Design Methodology for Synthesizing Clock Distribution Networks Exploiting Nonzero Localized Clock Skew

José Luis Neves and Eby G. Friedman

Abstract—**An integrated top-down design methodology is presented in this brief for synthesizing high performance clock distribution networks based on application dependent localized clock skew. The methodology is divided into four phases: 1) determining an optimal clock skew schedule composed of a set of nonzero clock skew values and the related minimum clock path delays; 2) designing the topology of the clock distribution network with delays assigned to each branch based on the circuit hierarchy, the aforementioned clock skew schedule, and minimizing process and environmental delay variations; 3) designing circuit structures to emulate the delay values assigned to the individual branches of the clock tree; and 4) designing the physical layout of the clock distribution network. The clock distribution network synthesis methodology is based on CMOS technology. The clock lines are transformed from distributed resistive-capacitive interconnect lines into purely capacitive interconnect lines by partitioning the *RC* interconnect lines with inverting repeaters. Variations in process parameters are considered during the circuit design of the clock distribution network to guarantee a race-free circuit. Nominal errors of less than 2.5% for the delay of the clock paths and 7% for the clock skew between any two registers belonging to the same global data path as compared with SPICE Level-3 are demonstrated.**

Index Terms— **Clock distribution networks, clock scheduling, clock skew, clock tree, CMOS inverters, repeaters, topology, VLSI.**

I. Introduction

Most existing digital systems utilize fully synchronous timing, requiring a reference signal to control the temporal sequence of operations. Globally distributed signals, such as clock signals, are used to provide this synchronous time reference. These signals can dominate and limit the performance of VLSI-based systems. The importance of these global signals is, in part, due to the continuing reduction of feature size concurrent with increasing chip dimensions. Thus interconnect delay has become increasingly significant, perhaps of greater importance than active device delay. Therefore, new design approaches are required for efficiently distributing these global signals, particularly the clock distribution network, while preventing any deleterious effects within the circuit. Furthermore, the design of the clock distribution network, particularly in high speed applications, requires significant amounts of time, inconsistent with the high design turnaround of the more common data flow synthesis phase of ASIC semi-custom and VLSI structured custom design methodologies.

Several techniques have been developed to improve the performance and design efficiency of clock distribution networks, such as placing distributed buffers within clock tree layouts [1] to control the propagation delay and power consumption of the clock distribution networks, using symmetric distribution networks, such as H-tree structures [2], to minimize clock skew, and applying zero-skew clock routing algorithms (i.e., [3], [4]) to the automated layout of high speed clock distribution networks in cell-based designs. A common feature of these approaches is that the clock distribution network is designed to ensure minimal (or zero) clock skew between every register, not recognizing that clock skew is both local [5], [6] and can be used to improve circuit performance while minimizing the likelihood of any race conditions.

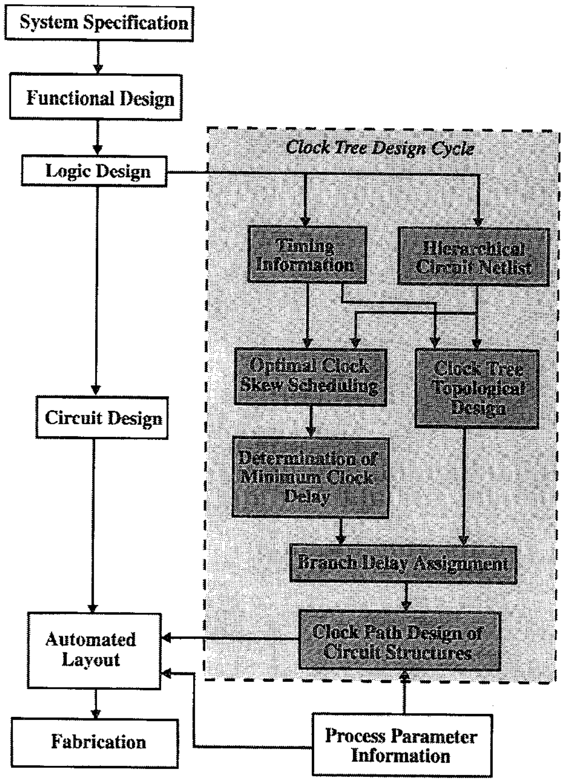

System Specification
Functional Design
Logic Design
Circuit Design
Automated Layout
Fabrication
Clock Tree Design Cycle
Timing Information
Hierarchical Circuit Netlist
Optimal Clock Skew Scheduling
Clock Tree Topological Design
Determination of Minimum Clock Delay
Branch Delay Assignment
Clock Path Design of Circuit Structures
Process Parameter Information

