A and B in Figure 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 re-asserted, 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).

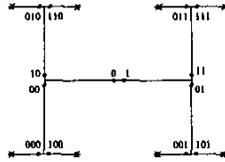


Figure 6: Address assignment for return path tri-stateable inverters

Addressing Mechanism: The address of the 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 3-level buffered H-tree shown in Figure 6. In this buffered H-tree, the bullets refer to a tri-stateable buffer. The terminal tri-stateable buffer (marked \times) is always enabled. This addressing scheme, when applied to this buffered H-tree, yields the addresses shown against each of the tri-stateable buffers of Figure 6.

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 (6 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.

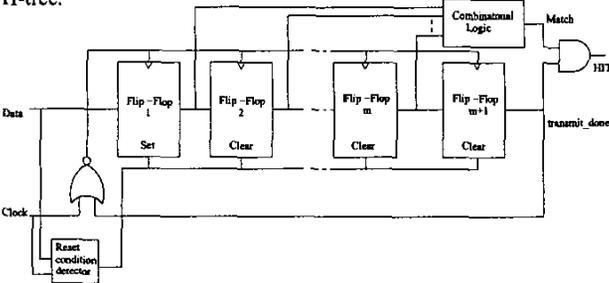


Figure 7: The M -bit address decoder

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 Figure 7. 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 Figure 7 implements the function $FF1 \overline{FF2} \overline{FF3} FF4$. 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 de-assert 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 Figure 7.

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.

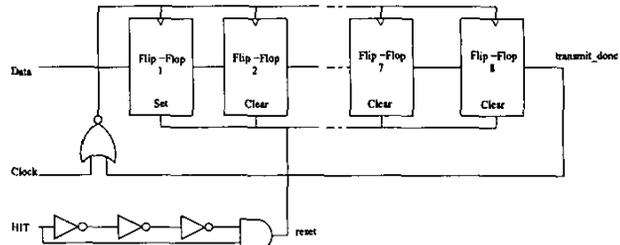


Figure 8: Capacitor Bank Controller logic

Capacitor Bank Controller: Figure 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 Figure 7. 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 7 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 7 data bits that are read in now enable the appropriate 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.

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.

4 Experimental Results

Several experiments were performed to verify the utility of our new buffered H-tree network with dynamic clock de-skewing capability. The wire sizes for our buffered H-tree network are as described in Table 1. We performed all simulations in SPICE3f5 [9], using the *bsim100* model cards [13]. The regenerators at each leaf were sized to drive a load of 6pF, as per [5]. The smallest capacitor in our capacitor bank had a value of 3fF. These capacitors are implemented as gate capacitances, using square devices with their source/drain connected

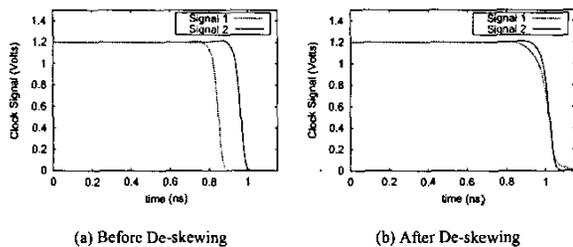


Figure 9: Clock Signals Before and After De-skewing

to ground. Process variations based on values in [5] were introduced in each segment of the network. In particular, we changed the values of t_{ox} , μ , t_{eff} and V_T as suggested in [5].

The plot in Figure 4 a) shows two leaf node signals before de-skewing. The plot in Figure 4 b) shows the two signals after de-skewing. While the skew between two return signals is 6ps, the skew between the forward clock signals (which are used by the regenerators at the leaf nodes) is 3ps. This is because the delay of the forward clock signal is half the round trip delay. The skew of the clock signals shown in Figure 4 a) was 115ps. Our configuration was able to de-skew clock signals which were up to 300ps 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.

