Topological Design of Clock Distribution Networks Based on Non-Zero Clock Skew Specifications

Jose Luis Neves and Eby G. Friedman

Department of Electrical Engineering
University of Rochester
Rochester, New York 14627

Abstract - A methodology is presented in this paper for designing clock distribution networks based on application dependent clock skew information. Contrary to previous approaches, the clock distribution network is designed such that the clock skew between two sequentially adjacent registers can be non-zero in order to maximize synchronous performance. The clock skew information is used to define the network topology and determine the delay values of the network branches. The design strategy for determining the topology of the clock distribution network considers the hierarchical description of the circuit to define the branching points of the clock tree. Algorithms to determine the optimal clock delay to each register and the network topology, given the non-zero clock skew specifications, are described and clock distribution networks are developed for example circuits.

1. INTRODUCTION

The distribution of clock signals is one of the primary limitations to maximizing the performance of synchronous integrated circuits. With the continuing reduction of feature size concurrent with increasing chip dimensions, interconnect delay has become increasingly significant, perhaps of greater importance than active device delay. Therefore, globally distributed signals, such as clock signals, can dominate and limit system performance. In this paper, focus is placed on the topological design of clock distribution networks, using a specified clock skew schedule to define the optimal clock delay to each register.

Several techniques have been developed to improve the performance of clock distribution networks, such as repeater insertion [1] to convert highly resistive-capacitive networks into effectively capacitive networks, symmetric distribution networks such as H-tree structures [2] to ensure minimal clock skew, and zero skew clock routing algorithms [3,4] to automatically layout these high speed networks in cell-based designs. These and similar approaches implement the clock distribution network so as to minimize the clock skew between each leaf of the clock distribution tree. However, these approaches neglect two fundamental properties of clock distribution networks. First, clock skew is only meaningful between sequentially adjacent registers, therefore there is no need to eliminate the skew between two clock signals that do not belong to the same data path. Second, it has been shown that clock skew [5] can be used to improve circuit performance by evening out the delay between slow and fast paths. This strategy permits the system clock frequency to be determined by the average path delay, instead of by the critical worst case path.

Clock skew is manifested by a lead/lag relationship between the clock signals that control a local data path, where a local data path is composed of two sequentially adjacent registers with, typically, combinational logic between them. If the clock delay to the initial register is less than the clock delay to the final register, the clock skew is described as negative. Likewise, if the clock delay to the initial register is greater than the clock delay to the final register, the clock skew is described as positive [6].

In this paper a methodology is proposed for designing synchronous clock distribution networks based on the optimal scheduling of the clock signals [7,8]. Given technology related information and the timing requirements of the circuit, such as the maximum permissible clock path delay and the aforementioned clock skew information, a topology of the clock distribution network is produced. The overall structure of the clock distribution network and the minimum delay values of each branch of the network are defined. The topological design of the clock distribution network exploits the hierarchical description of the circuit, although an approach is also presented for those circuits with minimal or no hierarchical information.

A theoretical background and an algorithm for determining the minimum clock delay of each clock signal, based on the optimal scheduling of the clock skew, is presented in section 2. In section 3, the delay values and the hierarchical information contained in the description of the circuit are used to design the topology of the clock distribution network. Experimental data describing several examples and discussion of these results are presented in section 4. Finally, section 5 summarizes the primary achievements of this paper and outlines further research.

2. DETERMINATION OF CLOCK PATH DELAY

The design of a clock distribution network requires a description of the circuit at the register transfer level and the desired clock skew between each pair of sequentially adjacent registers. The temporal characteristics of the functional data between each application-specific local data path are implicitly known from the clock skew information. The minimum clock delay of each register based on the clock skew data is obtained by a process called *negative clock skew* or *cycle stealing*, investigated independently by [9] and [10]. This approach shifts delay from the faster neighboring local data paths into the slower critical paths, thereby reducing the system-wide clock period and improving overall circuit performance.

2.1 Theoretical Background

To determine the minimum clock delay from the clock source to two sequentially adjacent registers, it is important to investigate if a relationship exists between the clock skews of the sequentially adjacent registers occurring within a global data path. Furthermore, it is necessary to describe the clock skew of global data paths which contain feedback paths. Before these issues are discussed, the concept of clock skew is first defined.