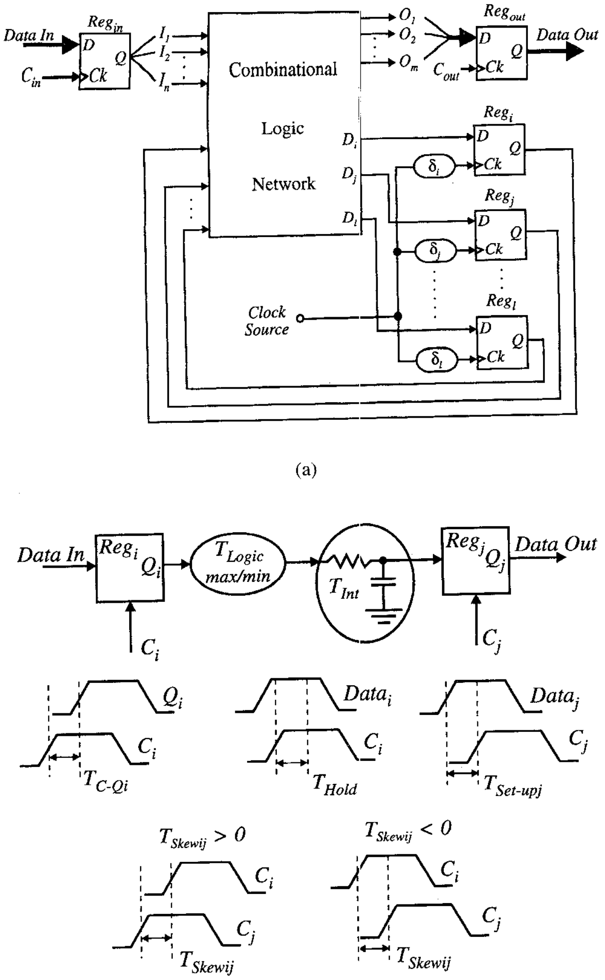
Fig. 1. Block diagram of the clock tree design cycle integrated with standard IC design flow.

A novel top-down methodology is therefore presented in this brief for efficiently synthesizing distributed buffer, tree-structured clock distribution networks that exploit nonzero localized clock skew to improve circuit performance. This methodology is illustrated in terms of the IC design process cycle shown in Fig. 1. The clock distribution network design methodology is composed of four major phases. In the first phase, summarized in Section II, a localized clock skew schedule is determined which maximizes circuit performance and reliability. This clock skew schedule is converted into individual clock path delays, i.e., the propagation delay from the clock source to each register. In the second phase, described in Section III, a topological graph of the clock distribution network is obtained, producing a clock tree with minimum delay values assigned to each branch. In the third phase, presented in Section IV, the circuit structures which implement the individual branch delay values are designed. Furthermore, variations in process parameters guide the design process to produce a clock tree free of race-conditions. The final phase, the physical layout of the clock distribution network, is not addressed in this brief. Simulations demonstrating the accuracy of this methodology are presented in Section V and some conclusions are drawn in Section VI.

Manuscript received April 5, 1994; revised December 21, 1994, July 28, 1995 and August 16, 1995. This work was supported by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brasil) under Grant 200484/89.3, the National Science Foundation under Grant MIP-9208165 and Grant MIP-9423886, the Army Research Office under Grant DAAH04-93-G-0323, and by a grant from the Xerox Corporation.

The authors are with the Department of Electrical Engineering, University of Rochester, Rochester, NY 14627 USA.

Publisher Item Identifier S 1063-8210(96)04052-8.



(a)

(b)

Fig. 2. (a) Model of a synchronous system. (b) Timing model of a local data path.

II. Optimal Clock Skew Scheduling

A synchronous digital system can be modeled by a combinational logic block and a data storage section, as shown in Fig. 2(a). The data storage section is composed of I/O and internal registers. Clock skew is manifested by a lead/lag relationship between the clock signals that control a *local data path* [see Fig. 2(b)] where a local data path is composed of two sequentially adjacent registers and a sequentially adjacent path is a path with only combinational logic and/or interconnect between the two registers. More specifically, given two sequentially adjacent registers, R_i and R_j, the clock skew between these two registers is defined as $T_{\mathrm{Skew}ij} = T_{\mathrm{CD}i} - T_{\mathrm{CD}j}$, where $T_{\mathrm{CD}i}$ and $T_{\mathrm{CD}j}$ are the clock delays from the clock source to t'e registers R_i and R_j, respectively. If the clock delay to the initial register $T_{\mathrm{CD}i}$ is greater than the clock delay to the final register $T_{\mathrm{CD}j}$, the clock skew is described as *positive*. Similarly, if the clock delay to the initial register $T_{\mathrm{CD}i}$ is less than the clock delay to the final register $T_{\mathrm{CD}j}$, the clock skew is described as *negative* (see Fig. 2(b)) [5]. Finally, a *global data path* is a data path consisting of one or more local data paths.

