

Clock Distribution Networks in Synchronous Digital Integrated Circuits

EBY G. FRIEDMAN

Invited Paper

Clock distribution networks synchronize the flow of data signals among synchronous data paths. The design of these networks can dramatically affect system-wide performance and reliability. A theoretical background of clock skew is provided in order to better understand how clock distribution networks interact with data paths. Minimum and maximum timing constraints are developed from the relative timing between the localized clock skew and the data paths. These constraint relationships are reviewed, and compensating design techniques are discussed.

The field of clock distribution network design and analysis can be grouped into a number of subtopics: 1) circuit and layout techniques for structured custom digital integrated circuits; 2) the automated layout and synthesis of clock distribution networks with application to automated placement and routing of gate arrays, standard cells, and larger block-oriented circuits; 3) the analysis and modeling of the timing characteristics of clock distribution networks; and 4) the scheduling of the optimal timing characteristics of clock distribution networks based on architectural and functional performance requirements. Each of these areas is described, the clock distribution networks of specific industrial circuits are surveyed, and future trends are discussed.

Keywords—Clock distribution networks, clock trees, clock skew, clock skew scheduling, CMOS, H-trees, interconnect delay, process variations, RLC impedances, synchronization, timing optimization.

I. INTRODUCTION

In a synchronous digital system, the clock signal is used to define a time reference for the movement of data within that system. Since this function is vital to the operation of a synchronous system, much attention has been given to the characteristics of these clock signals and the networks used in their distribution. Clock signals are often regarded as simple control signals; however, these signals have some very special characteristics and attributes. Clock signals are typically

loaded with the greatest fanout, travel over the longest distances, and operate at the highest speeds of any signal, either control or data, within the entire system. Since the data signals are provided with a temporal reference by the clock signals, the clock waveforms must be particularly clean and sharp. Furthermore, these clock signals are particularly affected by technology scaling, in that long global interconnect lines become much more highly resistive as line dimensions are decreased. This increased line resistance is one of the primary reasons for the growing importance of clock distribution on synchronous performance. Finally, the control of any differences in the delay of the clock signals can severely limit the maximum performance of the entire system as well as create catastrophic race conditions in which an incorrect data signal may latch within a register.

Most synchronous digital systems consist of cascaded banks of sequential registers with combinatorial logic between each set of registers. The functional requirements of the digital system are satisfied by the logic stages. The global performance and local timing requirements are satisfied by the careful insertion of pipeline registers into equally spaced time windows to satisfy critical worst case timing constraints. The proper design of the clock distribution network further ensures that these critical timing requirements are satisfied and that no race conditions exist [1]–[25], [173], [174]. With the careful design of the clock distribution network, system-level synchronous performance can actually increase, surpassing the performance advantages of asynchronous systems by permitting synchronous performance to be based on average path delays rather than worst case path delays, without incurring the handshaking protocol delay penalties required in most asynchronous systems.

In a synchronous system, each data signal is typically stored in a latched state within a bistable register [175] awaiting the incoming clock signal, which determines when the data signal leaves the register. Once the enabling clock signal reaches the register, the data signal leaves the bistable register and propagates through the combinatorial network

Manuscript received May 22, 2000; revised December 18, 2000.

The author is with the Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY 14627 USA (e-mail: friedman@ece.rochester.edu).

Publisher Item Identifier S 0018-9219(01)03966-4.

and, for a properly working system, enters the next register and is fully latched into that register before the next clock signal appears. Thus, the delay components that make up a general synchronous system are composed of the following three individual subsystems [147], [148], [176]:

- 1) memory storage elements;
- 2) logic elements;
- 3) clocking circuitry and distribution network.

The interrelationships among these three subsystems of a synchronous digital system are critical to achieving maximum levels of performance and reliability. The important area of clock generation, as compared to clock distribution which is the primary topic of this paper, bears separate focus.

The paper is organized as follows. In Section II, an overview of the operation of a synchronous system is provided. In Section III, fundamental definitions and the timing characteristics of clock skew are discussed. The timing relationships between a local data path and the clock skew of that path are described in Section IV. The interplay among the aforementioned three subsystems making up a synchronous digital system is described in Section V; particularly, how the timing characteristics of the memory and logic elements constrain the design and synthesis of clock distribution networks. Different forms of clock distribution networks, such as buffered trees and H-trees, are discussed. The automated layout and synthesis of clock distribution networks are described in Section VI. Techniques for making clock distribution networks less sensitive to process parameter variations are discussed in Section VII. Localized scheduling of the clock delays is useful in optimizing the performance of high-speed synchronous circuits. The process for determining the optimal timing characteristics of a clock distribution network is reviewed in Section VIII. The application of clock distribution networks to high-speed circuits has existed for many years. The design of the clock distribution network of certain important VLSI-based systems has been described in the literature, and some examples of these circuits are described in Section IX. In an effort to provide some insight into future and evolving areas of research relevant to high-performance clock distribution networks, some potentially important topics for future research are discussed in Section X. Finally, a summary of this paper with some concluding remarks is provided in Section XI.

II. SYNCHRONOUS SYSTEMS

A digital synchronous circuit is composed of a network of functional logic elements and globally clocked registers. For an arbitrary *ordered pair* of registers (R_1, R_2) , one of the following two situations can be observed: either 1) the input of R_2 cannot be reached from the output of R_1 by propagating through a sequence of logical elements only or 2) there exists at least one sequence of logic blocks that connects the output of R_1 to the input of R_2 . In the former case, switching events at the output of the register R_1 do not affect the input of the register R_2 during the same clock period. In the latter case—denoted by $R_1 \Rightarrow R_2$ —signal switching at the output

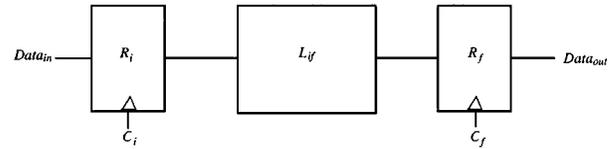


Fig. 1. Local data path.

of R_1 will propagate to the input of R_2 . In this case, (R_1, R_2) is called a *sequentially-adjacent* pair of registers which make up a *local data path* (see Fig. 1).

1) *Delay Components of Data Path:* The minimum allowable clock period $T_{CP}(\min)$ between any two registers in a sequential data path is given by

$$\frac{1}{f_{\text{clk MAX}}} = T_{CP}(\min)T_{PD}(\max) + T_{\text{Skew}} \quad (1)$$

where

$$T_{PD}(\max) = T_{C-Q} + T_{\text{Logic}} + T_{\text{Int}} + T_{\text{Set-up}} = D(i, f) \quad (2)$$

and the total path delay of a data path $T_{PD}(\max)$ is the sum of the maximum time required for the data to leave the initial register once the clock signal C_i arrives, T_{C-Q} , the time necessary to propagate through the logic and interconnect, $T_{\text{Logic}} + T_{\text{Int}}$, and the time required to successfully propagate to and latch within the final register of the data path, $T_{\text{Set-up}}$. Observe that the latest arrival time is given by $T_{\text{Logic}(\max)}$ and the earliest arrival time is given by $T_{\text{Logic}(\min)}$, since data is latched into each register within the same clock period.

The sum of the delay components in (2) must satisfy the timing constraint of (1) in order to support the clock period $T_{CP}(\min)$, which is the inverse of the maximum possible clock frequency, $f_{\text{clk MAX}}$. Note that the clock skew $T_{\text{Skew}ij}$ can be positive or negative depending on whether C_f leads or lags C_i , respectively. The clock period is chosen such that the latest data signal generated by the initial register is latched in the final register by the next clock edge after the clock edge that activated the initial register. Furthermore, in order to avoid race conditions, the local path delay must be chosen such that, for any two sequentially-adjacent registers in a multistage data path, the latest data signal must arrive and be latched within the final register before the earliest data signal generated with the next clock pulse arrives. The waveforms depicted in Fig. 2 show the timing requirement of (1) being barely satisfied (i.e., the data signal arrives at R_f just before the clock signal arrives at R_f).

An example of a local data path $R_i \Rightarrow R_f$ is shown in Fig. 1. The clock signals C_i and C_f synchronize the sequentially-adjacent pair of registers R_i and R_f , respectively. Signal switching at the output of R_i is triggered by the arrival of the clock signal C_i . After propagating through the logic block $L_{i,f}$, this signal will appear at the input of R_f . Therefore, a *nonzero* amount of time elapses between the triggering event and the signal switching at the input of R_f . The minimum and maximum values of this delay are called the *short* and *long* delays, respectively, and are denoted by

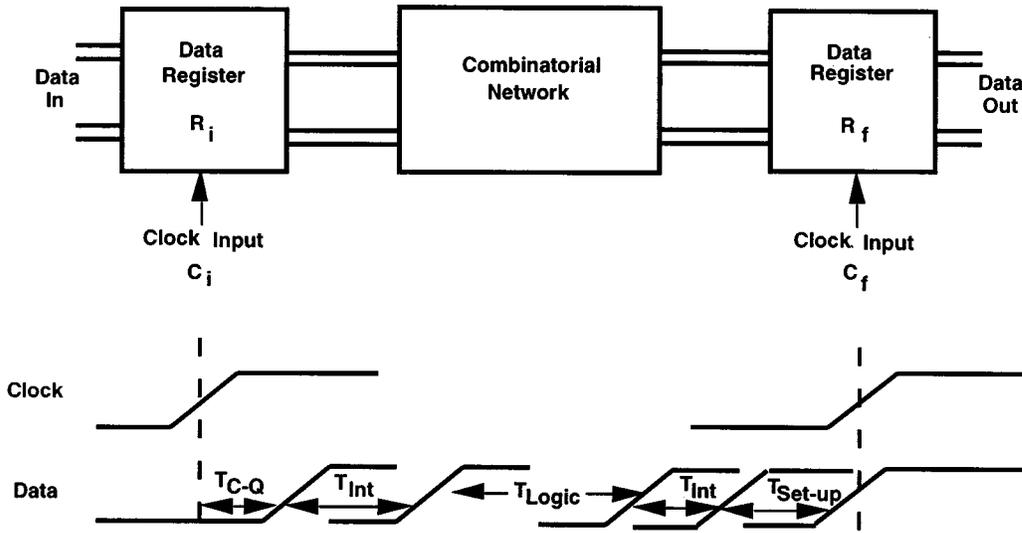


Fig. 2. Timing diagram of clocked data path.

$d(i, f)$ and $D(i, f)$, respectively. Note that both $d(i, f)$ and $D(i, f)$ are due to the accumulative effects of three sources of delay [4]. These sources are the clock-to-output delay of the register R_i , a delay introduced by the signal propagating through L_{if} , and an interconnect delay due to the presence of wires on the signal path, $R_i \Rightarrow R_f$.

A synchronous digital system can be modeled [177], [178] as a *directed graph* G with vertex set $V = \{v_1, \dots, v_{N_R}\}$ and edge set $E = \{e_1, \dots, e_{N_P}\}$. An example of a circuit graph G is illustrated in Fig. 3. The number of registers in a circuit is $|V| = N_R$ and vertex v_k corresponds to the register R_k . The number of local data paths in a circuit is $|E| = N_P$. There is an edge directed from v_i to v_j iff $R_i \Rightarrow R_j$. In the case where multiple paths exist between a sequentially-adjacent pair of registers $R_i \Rightarrow R_j$, only one edge connects v_i to v_j . The underlying graph G_u of the graph G is a nondirected graph that has the same vertex set V , where the directions have been removed from the edges. In both Fig. 3(a) and (b), an input or an output of the circuit is indicated by an edge that is incident to only one vertex.

III. THEORETICAL BACKGROUND OF CLOCK SKEW

A schematic of a generalized synchronized data path is presented in Figs. 1 and 2, where C_i and C_f represent the clock signals driving a sequentially-adjacent pair of registers, specifically, the initial register R_i and the final register R_f of a data path, respectively. Both clock signals originate from the same clock signal source. A pair of registers are sequentially-adjacent if only combinatorial logic (no sequential elements) exist between the two registers. The propagation delay from the clock source to the j th clocked register is the **clock delay**, T_{C_j} . The clock delays of the initial clock signal T_{C_i} and the final clock signal T_{C_f} define the time reference when the data signals begin to leave their respective registers. These clock signals originate from a clock distribution network which is designed to generate a specific clock signal waveform which is used to synchronize each register. This common clock distribution network structure is based

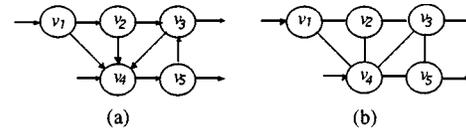


Fig. 3. Graph G of a circuit $N_R = 5$ registers. (a) The directed graph G . (b) The underlying graph G_u corresponding to the graph G in (a).

on *equipotential clocking*, where the entire network is considered a surface which must be brought to a specific voltage (clock signal polarity) at each half of the clock cycle. Ideally, clocking events occur at all registers simultaneously. Given this global clocking strategy, clock signal arrival times (at each register) are defined with respect to a universal time reference.

1) *Definition of Clock Skew*: The difference in clock signal arrival time between *two sequentially-adjacent registers*, as shown in (1), is the **clock skew**, T_{Skew} . If the clock signals C_i and C_f are in complete synchronism (i.e., the clock signals arrive at their respective registers at exactly the same time), the clock skew is zero. A definition of clock skew is provided below.

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

$$T_{\text{Skew}ij} = T_{C_i} - T_{C_j} \quad (3)$$

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

It is important to observe that the temporal skew between the arrival time of different clock signals is only relevant to sequentially-adjacent registers making up a single data path, as shown in Fig. 1. Thus, system-wide (or chip-wide) clock skew between nonsequentially connected registers, from an analysis viewpoint, has no effect on the performance and reliability of the synchronous system and is essentially meaningless. However, from a design perspective, system-wide global clock skew places constraints on the permissible local

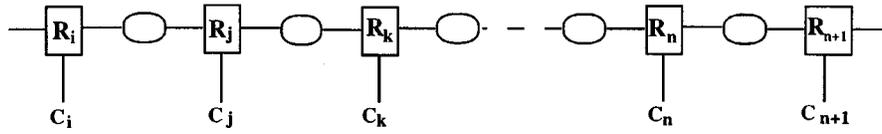


Fig. 4. Global data path composed of multiple local data paths.

clock skew. It should be noted that in [11], [179], Hatamian designates the lead/lag clock skew polarity (positive/negative clock skew) notation as the opposite of that used here.

Different clock signal paths can have different delays for a variety of reasons. Wann and Franklin [3] present the following causes of clock skew:

- 1) differences in line lengths from the clock source to the clocked register;
- 2) differences in delays of any active buffers (e.g., distributed buffers) within the clock distribution network [due to 3) and 4) below];
- 3) differences in passive interconnect parameters, such as line resistivity, dielectric constant and thickness, via/contact resistance, line and fringing capacitance, and line dimensions;
- 4) differences in active device parameters, such as MOS threshold voltages and channel mobilities, which affect the delay of the active buffers.

It should also be noted that, for a well-designed and balanced clock distribution network, the distributed clock buffers are the principal source of clock skew.

To determine the clock delay from the clock source to each register, it is important to investigate the relationship between the clock skews of the sequentially-adjacent registers occurring within a global data path. Furthermore, it is necessary to consider the effects of feedback within global data paths on the clock skew.

The path between two sequentially-adjacent registers is described in this paper as a local data path, as compared to a global data path, where a global data path can consist of one or more local data paths (see Fig. 4). The relationship between the clock skew of sequentially-adjacent registers in a global data path is called conservation of clock skew and is formalized below.

Theorem 1: For any given global data path, clock skew is conserved. Alternatively, the clock skew between any two registers in a global data path 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 same two 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 nonsequentially-adjacent registers belonging to the same global data path. It also illustrates that the clock skew between any two nonsequentially-adjacent registers which do not belong to the same global data path has no physical meaning since no functional data transfer between these registers occurs.

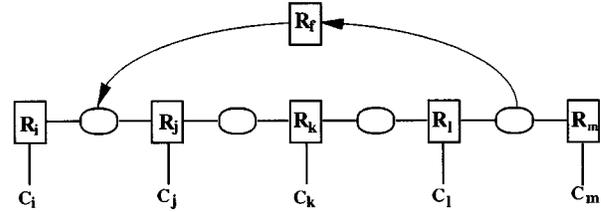


Fig. 5. Data path with feedback paths.

A typical sequential circuit may contain sequential feedback paths, as illustrated in Fig. 5. 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 Fig. 5, the initial and final registers in 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:

$$T_{\text{Skew feedback},lj} = -T_{\text{Skew forward},jl}. \quad (4)$$

Both Theorems 1 and 2 are useful for determining the optimal clock skew schedule within a synchronous digital system, specifically, the set of local clock skew values that maximizes system performance and reliability. The process for determining these clock skew values is discussed in Section VIII.

IV. TIMING CONSTRAINTS DUE TO CLOCK SKEW

The magnitude and polarity of the clock skew have a two-sided effect on system performance and reliability. Depending upon whether C_i leads or lags C_f and upon the magnitude of T_{Skew} with respect to T_{PD} , system performance and reliability can either be degraded or enhanced. These cases are discussed below.

A. Maximum Data Path/Clock Skew Constraint Relationship

For a design to meet its specified timing requirements, the greatest propagation delay of any data path between a pair of data registers, R_i and R_f , being synchronized by a clock distribution network must be less than the minimum clock period (the inverse of the maximum clock frequency) of the circuit as shown in (1) [7], [8], [11], [16], [28], [145], [176], [179], [180]. If the time of arrival of the clock signal at the final register of a data path T_{Cf} leads that of the time of

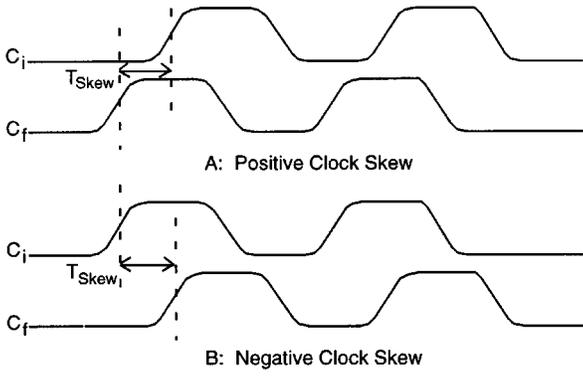


Fig. 6. Clock timing diagrams.

arrival of the clock signal at the initial register of the same sequential data path T_{C_i} [see Fig. 6(A)], the clock skew is referred to as **positive clock skew** and, under this condition, the maximum attainable operating frequency is decreased. Positive clock skew is the additional amount of time which must be added to the minimum clock period to reliably apply a new clock signal at the final register, where reliable operation implies that the system will function correctly at low as well as at high frequencies (assuming fully static logic). It should be noted that positive clock skew only affects the maximum frequency of a system and cannot create race conditions.

In the positive clock skew case, the clock signal arrives at R_f before it reaches R_i . From (1) and (2), the maximum permissible positive clock skew can be expressed as [7], [8], [11], [16], [28], [145], [176], [179], [180]

$$\begin{aligned} T_{\text{Skew}} &\leq T_{\text{CP}} - T_{\text{PD}(\text{max})} \\ &= T_{\text{CP}} - (T_{C-Q} + T_{\text{Logic}(\text{max})} + T_{\text{Int}} + T_{\text{Set-up}}) \\ &\quad \text{for } T_{C_i} > T_{C_f} \quad (5) \end{aligned}$$

where $T_{\text{PD}(\text{max})}$ is the maximum path delay between two sequentially-adjacent registers. This situation is the typical critical path timing analysis requirement commonly seen in most high-performance synchronous digital systems. If (5) is not satisfied, the system will not operate correctly at that specific clock period (or clock frequency). Therefore, T_{CP} must be increased for the circuit to operate correctly, thereby decreasing the system performance. In circuits where the tolerance for positive clock skew is small [T_{Skew} in (5) is small], the clock and data signals should be run in the same direction, thereby forcing C_f to lag C_i and making the clock skew negative.