Definition 1: Given two sequentially adjacent registers, R_i and R_j, the clock skew between these two registers is defined as

$$T_{SKEW_{ij}} = T_{CD_i} - T_{CD_j}\,, \tag{1}$$

where T_{CD_i} and T_{CD_j} are the clock delays from the clock source to the registers R_i and R_j, respectively.

The path between two sequentially adjacent registers is described in this paper as a *local data path*; this is compared to a *global data path*, where a global data path can consist of one or more local data paths. The relationship between the clock skew of sequentially adjacent registers in a global data path is called conservation of clock skew and is presented below.

Theorem 1: For any given global data path, clock skew is conserved. Alternatively, the clock skew between any two registers which are not necessarily sequentially adjacent is the sum of the clock skews between each pair of registers along the global data path between those two same registers.

Although clock skew is defined between two sequentially adjacent registers, Theorem 1 shows that clock skew can exist between any two registers in a global data path. Therefore, it extends the definition of clock skew introduced by Definition 1 to any two non-sequentially adjacent registers belonging to the same data path. It also illustrates that the clock skew between any two non-sequentially adjacent registers which do not belong to the same global data path has no physical meaning.

A typical sequential circuit may contain sequential feedback paths, as illustrated in Figure 1. It is possible to establish a relationship between the clock skew in the forward path and the clock skew in the feedback path, because the initial and final registers in the feedback path are also registers in the forward path. As shown in Figure 1, the initial and final registers in

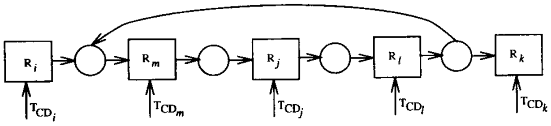

Figure 1: Data path with feedback paths

This research is base upon work supported by Grant 200484/89.3 from CNPq (Conselho Nacional de Desenvolvimento Cientifico e Tecnologico - Brasil) and the National Science Foundation under Grant No. MIP-9208165

CH 3381-1/93/$01.00 ©1993 IEEE

the feedback path R_l-R_j are the final and initial registers of the forward path R_j-R_k-R_l. This relationship is formalized below.

Theorem 2: For any given global data path containing feedback paths, the clock skew in a feedback path between any two registers, say R_l and R_j, is related to the clock skew of the forward path by the following relationship,

$$T_{SKEWfeedback,lj} = - T_{SKEWforward,jl} \qquad (2)$$

2.2 Clock Path Delay Algorithm

A synchronous circuit is formed by one or more global data paths. For each global data path, the clock delay of the registers is calculated by first choosing the local data path with the largest clock skew. If the clock skew is positive (negative), the clock delay of the final (initial) register of the local data path is assigned a constant *K*, and the clock delay of the initial (final) register is assigned the constant *K* plus (minus) the clock skew of the final (initial) register, where the constant *K* is the minimum clock delay of the circuit. The clock delay of the following register connected to the final register of the local data path is the clock delay of the final register minus the clock skew between them. Similarly, the clock delay of the previous register connected to the initial register is the clock delay of the initial register plus the clock skew between them. Proceeding in this fashion for the other registers, the clock delay of each of the registers in the global data path is calculated in terms of K.

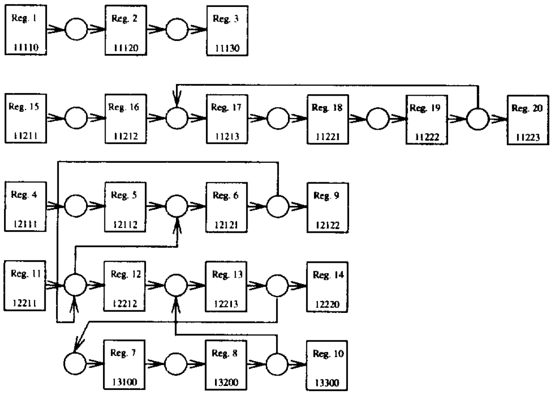

Figure 2: Example of a data path

To illustrate how clock delay is calculated, consider the synchronous circuit illustrated in Figure 2, where the rectangles represent registers and the circles represent combinational logic. Figure 3 illustrates the graph representation of this circuit, where a vertex represents a register instance, a directed edge represents a physical path between a pair of registers, and the edge weight represents the clock skew of each local data path. Consider, for example, the global data path, R_{15} to R_{20}. The maximum absolute clock skew is three for both local data paths (R_{15}, R_{16}) and (R_{16}, R_{17}), respectively, in Figure 2. Therefore, the clock delay to R_{16} from the clock source is the constant K, and the remaining clock delays are given as a function of K plus the local clock skew.