In the clock skew scheduling phase of the clock distribution network design methodology, the individual clock skew for each local data path is determined such that the clock period is minimized while avoiding all race conditions [5], [6]. The set of inequalities presented in Table I describes the timing relationships of each local and global data path, as illustrated in the data path composed of the path R_i to R_j shown in Fig. 2(b). In the inequalities shown in Table I, $T_{\mathrm{Skew}ij}$ is the clock skew between registers R_i and R_j, $T_{\mathrm{Hold}j}$ is the amount of time the input data must be stable after the clock signal changes state, $T_{\mathrm{PD(max)}}$ ($T_{\mathrm{PD(min)}}$) is the maximum (minimum) propagation delay between registers R_i and R_j, shown in (2) and (4), respectively, $T_{C\text{-}Qi}$ is the time required for the data to leave R_i once it is enabled by the clock pulse C_i, $T_{\mathrm{Logic(max)}}$ ($T_{\mathrm{Logic(min)}}$) is the maximum (minimum) propagation delay through the logic block between the registers R_i and R_j, T_{Int} is the interconnect delay between the registers R_i and R_j, $T_{\mathrm{Set-up}j}$ is the time required to successfully propagate to and latch the data within R_j, T_{CP} is the minimum clock period, and ζ_{ij} is the allowed tolerance in clock skew to account for process parameter and environmental variations, such as temperature, radiation, and power supply fluctuations. A procedure to manage these variations is described in greater detail in Section IV.

TABLE I
Timing Relationships for a Local Data Path, R_i to R_j

$T_{CP} \geq T_{Skewij} + T_{PD(max)}$	(1)
$T_{PD(max)} = T_{C-Qi} + T_{Logic(max)} + T_{Int} + T_{Set-upj}$	(2)
$T_{Skewij} \geq T_{Holdj} - T_{PD(min)} + \zeta_{ij}$	(3)
$T_{PD(min)} = T_{C-Qi} + T_{Logic(min)} + T_{Int} + T_{Set-upj}$	(4)
$T_{Skewin,out} = T_{Skew\,in,1} + \cdots + T_{Skew\,n,out} = 0$	(5)

Observe that (1) guarantees that the data signal latched in R_i is latched into R_j before the next clock pulse arrives at R_j, preventing *zero clocking* [6]. This effect becomes important when the clock skew is positive and greater in magnitude than the path delay. Also, (3) prevents latching an incorrect data signal into R_j by the clock pulse that latched the same data signal into R_i, or *double clocking* [6]. This race condition is created when the clock skew is negative and greater in magnitude than the path delay. If the clock skew is negative, but smaller than the path delay, this effect can be used to improve circuit performance and is called *cycle stealing* [5], [7]. Furthermore, for a given clock period T_{CP}, (1) and (3) provide a range within which each local clock skew $T_{\mathrm{Skew}ij}$ can vary. This tolerance range, described here as a *permissible range* between the minimum permissible clock skew $T_{\mathrm{Skew}ij(\mathrm{min})}$ and the maximum permissible clock skew $T_{\mathrm{Skew}ij(\mathrm{max})}$ (see Fig. 3), varies since it is dependent on the $T_{\mathrm{PD(min)}}$ and $T_{\mathrm{PD(max)}}$ of each local data path. $T_{\mathrm{Skew}ij(\mathrm{max})}$ is zero for those critical local data paths that define the minimum clock period T_{CP}. Finally, (5) describes the relationship between the on-chip and off-chip clock skew, where $T_{\mathrm{Skew}in,out}$ is the clock skew between the I/O off-chip registers. This relationship eliminates race conditions among different circuits controlled by the same clock source.

The inequalities presented in Table I are sufficient conditions to determine the optimal clock skew schedule, the associated minimum clock path delays, the allowed variation of each clock skew, and the minimum clock period such that the overall circuit performance is maximized while eliminating any race conditions. The optimal clock scheduling problem is usually described by a set of linear equations which are solved with standard linear programming techniques [6]–[9]. However, for determining an optimal clock skew schedule, a three phase graph-based algorithm has been developed. In the first phase, an optimal clock period is determined while ensuring that the clock skew constraints among the global data paths of the circuit are satisfied. In the second phase, the widest permissible range of each local data path is determined for a given clock period and the chosen