B. Minimum Data Path/Clock Skew Constraint Relationship

If the clock signal arrives at R_i before it reaches R_f [see Fig. 6(B)], the clock skew is defined as being negative. **Negative clock skew** can be used to improve the maximum performance of a synchronous system by decreasing the delay of a critical path; however, a potential minimum constraint can occur, creating a race condition [11], [12], [31], [138], [139], [145], [176], [179], [181]. In this case, when C_f lags C_i , the clock skew must be less than the time required for the data signal to leave the initial register, propagate through

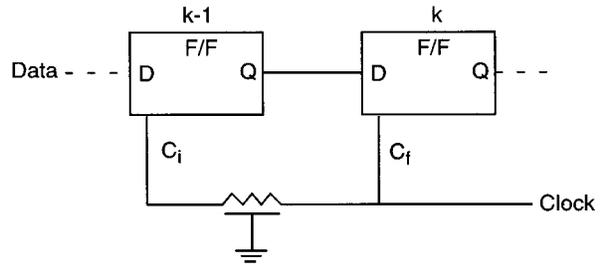


Fig. 7. K -bit shift register with positive clock skew.

the interconnect, combinatorial logic, and setup in the final register (see Fig. 1). If this condition is not met, the data stored in register R_f is overwritten by the data that had been stored in register R_i and has propagated through the combinatorial logic. Furthermore, a circuit operating close to this condition might pass system diagnostics but malfunction at unpredictable times due to fluctuations in ambient temperature or power supply voltage [145]. Correct operation requires that R_f latches data which correspond to the data R_i latched during the previous clock period. This constraint on clock skew is

$$\begin{aligned} |T_{\text{Skew}}| &\leq T_{\text{PD}(\text{min})} \\ &= T_{C-Q} + T_{\text{Logic}(\text{min})} + T_{\text{Int}} + T_{\text{Hold}} \\ &\quad \text{for } T_{C_f} > T_{C_i} \quad (6) \end{aligned}$$

where $T_{\text{PD}(\text{min})}$ is the minimum path delay, denoted by $d(i, f)$, between two sequentially-adjacent registers and T_{Hold} is the amount of time the input data signal must be stable once the clock signal changes state.

An important example in which this minimum constraint can occur is in those designs which use cascaded registers, such as a serial shift register or a k -bit counter, as shown in Fig. 7 (note that a distributed RC impedance is between C_i and C_f). In cascaded register circuits, $T_{\text{Logic}(\text{min})}$ is zero and T_{Int} approaches zero (since cascaded registers are typically designed, at the geometric level, to abut). If $T_{C_f} > T_{C_i}$ (i.e., negative clock skew), then the minimum constraint becomes

$$|T_{\text{Skew}}| \leq T_{C-Q} + T_{\text{Hold}}, \quad \text{for } T_{C_f} > T_{C_i} \quad (7)$$

and all that is necessary for the system to malfunction is a poor relative placement of the flip flops or a highly resistive connection between C_i and C_f . In a circuit configuration such as a shift register or counter, where negative clock skew is a more serious problem than positive clock skew, provisions should be made to force C_f to lead C_i , as shown in Fig. 7.

As higher levels of integration are achieved in high-complexity VLSI circuits, on-chip testability [182] becomes necessary. Data registers, configured in the form of serial set/scan chains when operating in the test mode, are a common example of a design for testability (DFT) technique. The placement of these circuits is typically optimized around the functional flow of the data. When the system is reconfigured to use the registers in the role of the set/scan function, different local path delays are possible. In particular, the clock skew of the reconfigured local data path

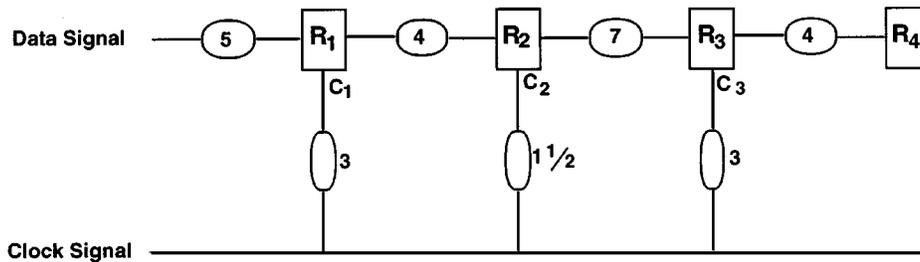


Fig. 8. Example of applying localized negative clock skew to synchronous circuit.

can be negative and greater in magnitude than the local register delays. Therefore, with increased negative clock skew, (7) may no longer be satisfied and incorrect data may latch into the final register of the reconfigured local data path. Therefore, it is imperative that attention be placed on the clock distribution of those paths that have nonstandard modes of operation.

In ideal scaling of MOS devices, all linear dimensions and voltages are multiplied by the factor $1/S$, where $S > 1$ [16], [183]–[185]. Device dependent delays, such as T_{C-Q} , T_{Set-up} , and T_{Logic} , scale as $1/S$ while interconnect dominated delays such as T_{Skew} remain constant to first order, and if fringing capacitance and electromigration are considered, actually increase with decreasing dimensions. Therefore, when examining the effects of dimensional scaling on system reliability, (6) and (7) should be considered carefully [186]. One straight forward method to avoid the effect of technology scaling on those data paths particularly susceptible to negative clock skew is to not scale the clock distribution lines. Svensson and Afghahi [136] show that by using courser than ordinary lines for the global clock distribution, 20-mm-wide chip sizes with CMOS circuits scaled to $0.3 \mu\text{m}$ polysilicon lines would have comparable logic and cross-chip interconnect delays (on the order of 0.5 ns), making possible synchronous clock frequencies of up to 1 GHz. Therefore, the scaling of device technologies can severely affect the design and operation of clock distribution networks, necessitating specialized strategies and compensation techniques.

C. Enhancing Synchronous Performance by Applying Localized Clock Skew

Localized clock skew can be used to improve synchronous performance by providing more time for the critical worst case data paths [142], [145], [147], [148], [176], [181]. By forcing C_i to *lead* C_f at each critical local data path, excess time is shifted from the neighboring less critical local data paths to the critical local data paths. This negative clock skew represents the additional amount of time that the data signal at R_i has to propagate through the logic stages and interconnect sections and into the final register. Negative clock skew subtracts from the logic path delay, thereby decreasing the minimum clock period. Thus, applying negative clock skew, in effect, increases the total time that a given critical data path has to accomplish its functional requirements by giving the data signal released from R_i more time to propagate through

the logic and interconnect stages and latch into R_f . Thus, the differences in delay between each local data path are minimized, thereby compensating for any inefficient partitioning of the global data path into local data paths that may have occurred, a common situation in many practical systems. Different terms have been used in the literature to describe negative clock skew, such as “double-clocking” [145], “deskewing data pulses” [31], “cycle stealing” [141], [142], “useful clock skew” [185], and “prescribed skew” [78].

The maximum permissible negative clock skew of a data path, however, is dependent upon the clock period itself as well as the time delay of the previous data paths. This results from the structure of the serially cascaded local data paths making up the global data path. Since a particular clock signal synchronizes a register which functions in a dual role, as the initial register of the next local data path and as the final register of the previous data path, the earlier C_i is for a given data path, the earlier that same clock signal, now C_f , is for the previous data path. Thus, the use of negative clock skew in the i th path results in a positive clock skew for the preceding path, which may then establish the new upper limit for the system clock frequency.

1) *An Example of Applying Localized Negative Clock Skew to Synchronous Circuits:* Consider the nonrecursive synchronous circuit shown in Fig. 8 where the horizontal ovals represent logic elements with logic delays and the vertical ovals represent clock delays. Since the local data path from R_2 to R_3 represents the worst case path (assuming the register delays are equal), by delaying C_3 with respect to C_2 , negative clock skew is added to the R_2 - R_3 local data path. If C_1 is synchronized with C_3 , the R_1 - R_2 local data path receives some positive clock skew. Thus, assuming the register delays are both 2 ns, C_2 should be designed to lead C_3 by 1.5 ns, forcing both paths to have the same total local path delay, $T_{PD} + T_{Skew} = 7.5$ ns. The delay of the critical path of the synchronous circuit is temporally refined to the precision of the clock distribution network and the entire system (for this simple example) could operate at a clock frequency of 133.3 MHz, rather than 111.1 MHz if no localized clock skew is applied. The performance characteristics of the system, both with and without the application of localized clock skew, are summarized in Table 1.

Note that $|T_{Skew}| < T_{PD}$ ($|-1.5 \text{ ns}| < 9 \text{ ns}$) for the R_2 - R_3 local data path; therefore, the correct data signal is successfully latched into R_3 and no minimum data path/clock skew constraint relationship exists. This design technique of applying localized clock skew is particularly

Table 1
Performance Characteristics of Circuit Shown in Fig. 8 Without and With Localized Clock Skew

Local Data Path	$T_{PD}(\text{min})$ - zero skew	T_{Ci}	T_{Cf}	T_{Skew}	$T_{PD}(\text{min})$ - with non-zero skew
R_1 to R_2	$4 + 2 + 0 = 6$	3	1.5	1.5	$4 + 2 + 1.5 = 7.5$
R_2 to R_3	$7 + 2 + 0 = 9$	1.5	3	-1.5	$7 + 2 - 1.5 = 7.5$
f_{Max}	111.1 MHz.				133.3 MHz.

- All time units are in nanoseconds

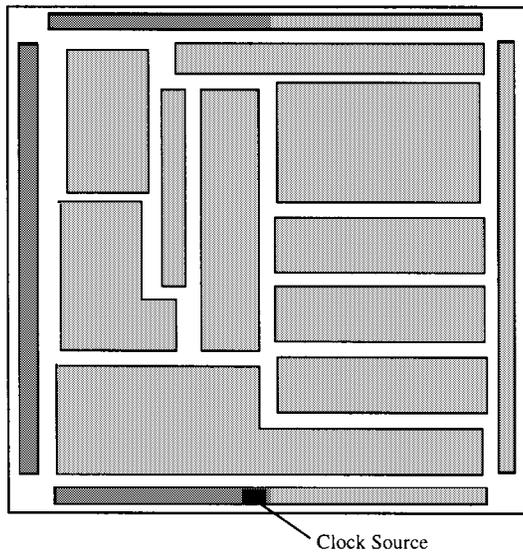


Fig. 9. Floorplan of structured custom VLSI circuit requiring synchronous clock distribution.

effective in sequentially-adjacent temporally irregular local data paths; however, it is applicable to any type of synchronous sequential system, and for certain architectures, a significant improvement in performance is both possible and likely.

The limiting condition for applying localized negative clock skew is determined by the control of the clock skew variations and by the difference in path delay between neighboring local data paths. These clock skew variations are due to power supply variations, process tolerances, where process parameters may vary over a specified range, and environmental effects, such as temperature or radiation, which, for example, can shift both MOS threshold voltages and channel mobilities.

V. CLOCK DISTRIBUTION DESIGN OF STRUCTURED CUSTOM VLSI CIRCUITS

Many different approaches, from *ad hoc* to algorithmic, have been developed for designing clock distribution networks in synchronous digital integrated circuits. The requirement of distributing a tightly controlled clock signal to each synchronous register on a large nonredundant hierarchically structured integrated circuit (an example floorplan is shown in Fig. 9) within specific temporal bounds is

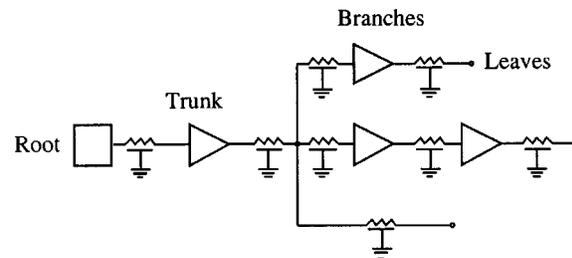


Fig. 10. Tree structure of clock distribution network.

difficult and problematic. Furthermore, the tradeoffs that exist among system speed, physical die area, and power dissipation are greatly affected by the clock distribution network. The design methodology and structural topology of the clock distribution network should be considered in the development of a system for distributing the clock signals. Therefore, various clock distribution strategies have been developed. The most common and general approach to equipotential clock distribution is the use of buffered trees, which are discussed in Section V-A. In contrast to these asymmetric structures, symmetric trees, such as H-trees, are used to distribute high-speed clock signals. This topic is described in Section V-B. In developing structured custom integrated circuits, such as is illustrated by the floorplan pictured in Fig. 9, specific circuit design techniques are used to control the signal properties within the clock distribution network. One important compensation technique is described in Section V-C. Low-power design techniques are an area of significant currency and importance. Some examples of different strategies for reducing the power dissipated within the clock distribution network are reviewed in Section V-D.

A. Buffered Clock Distribution Trees

The most common strategy for distributing clock signals in VLSI-based systems is to insert buffers either at the clock source and/or along a clock path, forming a tree structure. Thus, the unique clock source is frequently described as the root of the tree, the initial portion of the tree as the trunk, individual paths driving each register as the branches, and the registers being driven as the leaves. This metaphor for describing a clock distribution network is commonly accepted and used throughout the literature and is illustrated in Fig. 10. Occasionally, a mesh version of the clock tree structure is

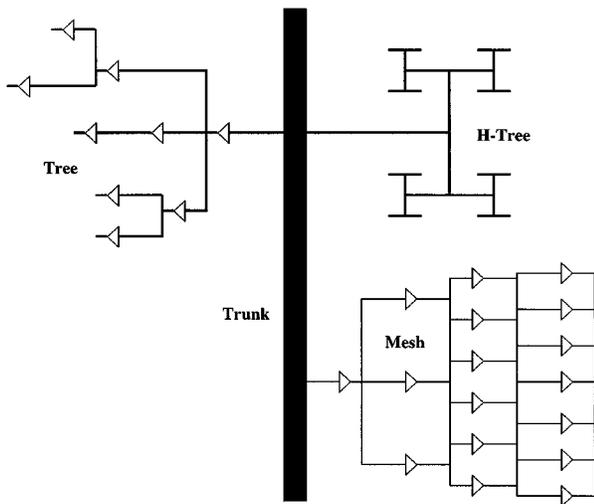


Fig. 11. Common structures of clock distribution networks including a trunk, tree, mesh, and H-tree.

used in which shunt paths further down the clock distribution network are placed to minimize the interconnect resistance within the clock tree. This mesh structure effectively places the branch resistances in parallel, minimizing the clock skew. An example of this mesh structure is described and illustrated in Section IX-B. The mesh version of the clock tree is considered in this paper as an extended version of the standard, more commonly used clock tree depicted in Fig. 10. The clock distribution network is therefore typically organized as a rooted tree structure [176], [187], as illustrated in Fig. 10 (and later in Fig. 16) [176]. Various forms of a clock distribution network, including a trunk, tree, mesh, and H-tree (see Section V-B for more information on symmetric H-tree structures) are illustrated in Fig. 11.

If the interconnect resistance of the buffer at the clock source is small as compared to the buffer output resistance, a single buffer is often used to drive the entire clock distribution network. This strategy may be appropriate if the clock is distributed entirely on metal, making load balancing of the network less critical. The primary requirement of a single buffer system is that the buffer should provide sufficient current to drive the network capacitance (both interconnect and fanout) while maintaining high-quality waveform shapes (i.e., short transition times) and minimizing the effects of the interconnect resistance by ensuring that the output resistance of the buffer is much greater than the resistance of the interconnect section being driven.

An alternative approach to using only a single buffer at the clock source is to distribute buffers throughout the clock distribution network, as shown in Fig. 10. This approach requires additional area but greatly improves the precision and control of the clock signal waveforms and is necessary if the resistance of the interconnect lines is nonnegligible. The distributed buffers serve the double function of amplifying the clock signals degraded by the distributed interconnect impedances and isolating the local clock nets from upstream load impedances [28]. A three-level buffer clock distribution network utilizing this strategy is shown in Fig. 12. In this approach a single buffer drives multiple clock paths (and

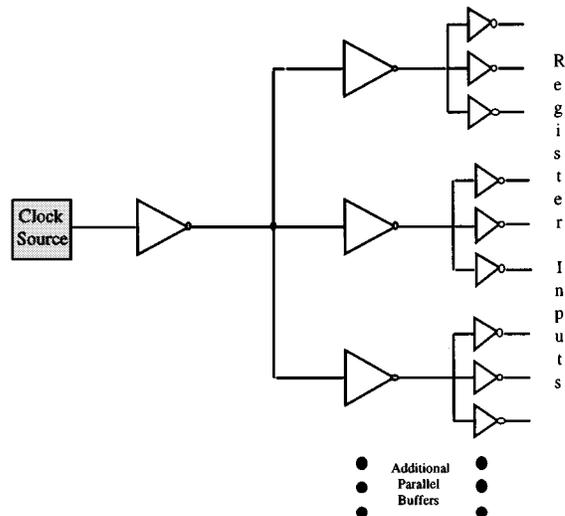


Fig. 12. Three-level buffer clock distribution network.

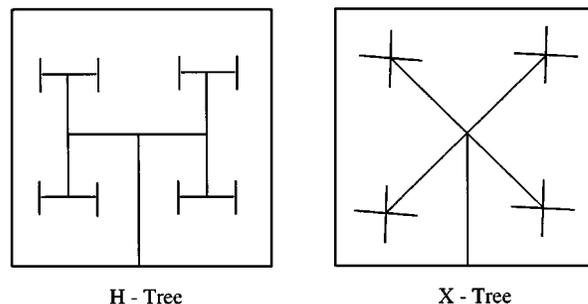


Fig. 13. Symmetric H-tree and X-tree clock distribution networks.

buffers). The number of buffer stages between the clock source and each clocked register depends upon the total capacitive loading, in the form of registers and interconnect, and the permissible clock skew [39]. It is worth noting that the buffers are a primary source of the total clock skew within a well-balanced clock distribution network since the active device characteristics vary much more greatly than the passive device characteristics. The maximum number of buffers driven by a single buffer is determined by the current drive of the source buffer and the capacitive load (assuming an MOS technology) of the destination buffers. The final buffer along each clock path provides the control signal of the driven register.

Historically, the primary design goal in clock distribution networks has been to ensure that a clock signal arrives at every register within the entire synchronous system at precisely the same time. This concept of zero clock skew design has been extended, as is explained in Section IV, to provide either a positive or a negative clock skew at a magnitude depending upon the temporal characteristics of each local data path in order to improve system performance and enhance system reliability.