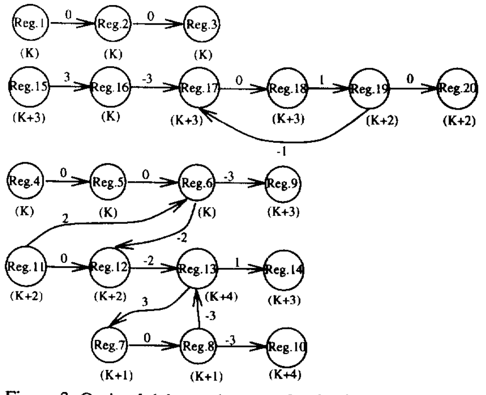

Figure 3: Optimal delay assignment for the data path example

The procedure to calculate the minimum clock delay is formalized in an algorithm and summarized in Figure 4 and called *Path_Delay*. After finding the local data path with the greatest clock skew, each of the clock delays to the other registers is calculated by traversing the global data path and attributing to each node the clock delay values that enforce the desired clock skew specification. If the initial clock skew specification satisfies Definition 1 and Theorems 1 and 2, each node is visited only once and the time complexity of the algorithm is *O(n)*, where *n* is the total number of nodes in the graph. The clock delay of each register obtained with this algorithm is represented by an expression in terms of *K* attached to each vertex, as illustrated in Figure 3.

```
Algorithm Path_Delay()
begin
   Circuit = $G(v,e)$;
   for each Datapath $D$ in Circuit $G$ do
    find MAX($T_{SKEWij}$);
    if $T_{SKEWij} < 0$ $v_i = K$; $v_j = K - T_{SKEWij}$ ;
    else $v_j = K$; $v_i = K + T_{SKEWij}$ ;
    while ($v_i$,$v_j$ not visited) and ($v_i$,$v_j$ in Datapath $D$) do
        Delay $v_{j+1}$ = delay $v_j + T_{SKEWj,j+1}$;
        Delay $v_{i-1}$ = delay $v_i + T_{SKEWi-1,i}$;
        Mark $v_{i-1}$, $v_{j+1}$ as visited;
end Path_Delay
```

Figure 4: Algorithm to find the optimal clock delay to each register

Assuming a single clock source and a clock distribution network that produces the delay values found from the previous procedure, the clock skew requirements of the circuit can be satisfied. Such a network is illustrated in Figure 5 for the example shown in Figure 2, where the numerals inside the rectangles represent clock delays.

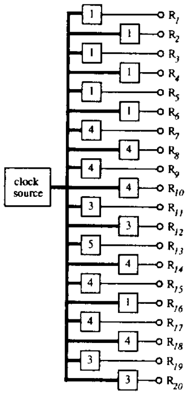

Figure 5: Clock distribution network for the data path example

3. TOPOLOGY OF CLOCK DISTRIBUTION NETWORK

In this section a methodology is presented for designing the topology of a clock distribution network that implements the delay values obtained from the algorithm *Path_Delay* introduced in section 2. A reasonable initial strategy is to provide an independent path delay for each register. This approach has the advantage of isolating each clock signal and requires a simple circuit composed of cascaded inverters and interconnect segments for which the behavior and delay models have been previously studied in great detail [1,11]. However, having independent clock paths drive each register is impractical because a typical VLSI circuit may contain many thousands of registers and, if each clock signal requires an independent clock path, significant chip area would be expended.

In this paper the clock distribution network for a specific circuit is designed in a tree structure using the hierarchical information of the original circuit description and the clock delay values obtained from algorithm *Path_Delay*. Subsection 3.1 illustrates the construction of the clock tree based on the hierarchical description of the circuit. The techniques to calculate the branch delays are presented in subsection 3.2. Subsection 3.3 describes a methodology for constructing clock distribution networks when minimal or no hierarchical information is available.