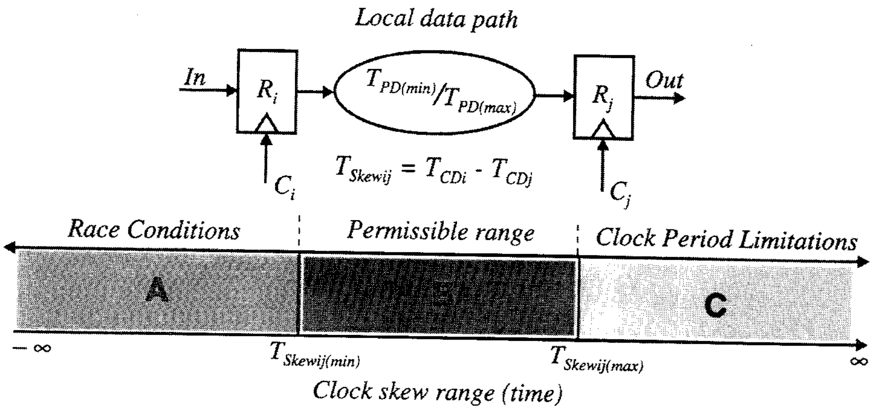


Fig. 3. Permissible and nonpermissible regions for the clock skew of a local data path.

clock skew value is the central value within the permissible range of each local data path. Finally, in the third phase, a schedule of clock delays is determined from the selected clock skew values.

Providing independent clock path delays for each register is impractical due to the large capacitive load placed on the clock source and the inefficient use of die area. A tree structured clock distribution network is more appropriate, where the branching points are selected according to the delay of each clock path, the relative physical position of the clocked registers, and the sensitivity of each local data path to delay variations. Such an approach for determining the structural topology of a clock distribution network is described in the following section.

III. Topological Design

The topology of a clock tree derived from a clock skew schedule must ensure that the clock path delays are accurately implemented while considering the effects of process parameter variations. A tree-structured topology can be based on the hierarchical description of the circuit netlist, on implementing a balanced tree with a fixed number of branching levels from the clock source to each register with a pre-defined number of branching points per node (an example of this approach is a binary tree with n levels for 2^n registers with two branching points per node), on reducing the effects of process parameter variations by driving common local data paths by the same sub-tree, or by implementing each clock path delay with pre-defined delay segments, the number of segments chosen to reduce the layout area of the clock tree.

In the methodology presented in this brief for synthesizing clock distribution networks, 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, and the leaf vertices are the registers. This approach has also been extended to derive the topology of the clock distribution network from a flat nonhierarchical circuit description. In either approach, the clock tree is designed such that both registers belonging to the same critical data path are driven by the same sub-tree in order to minimize the effects of process parameter variations as well as fluctuations in temperature and power supply. Thus, a greater portion of the clock tree driving both registers of a critical local data path are in common, minimizing the effects of process and environmental variations on the local clock skew of the critical data paths.

Once the tree structure (or topology) is obtained, the delay values of each branch are determined such that the clock skew specifications are satisfied. For the purpose of calculating the individual branch delays, the branches of the tree are classified as either *external or internal*. The external branches are those branches connected directly to the registers. All other branches are classified as internal.

The algorithm for determining the individual branch delays is composed of the following four 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. Thus, the minimum delay values determined from the clock skew scheduling phase are assigned to these external branches.

Delay of Internal Branches: The delay of the internal branches is determined by assigning variables to each of the branches and solving the system of linear equations so as to satisfy the clock skew specifications [10].

Delay Equalization: By restricting the clock skew among the input and output off-chip registers to zero, board level race conditions are avoided. Applying this restriction requires the clock path delay from the clock source to all I/O registers to be equal. This process is accomplished by finding the I/O register with the greatest clock path delay and equalizing the delay of the remaining I/O registers while satisfying the desired clock skew schedule.