B. Symmetric H-Tree Clock Distribution Networks

Another approach for distributing clock signals, a subset of the distributed buffer approach depicted in Fig. 10, utilizes a hierarchy of planar symmetric H-tree or X-tree structures (see Fig. 13) [30], [44], [185] to ensure zero clock skew

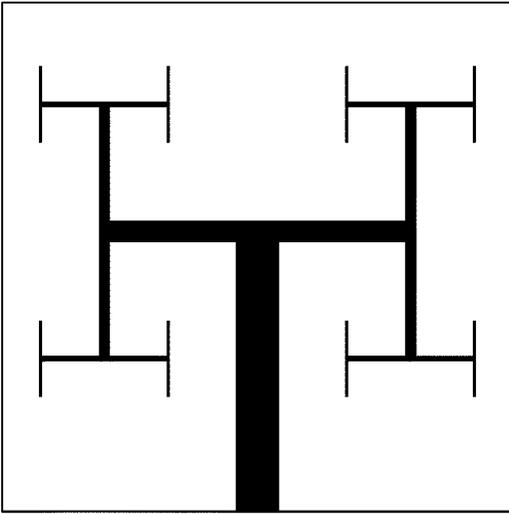


Fig. 14. Tapered H-tree clock distribution network.

by maintaining the distributed interconnect and buffers to be identical from the clock signal source to the clocked register of each clock path. In this approach, the primary clock driver is connected to the center of the main “H” structure. The clock signal is transmitted to the four corners of the main “H.” These four close to identical clock signals provide the inputs to the next level of the H-tree hierarchy, represented by the four smaller “H” structures. The distribution process continues through several levels of progressively smaller “H” structures. The final destination points of the H-tree are used to drive the local registers or are amplified by local buffers which drive the local registers. Thus, each clock path from the clock source to a clocked register has practically the same delay. The primary delay difference between the clock signal paths is due to variations in process parameters that affect the interconnect impedance and, in particular, any active distributed buffer amplifiers. The amount of clock skew within an H-tree structured clock distribution network is strongly dependent upon the physical size, the control of the semiconductor process, and the degree to which active buffers and clocked latches are distributed within the H-tree structure.

The conductor widths in H-tree structures are designed to progressively decrease as the signal propagates to lower levels of the hierarchy. This strategy minimizes reflections of the high-speed clock signals at the branching points. Specifically, the impedance of the conductor leaving each branch point Z_{K+1} must be twice the impedance of the conductor providing the signal to the branch point Z_K for an H-tree structure [30], [44], [158], [185] and four times the impedance for an X-tree structure. This tapered H-tree structure is illustrated in Fig. 14:

$$Z_k = \frac{Z_{k+1}}{2}, \quad \text{for an H-tree structure} \quad (8)$$

The planar H-tree structure places constraints on the physical layout of the clock distribution network as well as on the design methodology used in the development of the VLSI system. For example, in an H-tree network, clock lines must

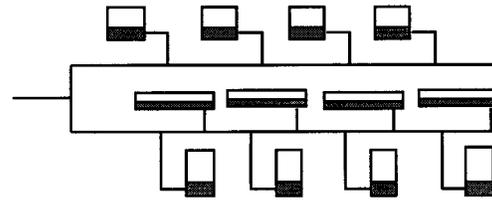


Fig. 15. Trunk-structured clock distribution network for structured custom VLSI circuit.

be routed in both the vertical and horizontal directions. For a standard two-level metal CMOS process, this manhattan structure creates added difficulty in routing the clock lines without using either resistive interconnect or multiple high-resistance vias between the two metal lines. This aspect is a primary reason for the development of three or more layers of metal in logic-based CMOS processes. Furthermore, the interconnect capacitance (and therefore the power dissipation) is much greater for the H-tree as compared with the standard clock tree since the total wire length tends to be much greater [171]. This increased capacitance of the H-tree structure exemplifies an important tradeoff between clock delay and clock skew in the design of high-speed clock distribution networks. Symmetric structures are used to minimize clock skew; however, an increase in clock signal delay is incurred. Therefore, the increased clock delay must be considered when choosing between buffered tree and H-tree clock distribution networks. Also, since clock skew only affects sequentially-adjacent registers, the obvious advantages to using highly symmetric structures to distribute clock signals are significantly degraded. There may, however, be certain sequentially-adjacent registers distributed across the integrated circuit. For this situation, a symmetric H-tree structure may be appropriate, particularly to distribute the global portion of the clock network.

Another consideration in choosing a clock distribution topology is that the H-tree and X-tree clock distribution networks are difficult to implement in those VLSI-based systems which are irregular in nature, such as pictured in Figs. 9 and 15. In these types of systems, buffered tree topologies integrated with structured custom design methodologies [188] should be used in the design of the clock distribution networks in order to maximize system clock frequency, minimize clock delay, and control any deleterious effects of local (particularly, negative) clock skew.

C. Compensation Techniques for Controlling Clock Skew

One structured custom approach, oriented for hierarchical VLSI-based circuits, utilizes compensation techniques to minimize the variation of interconnect impedances and capacitive loads between clock signal paths [28], [31], [34], [36]. A general schematic of a clock distribution network is shown in Fig. 16, in which the nodes $i, j,$ and k represent different clock signal destinations (i.e., clocked registers). Different clock paths could conceivably have different levels of buffering, where each buffer drives a localized distributed RC impedance. The location of these buffers is often chosen so that the active buffer output impedance is

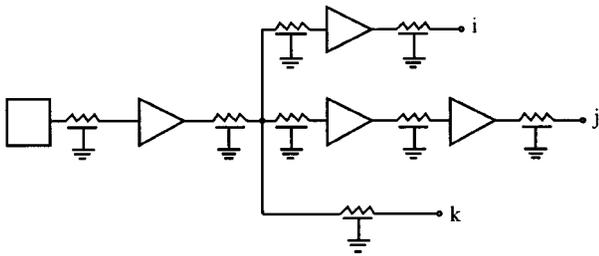


Fig. 16. Clock distribution network.

comparable to or greater than the interconnect resistance seen at the buffer output. This characteristic ensures that the locally distributed RC interconnect section can be accurately modeled as being mostly capacitive. The use of distributed buffers in this manner is described as buffer repeaters [185], [189], [199], [211]. However, in general, the interconnect impedance should be modeled as a distributed resistive-capacitive section of interconnect.

The difficulty with applying symmetric clock distribution strategies is that they do not easily support the ability to partition large VLSI systems into hierarchically structured functional blocks. Preferably, each large functional block would contain its own locally optimized clock distribution network to satisfy the local timing and loading of that particular functional block. For a globally synchronous system, however, local optimization within a functional element does not necessarily lead to global optimization of the overall on-chip clock distribution system.

If the interconnect resistance of the global clock distribution network is relatively small, a chip-level centralized clock buffer circuit can be used to satisfy the synchronization requirements of a VLSI circuit. However, in most large VLSI circuits, the physical distances are such that line resistances coupled with any via/contact resistances and the significant line and coupling capacitances will create large interconnect impedances. Therefore, even with a centrally located clock generation and distribution circuit, additional techniques are required to compensate for variations in interconnect and register loading.

In order to control the delay of each clock signal path and to minimize the skew between these paths, passive RC delay elements [31] or geometrically sized transistor widths [28] are used to compensate for the variation of the delay of each clock signal path caused by different on-chip locations (i.e., different path dependent interconnect impedances) and capacitive loading of the clock destinations (i.e., the number and load of the clocked registers per clock signal path). Clock buffers are placed along the clock path such that the highly resistive interconnect lines (typically long lines) drive loads with low capacitance while the low-resistance interconnect lines (typically short lines) drive loads with high capacitance. Thus, either a centralized module of clock buffer drivers can be used or those clock buffers driving large capacitive loads can be placed close to the registers, thereby decreasing the interconnect resistance. This design strategy of using compensation techniques to control the local clock skew is graphically depicted in Fig. 16. The variation of clock delay be-

tween each of the functional elements is compensated for by parameterizing the current drive of each of the functional block clock buffers resident in the centrally located clock buffering circuit (see Fig. 17). If feedback circuitry is being used to further control the delays and skews within the clock distribution network, as in on-chip phase lock loops (PLLs), taps are placed close to the register which are fed back to maintain lock.

In order to ensure that the clock distribution network is successfully designed, the following practices should be followed: 1) the number of stages of clock buffering within each of the functional blocks should be the same to maintain equal polarity; 2) the maximum clock signal rise and fall times within each functional block should be specified and controlled; and 3) the maximum clock skew of each functional block should be specified and controlled using the same hierarchical clock distribution strategy as is used at the global VLSI system level [28].

1) Advantages and Disadvantages of Compensation Technique: The primary advantage of using a compensation technique is controlling (and reducing) on-chip clock skew. Also, the clock delay from the clock source to the clocked registers is reduced. This behavior is due to improved partitioning of the RC loads. Since the inverters located within each functional block drive large capacitive loads, the interconnect impedance, and in particular, the interconnect resistance driven by any specific clock buffer is small in comparison to the buffer output impedance. The fairly long distances of the intrablock clock signal paths are fairly resistive. These paths, however, are isolated from the highly capacitive loads. Thus, the RC time constants are reduced, reducing the overall clock delay. Another important advantage of this design technique is the ease of partitioning the clock distribution problem among a team of VLSI circuit designers. The overall VLSI system design can be partitioned hierarchically into a manageable domain of information while still providing a strategy for implementing optimal clock distribution networks. The usefulness of this compensation technique is dependent upon the ability to characterize the device and interconnect impedances within the VLSI circuit. With an accurate estimate of these impedances, the parameterized buffers can be designed so as to satisfy a specific clock skew schedule.

It is important to note an important disadvantage of this compensation technique. Unlike interconnect impedances, transistor conductances tend to be highly sensitive to variations in supply voltage and to process and environmental conditions (e.g., temperature, radiation). A clock signal path whose delay is dominated by interconnect impedances may vary differently than a clock signal path whose delay is dominated by device impedances [28].

Several specific examples of clock distribution networks are discussed in the literature (e.g., [1], [11], [28], [35], [36], [39], [163]). Certain examples of clock distribution networks applied to high-speed circuits are described in Section X. In each of these clock distribution networks, significant effort has been placed on accurately estimating the magnitude of the resistive and capacitive interconnect impedances to deter-

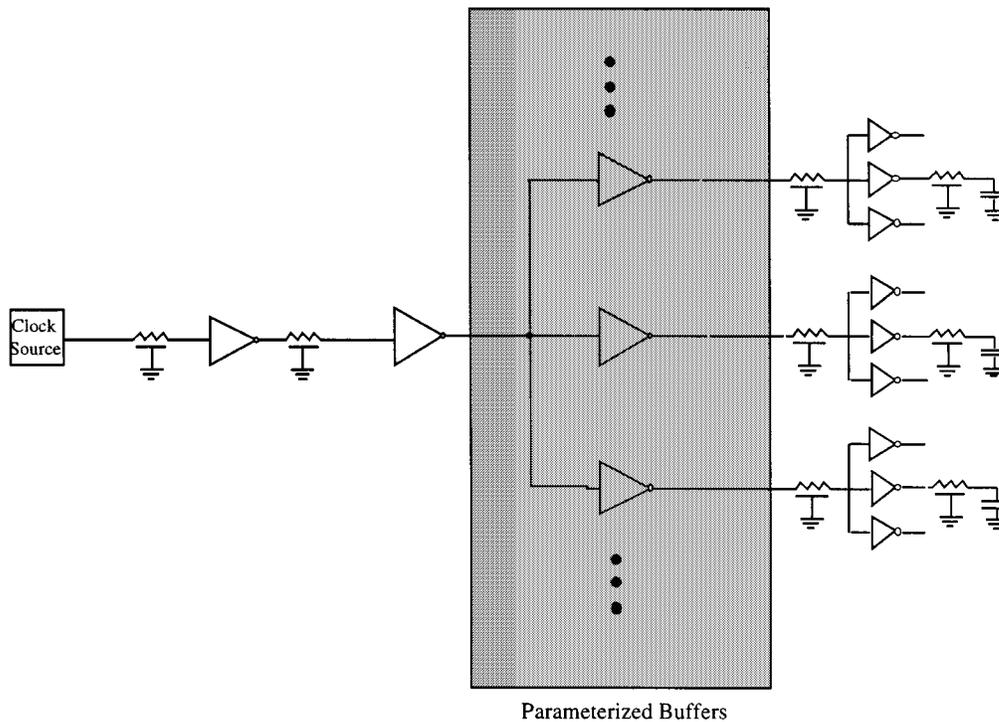


Fig. 17. Parameterized buffers integrated into a clock distribution network to control local clock skew [28].

mine the effect of these RC loads on the shape of the clock signal waveform. This information is typically back annotated into a SPICE-like circuit simulator to adjust the clock delays for minimum clock skew [190]. Minimal work exists, however, in developing circuit procedures and algorithms for automating the circuit design of clock distribution networks in structured custom VLSI circuits [119]–[122]. One primary requirement for developing these algorithms is a physical model for estimating the delay of the clock signal path. An important observation is that the accuracy required to calculate delay differences (as in clock skew) is much greater than the accuracy required when calculating absolute delay values (as in the delay of a clock path).

D. Design of Low-Power Clock Distribution Networks [52]–[54]

In a modern VLSI system, the clock distribution network may drive thousands of registers, creating a large capacitive load that must be efficiently sourced. Furthermore, each transition of the clock signal changes the state of each capacitive node within the clock distribution network, in contrast with the switching activity in combinational logic blocks, where the change of logic state is dependent on the logic function. The combination of large capacitive loads and a continuous demand for higher clock frequencies has led to an increasingly larger proportion of the total power of a system dissipated within the clock distribution network, in some applications much greater than 25% of the total power [53], [167].

The primary component of power dissipation in most CMOS-based digital circuits is dynamic power. It is possible

to reduce CV^2f dynamic power by lowering the clock frequency, the power supply, and/or the capacitive load of the clock distribution network. Lowering the clock frequency, however, conflicts with the primary goal of developing high-speed VLSI systems. Therefore, for a given circuit implementation low dynamic power dissipation is best achieved by employing certain design techniques that either minimize the power supply and/or the capacitive load.

De Man [52] introduced a technique for designing clock buffers and pipeline registers such that the clock distribution network operates at half the power supply swing, reducing the power dissipated in the clock tree by 60% without compromising the clock frequency of the circuit. Kojima *et al.* [53] describe a similar strategy in which the clock signals also operate only over half of the power supply rail, reducing the power dissipated in the clock tree by ideally 75%. The degradation in system speed is very small since, unlike the clock signals, the data signals operate over the full-power supply rail. Thus, the voltage is reduced only in the clocking circuitry, resulting in significantly reduced power with a minimal degradation in system speed. Experimentally derived savings of 67% were demonstrated on a test circuit (a 16-stage shift register) fabricated in a 0.5- μm CMOS technology with only a 0.5-ns degradation in speed using this half-swing clocking scheme.

Other approaches exist for reducing the power dissipated within a clock distribution network. These approaches reduce power by decreasing the total effective capacitance required to implement a clock tree. Reductions of 10%–25% in power dissipated within the clock tree are reported with no degradation in clock frequency [54]. As described in Section X, the development of design strategies for minimizing the power

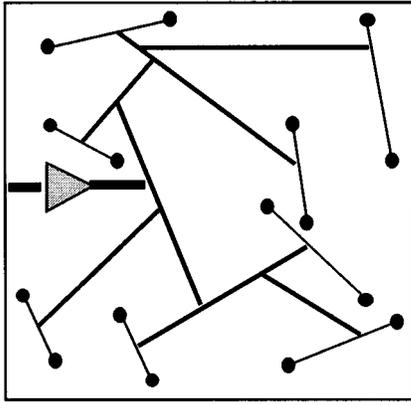


Fig. 18. Geometric matching for zero clock skew.

dissipated both internal to the clock tree as well as the overall system being synchronized is a research topic of great relevance to a variety of important applications.

VI. AUTOMATED SYNTHESIS AND LAYOUT OF CLOCK DISTRIBUTION NETWORKS

Different approaches have been taken in the automated synthesis and layout of clock distribution networks, ranging from procedural behavioral synthesis of pipelined registers [124]–[129] to the automated layout of clock distribution nets for application to gate array and standard cell-based integrated circuits [35], [61]–[91], [96], [185]. The area of automated layout is described below in Section VI-A. The integration of the effects of clock distribution into behavioral synthesis methodologies is described in Section VI-B.

A. Automated Layout of Clock Distribution Networks

A second research path in the area of automated layout has been the development of algorithms which carefully control the variations in delay between clock signal net length so as to minimize clock skew (e.g., [65], [68]–[73], [78], [84]). The strategy used is to construct binary tree-like structures with the clock pins at the leaf nodes. Minimal skew clock distribution networks are created using a recursive bottom-up approach. At each of the clock pins of the registers, which represent the leaves of the clock distribution tree, a clock net is defined. The point where two zero-skew clock nets connect is chosen such that the effective delay from that point to each clocked register is identical (within the accuracy of the delay model). This process continues up the clock distribution tree; the point of connection of each new branch being chosen to satisfy the zero-skew design goal. The layout process terminates when the root (or source) of the clock tree is reached. The schematic diagram of this geometric matching process is illustrated in Fig. 18. Thus, the automated layout algorithm attempts to balance the delay of each clock branch in a recursive manner, moving from the leaves to the root of the tree. The appropriate branching points of the zero skew subtree are chosen so as to maintain equal delay. If the zero skew tapping location falls directly on an unroutable location, such as an existing macrocell, a nonzero clock skew would be realized [96]–[98].

Some early clock routing algorithms [65], [71], [72] define the delay as a measure of the total wire length along a path. These algorithms attempt to equalize the lengths of each net from the root of the clock tree to each of the leaf nodes. Thus, the clock skew is minimized during the routing phase of the layout process. No attempt is made to postprocess the layout database to further improve the skew characteristics.

In [96]–[98], the automated layout of the clock tree is composed of a two-phase process. The clock net is initially routed in a binary-tree manner with the clock pins as leaf nodes and the clock buffer as the root. This layout phase is followed by a post-layout phase in which the clock nets are widened according to the zero-skew specifications, thereby giving the clock layout system additional flexibility in routing around possible blockages. The choice of which clock net to widen is determined by analyzing the sensitivity of the clock net impedance. These sensitivities provide a means of choosing those nets which will decrease the average delay of the RC trees as near as possible to a specified target delay. Those nets whose delay must be increased and are less sensitive to increasing capacitance are widened. However, if all the clock nets are relatively thin, statistical variations in the widths of those wires closest to the clock driver may affect the actual clock skew the most [96]. This occurs since the section of interconnect closest to the driver sees the greatest portion of the distributed RC impedance of the interconnect line. Therefore, the greatest change in delay will occur as the width varies.

These automated clock layout algorithms tend to use simplified delay models, such as linear delay, where the delay is linearly related to the path length, or the Elmore delay [191]–[193], where the delay along a path is the summation of the products of the branch resistance and the downstream capacitance of every branch on the path from the root to the downstream node i (the clock pin of the register), and is

$$T_{Di} = \sum_k R_{ki} C_k \quad (10)$$

where C_k is the capacitance at node k and R_{ki} is the resistance of the portion of the (unique) path between the input and the output node i , that is common with the (unique) path between the input and node k . The Elmore delay is a first-order step response approximation of the delay through a distributed resistive-capacitive interconnect section. For slow input waveforms, the Elmore delay approximation can become highly inaccurate, since the shape and magnitude of the clock waveforms are not considered.

The fundamental difficulty with these delay models, however, is the inability to accurately consider the effects of active devices, such as distributed buffers, when estimating delay as well as more subtle considerations, such as bias dependent loading and varying waveform shapes. The primary focus of the existing research into the automatic layout of clock distribution networks has been placed on minimizing total wirelength, metal-to-metal contacts and crossovers, as well as attaining zero system-wide clock skew (assuming nominal conditions).

B. Integration of Clock Distribution Issues into the Behavioral Synthesis Process

Localized clock distribution has been considered only minimally in automated layout or physical synthesis. However, early work in applying local clock skew to behavioral synthesis is described in [119]–[129]. These papers represent early efforts to develop strategies that consider the effects of clock distribution networks in the behavioral synthesis process, rather than after the circuit has been partitioned into logic (or register transfer level) blocks. This capability will improve high-level exploratory design techniques as well as optimize the performance of circuits implemented with high-level synthesis tools.