3.1 Construction of the Clock Tree Structure

Almost any large VLSI circuit is composed of several levels of hierarchy, where the number of levels is defined by the size and organization of the system and related factors such as the complexity of the circuit, the number of transistors, and the design methodology. During circuit placement, the elements of each hierarchical module are typically placed physically close to each other. Based on this reasoning, the hierarchical description of the circuit can be used to constrain the structure of the clock distribution network.

To build the clock distribution network, the hierarchy of the circuit is extracted from the circuit netlist and represented as a tree structure. The root vertex is the clock source, the internal vertices are the branching points derived from the hierarchy, and the leaf vertices are the registers. How the tree is built depends upon the unique code assigned to each register, which locates a register within the hierarchy of the circuit netlist. Figure 6 illustrates the hierarchical tree structure for the circuit example shown in Figure 2. For the purpose of calculating the individual clock delays, the branches of the tree are classified as either external or internal. The external branches are the branches connected directly to the registers. All other branches are classified as internal.

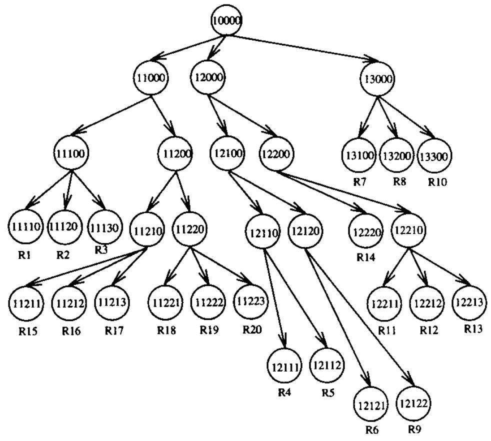

Figure 6: Hierarchical representation of the data path example

3.2 Calculation of Branch Delay

The minimum clock delays obtained from the algorithm *Path_Delay* are calculated without regard to hierarchical information. However, when the hierarchy is considered, these delay values are insufficient to completely determine the delay of each branch of the clock distribution network because of two characteristics of the hierarchical representation:

1) The clock skew between sequentially adjacent registers may not depend on the delay of the internal branches. Consider, for example, the global data path R_1 to R_3 in Figure 2. From Figure 6, these registers are driven by the same branching point, meaning that the clock skew specifications are satisfied solely by the delay of the individual external branches, driving R_1, R_2, and R_3.

2) The clock skew specifications may be satisfied if the delay of some of the internal branches is known. As an example, the global data path formed by R_{15} to R_{20} contains three registers, R_{15} to R_{17}, driven by one branching point while the remaining three registers, R_{18} to R_{20}, are driven by a second branching point. The clock skew between registers R_{17} and R_{18} can only be satisfied if the delay of the internal and external branches driving these two registers is known.

The process of determining the individual branch delays is divided into three steps:

Delay of External Branches - When both registers within a local data path are driven by the same branching point, the clock skew specifications are completely satisfied by the delay values assigned to the external branches. These delay values are provided by *Path_Delay* and are expressed in absolute terms, provided that a value is assigned to the constant K. Since the delay of a branch cannot be zero, unit delay is assigned to K without loss of generality.

Delay of Internal Branches - It is possible however, to have a global data path driven by more than one branching point. For example, the global data path R_{15} to R_{20} is driven by two separate branches of the clock distribution network. Starting with the common vertex that drives the data path (vertex V_{11200}), variables are assigned to each of the branches, as illustrated in Figure 7(a). The clock delay equations are chosen to satisfy the clock skew specifications. The set of equations are

$$c - d = 3$$
$$d - e = -3$$
$$a + e = b + f$$
$$f - g = 1$$
$$g = h$$

This system of equations has multiple solutions, since more unknowns exist than constraints. To obtain a solution, the delay of the external branches is calculated initially, providing a solution for variables c to h, respectively. Rewriting these equations, it is found that $a = b$. Attributing unit delay to these variables, a solution, shown in Figure 7(b), satisfying these clock skew specifications is found.