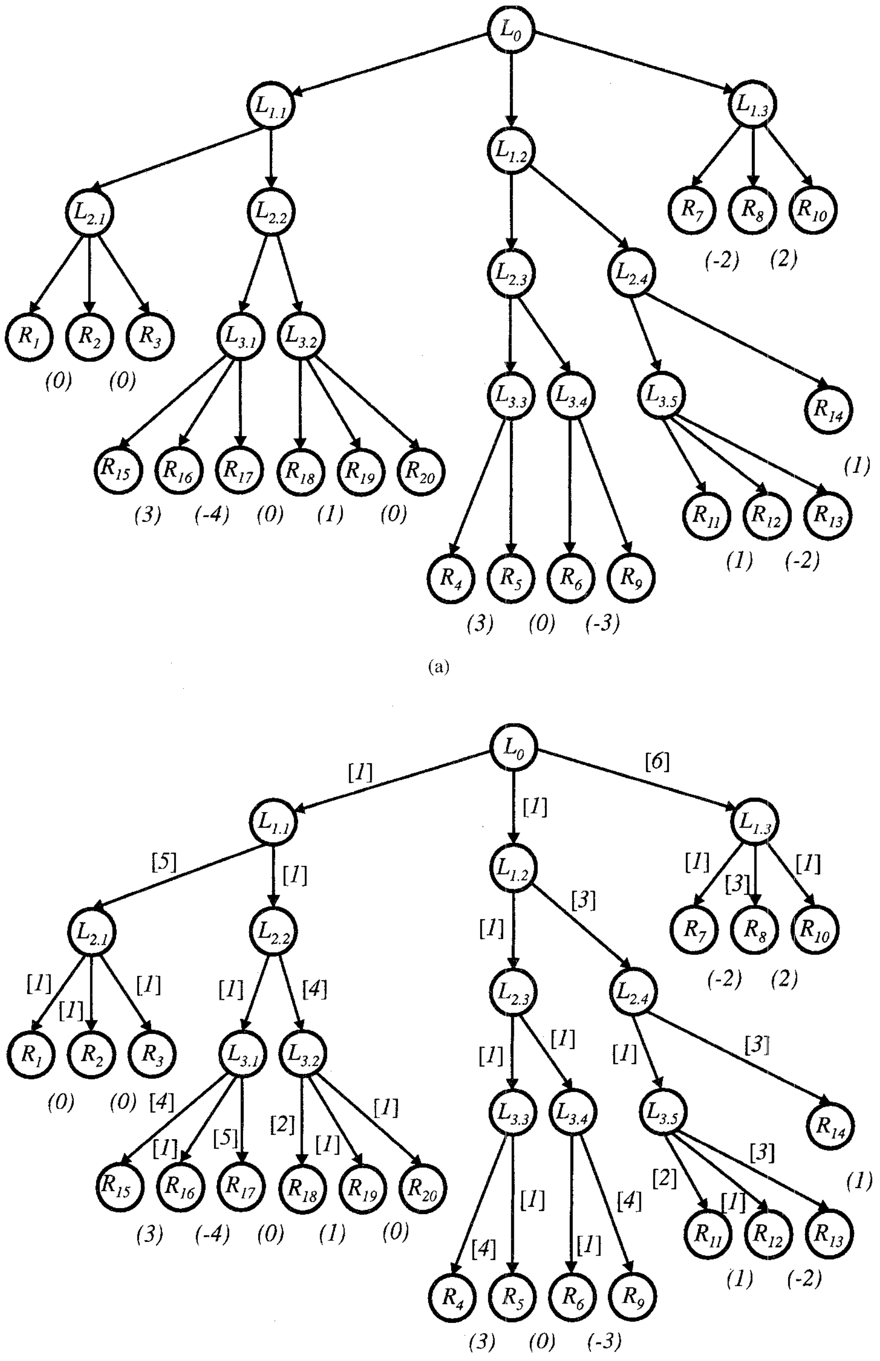
Delay Shifting: It is possible to shift the delay of the external branches to the internal branches of the sub-tree, thereby reducing the size of the active buffers within the clock distribution network. Another advantage of shifting the delay is the increased flexibility of the circuit implementation. Since the delay can be shifted among branches, different variations of the layout placement can be accommodated.

An example of a clock distribution network topology before and after branch delay assignment is illustrated in Fig. 4. The numbers in the parentheses are the initial clock skew schedules, while the numbers in the brackets are the assigned branch delay values.

IV. Design of Circuit Delay Elements

The delay of the circuit structures that emulates the delay values associated with each branch of the network requires high precision, because variations in the delays of the internal branches are propagated throughout the network, causing unacceptable variations in the desired clock skew. Note that it is much more difficult and important to accurately satisfy the clock skew *between* any two clock paths rather than to accurately satisfy each individual clock path delay.

Implementing the clock distribution network in a large VLSI-complexity system as a passive RC network is unacceptable for



(a)

(b)

Fig. 4. Clock skew and delay assignment for an example clock distribution network. (a) With local clock skew specified. (b) With branch delays assigned to satisfy the target clock skew schedule.

several reasons as follows: 1) the delay of each branch would be highly dependent on the delay of every other branch, 2) the clock signal waveform would degrade, limiting system performance and reliability, 3) an accurate delay model of a passive clock distribution network driving thousands of registers is difficult to obtain, and 4) the layout of the passive RC network is highly sensitive to small variations in position or length of the clock lines, producing unacceptable variations in the localized clock skew. It is preferable to make the delay of each branch independent of the delay of the remaining branches, and the clock branches designed such that the physical layout constraints can be easily satisfied.