As described in [100]–[108], [119]–[121], the automated synthesis of clock distribution networks can be broken up into four phases: 1) optimal clock scheduling; 2) topological design; 3) circuit design; and 4) physical layout. Optimal scheduling represents a primary research activity in clock distribution networks and is discussed further in Section VIII. The area of topological design, in which the structure of the clock distribution network is derived from the scheduling of the local clock skew schedule, is discussed in [101], [119]. Algorithms are developed for converting the clock skew information into path specific clock delays. With this information and some additional information describing the hierarchy of the circuit function, a clock distribution tree is developed with delay values assigned to each branch of the tree. With the topological structure and delay information determined, circuit delay elements are synthesized which satisfy the individual branch delays. Circuit techniques to implement the clock tree delay elements are discussed further in [101], [108], [120], [121]. Finally, a variety of techniques exist to lay out the clock distribution trees. Some of these layout techniques are discussed in Section VI-B. This work represents early research in the development of a systematic methodology for synthesizing tree structured clock distribution networks which contain distributed cascaded buffers and, furthermore, exploit nonzero localized clock skew.

In [125]–[127], [130], a delay model characterizing the timing components of a local data path, similar to (2) and (3), are used to incorporate the effects of local clock distribution delays into the retiming process. This is accomplished by assuming that physical regions of similar clock delay exist throughout an integrated circuit. Retiming is an automated synthesis process for relocating pipeline registers such that the critical worst case path delay is minimized, creating a synchronous system with the highest possible clock frequency while maintaining the function and latency of the original system. Previous work in the area of retiming ignored clock skew in the calculation of the minimum clock period. In the algorithm presented in [126], [127], [130], clock delays are attached to individual paths between logic gates. As a register is placed on a new path, it assumes the clock delay of that path. Since each sequentially-adjacent pair of registers defines a local data path, as registers are moved from one region to another during the retiming process, the displaced registers assume the clock delay of the new physical

region. Thus, the local clock skews of each data path are determined at each iteration of the retiming process, permitting both increased accuracy in estimating the maximum clock frequency and detection and elimination of any catastrophic race conditions. If a choice of register locations does not satisfy a particular clock period or a race condition is created, that specific register instantiation is disallowed. This algorithm, therefore, integrates the effects of clock skew (and variable register and interconnect delays) directly into the synchronous retiming process.

It is interesting to note that adding clock delay to a clock path (applying localized clock skew) has an effect similar to retiming, where the register crosses logic boundaries. Thus, time can be shifted by moving the registers or changing the local clock delays, where retiming is discrete in time and localized clock skew is continuous in time. In general, the two methods complement each other [145]. As Fishburn mentions in [145], since both methods are linear, “it is likely that efficient procedures could be given for optimizing systems by jointly considering both sets of variables.” These two methods of synchronous timing optimization have recently been integrated into a cohesive design methodology for sequential optimization [210]. It is shown by Liu *et al.* that the simultaneous application of retiming and clock skew scheduling can achieve higher clock frequencies than optimized circuits generated by applying either of the two techniques separately.

VII. ANALYSIS AND MODELING OF THE TIMING CHARACTERISTICS OF CLOCK DISTRIBUTION NETWORKS

This research area is composed of a number of disparate topics, all of which have in common the attributes of modeling the general performance characteristics of clock distribution networks. An important and active area of research in clock distribution networks is the design of circuits which are less sensitive to variations in process parameters. This topic is discussed in Section VII-A. Techniques for calculating clock skew are summarized in Section IV and Section VII-B.

A clock tree topology that implements a given clock schedule must enforce a clock skew $T_{\text{Skew}}(i, f)$ for each local data path $R_i \Rightarrow R_f$ of the circuit in order to ensure that both (5) and (6) are satisfied. Previous research [101], [106] has indicated that tight control over the clock skews rather than the clock delays is necessary for the circuit to operate reliably. Equations (5) and (6) are used in [106] to determine a permissible range of the allowed clock skew for each local data path. The concept of a permissible range for the clock skew $T_{\text{Skew}}(i, f)$ of a local data path $R_i \Rightarrow R_f$ is illustrated in Fig. 19. When $T_{\text{Skew}}(i, f) \in [d(i, f), T_{\text{CP}} - D(i, f)]$ —as shown in Fig. 19—(5) and (6) are satisfied. $T_{\text{Skew}}(i, f)$ is not permitted to be in either the interval $(-\infty, d(i, f))$ because a race condition will be created or the interval $(T_{\text{CP}} - D(i, f), +\infty)$ because the minimum clock period will be limited. It is this range of permissible clock skew rather than a specific target value of clock skew that provides one important opportunity to improve the tolerance of a

Fig. 19. Permissible range of the clock skew of a local data path, $R_i \Rightarrow R_f$. A clock hazard exists if $T_{Skew}(i, f) \notin [d(i, f), T_{CP} - D(i, f)]$.

synchronous digital system to process parameter variations. Furthermore, this decreased sensitivity to process parameter variations by exploiting the localized permissible range of each local data path is completely compatible with the design techniques described in Section VII-A.

A. Design of Process Insensitive Clock Distribution Networks

A primary disadvantage of clock distribution networks is that the delay of each of the elements of a clock path, the distributed buffers and the interconnect impedances, are highly sensitive to geometric, material, and environmental variations that exist in an implementing technology. Thus, as device and interconnect parameters vary from process lot to process lot, the specific performance characteristics of the clock distribution network may change. This phenomenon can have a disastrous effect on both the performance and the reliability of a synchronous system and limit both the precision and the design methodology of the clock distribution network. It is essential for a robust clock distribution network to exhibit a certain degree of tolerance to variations in process parameters and environmental conditions. In an effort to overcome this problem, various approaches have been developed which mitigate the effect of process tolerances on the design of clock distribution networks while maintaining an effective methodology for designing these networks.

1) *Threshold Tracking to Control Clock Skew:* An interesting circuit design technique for making clock distribution networks less process-sensitive is described by Shoji [131]. The technique uses the MOS circuit characteristic that *N*-channel and *P*-channel parameters tend not to track each other as a process varies. Interestingly, the response times of these devices tend to move in opposite directions, since the cause of a positive threshold voltage shift in one type of MOS transistor, e.g., an *N*-channel device, will typically cause the *P*-channel threshold voltage to shift in the opposite direction. Shoji quantitatively describes how the delays of the *P*-channel and *N*-channel transistors within the distributed buffers of a clock distribution network should be individually matched to ensure that as the process varies, the path delays between different clock paths will continue to track each other.

The primary objective of this process insensitive circuit design technique is to match the two clock edges (of either a *P*-channel or an *N*-channel transistor) as the process parameters vary. Shoji presents two rules to minimize the effects of process variations on clock skew. The rules are:

- 1) match the sum of the pull-up delays of the *P*-channel MOSFET with the pull-up delays of any related clock signal paths;

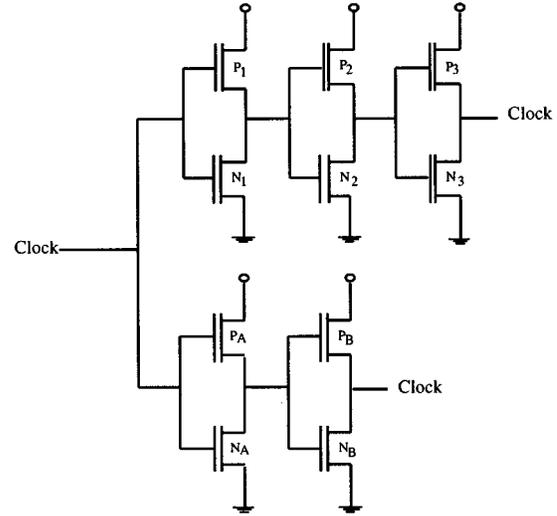


Fig. 20. Elimination of process-induced clock skew by scaling matched transistor types.

- 2) match the sum of the pull-down delays of the *N*-channel MOSFET with the pull-down delays of any related clock signal paths.

Although process variations may change the total clock delay along a given path, the difference in delay between paths will track each other, keeping the skew small.

A circuit utilizing this technique is shown in Fig. 20. Delay times T_1, T_3 , and T_A are directly related to the conductances of the *N*-channel devices, N_1, N_3 , and N_A , respectively. Delay times T_2 and T_B are directly related to the conductances of the *P*-channel devices, P_2 and P_B , respectively. The conductance of each of these devices is proportional to the size of the MOSFET geometric width. In conventional CMOS circuit design, the transistor widths are adjusted to satisfy

$$T_A + T_B = T_1 + T_2 + T_3 \quad (11)$$

thereby ensuring that the skew at the output of each clock branch is close to zero. If, for example, either the *N*-channel or the *P*-channel threshold voltages varies significantly from its assumed value, (11) would no longer be satisfied, and a large skew would develop between the outputs, clock and clock', shown in Fig. 20. Instead of designing the circuit in Fig. 20 to satisfy (11), the circuit is designed to satisfy (12) and (13). Thus, the primary objective of (11) (zero clock skew) is maintained while the added more stringent design constraint makes the overall clock distribution circuit more tolerant to process variations. This increased tolerance occurs since by satisfying both (12) and (13), the *N*-channel and *P*-channel transistors of both branches individually track

each other, making the system more tolerant to variations in the N -channel and P -channel transistor characteristics.

$$T_A = T_1 + T_3, \quad (12)$$

$$T_B = T_2, \quad (13)$$

This design technique can be used to make circuits less sensitive to process variations and environmental conditions even if the circuits are not inverters but are more general forms of logic gates. The technique also ensures similar behavior when interconnect impedances are included within the circuit. Simulated worst case clock skews of circuits using this technique exhibit skews that are 10% less than that of conventionally designed circuits [131].

2) *Interconnect Widening to Minimize Clock Skew Sensitivity*: As described in Section VI-A, one approach for the automated layout of clock nets is to lengthen specific clock nets to equalize the length of every clock line, thereby keeping the clock skew close to zero. A disadvantage of this approach is that these minimum width lines are very susceptible to variations in the etch rate of the metal lines, as well as to mask misalignment or local spot defects. Therefore, the effective interconnect impedance (and the delay) of these long thin clock nets can vary greatly from wafer to wafer as these linewidths vary. In order to design these clock nets to be less sensitive to process variations, Pullela *et al.* [96]–[98] have developed an automated layout algorithm that widens the clock nets rather than lengthens the nets while equalizing the line delays. These nets are therefore less sensitive to both under- and overetching during the metal patterning process. By widening the clock lines, the interconnect resistance is decreased; however, the interconnect capacitance increases. It is interesting to note that increasing the linewidth of those branches closer to the root of the RC tree has a greater effect on the clock path delay than increasing the widths closer to the leaf nodes (the clocked registers). Thus, decreasing the resistance at the source by increasing the linewidth affects the total path delay more significantly than decreasing the resistance at the leaf node, since more capacitance is seen by the large source resistance than if the resistance is greater near the leaf. Therefore, the clock skew is particularly sensitive to changes in linewidth close to the clock source. One approach to making the clock lines more tolerant of process variations is to make the width of the clock interconnect lines widest near the clock source and thinner as the leaf nodes are approached. This strategy would provide a reasonable tradeoff between controlling the effect of process variations (particularly, metal etch rates) on the clock skew and minimizing line dimensions for process yield and circuit layout efficiency. The relative sensitivities of each net to changes in the capacitive and resistive interconnect impedances are analyzed and integrated into the Elmore delay model [191]–[193]. One of the primary advantages of this approach is that the process of automatically laying out the clock nets is separated from the clock skew reduction process. Thus, local layout techniques, such as widening the clock nets, can be used to make the overall circuit less sensitive to variations in process parameters.

B. Deterministic Models for Estimating Clock Skew

A clock signal path within a clock distribution network has a single input and, although paths branch off from the trunk of the tree, a single path (or branch) exists from the clock source to the clock input of a register. This branch is typically composed of distributed buffers and interconnect sections, as shown in Fig. 16. In order to simplify the calculation of the path delay and to provide simple closed form delay models of the path, it is typically assumed that the buffer on-resistance is much greater than the interconnect resistance that the buffer is driving. This permits the distributed RC interconnect section to be modeled as a simple lumped capacitor. Percent errors reflecting this assumption are provided in [194], where the errors are dependent upon the ratio of the load resistance to the output buffer on-resistance. However, if the line resistance is not significantly smaller than the buffer output resistance, repeaters [185], [189] are often inserted at a point within the clock line to ensure that the output resistance of the repeater buffer is much larger than the local line resistance of the interconnect section between the repeaters [16].

In order to calculate the clock path delay and skew, a simple model of a CMOS inverter driving another inverter with line resistance and capacitance between the two inverters is often used. A well-known empirical estimate of the rise (or fall) time of a single CMOS inverter driving an RC interconnect section with a capacitive load (representing the following CMOS inverter) is [184], [185], [194]

$$T_{R/F} = 1.02R_{\text{Int}}C_{\text{Int}} + 2.21(R_{\text{Tr}}C_{\text{Int}} + R_{\text{Tr}}C_{\text{Tr}} + R_{\text{Int}}C_{\text{Tr}}) \quad (14)$$

where R_{Int} and C_{Int} is the resistance and capacitance of the interconnect section, respectively, and R_{Tr} and C_{Tr} is the output on-resistance of the driving buffer and the input load capacitance ($=C_{\text{OX}}WL$) of the following buffer, respectively. Note that C_{OX} is the oxide capacitance per unit area, and W and L are the width and length, respectively, of the following buffer. An approximate estimate of the output resistance of the buffer may be obtained from [184], [185], [194]

$$R_O = \frac{L/W}{\mu C_{\text{OX}}(V_{\text{DD}} - V_T)} \quad (15)$$

where

- μ channel mobility;
- V_{DD} power supply voltage;
- V_T device threshold voltage.

Equation (15) is derived from the large-signal current–voltage (I – V) equation of a MOSFET operating in the saturation region close to the linear region and is accurate for small channel geometries, since velocity saturation decreases the quadratic behavior of the MOS device operating in the saturation region. The physical delay model represented by (14) and (15) is a fairly simple approximation of the delay of a CMOS inverter driving a distributed

RC impedance. More complex and accurate delay models exist. This area of inquiry represents an important topic of intensive research unto itself and is discussed in great detail throughout the literature.

An important research area in VLSI circuits is timing analysis, where simplified RC models are used to estimate the delay through a CMOS circuit. Clock characteristics are provided to a timing analyzer to define application specific temporal constraints, such as the minimum clock period or hold time, on the functional timing of a specific synchronous system [133]. In [141], [142], Tsay and Lin continue this approach by describing an innovative timing analyzer which exploits negative clock skew, i.e., time is “stolen” from adjacent data paths to increase system performance. Therefore, the descriptive term “cycle stealing” is used to describe this process. In [134], [135], Dagenais and Rumin present a timing analysis system which determines important clocking parameters from a circuit specification of the system, such as the minimum clock period and hold time. This approach is useful for top-down design when performing exploratory estimation of system performance.

VIII. SPECIFICATION OF THE OPTIMAL TIMING CHARACTERISTICS OF CLOCK DISTRIBUTION NETWORKS

An important element in the design of clock distribution networks is choosing the minimum local clock skews that increase circuit performance by reducing the maximum clock period while ensuring that no race conditions exist. This design process is called *optimal clock skew scheduling* and has been extensively studied in [145]–[156], [208], [209] and is described in this section. Starting with the timing characteristics of the circuit, such as the minimum and maximum delay of each combinational logic block and register, it is possible to obtain the localized clock skews and the minimum clock period. This information is determined by formulating the optimal clock scheduling problem as a linear programming problem and solving with linear programming techniques [145], [152].

The concept of scheduling the system-wide clock skews for improved performance while minimizing the likelihood of race conditions was first presented by Fishburn in 1990 [145], although the application of localized clock skew to increase the clock frequency and to eliminate race conditions was known previously [179], [181]. Fishburn presents a methodology in which a set of linear inequalities are solved using standard linear programming techniques in order to determine each clock signal path delay from the clock source to every clocked register. Two clocking hazards, identical to the constraint relationships described in Section IV, are eliminated by solving the set of linear inequalities derived from (5) and (6) for each local data path. The deleterious effect of positive clock skew (the maximum data path/clock skew constraint relationship) is described as *zero-clocking*, while the deleterious effect of negative clock skew (the minimum data path/clock skew constraint relationship) is described as *double-clocking*. This approach is demonstrated on a 4-bit ripple-carry adder with accumulation and input register in a

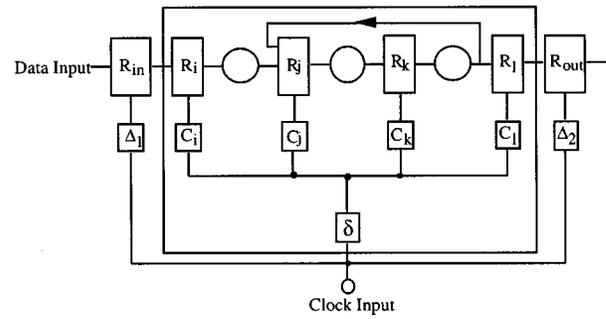


Fig. 21. Multistage block diagram of a synchronous digital system.

1.25- μm CMOS technology. The minimum clock period is decreased from 9.5 ns with zero clock skew to 7.5 ns with localized clock skew [145]. Szymanski improves this methodology for determining an optimal clock skew schedule by selectively generating the short path constraints, permitting the inequalities describing the timing characteristics of each local data path to be solved more efficiently [151].

In order to describe the process for determining an optimal clock skew schedule, a system timing model is presented. A block diagram of a multistage synchronous digital system is depicted in Fig. 21. Between each stage of registers there is, typically, a block of combinational logic with possible feedback paths between the registers. Each register is either a multi- or single-bit register, and all the inputs are assumed to change at the same time point of the transition of the clock signal. Only one single-phase clock signal source in the circuit is assumed. The registers are composed of edge-triggered flip-flops and considered to assume a single value for each clock cycle. The combinational logic block is described in terms of the maximum and minimum delay values, $T_{\text{Logic}(\text{max})}$ and $T_{\text{Logic}(\text{min})}$. These logic delay values are obtained by considering the delay of all possible input to output paths within each combinational logic block. For simplicity and without loss of generality, the block diagram in Fig. 21 only considers a single-input, single-output circuit. In this figure, Δ_1 and Δ_2 are clock delays outside the VLSI-based system and δ is that portion of the on-chip clock delay that is shared by each of the clock paths (the initial trunk of the clock tree). The registers R_{in} and R_{out} make up one set of registers placed at the input and output, respectively, of the VLSI-based system. The circuit is composed of the registers, R_i to R_l , where the logic blocks are between the registers, and the signal clock delays are C_i to C_l .

1) *Off-Chip Clock Skew*: This circuit model, pictured in Fig. 21, also considers the relationship between off-chip and on-chip clock skew. The registers R_{in} and R_{out} symbolize off-chip registers and are controlled by the off-chip clock source, which also provides the on-chip clock signals, since the circuit is assumed to be a fully synchronous system. This relationship is represented by

$$T_{\text{Skew in,out}} = T_{\text{Skew in},i} + T_{\text{Skew }i,j} + \dots + T_{\text{Skew }l,\text{out}} = 0. \quad (18)$$

Therefore, to satisfy (18) in Fig. 21, $\Delta_1 = \Delta_2 = \Delta$.