A comparison of the power consumption of our clock network was performed against a traditional H-tree network. The size (W) of the traditional 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 power consumption of the traditional network was 126.12mW as compared to 116.7mW for our design, a 7.9% improvement. Note that this is despite the fact that the *traditional H-tree we considered had no de-skewing mechanism*. With higher clock frequencies the increase in power consumption of the traditional H-tree would be higher than our approach since the wire loads involved are much smaller in our clock network.

We also compared the area overheads of our buffered clock tree network over the traditional H-tree network. Table 2 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 used a clock rate of 100MHz for de-skewing operations. 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 1ms.

We studied the area overhead in three categories - wiring area, active logic area, and the area to implement capacitor banks. The total wiring area was calculated by adding the areas for the forward and return paths for our buffered H-tree distribution approach. Our method exhibits a wiring area overhead of about 35%. This area can be reduced by avoiding a reverse tapered sizing approach of the H-tree (Table ??), while still using fixed size tri-state buffers in the entire network. The total active logic area for our method is about 25% larger than the traditional H-tree approach. 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. Note that our method is able to de-skew a

clock distribution network, while the traditional H-tree method is not capable of this.

Category	Orig. Area	Our Area	Ovh.
Wiring	1.635×10^5	2.205×10^5	34.86%
Central Ck Driver	480	-	-
Regenerators	18432	18432	-
TS inverters	-	4408	24.56%
TS controllers	-	307	-
Cap controllers	-	410	-
Capacitors	-	4880	-

Table 2: Area overheads

5 Conclusions

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 6-level buffered H-tree network, demonstrating the ability of our method to de-skew clock signals with up to 300ps of initial skew to within 3ps. The power consumption of our scheme is about 8% lower, and the wiring area overhead of our scheme is about 35% over a traditional H-tree distribution network. The active logic area overhead of our scheme is about 25%. Note that the traditional H-tree distribution network *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," *IEEE Press*, pp. 1-36, 1995.
- [2] P. J. Restle and A. Deutsch, "Designing the best clock distribution network," in *1998 Symposium on VLSI Circuits*, June 1998.
- [3] Ramanathan, Dupont, and Shin, "Clock distribution in general VLSI circuits," *IEEE TCAS: IEEE Transactions on Circuits and Systems*, vol. 41, 1994.
- [4] J. Rabaey, *Digital Integrated Circuits: A Design Perspective*. Prentice Hall Electronics and VLSI Series, Prentice Hall, 1996.
- [5] P. Zarkesh-Ha, T. Mule, and J. D. Meindl, "Characterization and modelling of clock skew with process variation," in *IEEE 1999 Custom Integrated Circuits Conference*, 1999.
- [6] C. E. Dike, N. A. Kurd, P. Patra, and J. Barkatullah, "A design for digital, dynamic clock deskew," in *2003 Symposium on VLSI Circuits*, pp. 12-14, June 2003.
- [7] G. Giannopoulos and X. Dai, "An adaptive digital deskewing circuit for clock distribution networks," in *45th IEEE International Solid-State Circuits Conference*, pp. 400-401, Feb 1998.
- [8] H. B. et al, "A symmetric clock distribution tree and optimized high speed interconnections for reduced clock skew in vlsi and wsi circuits," in *IEEE Int'l Conf. Computer Design*, pp. 118-122, Oct 1986.
- [9] L. Nagel, "Spice: A computer program to simulate computer circuits," in *University of California, Berkeley UCB/ERL Memo M520*, May 1995.
- [10] "Physical Design Modelling and Verification Project (SPACE Project)." <http://cam.et.tuoflt.nl/xcsearch/space/html>.
- [11] S. Khatri, R. Brayton, and A. Sangiovanni-Vincentelli, *Cross-Talk Noise Immune VLSI design using regular layout Fabricat*. Kluwer Academic Publishers, 2001. ISBN 0-7923-7407-X.
- [12] C. Duan, A. Tirumala, and S. Khatri, "Analysis and avoidance of cross-talk in on-chip buses," in *Hot Interconnects 9*, (Stanford, CA), pp. 133-138, Aug 2001.
- [13] "BSIM3 Homepage" <http://www-device.eecs.berkeley.edu/~bsim3/intro.html>.