To satisfy these criteria, the delay segments are implemented with active elements, specifically CMOS inverters. Due to the high input impedance of a CMOS inverter, the inverter effectively isolates each clock branch from each other. Additionally, the interconnect lines are constrained to behave as purely capacitive lines by appropriately inserting these distributed CMOS inverters as repeaters along the clock signal path [11]. The insertion points are chosen such that

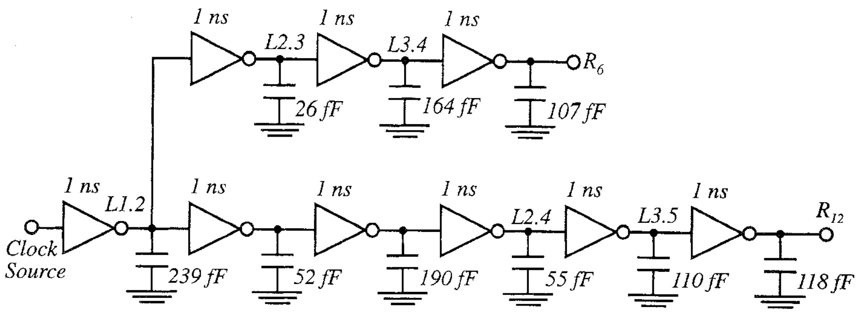


Fig. 5. Circuit implementation of the clock paths driving registers R_6 and R_{12}.

TABLE II
INVERTER DESIGN EQUATIONS

$C_{Li} = \frac{2I_{DO}}{V_{DD}}\left[t_{di} - \left(\frac{1}{2} - \frac{1-v_T}{1+\alpha}\right)t_{Ti-1}\right], \quad where\ v_T = \frac{V_{th}}{V_{DD}}$	(6)
$t_{Ti} = \frac{t_{0.9} - t_{0.1}}{0.8} = \frac{C_{Li}V_{DD}}{I_{DO}}\left(\frac{0.9}{0.8} + \frac{V_{DO}}{0.8V_{DD}} ln \frac{10V_{DO}}{eV_{DD}}\right)$	(7)

TABLE III
COMPARISON BETWEEN CALCULATED AND SIMULATED CLOCK PATH DELAYS

Clock Path	Specified	SPICE (ns)	Error (%)	
	Delay (ns)	min/nom/max	nom	worst case
R_1, R_2	7.0	6.49/6.84/7.15	2.3	7.3
R_{15}	7.0	6.59/7.11/7.63	1.6	9.0
R_{18}	8.0	7.28/8.10/8.98	1.3	12.3
R_5, R_6	4.0	3.70/4.06/4.43	1.5	11.3
R_{17}	8.0	7.20/8.06/8.91	0.8	11.4
R_{16}	4.0	3.96/4.06/4.13	1.3	3.3
R_9	7.0	6.53/7.09/7.61	1.3	8.7
R_{12}	6.0	5.44/6.12/6.81	2.0	13.5
R_7	7.0	6.77/6.84/6.93	2.3	3.3
R_8	9.0	8.63/8.98/9.35	0.3	4.1

TABLE IV
COMPARISON BETWEEN SPECIFIED AND SIMULATED CLOCK SKEW VALUES. BOLD VALUES ARE THOSE CLOCK SKEWS OUT OF THE PERMISSIBLE RANGE

Local Data	Specified Skew (ns)		Measured skew (ns)		Error (%)	
Path	range	scheduled	nom	worst case	nom	worst case
$R_1 - R_2$	[-4,2]	0.0	0.0	0.66	0.0	--
$R_{15} - R_{16}$	[-3,4]	3.0	3.0	3.67	0.0	23.0
$R_{16} - R_{17}$	[-7,-3]	-4.0	-4.04	-3.07	1.0	23.3
$R_{17} - R_{18}$	[-2,1]	0.0	0.04	**1.63**	--	--
$R_5 - R_6$	[-5,0]	0.0	0.0	**0.73**	0.0	--
$R_6 - R_9$	[-8,-2]	-3.0	-3.0	-2.10	0.0	30.0
$R_7 - R_8$	[-6,0]	-2.0	-2.14	-1.70	7.0	15.0
$R_6 - R_{12}$	[-4,-2]	-2.0	-2.06	**-1.01**	3.0	49.5

the output impedance of each inverter is much greater than the resistance of that portion of the interconnect section being driven. This strategy permits the length of a single interconnect line to be accurately modeled as a lumped capacitance with negligible resistance. However, the strategy also places a maximum constraint on the resistance of the section of the line (or effectively the length of the interconnect section between inverting repeaters), thereby limiting the physical placement of a clock branch within the circuit layout.