Although it is possible to have off-chip nonzero clock skew, it is desirable to ensure that the clock skew between VLSI input–output (I/O) approaches zero in order to avoid complicating the design of a circuit board or the specification of the interface of the circuit with other components also controlled by the same clock source. For example, a circuit with intentional nonzero clock skew requires that the clock distribution network of any other synchronous circuit sharing the same global clock be offset by the same amount of temporal skew, otherwise race conditions, such as described in Section IV-B, may occur at the board level. This strategy of minimizing clock skew as the level of design complexity shifts should be applied at each higher level of design, such as from the board level to the multiboard level.

A fully synchronous circuit must generate data at a rate defined by the clock period of the clock source, otherwise race conditions may occur at the interface of the circuit with other parts of the system. These race conditions occur when there is a negative clock skew, intentionally or unintentionally introduced into the circuit. Observe that every circuit has unintentional clock skew caused by several factors, one common cause being variations in process parameters. This type of clock skew must be considered during the design of the circuit, and should be less than the intentional on-chip negative clock skew introduced to increase the performance of the circuit. Furthermore, the magnitude of the intentional clock skew at each register I/O may vary substantially, according to the optimization applied to the data path. Therefore, clock skew at the system level of a VLSI circuit should be constrained to approach zero, in order to allow the circuit to communicate correctly with other board-level circuits. For example, a symmetric zero clock skew distribution system should be used for the external registers (symmetric networks are discussed in Section V-B). Observe that restricting the off-chip clock skew to zero does not preclude the circuit from being optimized with localized clock skew. The primary deleterious effect is that the performance improvement is less than that obtained without this constraint.

Observe that (18) is valid only if the interface circuitry is controlled by the same clock source. The restriction does not apply to asynchronous circuits or synchronous circuits that communicate asynchronously, i.e., globally asynchronous, locally synchronous systems.

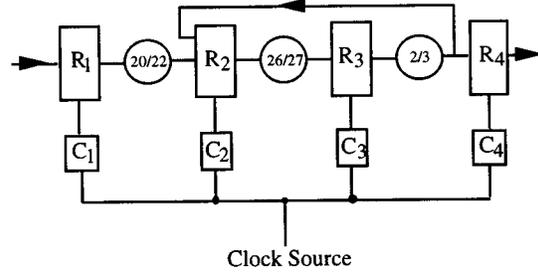


Fig. 22. Circuit example with feedback.

2) *Global and Local Timing Constraints:* As described in Section IV, in order to avoid either types of clock hazard (either a maximum or minimum data path/clock skew constraint relationship), a set of inequalities must be satisfied for each local data path in terms of the system clock period T_{CP} and the individual delay components within the data path. To avoid limiting the maximum clock rate between two sequentially-adjacent registers, R_i and R_j , (5) must be satisfied. To avoid race conditions between two sequentially-adjacent registers, R_i and R_j , (6) must be satisfied.

The system-wide clock period is minimized by finding a set of clock skew values that satisfies (5) and (6) for each local data path and (18) for each global data path. These relationships are sufficient conditions to determine the optimal clock skew schedule such that the overall circuit performance is maximized while eliminating any race conditions.

The timing characteristics of each local data path are assumed to be known. The minimum clock period is obtained when the problem is formalized as a linear programming problem, such as the equation at the bottom of the page.

3) *An Example of Determining the Optimal Clock Schedule of a Pipelined System:* An example circuit used to exemplify the process for determining the minimum clock period of a multistage system with feedback paths is the circuit illustrated in Fig. 22. This example is similar to that used in [27], adapted to consider zero clock skew between off-chip registers. The numbers inside the logic blocks are the minimum and maximum delays of each block, respectively. Similar to the approach taken in [23], for simplicity all the register timing parameters are assumed to be zero.

Minimize

$$T_{CP}$$

subject to the local and global timing constraints:

$$T_{Skew} \leq T_{CP} - T_{PD(max)} = T_{CP} - (T_{C-Q} + T_{Logic(max)} + T_{Int} + T_{Set-up}), \quad \text{for } T_{Ci} > T_{Cf}$$

$$|T_{Skew}| \leq T_{PD(min)} = T_{C-Q} + T_{Logic(min)} + T_{Int} + T_{Hold}, \quad \text{for } T_{Cf} > T_{Ci}$$

$$T_{CP} \geq T_{PD} + T_{Skew_{i,j}}$$

$$T_{PD(max)} = T_{C-Q_i} + T_{Logic(max)} + T_{Int} + T_{Set-up_j}$$

$$T_{Skew_{in,i}} + T_{Skew_{i,j}} + T_{Skew_{j,k}} + \dots + T_{Skew_{n,out}} = 0$$

The linear program that produces the minimum clock period and the optimal clock skew schedule for the circuit shown in Fig. 22 is

$$\begin{aligned}
& \text{Minimize} \\
& T_{\text{CP}} \\
& \text{Subject to:} \\
& R_1 - R_2 : C_1 - C_2 = T_{\text{Skew12}} \geq -20 \text{ ns} \\
& T_{\text{Skew12}} - T_{\text{CP}} \leq -22 \text{ ns} \\
& R_2 - R_3 : C_2 - C_3 = T_{\text{Skew23}} \geq -26 \text{ ns} \\
& T_{\text{Skew23}} - T_{\text{CP}} \leq -27 \text{ ns} \\
& R_3 - R_4 : C_3 - C_4 = T_{\text{Skew34}} \geq -2 \text{ ns} \\
& T_{\text{Skew34}} - T_{\text{CP}} \leq -3 \text{ ns} \\
& R_3 - R_2 : C_3 - C_2 = T_{\text{Skew32}} \geq -2 \text{ ns} \\
& T_{\text{Skew32}} - T_{\text{CP}} \leq -3 \text{ ns} \\
& R_1 - R_4 : T_{\text{Skew12}} + T_{\text{Skew23}} + T_{\text{Skew34}} = 0
\end{aligned}$$

where the optimal clock schedule is

$$\begin{aligned}
T_{\text{Skew12}} &= -3 \text{ ns} \\
T_{\text{Skew23}} &= -12 \text{ ns} \\
T_{\text{Skew34}} &= 15 \text{ ns} \\
T_{\text{CP}} &= 19 \text{ ns.}
\end{aligned}$$

If zero clock skew between off-chip registers is not considered, the minimum clock period is $T_{\text{CP}} = 15 \text{ ns}$. Although the restriction of zero clock skew increases the clock period, there is still an improvement in performance by applying intentional localized nonzero clock skew to the circuit. With zero clock skew, the minimum period is $T_{\text{CP}} = 27 \text{ ns}$ due to the worst case path delay of the local data path between registers R_2 and R_3 .

IX. EXAMPLE IMPLEMENTATIONS OF CLOCK DISTRIBUTION NETWORKS

A number of interesting and innovative examples of high-performance fully synchronous clock distribution networks for highly specialized and high-performance commercial processors have been described in the literature. These VLSI-based systems required an unusual combination of methodologies and practices consistent with large design teams while maintaining the localized circuit optimization requirements important to very high-speed VLSI circuit design. The process of designing the clock distribution networks used to synchronize some well-known industrial circuit examples is discussed in this section. Effort is also made to present a historical perspective to the general clock distribution design problem and related design tradeoffs and issues as technology and circuit complexities have evolved over the past two decades.

A. The Bell Telephone WE32100 32-Bit Microprocessor [159], [160]

In the early 1980s, a family of 32-bit microprocessors was developed at Bell Laboratories using a variety of advanced

CMOS circuit design techniques and methodologies. Since performance was of fundamental importance, significant attention was placed on synchronization, particularly the design of the global clock distribution network. In 1982, Shoji [159] described the clock distribution of the BELLMAC-32A, a 146000 transistor CPU operating at 8 MHz and built using a 2.5- μm single-level metal silicide CMOS technology. The clock delay and maximum tolerance of the clock distribution network is specified at $15 \text{ ns} \pm 3.5 \text{ ns}$, defining the maximum permissible clock skew. A four-phase clocking strategy is utilized; each phase synchronizes in order, the slave latches, the slave latch logic, the master latches, and the master latch logic. Each time a clock signal crosses a power bus, a silicide crossunder is used to route the clock signal. In order to equalize the series resistance of each clock path, each clock path is routed with an identical number of three power bus crossunders from either of the two primary clock lines around the chip periphery. Buffers are strategically placed after each crossunder to amplify the degraded clock signal. Using this clock distribution strategy, the circuit satisfied the clock frequency specification of 8 MHz at 70 °C with the clock skew not exceeding $\pm 3.5 \text{ ns}$. It is worth noting that due to the significantly increased complexity encountered when distributing four separate clock signals, a four-phase clocking strategy is not particularly appropriate for higher density, higher speed VLSI-based systems. This perspective is consistent with the processor described next.

In 1986, Shoji [160] reported on the electrical design of the WE32100 CPU built using a 1.75- μm CMOS technology. The approach used in designing the WE32100 synchronizing clock system is described; local clock skew is optimized for a small number of the critical paths by applying negative clock skew. This strategy is consistent with the customized design methodology used in the design of the CPU. The clock distribution network utilizes a standard tree structure where the input clock signal is buffered by the clock driver and distributed over the entire circuit. Buffers are again placed after each crossunder. A strategy very similar to the approach presented in Section V-C [28] and depicted in Fig. 15 is used to compensate for the variation in interconnect impedance and register load of each clock line. Clock edges at each register are further synchronized by adjusting the MOSFET transistor geometries of the distributed buffers within the clock distribution network.

Another circuit technique used in the WE32100 to minimize the dependence of the clock skew on any process variations is discussed in Section VII-A. This technique minimizes process-induced clock skew caused by asymmetric variations of the device parameters of the N -channel and P -channel MOSFETs. Clock distribution networks with skews an order of magnitude less than conventionally designed circuits have been simulated with this technique [131].

The issue of chip-to-chip synchronization is important in the design of integrated circuits because these circuits make up the components of a larger computing system. Since individual integrated circuits are processed in different wafer

Table 2
Technology and Performance Characteristics of Three Alpha Microprocessor Generations

Generation	21064 (I)	21164(II)	21264 (III)
Feature size (μm)	0.75	0.5	0.35
Transistors (million)	1.68	9.3	15.2
Die size (square millimeters)	16.8 X 13.9	18.1 X 16.5	16.7 X 18.8
Frequency (MHz)	200	300	600
Capacitive clock load (nF)	3.25	3.75	-----
Power supply (volts)	3.3	3.3	2.2
Supply current (amperes)	< 10	~ 15	> 30
Total power (watts)	~ 30	~ 50	~ 72
% Power dissipated within clock distribution network	40%	40%	44%

lots, the device parameters may differ, and therefore any internal delays, such as the clock lines, will also differ. If the clock delay is T_{CD} for the slowest integrated circuit, as much as $T_{CD}/2$ clock skew may be developed when the fastest and slowest chips communicate synchronously. Therefore, a preferable strategy is to reduce the on-chip clock delay T_{CD} to a minimum and to preselect chips for similar clock delays [160].

B. The DEC/Compaq 64-Bit Alpha Microprocessor Family [167], [168], [206], [207]

A defining application of high-speed clock distribution networks is high-performance microprocessors. The performance of these circuits is often limited by the clocking strategy used in their implementation. The family of DEC/Compaq Alpha microprocessors represents significant milestones in high-speed microprocessor technology. These high-speed integrated circuits operate from 200 MHz with a 3.3-V power supply implemented in 0.75- μm CMOS three-level metal technology to 600 MHz with a 2.2-V power supply in 0.35- μm CMOS technology (see Table 2 for more specific technology and performance characteristics of three examples of the Alpha microprocessor family).

The 21064 microprocessor operates at a 200-MHz clock frequency; therefore, a clock period of 5 ns must be satisfied for each individual local data path, approximately 16 gate delays per clock period including the registers [206]. Therefore, the clock skew must be of sufficiently small magnitude such that all long path constraints are satisfied (e.g., less than 0.5 ns for a 10% positive clock skew requirement). This strategy assumes the clock skew to be a global effect rather than a local effect. Thus, careful attention to modeling the individual active gates and passive interconnects is required in order to design and analyze these high-speed systems. The Alpha 21064 microprocessor contains 1.68 million transistors and supports a fully pipelined 64-bit data structure. The functional attributes of the microprocessor are de-

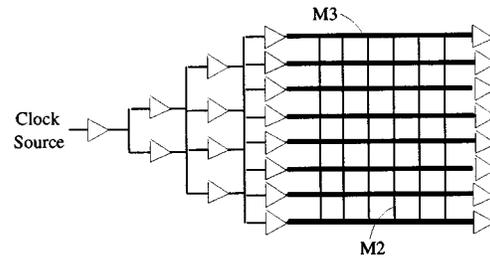


Fig. 23. Clock distribution network of DEC Alpha microprocessor.

scribed in much greater detail in [167]. The focus here is on the clocking strategy used within the circuit.

In designing the Alpha 21064 microprocessor, significant attention has been placed on the circuit implementation. The single-phase clock signal is distributed globally on the top-most level of the trilayer metal process, as is the power distribution, since the third layer of metal is thicker and wider (a 7.5- μm pitch with contacts is used as compared to 2.625 μm and 2.25 μm for the second and first layers of metal, respectively). Therefore, the resistivity per unit length of the third layer of metal and the metal to substrate capacitance is less. A number of inherent difficulties exists within the clock distribution requirements of the Alpha 21064 microprocessor. For example, a substantial capacitive load must be driven at high speed by the clock distribution network, 3250 pF (3.25 nF). Also, for both latch design and power dissipation reasons, so as to minimize short-circuit current, a fast clock edge rate (<0.5 ns) must be maintained throughout the clock distribution network. The huge capacitive load is due to the 63 000 transistor gates being driven by the clock distribution system. The distribution of the loads is asymmetric, necessitating a specialized strategy for distributing the clock.

The single 200-MHz clock signal is distributed through five levels of buffering, where the total network consists of 145 separate elements. Each of the elements contains four levels of buffering with a final output stage locally driving the clocked registers. These distributed buffers are configured as

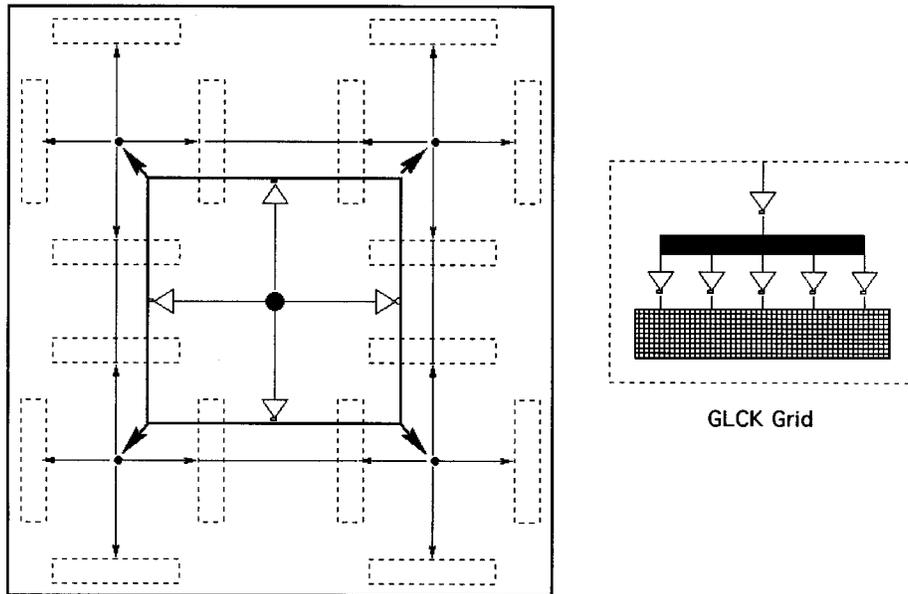


Fig. 24. Global clock distribution network in 600-MHz Alpha microprocessor [207].

a tree as shown in Fig. 23 [43]. Vertical straps are placed on the second level of metal (M2) to minimize any skew that may develop within the initial four-stage portion of the buffer tree. The primary signal-wide distribution is on the third level of metal (M3), designed to be particularly thick to minimize any line resistance as well as to improve process yield.

The approach used in designing the clock distribution of the Alpha 21064 microprocessor is to permit only positive clock skew, thereby assuring that no catastrophic race conditions induced by negative clock skew can occur. Thus, only a graceful degradation in maximum clock rate caused by localized positive clock skew is possible. This strategy is accomplished by centrally locating the clock generation circuitry within the integrated circuit. The clock signal is then radially distributed from the center of the chip die to the periphery of the IC. By carefully monitoring the design of this clock distribution methodology, the likelihood of developing a catastrophic amount of negative clock skew (i.e., $|T_{Skew}| > T_{PD(min)}$) is minimized.

The second generation Alpha microprocessor, the 21164, is manufactured in a $0.5\text{-}\mu\text{m}$ CMOS process and operates at 300 MHz with a 3.3-V power supply. The third generation of the Alpha family is the 21264. This microprocessor is manufactured in a $0.35\text{-}\mu\text{m}$ CMOS technology and operates at 600 MHz. Note that the number of gate delays per clock period is reduced from 16 (for the 21064) to 14 (for the 21164) to 12 (for the 21264). Both the 21164 and 21264 uses a four-level metal process with the metal four pitch decreasing from $6.0\ \mu\text{m}$ to $2.8\ \mu\text{m}$. As listed in Table 2, a primary issue in the design of the Alpha clock distribution networks is the power dissipated by the clock network. Between 40% and 44% of the total power dissipated by the various Alpha microprocessors is due to the clock distribution network.

Unlike the Alpha 20164 in which a single bank of primary clock drivers are centrally placed within the IC, in the 21164, the bank of clock drivers are divided into two in

order to reduce the significant thermal gradients that develop in the Alpha 21064. This strategy is further exploited in the Alpha 21264 in which a gridded two-phase global clock signal (GCLK) is distributed over the entire IC in order to maintain a low-resistance reference clock signal and to distribute the power dissipated by the clock distribution network across the die area (see Fig. 24) [206]. The Alpha 21264 microprocessor is the first version of the Alpha family to utilize a hierarchy of clocks. The global clock signal GCLK is the source of thousands of buffered and conditional (or gated) clock signals driving registers across the IC as shown in Fig. 25 [206], [207]. Local clocks and local conditioning clocks operate several stages past GCLK [207]. This strategy is used to reduce the power dissipated by the clock distribution network while providing clock signals that can be controlled at the local level so that the clock signals can be locally delayed in order to schedule the local clock skews and apply time borrowing, thereby increasing the overall clock frequency. The maximum GCLK skew is less than 75 ps. As shown in Fig. 25, an on-chip PLL is used to generate the clock signal. As shown in Fig. 24, the clock signal is routed as a trunk to the center of the die and distributed by X- and H-trees (see Section V-B) to 16 distributed clock drivers. The final portion of the clock distribution network uses a tree configuration as compared to an H-tree to save power and area. All interconnect within the GCLK grid is shielded both laterally and vertically by a power/ground line. Clock skews of 65 ps operating at $0\ ^\circ\text{C}$ ambient and 2.2 V have been experimentally verified by an e-beam tester [207].