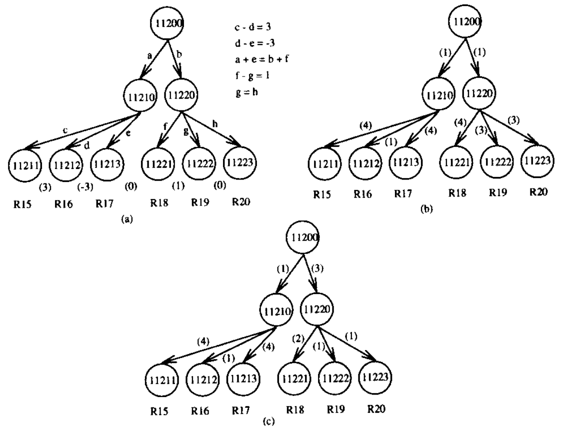

Figure 7: Calculation of the internal branch delays

Delay Shifting - It is possible to reorganize the delay of the external branches to reduce the total number of delay units needed to build the clock distribution network, thereby reducing the die area. Consider, for example, Figure 7(b). The external branches connected to registers R_{18} to R_{20} may each have their branch delay reduced by two, if the delay of the internal branch driving each of these registers is increased by two. Another advantage of shifting the delay is the increased flexibility of the circuit implementation, since the delay can be shifted between branches to accommodate for variations in the layout placement. An example of delay shifting is illustrated in Figure 7(c).

Extending this procedure to the other global data paths in the circuit of Figure 2, it is possible to define the minimum delay values for each branch of the clock distribution network. The complete delay specification for the clock distribution network required to synchronize the system depicted in Figure 2 is illustrated in Figure 8.

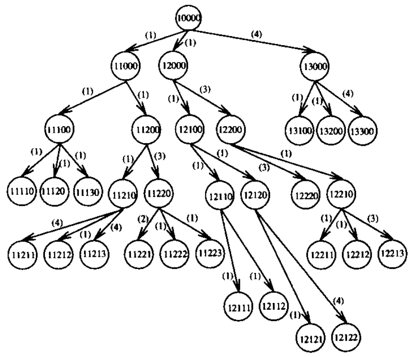

Figure 8: Delay assignment for each branch of the data path example

3.3 Reorganization of the Clock Tree

The methodology for constructing the clock distribution network presented in section 3 assumes that the circuit netlist is described hierarchically. It is possible, however, to have circuit descriptions which are completely or partially flat. For those circuits described non-hierarchically, the output of the previous section is a clock distribution network with independent clock paths for each register, similar to that shown in Figure 5. In this subsection, an approach for transforming these types of inefficient clock distribution networks into tree structures is presented. As an example, the clock distribution network for the flat representation of the circuit in Figure 2 is transformed below in Figure 9 into a clock tree with several levels of hierarchy.

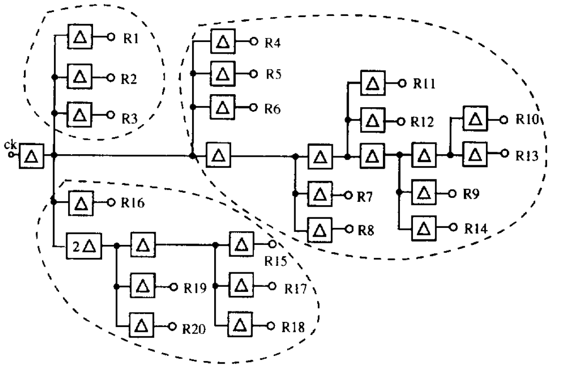

Figure 9: Delay assignment of non-hierarchically defined clock distribution network

Without regard to placement information, the clock distribution tree can be built by placing the fast clock paths in branches close to the clock source and the slow clock paths in branches farther from the clock source. For this purpose, the branch with the longest delay is partitioned into a series of segments, where each segment emulates a precise quantized delay value, Δ. Between any two segments, there is a branching point to other registers or sub-trees of the clock tree, where several segments with delay greater or equal to Δ are cascaded to provide the appropriate final delay at each leaf node. The result of this approach is illustrated in Figure 9, where a significant saving of segment delays, as compared to Figure 5, is obtained (from 56 to 28).

4. Results

The algorithms for determining the minimum clock delay of each register and for calculating the clock delay and circuit topology of the clock distribution network have been implemented in C. Table 1 illustrates some of the features of the algorithms for several circuit examples. The circuit illustrated in Figure 2 is example *cir1*, cited in Table 1. The second example, *cir2*, illustrates a clock distribution network in which each of the registers of a global data path are all interconnected. The final circuits, *cir3* and *cir4*, exemplify the effects of having large number of registers driven by the same branching point.