The delay equations describing the inverting repeater are used to determine the effective capacitive load of each branch and are based on the MOSFET α-power law I-V model developed by Sakurai and Newton [12]. Equation (6) in Table II, derived from [12], describes the load capacitance C_L in terms of the delay of a CMOS inverter, where I_{DO} is the drain current at $V_{GS} = V_{DS} = V_{DD}$. V_{DO} is the drain saturation voltage at $V_{GS} = V_{DD}$, V_{th} is the threshold voltage, α is the velocity saturation index, and V_{DD} is the power supply. The output waveform of the driving inverter is approximately the same as the input signal to all the branches connected to this inverter since the line resistance is negligible (due to the insertion of the repeaters), and is approximated by a ramp shaped waveform with transition time t_{Ti} given by (7) in Table II.

For each clock path within the clock tree, the CMOS inverters are designed as follows [13]: 1) the load of the initial branch of a clock path is determined from (6), assuming a step input clock source; 2) the slope of the output signal is calculated from (7) and applied in (6) to determine the load of the following branch, and (7) is used again to determine the slope of the output signal; and 3) step 2 is repeated for each subsequent branch of the clock path. The clock paths driving R_6 and R_{12} shown in Fig. 4 are depicted in Fig. 5 to illustrate this design procedure.

A. Process Parameter and Environmental Variations

Every semiconductor fabrication process can be characterized by process variations and environmental variations, such as temperature, supply voltage, and radiation. These variations may compromise both the performance and the accuracy of the clock distribution network. Each clock path delay can be modeled as the sum of two components: a deterministic delay component and a probabilistic delay component. The probabilistic component may significantly affect the accuracy of the implementation of the clock skew values, particularly the critical local data paths, since the permissible clock skew range of these critical paths is smaller.

In the circuit design phase of the clock tree, the cumulative effects of variations in the device parameters, such as the threshold voltage and channel mobility, can be collected into a single parameter, I_{DO} [12]. The clock tree example illustrated in Fig. 4 is designed with I_{DO} varying by ±15%. Simulations of clock path delay and clock skew based on these variations are presented in Tables III and IV, respectively, and are discussed in the following section.

After calculating the worst case variation of the clock delays, three clock skew cases are possible for each local data path. The first case occurs when the clock skew is within a permissible range, shown as region B in Fig. 3, and, in this region, the implementation of the clock tree is valid. In the second case, the implementation of at least one local data path within the system does not satisfy (1), shown as region C in Fig. 3, causing zero clocking. By increasing the clock period T_{CP}, the permissible clock skew range for each local data path is also increased ($T_{Skew ij(max)}$ is increased), permitting those local data paths previously in region C to satisfy (1). In the final case, certain local data paths may violate (3), creating race conditions (double clocking), shown as region A in Fig. 3. This situation is more critical since race conditions occur independently of the clock frequency, forcing the circuit to function improperly. Race conditions can be prevented during the design of the clock tree by inserting a "safety term" into *each* local data path, designated by the ζ_{ij} term in (3). This term is initially equal to zero for each local data path, and is progressively increased for those local data paths which violate (3). Observe that by changing ζ_{ij}, a new set of clock skew values and a minimum clock period T_{CP} must be calculated. This iterative process continues until the worst case variations of the selected clock skews no longer violate the corresponding permissible ranges.

V. Simulated Results

The accuracy of this design methodology can be measured by comparing the individual clock path delay and clock skew values determined from the clock skew scheduling, topological, and circuit synthesis phases with delay values derived from SPICE. In Table III, the difference between the calculated and simulated clock path delays for the clock tree topology shown in Fig. 4 is compared. The target delay obtained from the topological and circuit synthesis of the clock distribution network is depicted in the second column. The minimum, nominal, and maximum delay values of each clock path derived from SPICE circuit simulation using Level-3 device models are shown in the third column, while the per cent error between the specified and the nominal and worst case delay are depicted in columns four and five, respectively. Note that the maximum error for nominal conditions is less than 2.5% and for worst case conditions is 13.5%.

A more significant measure of the effectiveness of this design methodology is to guarantee that the clock skew of any local data path is accurately implemented assuming nominal clock delay values while remaining within the permissible range assuming worst case conditions. The clock skew between registers for the same circuit example is illustrated in Table IV. The permissible range and the scheduled clock skew are shown in columns two and three, respectively, for the local data path listed in column one. The values obtained from SPICE circuit simulations for the clock skew obtained under nominal and worst case clock delay values are depicted in columns four and five, while the per cent error between the scheduled clock skew and the simulated clock distribution network with nominal and worst case clock delay values are shown in columns six and seven, respectively. Assuming nominal conditions, the maximum error of the clock skew for this example is 7%, a number well within practical and useful limits, particularly for those paths in which the clock paths driving two sequentially adjacent registers do not share a significant portion of the clock distribution network.