C. 8 Bit \times 8 Bit Pipelined Multiplier [11], [157], [161], [162], [171], [196]

Another application area that requires sophisticated clock distribution is heavily pipelined digital signal processors

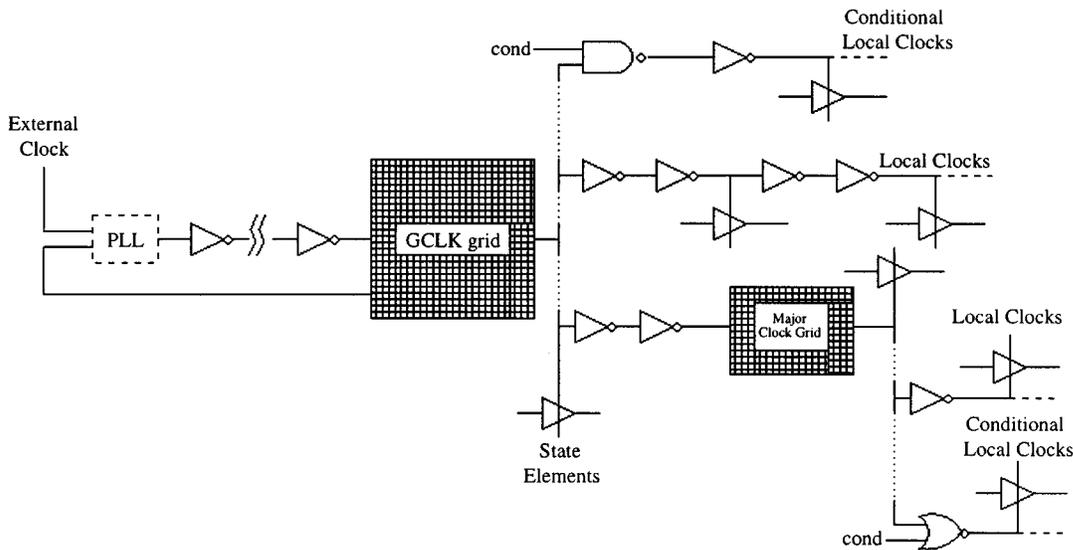


Fig. 25. Hierarchy of the clock distribution network in the 21264 Alpha microprocessor [206].

(DSP), such as FIR/IIR digital filters, multiply-adders, multiply-accumulators, and frequency synthesizers. These types of circuits repeatedly use similar circuit elements, such as multipliers, adders, and registers. Careful attention is placed on developing high-performance customized versions of these circuit elements and these circuit components are repeatedly used within larger DSP systems. The primary difference between different versions of these DSP circuit components is typically the size of the bit slice (e.g., 8 bit versus 64 bit) and the degree of pipelining. Substantial pipelining is applied to these circuits to increase the system clock frequency (see [147], [148], [176]).

The multiplier function is a good example of a complex circuit element capable of significant improvement in clock rate with high levels of pipelining. Since the data flow is nonrecursive, fewer pipeline registers are required as compared to those structures which contain substantial feedback. Furthermore, the multiplier tends to be the critical element (in terms of speed, area, and power) in most DSP circuits. Heavily pipelined multipliers requiring sophisticated clock distribution networks are the focus of considerable research. In this section, specific examples of clock distribution networks in highly pipelined DSP-based multipliers implemented in VLSI technologies are described.

A common feature of these VLSI-based multipliers (and many VLSI-based systems) is the repetitive organization of the physical layout. Repetitive parallel arrays of abutted adder cells, pipelined at each bit (a register placed between each adder cell), provide worst case path delays of only a single adder and a register delay (T_{C-Q} and T_{Set-up}), permitting very high multiplication throughput. Specialized architectures, which are beyond the scope of this paper, such as carry save addition, are used to improve the throughput of these VLSI-based multipliers (see e.g., [11], [157], [161], [162], [171], [196]).

In these types of highly arrayed structures, clock skew can appear both horizontally (serial skew), in the direction of the data flow, and vertically (parallel skew), orthogonal to the

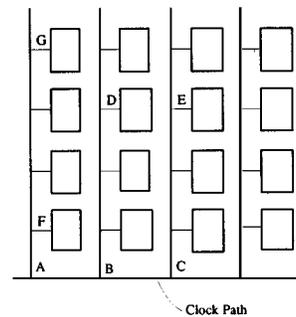


Fig. 26. Clock distribution network of a multiplier array [157], [161].

data flow. As described by Hatamian and Cash in [157], [161] and pictured in Fig. 26, assuming the clock source originates from point A, the clock skew between points D and E at the cell inputs is quite close to the skew between points B and C. As long as this horizontal clock skew is less than the local data path delay between cells, no negative clock skew condition will occur [see Section IV-B and (6)], and the multiplier array will operate properly. Furthermore, additional cells can be added to the array without creating any race conditions as long as the same constraint is maintained. Unlike the horizontal skew, however, the vertical skew is cumulative. The clock skew increases as the signal propagates vertically from point A. Thus, the cumulative skew between points F and G dominates over the horizontal skew, again assuming the clock source originates from point A. It is worth noting that the highly arrayed structure of the multiplier, which makes it particularly amenable to a VLSI implementation, also constrains and limits the topology and layout of the clock distribution network used to synchronize the circuit.

In the 8 bit \times 8 bit multiplier described in [157], [161], which is implemented in a 2.5- μ m CMOS technology and operates up to 70 MHz, the clock signals are distributed entirely on metal, except where short polysilicon crossunders are used to transverse the power lines. A two-level buffer

clock distribution network is used, where the clock load is balanced at the output of the second buffer, and the path before the second buffer is kept symmetric. This clock distribution strategy is similar to that described in [162]. Also an 8 bit \times 8 bit multiplier, the multiplier described in [162], is implemented in a 1.0- μm NMOS technology and operates up to 330 MHz at room temperature and up to 600 MHz with liquid nitrogen cooling. Pipelining after each one bit addition, the multiplier architecture is very similar to that described in [157]. The clock signals are distributed in metal outside the multiplier array and in polycide (strapped polysilicon with silicide to minimize the line resistance) inside the multiplier array. Two-phase clocking is used with a total master and slave register fanout of 8 pf. No special circuitry to prevent overlap of the two-phase clock is used, since this degrades the active-high portion of the clock signal.

Maximal pipelining of multiplying architectures is taken one step further by pipelining each half bit of an 8 bit \times 8 bit multiplier [171]. Operating at 230 MHz and implemented in a 1.6 μm double-level metal CMOS process, the multiplier architecture is similar to that used in [157], [161], [162]. The circuit uses a single-phase clocking scheme [197]. A standard three-level buffer network is used to distribute the clock signal. Each row of the multiplier provides a capacitive load of 5.3 pf. A common clock line runs horizontally in metal one (M1) and is driven by a large-sized buffer. The complete buffer tree is composed of 14 buffers, where ten of these buffers drive the registers (i.e., the leaves of the tree). The maximum skew measured between two clock lines is 150 ps [171].

These three multiplier examples are intended to provide some insight into distributing clock signals within highly arrayed VLSI-based DSP systems. The primary attributes of the multiplier; repetitive circuit elements, abutted or closely spaced layouts, extremely high throughput (i.e., many hundreds of megahertz) due to heavy pipelining, make the multiplier an important class of VLSI system that requires highly specialized clock distribution networks.

D. The Intel IA-64 Microprocessor [212], [213]

The fundamental design trend in current microprocessors has been the development of systems operating at high clock frequencies; specifically, greater than 1-GHz operating frequencies. The primary difficulty in satisfying the goal of achieving these high clock rates is the delay uncertainty caused by process and environmental variations such as on-chip process parameter and temperature variations and related design-oriented issues such as voltage gradients, capacitive and inductive coupling, and load impedance mismatches. Thus managing the design problem of achieving hundreds of picosecond clock periods in order to operate at multigigahertz clock frequencies while understanding, minimizing, and compensating for the large variety of delay uncertainty is the primary issue today in the design of high-performance clock distribution networks.

A recent example of a microprocessor designed to operate beyond the gigahertz frequency level is described in [212].

This circuit, designed and manufactured by Intel, is called the IA-64. The general clock distribution design strategy applied in this microprocessor is to minimize the clock skew through the use of distributed programmable deskew circuits while supporting local optimization of the clock distribution network.

The architecture of the IA-64 clock distribution network consists of three components: a balanced tree structure to globally distribute the clock signal, multiple deskew circuits distributed regionally with balanced clock trees, and multiple local clock buffers that drive the individual circuits and registers. In addition, a separate reference clock signal is distributed along with the global clock signal for use within the deskew circuit [212].

The global portion of the clock distribution network originates from the clock source (an on-chip PLL), distributing the clock signal through a network structured as an H-tree (see Section V-B) until the tree reaches one of eight deskew clusters (each containing up to four deskew circuits). In order to minimize the effects of any capacitive and inductive coupling between the clock lines and any adjacent signal lines, the global clock signal lines are completely shielded by adjacent power and ground lines to enhance the noise immunity of the clock signal while also controlling the inductive current return paths.

The deskew buffer consists of a phase detector, a finite state machine (FSM), and a digitally controlled analog delay line. The buffer provides local delay compensation by comparing the reference clock signal to a local feedback signal, permitting the delay of the clock signal to be adjusted through the analog delay line and the FSM. The delay variation in the clock signal which causes clock skew in the data paths is therefore due to the skew of the reference clock and the uncertainty of the phase detector rather than the much greater delay uncertainties due to the distribution and load mismatches of the global and intermediate clock distribution networks. The delay uncertainty of the reference clock signal is also much smaller than the global clock signal since the topology of the network distributing the reference clock signal is both balanced and much smaller with fixed known loading characteristics [212].

The output signal of each deskew circuit is distributed through a balanced tree network to a set of local buffers. These buffers drive a regional grid of clock ports, permitting the clock signal to be tapped by the underlying active circuits from a local clock buffer. The maximum clock skew within a regional clock grid is estimated (by circuit simulation) to be less than 25 ps with this clock distribution architecture. Each local clock distribution network is managed by the design team developing that circuit block. Thus, by applying this intermediate compensation technique, the global clock distribution design problem can be treated as a local design problem, making the entire chip integration process significantly easier and more manageable. Localized clock skew scheduling and clock gating for power management are also supported at this level of the clock distribution network. Experimental measurements characterizing the maximum clock skew among all deskew buffers is 28 ps (as compared to more

than 110 ps if the deskew circuitry is not used) [212]. Thus, the clock distribution network of this microprocessor utilizes a number of sophisticated design techniques and strategies to support gigahertz clock frequencies while being developed within a design environment consistent with the large teams common to microprocessor design.

The application of clock skew scheduling has also been demonstrated on a similar microprocessor architecture in [213]. Both speed and power are enhanced through the use of clock skew scheduling. The circuit speed is increased by up to an additional 18% in specific highly tuned circuit blocks. More significantly, the power dissipation is decreased by more than 80% by exploiting clock skew scheduling in the less time critical local data paths. The strategy used in this low-power design technique is to slow down the fast paths that do not limit the maximum clock frequency by replacing the high-speed drivers with lower current (and therefore slower) buffers. Thus, power dissipation can be significantly decreased in addition to the speed enhanced by applying local clock skew scheduling even on highly tuned, high-performance microprocessors such as the Intel IA-64 microprocessor.

X. DIRECTIONS FOR FUTURE RESEARCH IN THE DESIGN OF CLOCK DISTRIBUTION NETWORKS

Significant research opportunities remain in the design, synthesis, and test of clock distribution networks. Some examples of these research areas are briefly described in this section.

1) *Automated Synthesis of Clock Distribution Networks*: Much of the current research focuses on automating the synthesis of clock distribution networks to support higher performance requirements. The optimal placement of localized distributed buffers (i.e., repeaters [211]), improved delay models which account for nonlinear active transistor behavior, the appropriate application of localized shielding, the use of localized clock skew to increase circuit speed, and the integration of RC and RLC interconnect and buffer physical delay models for more accurate delay analysis must be considered in the automated design and layout of clock distribution networks. The effects of clock skew, both positive and negative, must also be integrated into behavioral synthesis and RC timing analyzers to detect race conditions as well as to satisfy local and global performance constraints. Synchronous timing constraints must be integrated into high-level behavioral synthesis algorithms, thereby improving the accuracy and generality of these synthesis (and simulation) tools.

Most clock distribution networks are tree structured; however, in many customized VLSI circuits, certain portions of the network are strapped in a mesh configuration to minimize interconnect resistance. These mesh networks decrease clock skew as well as improve circuit reliability. Therefore, one area of future research is the automated layout of clock

meshes. Both timing models and physical layout must be integrated to handle this important extension of tree structured clock distribution networks.

2) *Design of Process Insensitive Clock Distribution Networks*: A research area of primary importance to the practical design of high-speed clock distribution networks is improving the tolerance of these networks to both process and environmental variations. Since functional clock periods are now commonly less than a nanosecond (i.e., greater than a 1-GHz clock frequency), variations in delay of tens of picoseconds can seriously degrade the performance and reliability of these high-speed synchronous systems. Variations of this magnitude, however, are quite common in modern semiconductor technologies. This topic is, therefore, of immediate and significant importance to the high-speed processor community.

3) *Design of Microwave Frequency Clock Distribution Networks*: As system-wide clock frequencies increase beyond the 1–2-GHz frequency levels, transmission line effects will begin to influence the performance characteristics of clock distribution networks. Models of interconnect sections will require the inclusion of more accurate lossy transmission line behavior including the effects of on-chip inductance [198]–[200], [214]. The on-chip inductance will greatly affect both the signal delay and power dissipation characteristics of the gigahertz frequency clock distribution networks. In certain cases, the power dissipated within the clock distribution network may actually decrease due to the nonquadratic behavior of the RLC interconnect with interconnect length (actually between one and two [199]); therefore, fewer buffer repeaters will be required to provide the signal distribution within the clock network. The basic problem of transmission line effects in the design and analysis of high-speed clock distribution networks is expected to become a problem of general near term importance.

These microwave effects will also become significantly more apparent as ultrahigh-speed digital technologies, such as the superconductive digital electronic technology, single flux quantum (SFQ) logic [51], [201]–[205], become more readily available. SFQ circuits typically operate at frequencies well into the 10–100-GHz frequency range [202].

4) *System Architectural Issues in the Design of Clock Distribution Networks*: System issues in the design of clock distribution networks also necessitate focused research. Important and developing architectural structures, such as parallel processors, neural networks, supercomputers, hybrid circuits (or monolithic multichip modules [172]), and wafer scale integration all require specialized synchronization strategies. Related systems issues, such as ultralow-power circuits which require extremely low-power clock distribution networks [52]–[54], are becoming increasingly important. Improving the power dissipation characteristics of clock distribution networks is particularly important, since these networks have been clearly demonstrated to dissipate a large portion of the total system-wide power budget [206], [207]. Topics, such as distributing small differential

signals and task monitoring (or clock gating) strategies, are important areas of research in low-power and/or low-voltage systems that are highly applicable to the design of effective clock distribution networks.

5) *Debug and Production Test of Clock Distribution Networks*: A necessary requirement in developing a product is evaluating the quality of that product. Both debug and production test of high-performance clock distribution networks are of fundamental importance. Minimal research exists that describes how best to debug gigahertz frequency clock distribution networks as well as how to best test these networks in a production environment.

XI. SUMMARY AND CONCLUSION

All electronic systems are fundamentally asynchronous in nature; by the careful insertion of precise localized timing relationships and storage elements, an asynchronous system can be adapted to appear to behave synchronously. As long as the specific local timing and functional relationships are satisfied, synchronous design methodologies can be applied, easing the timing constraints on the data flow, albeit, requiring a clock distribution network to provide the synchronizing reference signal. By synchronizing with distributed clock signals, the clock frequency, a measure of how often new data appear at the output of a system, will remain as the primary performance metric in synchronous systems. Furthermore, systems can be developed which operate at performance levels that would be otherwise unattainable without requiring significant architectural or technological improvements.

It is often noted that the design of the clock distribution network represents the fundamental circuit-based performance limitation in high-speed synchronous digital systems. The local data path-dependent nature of clock skew, rather than any global characteristics, requires extreme care in the design, analysis, and evaluation of high-speed clock distribution networks. The design complexity and difficulty in scaling these networks to finer geometries are the primary reasons for the recent effort placed on developing asynchronous design methodologies. Clearly, however, synchronous systems will be commonplace for a long time to come, necessitating improved techniques for designing and implementing high-speed, highly reliable clock distribution networks. Furthermore, as tighter control of the clocking parameters improves, approaches such as localized clock skew will be more generally used in the design of clock distribution networks to further enhance system performance.

Summarizing this paper on synchronous clock distribution, fundamental timing relationships in synchronous digital integrated circuits have been examined and timing-based constraint relationships developed. Various circuit-based design methodologies and techniques for distributing the clock signals have been suggested and practical circuit applications have been reviewed. It is the intention of this paper to integrate these various topics and to provide some sense of cohe-

siveness to the field of clocking and specifically, clock distribution networks.

REFERENCES

- [1] F. Anceau, "A synchronous approach for clocking VLSI systems," *IEEE J. Solid-State Circuits*, vol. SC-17, pp. 51–56, Feb. 1982.
- [2] M. A. Franklin and D. F. Wann, "Asynchronous and clocked control structures for VLSI based interconnection networks," in *Proc. 9th Annu. Symp. Computer Architecture*, Apr. 1982, pp. 50–59.
- [3] D. Wann and M. Franklin, "Asynchronous and clocked control structures for VLSI based interconnection networks," *IEEE Trans. Comput.*, vol. C-32, pp. 284–293, Mar. 1983.
- [4] S. Dhar, M. Franklin, and D. Wann, "Reduction of clock delays in VLSI structures," in *Proc. IEEE Int. Conf. Computer Design*, Oct. 1984, pp. 778–783.
- [5] S. Unger and C.-J. Tan, "Optimal clocking schemes for high speed digital systems," in *Proc. IEEE Int. Conf. Computer Design*, Oct. 1983, pp. 366–369.
- [6] J. Beausang and A. Albicki, "A method to obtain an optimal clocking scheme for a digital system," in *Proc. IEEE Int. Conf. Computer Design*, Oct. 1983, pp. 68–72.
- [7] S. H. Unger and C.-J. Tan, "Clocking schemes for high-speed digital systems," *IEEE Trans. Comput.*, vol. C-35, pp. 880–895, Oct. 1986.
- [8] D. Noice, R. Mathews, and J. Newkirk, "A clocking discipline for two-phase digital systems," in *Proc. IEEE Int. Conf. Circuits and Computers*, Sept. 1982, pp. 108–111.
- [9] C. Svensson, "Signal resynchronization in VLSI system," *Int. VLSI J.*, vol. 4, pp. 75–80, Mar. 1986.
- [10] M. S. McGregor, P. B. Denyer, and A. F. Murray, "A single-phase clocking scheme for CMOS VLSI," in *Proc. Stanford Conf. Advanced Research in VLSI*, Mar. 1987, pp. 257–271.
- [11] M. Hatamian and G. L. Cash, "Parallel bit-level pipelined VLSI designs for high-speed signal processing," *Proc. IEEE*, vol. 75, pp. 1192–1202, Sept. 1987.
- [12] M. Hatamian, L. A. Hornak, T. E. Little, S. T. Tewksbury, and P. Franzon, "Fundamental interconnection issues," *AT&T Tech. J.*, vol. 66, pp. 13–30, July/Aug. 1987.
- [13] K. D. Wagner, "Clock system design," *IEEE Des. Test Comput.*, pp. 9–27, Oct. 1988.
- [14] A. F. Champernowne, L. B. Bushard, J. T. Rusterholz, and J. R. Schomburg, "Latch-to-latch timing rules," *IEEE Trans. Comput.*, vol. 39, pp. 798–808, June 1990.
- [15] D. G. Messerschmitt, "Synchronization in digital system design," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 1404–1419, Oct. 1990.
- [16] M. Afghahi and C. Svensson, "Performance of synchronous and asynchronous schemes for VLSI systems," *IEEE Trans. Comput.*, vol. 41, pp. 858–872, July 1992.
- [17] V. Chi, "Designing salphasic clock distribution systems," in *Proc. Symp. Integrated Systems*, Mar. 1993, pp. 219–233.
- [18] V. L. Chi, "Salphasic distribution of clock signals for synchronous systems," *IEEE Trans. Comput.*, vol. 43, pp. 597–602, May 1994.
- [19] M. C. Papaefthymiou and K. H. Randall, "Edge-triggering vs. two-phase level-clocking," in *Proc. Symp. Integrated Systems*, Mar. 1993, pp. 201–218.
- [20] W. K. C. Lam, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "Valid clocking in wavepipelined circuits," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1992, pp. 124–131.
- [21] D. A. Joy and M. J. Ciesielski, "Clock period minimization with wave pipelining," *IEEE Trans. Computer-Aided Design*, vol. 12, pp. 461–472, Apr. 1993.
- [22] C. T. Gray, W. Liu, and R. K. Cavin III, "Timing constraints for wave-pipelined systems," *IEEE Trans. Computer-Aided Design*, vol. 13, pp. 987–1004, Aug. 1994.
- [23] X. Zhang and R. Sridhar, "Synchronization of wave-pipelined circuits," in *Proc. IEEE Int. Conf. Computer Design*, Oct. 1994, pp. 164–167.
- [24] E. G. Friedman, "Clock distribution design in VLSI circuits—An overview," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 1993, pp. 1475–1478.
- [25] K. A. Sakallah, T. N. Mudge, T. M. Burks, and E. S. Davidson, "Synchronization of pipelines," *IEEE Trans. Computer-Aided Design*, vol. 12, pp. 1132–1146, Aug. 1993.
- [26] E. G. Friedman, Ed., *High Performance Clock Distribution Networks*. Norwell, MA: Kluwer, 1997.