Circuit	Registers	Minimum clock delay		Total delay units	
		No hierarchy	Hierarchy	No hierarchy	Hierarchy
cir1	20	5	8	56	57
cir2	15	7.2	9.2	57	53
cir3	56	11.1	13.2	184	115
cir4	37	8	10.6	163	124

Table 1: Minimum clock delay and total number of delay units for several example circuits

The second column in Table 1 shows the number of registers within each circuit. The third and fourth columns depict the maximum clock delay of the clock distribution network based on a non-zero clock skew schedule. The fourth column describes the maximum delay once the hierarchy of the circuit description is considered. The final two columns describe the total number of delay units required to implement the circuit without and with using the hierarchical information of the circuit. The total number of delay units is an indirect measure of the layout area required to implement the clock distribution network.

5. Conclusions

In this paper a strategy for designing the topology of a clock distribution network based on non-zero clock skew is presented. The primary result of this work is a description of the topology of the network with minimum delay values assigned to each branch of the clock tree, such that the clock skew specifications between any two sequentially adjacent registers are satisfied. The design of the topology exploits the hierarchical description of the circuit, although an approach for developing the topology of clock distribution networks for circuits with minimal or no hierarchical information is also described.

It is shown that the clock skew between any two registers belonging to the same global data path is the sum of the individual clock skews of the sequentially adjacent registers between those two registers. It is also shown that the clock skew of any feedback path between two registers is the negative of the clock skew of the forward path between the same two registers. These results are used in an algorithm that provides the optimal clock delay values for each individual clock path. Examples are described which depict the significant area advantages of tree structured clock distribution networks with a minimal increase in clock delay.

Future research includes both determining the optimal non-zero clocking characteristics and the design of circuit structures to emulate the delay values of the network branches. The circuit design process requires accurate delay models for both the buffers and the interconnect lines. The buffer delay model must consider the effects of ramp shaped waveforms and short-channel effects, such as velocity saturation and channel length modulation. The buffer delay model must also be integrated with an interconnect delay model which includes the effects of both parasitic capacitance and resistance. These results will permit the efficient synthesis of high speed clock distribution networks for application to high performance VLSI/ULSI-based synchronous systems.

References

[1] H. B. Bakoglu, *Circuits, Interconnections and Packaging for VLSI*, Addison-Wesley, 1990.

[2] H. B. 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," *Proceedings of the IEEE International Conference on Computer Design*, pp. 118-122, October 1986.

[3] T.-H. Chao, Y.-C. Hsu, J.-M. Ho, K. D. Boese, and A. B. Kahng, "Zero Skew Clock Routing with Minimum Wirelength," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, Vol. CAS-39, No. 11, pp. 799-814, November 1992.

[4] R.-S. Tsay, "An Exact Zero-Skew Clock Routing Algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. CAD-12, No. 2, pp. 242-249, February 1993.

[5] E. G. Friedman and J. H. Mulligan Jr., "Clock Frequency and Latency in Synchronous Systems," *IEEE Transactions on Signal Processing*, Vol. SP-39, pp. 930-934, April 1991.

[6] E. G. Friedman, "Clock Distribution Design in VLSI Circuits - an Overview," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 1475-1478, May 1993.

[7] K. A. Sakallah, T. N. Mudge, O. A. Olukotun, "*checkTc* and *minTc*: Timing Verification and Optimal Clocking of Synchronous Digital Circuits," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 111-117, June 1990.

[8] T. G. Szymanski, "Computing Optimal Clock Schedules," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 399-404, June 1992.

[9] E. G. Friedman, *Performance Limitations in Synchronous Digital Systems*, Ph. D. Dissertation, University of California, Irvine, California, June 1989.

[10] J. P. Fishburn, "Clock Skew Optimization," *IEEE Transactions on Computers*, Vol. C-39, No. 7, pp. 945-951, July 1990.

[11] T. Sakurai, "Closed-Form Expressions for Interconnection Delay, Coupling, and Crosstalk in VLSI's," *IEEE Transactions on Electron Devices*, Vol. ED-40, No. 1, pp. 118-124, January 1993.