The difference between the simulated and the chosen clock skew values becomes greater when the circuit is analyzed under worst case conditions, exhibiting variations of up to 50%. However, in order to obtain a race-free clock distribution network, the clock skew obtained under best case, nominal, and worst case conditions of operation must fall within the permissible range calculated for each local data path. In the example circuit, this requirement is not satisfied for the clock skew values obtained for the local data paths R_{17}–R_{18}, R_5–R_6, and R_6–R_{12}, represented by the bold values in column five of Table IV. These values fall within region C of Fig. 3 and, as described in Section IV, the clock period must be increased until (1) is satisfied for every local data path. Therefore, the minimum clock period determined for this example circuit must be increased from 8 to 9 ns, providing a permissible range of [-4, 2] for R_{17}–R_{18}, [-5, 1] for R_5–R_6, and [-8, -1] for R_6–R_{12}, which encompasses the maximum clock skew for each local data path. Note that if zero clock skew is applied to every local data path the minimum clock period for this circuit is 11 ns, 22% greater than the 9 ns clock period derived from exploiting nonzero localized clock skew while considering the effects of delay variations.

VI. Conclusion

Synchronous digital systems require the efficient synthesis of high speed clock distribution networks in order to obtain higher levels of circuit performance and reliability and improved design efficiency. In this brief, circuit performance and reliability are enhanced by using nonzero localized clock skew to reduce the minimum clock period and to eliminate race conditions. An integrated top-down methodology is presented for synthesizing clock distribution networks. This methodology is composed of four phases, 1) optimal clock scheduling, 2) topological design, 3) design of the circuit delay elements, and 4) physical layout. Distributed buffers are included during the top-down synthesis process, permitting the clock distribution network to be optimized for the specific performance requirements of the circuit application, while ensuring that the clock tree is tolerant to process parameter variations. The clock skew is constrained within a permissible range for locally minimizing the effects of parameter variations under worst case conditions. Simulations exhibit excellent agreement between the synthesized clock skew and the clock skew values obtained from SPICE.

Thus, an integrated top-down methodology for synthesizing tree-structured process tolerant clock distribution networks for high speed VLSI/ULSI CMOS circuits is presented. This methodology, based on inserted delay elements, accurately synthesizes localized nonzero clock skews, thereby increasing the system clock frequency while eliminating race conditions.

References

[1] E. G. Friedman and S. Powell, "Design and analysis for a hierarchical clock distribution system for synchronous standard cell/macrocell VLSI," *IEEE J. Solid-State Circuits*, vol. SC-21, pp. 240–246, Apr. 1986.

[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," in *Proc. IEEE Int. Conf. Comput. Design*, Oct. 1986, pp. 118–122.

[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 Trans. Circuits Syst. II*, vol. 39, pp. 799–814, Nov. 1992.

[4] A. B. Kahng and G. Robins, *On Optimal Interconnections for VLSI*. Boston, MA: Kluwer Academic, 1995.

[5] E. G. Friedman, *Clock Distribution Networks in VLSI Circuits and Systems*. Piscataway, NJ: IEEE Press, 1995.

[6] J. P. Fishburn, "Clock skew optimization," *IEEE Trans. Comput.*, vol. 39, pp. 945–951, July 1990.

[7] I. Lin, J. A. Ludwig, and K. Eng, "Analyzing cycle stealing on synchronous circuits with level-sensitive latches," in *Proc. IEEE/ACM Design Automation Conf.*, June 1992, pp. 393–398.

[8] K. A. Sakallah, T. N. Mudge, and O. A. Olukotun, "*checkTc* and *minTc*: Timing verification and optimal clocking of synchronous digital circuits," in *Proc. IEEE/ACM Design Automation Conf.*, June 1990, pp. 111–117.

[9] T. G. Szymanski, "Computing optimal clock schedules," in *Proc. IEEE/ACM Design Automation Conf.*, June 1992, pp. 399–404.

[10] J. L. Neves and E. G. Friedman, "Topological design of clock distribution networks based on nonzero clock skew specifications," in *Proc. IEEE 36th Midwest Conf. Circuits Syst.*, Aug. 1993, pp. 468–471.

[11] S. Dhar and M. A. Franklin, "Optimum buffer circuits for driving long uniform lines," *IEEE J. Solid State Circuits*, vol. 26, pp. 32–40, Jan. 1991.

[12] T. Sakurai and A. R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," *IEEE J. Solid State Circuits*, vol. 25, pp. 584–594, Apr. 1990.

[13] J. L. Neves and E. G. Friedman, "Circuit synthesis of clock distribution networks based on nonzero clock skew," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1994, pp. 4.175–4.178.