- [27] E. G. Friedman, Ed., *Clock Distribution Networks in VLSI Circuits and Systems*. Piscataway, NJ: IEEE Press, 1995.
- [28] E. G. Friedman and S. Powell, "Design and analysis of a hierarchical clock distribution system for synchronous standard cell/macrocell VLSI," *IEEE J. Solid-State Circuits*, vol. SC-21, pp. 240–246, Apr. 1986.
- [29] E. Friedman, "A partitionable clock distribution system for sequential VLSI circuits," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 1986, pp. 743–746.
- [30] 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. Computer Design*, Oct. 1986, pp. 118–122.
- [31] "Method of deskewing data pulses," *IBM Tech. Disclosure Bull.*, vol. 28, pp. 2658–2659, Nov. 1985.
- [32] M. Afghahi and C. Svensson, "A scalable synchronous system," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 1988, pp. 471–474.
- [33] —, "A unified single-phase clocking scheme for VLSI systems," *IEEE J. Solid-State Circuits*, vol. 25, pp. 225–233, Feb. 1990.
- [34] B. Wu and N. A. Sherwani, "Effective buffer insertion of clock tree for high-speed VLSI circuits," *Microelectron. J.*, vol. 23, pp. 291–300, July 1992.
- [35] G. M. Blair, "Skew-free clock distribution for standard-cell VLSI designs," *Proc. Inst. Elect. Eng.*, vol. 139, pp. 265–267, Apr. 1992.
- [36] S. Padin, "Scheme for distributing high-speed clock signals in a large digital system," *Electron. Lett.*, vol. 25, pp. 92–93, Jan. 1989.
- [37] M. A. Cirit, "Clock skew elimination in CMOS VLSI," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 1990, pp. 861–864.
- [38] F. Minami and M. Takano, "Clock tree synthesis based on RC delay balancing," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 1992, pp. 28.3.1–28.3.4.
- [39] D. Mijuskovic, "Clock distribution in application specific integrated circuits," *Microelectron. J.*, vol. 18, pp. 15–27, July/Aug. 1987.
- [40] D. Renshaw and C. H. Lau, "Race-free clocking of CMOS pipelines using a single global clock," *IEEE J. Solid-State Circuits*, vol. 25, pp. 766–769, June 1990.
- [41] I. Deol, "Automatic analysis of circuits for tester skew and clock distribution for VLSI circuits," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Oct. 1987, pp. 350–353.
- [42] D. C. Keezer, "Design and verification of clock distribution in VLSI," in *Proc. IEEE Int. Conf. Communications*, Apr. 1990, pp. 811–816.
- [43] M. Horowitz, "Clocking strategies in high performance processors," in *Proc. IEEE Symp. VLSI Circuits*, June 1992, pp. 50–53.
- [44] M. Nekili, Y. Savaria, G. Bois, and M. Bennani, "Logic-based H-trees for large VLSI processor arrays: A novel skew modeling and high-speed clocking method," in *Proc. 5th Int. Conf. Microelectronics*, Dec. 1993, pp. 1–4.
- [45] W. Chuang, S. S. Sapatnekar, and I. N. Hajj, "A unified algorithm for gate sizing and clock skew optimization to minimize sequential circuit area," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1993, pp. 220–223.
- [46] C. Kraft, "Harmonic series analysis of digital clock distribution circuits," in *Proc. 32nd Midwest Symp. Circuits and Systems*, Aug. 1989, pp. 206–211.
- [47] M. Bußmann and U. Langmann, "Active compensation of interconnect losses for multi-GHz clock distribution networks," *IEEE Trans. Circuits Syst. II*, vol. 39, pp. 790–798, Nov. 1992.
- [48] Q. Zhu, W. W.-M. Dai, and J. G. Xi, "Optimal sizing of high speed clock networks based on distributed RC and lossy transmission line models," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1993, pp. 628–633.
- [49] P. Ramanathan, A. J. Dupont, and K. G. Shin, "Clock distribution in general VLSI circuits," *IEEE Trans. Circuits Syst. I*, vol. 41, pp. 395–404, May 1994.
- [50] W. D. Grover, "A new method for clock distribution," *IEEE Trans. Circuits Syst. I*, vol. 41, pp. 149–160, Feb. 1994.
- [51] K. Gaj, E. G. Friedman, M. J. Feldman, and A. Krasniewski, "A clock distribution scheme for large RSFQ circuits," *IEEE Trans. Appl. Superconduct.*, vol. 5, pp. 3320–3324, June 1995.
- [52] E. De Man and M. Schobinger, "Power dissipation in the clock system of highly pipelined ULSI CMOS circuits," in *Proc. Int. Workshop Low Power Design*, Apr. 1994, pp. 133–138.
- [53] H. Kojima, S. Tanaka, and K. Sasaki, "Half-swing clocking scheme for 75% power saving in clocking circuitry," in *Proc. IEEE Symp. VLSI Circuits*, June 1994, pp. 23–24.
- [54] J. L. Neves and E. G. Friedman, "Minimizing power dissipation in nonzero skew-based clock distribution networks," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 1995, pp. 1576–1579.
- [55] A. Vittal and M. Marek-Sadowska, "Power optimal buffered clock tree design," in *Proc. ACM/IEEE Design Automation Conf.*, June 1995, pp. 497–502.
- [56] —, "Low-power buffered clock tree design," *IEEE Trans. Computer-Aided Design*, vol. 16, pp. 965–975, Sept. 1997.
- [57] S. Pullela, N. Menezes, and L. T. Pillage, "Low power IC clock tree design," in *Proc. Custom Integrated Circuits Conf.*, May 1995, pp. 12.4.1–12.4.4.
- [58] M. Nekili, Y. Savaria, and G. Bois, "Design of clock distribution networks in presence of process variations," in *Proc. IEEE 8th Great Lakes Symp. VLSI*, Feb. 1998.
- [59] M. Nekili, G. Bois, and Y. Savaria, "Pipelined H-trees for high-speed clocking of large integrated systems in presence of process variations," *IEEE Trans. VLSI Syst.*, vol. 5, pp. 161–174, June 1997.
- [60] Q. Zhu and W. W. M. Dai, "High-speed clock network sizing optimization based on distributed RC and lossy RLC interconnect models," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 1106–1118, Sept. 1996.
- [61] Y. Ogawa, T. Ishii, Y. Shiraiishi, H. Terai, T. Kozawa, K. Yuyama, and K. Chiba, "Efficient placement algorithms optimizing delay for high-speed ECL masterslice LSI's," in *Proc. ACM/IEEE 23rd Design Automation Conf.*, June 1986, pp. 404–410.
- [62] S. Boon, S. Butler, R. Byrne, B. Setering, M. Casalanda, and A. Scherf, "High performance clock distribution for CMOS ASICs," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 1989, pp. 15.4.1–15.4.5.
- [63] A. Chao, "Clock tree synthesis for large gate arrays," in *High Performance Systems*, 1989, p. 32.
- [64] D. Y. Montuno and R. C. S. Ma, "A layout methodology for the synthesis of high speed global clock nets," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 1992, pp. 28.4.1–28.4.4.
- [65] P. Ramanathan and K. G. Shin, "A clock distribution scheme for nonsymmetric VLSI circuits," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1989, pp. 398–401.
- [66] K. D. Boese and A. B. Kahng, "Zero-skew clock routing trees with minimum wire-length," in *Proc. IEEE Int. Conf. ASICs*, Sept. 1992, pp. 1.1.1–1.1.5.
- [67] J. Burkis, "Clock tree synthesis for high performance ASICs," in *Proc. IEEE Int. Conf. ASICs*, Sept. 1991, pp. 9.8.1–9.8.4.
- [68] T.-H. Chao, Y.-C. Hsu, and J.-M. Ho, "Zero skew clock net routing," in *Proc. ACM/IEEE Design Automation Conf.*, June 1992, pp. 518–523.
- [69] J. Cong, A. B. Kahng, and G. Robins, "Matching-based methods for high-performance clock routing," *IEEE Trans. Computer-Aided Design*, vol. 12, pp. 1157–1169, Aug. 1993.
- [70] M. Edahiro, "A clock net reassignment algorithm using Voronoi diagrams," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1990, pp. 420–423.
- [71] M. A. B. Jackson, A. Srinivasan, and E. S. Kuh, "Clock routing for high performance ICs," in *Proc. ACM/IEEE Design Automation Conf.*, June 1990, pp. 573–579.
- [72] A. B. Kahng, J. Cong, and G. Robins, "High-performance clock routing based on recursive geometric matching," in *Proc. ACM/IEEE Design Automation Conf.*, June 1991, pp. 322–327.
- [73] R. S. Tsay, "Exact zero skew," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1991, pp. 336–339.
- [74] T. Saigo, S. Watanabe, Y. Ichikawa, S. Takayama, T. Umetsu, K. Mima, T. Yamamoto, J. Santos, and J. Buurma, "Clock skew reduction approach for standard cell," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 1990, pp. 16.4.1–16.4.4.
- [75] P. D. Ta and K. Do, "A low power clock distribution scheme for complex IC system," in *Proc. IEEE Int. Conf. ASICs*, Sept. 1991, pp. 1-5.1–1-5.4.
- [76] P. R. Mukund and D. W. Bouldin, "A graph theoretic approach to the clock distribution problem," in *Proc. IEEE Int. Conf. ASICs*, Sept. 1991, pp. 7-5.1–7-5.4.
- [77] J. Cong, A. Kahng, and G. Robins, "On clock routing for general cell layouts," in *Proc. IEEE Int. Conf. ASICs*, Sept. 1991, pp. 14-5.1–14-5.4.
- [78] 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.

- [79] N. A. Sherwani and B. Wu, "Clock layout for high-performance ASIC based on weighted center algorithm," in *Proc. IEEE Int. Conf. ASICs*, Sept. 1991, pp. 15-5.1-15-5.4.
- [80] D. A. Joy and M. J. Ciesielski, "Placement for clock period minimization with multiple wave propagation," in *Proc. ACM/IEEE 28th Design Automation Conf.*, June 1991, pp. 640-643.
- [81] Y.-M. Li and M. A. Jabri, "A zero-skew clock routing scheme for VLSI circuits," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1992, pp. 458-463.
- [82] W. Khan, M. Hossain, and N. Sherwani, "Zero skew clock routing in multiple-clock synchronous systems," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1992, pp. 464-467.
- [83] Q. Zhu and W. W.-M. Dai, "Perfect-balance planar clock routing with minimal path-length," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1992, pp. 473-476.
- [84] R.-S. Tsay, "An exact zero-skew clock routing algorithm," *IEEE Trans. Computer-Aided Design*, vol. 12, pp. 242-249, Feb. 1993.
- [85] W. Khan and N. Sherwani, "Zero skew clock routing algorithm for high performance ASIC systems," in *Proc. IEEE Int. Conf. ASICs*, Sept. 1993, pp. 79-82.
- [86] W. Khan, S. Madhwapathy, and N. Sherwani, "An hierarchical approach to clock routing in high performance systems," in *Proc. IEEE Int. Symp. Circuits and Systems*, May/June 1994, pp. 1.467-1.470.
- [87] S. Lin and C. K. Wong, "Process-variation-tolerant zero skew clock routing," in *Proc. IEEE Int. Conf. ASICs*, Sept. 1993, pp. 83-86.
- [88] M. Edahiro, "A clustering-based optimization algorithm in zero-skew routings," in *Proc. ACM/IEEE Design Automation Conf.*, June 1993, pp. 612-616.
- [89] N.-C. Chou and C.-K. Cheng, "Wire length and delay minimization in general clock net routings," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1993, pp. 552-555.
- [90] M. Edahiro, "Delay minimization for zero-skew routing," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1993, pp. 563-566.
- [91] —, "An efficient zero-skew routing algorithm," in *Proc. ACM/IEEE 31st Design Automation Conf.*, June 1994.
- [92] J. D. Cho and M. Sarrafzadeh, "A buffer distribution algorithm for high-speed clock routing," in *Proc. ACM/IEEE Design Automation Conf.*, June 1993, pp. 537-543.
- [93] G. E. Tellez and M. Sarrafzadeh, "Clock-period constrained minimal buffer insertion in clock trees," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1994.
- [94] A. B. Kahng and C.-W. A. Tsao, "Low-cost single layer clock trees with exact zero Elmore delay skew," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1994.
- [95] M. Seki, T. Inoue, K. Kato, K. Tsurusaki, S. Fukasawa, H. Sasaki, and M. Aizawa, "A specified delay accomplishing clock router using multiple layers," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1994.
- [96] S. Pullela, N. Menezes, and L. T. Pillage, "Reliable nonzero clock trees using wire width optimization," in *Proc. ACM/IEEE Design Automation Conf.*, June 1993, pp. 165-170.
- [97] N. Menezes, A. Balivada, S. Pullela, and L. T. Pillage, "Skew reduction in clock trees using wire width optimization," in *Proc. Custom Integrated Circuits Conf.*, May 1993, pp. 9.6.1-9.6.4.
- [98] S. Pullela, N. Menezes, J. Omar, and L. T. Pillage, "Skew and delay optimization for reliable buffered clock trees," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1993, pp. 556-562.
- [99] M. Mohsin and R. Feaver, "Clock tree generator meets the needs of high-speed deep-submicron designs," *Int. Syst. Des.*, pp. 36-40, Mar. 1998.
- [100] J. L. Neves and E. G. Friedman, "Buffered clock tree synthesis with nonzero clock skew scheduling for increased tolerance to process parameter variations," *J. VLSI Signal Process.*, vol. 16, pp. 149-161, June/July 1997.
- [101] —, "Design methodology for synthesizing clock distribution networks exploiting nonzero clock skew," *IEEE Trans. VLSI Syst.*, vol. 4, pp. 286-291, June 1996.
- [102] I. S. Kourtev and E. G. Friedman, "Topological synthesis of clock trees with nonzero clock skew," in *Proc. ACM/IEEE Int. Workshop Timing Issues in the Specification and Synthesis of Digital Systems*, Dec. 1997, pp. 158-163.
- [103] —, "Topological synthesis of clock trees for VLSI-based DSP systems," in *Proc. IEEE Workshop Signal Processing Systems*, Nov. 1997, pp. 151-162.
- [104] —, "Synthesis of clock tree topologies to implement nonzero skew schedule," *Proc. Inst. Elect. Eng.*, vol. 146, pp. 321-326, Dec. 1999.
- [105] —, "Simultaneous clock scheduling and buffered clock tree synthesis," in *Proc. IEEE Int. Symp. Circuits and Systems*, June 1997, pp. 1812-1815.
- [106] J. L. Neves and E. G. Friedman, "Optimal clock skew scheduling tolerant to process variations," in *Proc. ACM/IEEE Design Automation Conf.*, June 1996, pp. 623-628.
- [107] —, "Reduced sensitivity of clock skew scheduling to technology variations," in *Proc. 5th ACM/SIGDA Physical Design Workshop*, Apr. 1996, pp. 241-248.
- [108] —, "Buffered clock tree synthesis with optimal clock skew scheduling for reduced sensitivity to process parameter variations," in *Proc. TAU ACM/SIGDA Int. Workshop Timing Issues in the Specification and Synthesis of Digital Systems*, Nov. 1995, pp. 131-141.
- [109] A. Takahashi, W. Takahashi, and Y. Kajitani, "Clock-routing driven layout methodology for semi-synchronous circuit design," in *Proc. ACM/IEEE Int. Workshop Timing Issues in the Specification and Synthesis of Digital Systems*, Dec. 1997, pp. 63-66.
- [110] —, "Clock-tree routing realizing a clock-schedule for semi-synchronous circuits," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1997, pp. 234-241.
- [111] P. Ragon and A. Dugar, "Using hierarchical clock tree synthesis to generate balanced clock trees," *Integr. Syst. Des.*, pp. 40-46, Aug. 1997.
- [112] G. E. Tellez and M. Sarrafzadeh, "Minimal buffer insertion in clock trees with skew and slew rate constraints," *IEEE Trans. Computer-Aided Design*, vol. 16, pp. 333-342, Apr. 1997.
- [113] K. Zhu and D. F. Wong, "Clock skew minimization during FPGA placement," *IEEE Trans. Computer-Aided Design*, vol. 16, pp. 376-385, Apr. 1997.
- [114] S. Pullela, N. Menezes, and L. T. Pileggi, "Moment-sensitivity-based sizing for skew reduction in on-chip clock nets," *IEEE Trans. Computer-Aided Design*, vol. 16, pp. 210-215, Feb. 1997.
- [115] —, "Post-processing of clock trees via wiresizing and buffering for robust design," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 691-701, June 1995.
- [116] N. Ito, H. Sugiyama, and T. Konno, "ChipPRISM: Clock routing and timing analysis for high-performance CMOS VLSI chips," *Fujitsu Sci. Technol. J.*, vol. 31, pp. 180-187, Dec. 1995.
- [117] J. G. Xi and W. W.-M. Dai, "Buffer insertion and sizing under process variations for low power clock distribution," in *Proc. ACM/IEEE Design Automation Conf.*, June 1995, pp. 491-496.
- [118] A. B. Kahng and C.-W. A. Tsao, "Planar-DME: A single-layer zero-skew clock tree router," *IEEE Trans. Computer-Aided Design*, vol. 15, pp. 8-19, Jan. 1996.
- [119] J. L. Neves and E. G. Friedman, "Topological design of clock distribution networks based on nonzero clock skew specifications," in *Proc. IEEE Midwest Symp. Circuits and Systems*, Aug. 1993, pp. 468-471.
- [120] —, "Circuit synthesis of clock distribution networks based on nonzero clock skew," in *Proc. IEEE Int. Symp. Circuits and Systems*, May/June 1994, pp. 4.175-4.178.
- [121] —, "Automated synthesis of skew-based clock distribution networks," *VLSI Design*, vol. 7, pp. 31-57, 1998.
- [122] J. Chung and C.-K. Cheng, "Optimal buffered clock tree synthesis," in *Proc. IEEE Int. Conf. ASICs*, Sept. 1994, pp. 130-133.
- [123] —, "Skew sensitivity minimization of buffered clock tree," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1994.
- [124] N. Park and A. Parker, "Synthesis of optimal clocking schemes," in *Proc. ACM/IEEE 22nd Design Automation Conf.*, June 1985, pp. 489-495.
- [125] E. G. Friedman, "The application of localized clock distribution design to improving the performance of retimed sequential circuits," in *Proc. IEEE Asia-Pacific Conf. Circuits and Systems*, Dec. 1992, pp. 12-17.
- [126] T. Soyata, E. G. Friedman, and J. H. Mulligan Jr., "Integration of clock skew and register delays into a retiming algorithm," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 1993, pp. 1483-1486.
- [127] T. Soyata and E. G. Friedman, "Retiming with nonzero clock skew, variable register, and interconnect delay," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1994, pp. 234-241.
- [128] L.-F. Chao and E. H.-M. Sha, "Retiming and clock skew for synchronous systems," in *Proc. IEEE Int. Symp. Circuits and Systems*, May/June 1994, pp. 1.283-1.286.
- [129] B. Lockyear and C. Ebeling, "The practical application of retiming to the design of high-performance systems," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1993, pp. 288-295.

- [130] T. Soyata, E. G. Friedman, and J. H. Mulligan Jr., "Incorporating interconnect, register, and clock distribution delays into the re-timing process," *IEEE Trans. Computer-Aided Design*, vol. 16, pp. 105–120, Jan. 1997.
- [131] M. Shoji, "Elimination of process-dependent clock skew in CMOS VLSI," *IEEE J. Solid-State Circuits*, vol. SC-21, pp. 875–880, Oct. 1986.
- [132] S. Lin and C. K. Wong, "Process-variation-tolerant clock skew minimization," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1994.
- [133] E. Vanden Meersch, L. Claesen, and H. De Man, "Automated analysis of timing faults in synchronous MOS circuits," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 1988, pp. 487–490.
- [134] M. R. Dagenais and N. C. Rumin, "Automatic determination of optimal clocking parameters in synchronous MOS VLSI circuits," in *Proc. Stanford Conf. Advanced Research in VLSI*, Mar. 1988, pp. 19–33.
- [135] —, "On the calculation of optimal clocking parameters in synchronous circuits with level-sensitive latches," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 268–278, Mar. 1989.
- [136] C. Svensson and M. Afghahi, "On RC line delays and scaling in VLSI systems," *Electron. Lett.*, vol. 24, pp. 562–563, Apr. 28, 1988.
- [137] M. Afghahi and C. Svensson, "Calculation of clock path delay and skew in VLSI synchronous systems," in *Proc. IEEE Eur. Conf. Circuit Theory and Design*, Sept. 1989, pp. 265–269.
- [138] S. C. Menon and K. A. Sakallah, "Clock qualification algorithm for timing analysis of custom CMOS VLSI circuits with overlapping clocking disciplines and on-section clock derivation," in *Proc. 1st Int. Conf. Systems Integration*, Apr. 1990, pp. 550–558.
- [139] K. A. Sakallah, T. N. Mudge, and O. A. Olukoton, "Analysis and design of latch-controlled synchronous digital circuits," in *Proc. ACM/IEEE Design Automation Conf.*, June 1990, pp. 111–117.
- [140] —, "Analysis and design of latch-controlled synchronous digital circuits," *IEEE Trans. Computer-Aided Design*, vol. 11, pp. 322–333, Mar. 1992.
- [141] R.-S. Tsay and I. Lin, "Robin Hood: A system timing verifier for multi-phase level-sensitive clock designs," in *Proc. IEEE Int. Conf. ASICs*, Sept. 1992, pp. 516–519.
- [142] I. Lin, J. A. Ludwig, and K. Eng, "Analyzing cycle stealing on synchronous circuits with level-sensitive latches," in *Proc. ACM/IEEE Design Automation Conf.*, June 1992, pp. 393–398.
- [143] S. Narayan and D. D. Gajski, "System clock estimation based on clock wastage minimization," in *Proc. IEEE Eur. Design Automation Conf.*, Sept. 1992, pp. 66–71.
- [144] M. C. Papaefthymiou and K. H. Randall, "TIM: A timing package for two-phase, level-clocked circuitry," in *Proc. ACM/IEEE Design Automation Conf.*, June 1993, pp. 497–502.
- [145] J. P. Fishburn, "Clock skew optimization," *IEEE Trans. Comput.*, vol. 39, pp. 945–951, July 1990.
- [146] K. A. Sakallah, T. N. Mudge, T. M. Burks, and E. S. Davidson, "Optimal clocking of circular pipelines," in *Proc. IEEE Int. Conf. Computer Design*, Oct. 1991, pp. 642–646.
- [147] E. G. Friedman and J. H. Mulligan Jr., "Clock frequency and latency in synchronous digital systems," *IEEE Trans. Signal Processing*, vol. 39, pp. 930–934, Apr. 1991.
- [148] —, "Pipelining of high performance synchronous digital systems," *Int. J. Electron.*, vol. 70, pp. 917–935, May 1991.
- [149] K. A. Sakallah, T. N. Mudge, and O. A. Olukoton, "check T_c and min T_c : Timing verification and optimal clocking of synchronous digital circuits," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1990, pp. 552–555.
- [150] —, "Optimal clocking of synchronous systems," in *Proc. TAU-ACM Int. Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, Aug. 1990.
- [151] T. G. Szymanski, "Computing optimal clock schedules," in *Proc. ACM/IEEE Design Automation Conf.*, June 1992, pp. 399–404.
- [152] T. G. Szymanski and N. Shenoy, "Verifying clock schedules," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1992, pp. 124–131.
- [153] N. Shenoy and R. K. Brayton, "Graph algorithms for clock schedule optimization," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1992, pp. 132–136.
- [154] R. B. Deokar and S. S. Sapatnekar, "A graph-theoretic approach to clock skew optimization," in *Proc. IEEE Int. Symp. Circuits and Systems*, May/June 1994, pp. 1.407–1.410.
- [155] W. Chuang, S. S. Sapatnekar, and I. N. Hajj, "A unified algorithm for gate sizing and clock skew optimization to minimize sequential circuit area," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1993, pp. 220–223.
- [156] T. M. Burks, K. A. Sakallah, and T. N. Mudge, "Identification of critical paths in circuits with level-sensitive latches," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1992, pp. 137–141.
- [157] M. Hatamian and G. Cash, "A 70-MHz 8 bit \times 8 bit parallel pipelined multiplier in 2.5 μ m CMOS," *IEEE J. Solid-State Circuits*, vol. SC-21, pp. 505–513, Aug. 1986.
- [158] D. C. Keezer and V. K. Jain, "Clock distribution strategies for WSI: A critical survey," in *Proc. IEEE Int. Conf. Wafer Scale Integration*, Jan. 1991, pp. 277–283.
- [159] M. Shoji, "Electrical design of BELLMAC-32A microprocessor," in *Proc. IEEE Int. Conf. Circuits and Computers*, Sept. 1982, pp. 112–115.
- [160] —, "Reliable chip design method in high performance CMOS VLSI," in *Proc. IEEE Int. Conf. Computer Design*, Oct. 1986, pp. 389–392.
- [161] M. Hatamian and G. L. Cash, "High speed signal processing, pipelining, and VLSI," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, Apr. 1986, pp. 1173–1176.
- [162] T. G. Noll, D. Schmitt-Landsiedel, H. Klar, and G. Enders, "A pipelined 330-MHz multiplier," *IEEE J. Solid-State Circuits*, vol. SC-21, pp. 411–416, June 1986.
- [163] R. Maini, J. McDonald, and L. Spangler, "A clock distribution circuit with a 100 PS skew window," in *Proc. IEEE Bipolar Circuits and Technology Meeting*, Sept. 1987, pp. 41–43.
- [164] M. Usami, S. Ishii, S. Kawashima, B. Fujita, N. Masuda, and H. Itoh, "An automatic 5 ps skew-time control clock-pulse adjustment LSI for high-speed computers," in *Proc. IEEE Symp. VLSI Circuits*, May/June 1991, pp. 53–54.
- [165] G. M. Blair, "Bit-serial correlator with novel clocking scheme," in *Proc. Eur. Solid-State Circuits Conf.*, 1991, pp. 157–160.
- [166] —, "Self-generating clock using an augmented distribution network," *Proc. Inst. Elect. Eng.*, vol. 144, pp. 219–222, Aug. 1997.
- [167] D. W. Dobberpuhl *et al.*, "A 200-MHz 64-b dual-issue CMOS microprocessor," *IEEE J. Solid-State Circuits*, vol. 27, pp. 1555–1565, Nov. 1992.
- [168] J. Montanaro *et al.*, "A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor," *Digital Tech. J.*, vol. 9, no. 1, pp. 49–62, 1997.
- [169] W. J. Bowhill *et al.*, "Circuit implementation of a 300-MHz 64-bit second-generation CMOS alpha CPU," *Digital Tech. J.*, vol. 7, no. 1, pp. 100–118, 1995.
- [170] R. B. Watson Jr., H. A. Collins, and R. Iknaiian, "Clock buffer chip with absolute delay regulation over process and environmental variations," in *Proc. IEEE Custom Integrated Circuits Conf.*, May 1992, pp. 25.2.1–25.2.5.
- [171] D. Somasekhar and V. Visvanathan, "A 230-MHz half-bit level pipelined multiplier using true single-phase clocking," *IEEE Trans. VLSI Syst.*, vol. 1, pp. 415–422, Dec. 1993.
- [172] R. Reinschmidt and D. Leuthold, "Clocking considerations for a Pentium based CPU module with a 512 K Byte secondary cache," in *Proc. IEEE Multi-Chip Module Conf.*, Mar. 1994, pp. 26–31.
- [173] K. Wagner and E. McCluskey, "Tuning, clock distribution, and communication in VLSI high-speed chips," Stanford Univ., Stanford, CA, CRC Tech. Rep. 84-5, June 1984.
- [174] K. D. Wagner, "A Survey of Clock distribution techniques in high-speed computer systems," Stanford Univ., Stanford, CA, CRC Tech. Rep. 86-20, Dec. 1986.
- [175] E. G. Friedman, "Latching characteristics of a CMOS bistable register," *IEEE Trans. Circuits Syst. I*, vol. 40, pp. 902–908, Dec. 1993.
- [176] E. G. Friedman and J. H. Mulligan Jr., "Pipelining and clocking of high performance synchronous digital systems," in *VLSI Signal Processing Technology*, M. A. Bayoumi and E. E. Swartzlander Jr., Eds. Norwell, MA: Kluwer, 1994, pp. 97–133.
- [177] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1989.
- [178] D. B. West, *Introduction to Graph Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
- [179] M. Hatamian, "Understanding clock skew in synchronous systems," in *Concurrent Computations (Algorithms, Architecture and Technology)*, S. K. Tewksbury, B. W. Dickinson, and S. C. Schwartz, Eds. New York: Plenum, 1988, pp. 87–96.
- [180] J. A. Marques and A. Cunha, "Clocking of VLSI circuits," in *VLSI Architecture*, B. Randell and P. C. Treleaven, Eds. Englewood Cliffs, NJ: Prentice-Hall, 1983, pp. 165–178.

- [181] E. G. Friedman, "Performance limitations in synchronous digital systems," Ph.D. dissertation, Univ. California, Irvine, June 1989.
- [182] T. W. Williams and K. P. Parker, "Design for testability—A survey," *Proc. IEEE*, vol. 71, pp. 98–112, Jan. 1983.
- [183] R. H. Dennard, F. H. Gaensslen, H.-N. Yu, V. L. Rideout, E. Bassous, and A. R. LeBlanc, "Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE J. Solid-State Circuits*, vol. SC-9, pp. 256–268, Oct. 1974.
- [184] H. B. Bakoglu and J. D. Meindl, "Optimal interconnection circuits for VLSI," *IEEE Trans. Electron Devices*, vol. ED-32, pp. 903–909, May 1985.
- [185] H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI*. Reading, MA: Addison Wesley, 1990.
- [186] C. V. Gura, "Analysis of clock skew in distributed resistive-capacitive interconnects," University of Illinois, Urbana, SRC Tech. Rep. T87053, June 1987.
- [187] S. Y. Kung, *VLSI Array Processors*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [188] D. D. Gajski, *Silicon Compilation*. Reading, MA: Addison Wesley, 1988.
- [189] 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.
- [190] G. Yacoub, H. Pham, M. Ma, and E. G. Friedman, "A system for critical path analysis based on back annotation and distributed interconnect impedance models," *Microelectron. J.*, vol. 19, pp. 21–30, May/June 1988.
- [191] W. C. Elmore, "The transient response of damped linear networks with particular regard to wideband amplifiers," *J. Appl. Phys.*, vol. 19, pp. 55–63, Jan. 1948.
- [192] J. Rubinstein, P. Penfield Jr., and M. A. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 202–211, July 1983.
- [193] J. Vlach, J. A. Barby, A. Vannelli, T. Talkhan, and C. J. Shin, "Group delay as an estimate of delay in logic," *IEEE Trans. Computer-Aided Design*, vol. 10, pp. 949–953, July 1991.
- [194] T. Sakurai, "Approximation of wiring delay in MOSFET LSI," *IEEE J. Solid-State Circuits*, vol. SC-18, pp. 418–426, Aug. 1983.
- [195] C. T. Gray, "Optimal clocking of wave pipelined systems and CMOS applications," Ph.D. dissertation, North Carolina State Univ., July 1993.
- [196] M. R. Santoro, "Design and clocking of VLSI multipliers," Ph.D. dissertation, Stanford Univ., Palo Alto, CA, Oct. 1989.
- [197] Y. Jiren, I. Karlsson, and G. Svensson, "A true single-phase-clock dynamic CMOS circuit technique," *IEEE J. Solid-State Circuits*, vol. 22, pp. 899–901, Oct. 1987.
- [198] Y. I. Ismail, E. G. Friedman, and J. L. Neves, "Figures of merit to characterize the importance of on-chip inductance," *IEEE Trans. VLSI Syst.*, vol. 7, pp. 442–449, Dec. 1999.
- [199] Y. I. Ismail and E. G. Friedman, "Effects of inductance on the propagation delay and repeater insertion in VLSI circuits," *IEEE Trans. VLSI Systems*, vol. 8, no. 2, pp. 195–206, Apr. 2000.
- [200] Y. I. Ismail, E. G. Friedman, and J. Neves, "Equivalent Elmore delay for RLC trees," *IEEE Trans. Computer-Aided Design*, vol. 19, pp. 83–97, Jan. 2000.
- [201] K. K. Likharev and V. K. Semenov, "RSFQ logic/memory family: A new Josephson-junction technology for sub-terahertz clock-frequency digital systems," *IEEE Trans. Appl. Superconduct.*, vol. 1, pp. 3–28, Mar. 1991.
- [202] K. Gaj, E. G. Friedman, and M. J. Feldman, "Timing of multi-gigahertz rapid single flux quantum digital circuits," *J. VLSI Signal Process.*, vol. 16, pp. 247–276, June/July 1997.
- [203] ———, "Timing of large RSFQ digital circuits," in *Proc. 6th Int. Superconductive Electronics Conf.*, June 1997, pp. 299–301.
- [204] ———, "Two-phase clocking for medium to large RSFQ circuits," in *Proc. 6th Int. Superconductive Electronics Conf.*, June 1997, pp. 302–304.
- [205] ———, "Choice of the optimum timing scheme for RSFQ digital circuits," in *Proc. 5th Int. Workshop High-Temperature Superconducting Electron Devices*, May 1997, pp. 39–40.
- [206] P. E. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon, "High-performance microprocessor design," *IEEE J. Solid-State Circuits*, vol. 33, pp. 676–686, May 1998.
- [207] D. W. Bailey and B. J. Benschneider, "Clocking design and analysis for a 600-MHz alpha microprocessor," *IEEE J. Solid-State Circuits*, vol. 3, pp. 1627–1633, Nov. 1998.
- [208] I. S. Kourtev and E. G. Friedman, *Timing Optimization Through Clock Skew Scheduling*. Norwell, MA: Kluwer, 2000.
- [209] ———, "Clock skew scheduling for improved reliability via quadratic programming," in *Proc. IEEE Int. Conf. Computer-Aided Design*, Nov. 1999, pp. 239–243.
- [210] X. Liu, M. C. Papaefthymiou, and E. G. Friedman, "Maximizing performance by retiming and clock skew scheduling," *Proc. IEEE/ACM Design Automation Conf.*, pp. 231–236, June 1999.
- [211] V. Adler and E. G. Friedman, "Repeater design to reduce delay and power in resistive interconnect," *IEEE Trans. Circuits Syst. II*, vol. 45, pp. 607–616, May 1998.
- [212] S. Rusu and S. Tam, "Clock generation and distribution for the first IA-64 microprocessor," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2000, pp. 176–177.
- [213] D. Velenis, K. T. Tang, I. S. Kourtev, V. Adler, F. Baez, and E. G. Friedman, "Demonstration of speed and power enhancements through application of nonzero clock skew scheduling," in *Proc. ACM/IEEE Int. Workshop Timing Issues in the Specification and Synthesis of Digital Systems*, Dec. 2000, pp. 58–63.
- [214] Y. I. Ismail and E. G. Friedman, *On-Chip Inductance in High Speed Integrated Circuits*. Norwell, MA: Kluwer, 2001.



Eby G. Friedman (Fellow, IEEE) received the B.S. degree from Lafayette College in 1979, and the M.S. and Ph.D. degrees from the University of California, Irvine, in 1981 and 1989, respectively, all in electrical engineering.

From 1979 to 1991, he was with Hughes Aircraft Company, rising to the position of Manager of the Signal Processing Design and Test Department, responsible for the design and test of high-performance digital and analog ICs.

He has been with the Department of Electrical and Computer Engineering at the University of Rochester, Rochester, NY, since 1991, where he is a Professor, the Director of the High Performance VLSI/IC Design and Analysis Laboratory, and the Director of the Center for Electronic Imaging Systems. He was also a Visiting Professor at the Technion—Israel Institute of Technology. His current research and teaching interests are in high-performance synchronous digital and mixed-signal microelectronic design and analysis with application to high-speed portable processors and low-power wireless communications. He is the author of over 150 papers and book chapters and the author or editor of six books in the fields of high-speed and low-power CMOS design techniques, high-speed interconnect, and the theory and application of synchronous clock distribution networks.

Dr. Friedman is the current Editor-in-Chief of the IEEE TRANSACTIONS ON VLSI SYSTEMS, Regional Editor of the *Journal of Circuits, Systems, and Computers*, a Member of the editorial boards of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: ANALOG AND DIGITAL SIGNAL PROCESSING, *Journal of VLSI Signal Processing*, and *Analog Integrated Circuits and Signal Processing*, a Member of the Circuits and Systems (CAS) Board of Governors, and a Member of the technical program committee of a number of conferences. He previously was chair of the IEEE TRANSACTIONS ON VLSI SYSTEMS steering committee, a Member of the editorial board of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—II: ANALOG AND DIGITAL SIGNAL PROCESSING, CAS liaison to the Solid-State Circuits Society, Chair of the VLSI Systems and Applications CAS Technical Committee, Chair of the Electron Devices Chapter of the IEEE Rochester Section, Program Co-Chair of the 2000 SiSP conference, Chair of the VLSI track for ISCAS'96 and '97, Technical Co-Chair of the 1997 IEEE International Workshop on Clock Distribution Networks, editor of several special issues in a variety of journals, and a recipient of the Howard Hughes Masters and Doctoral Fellowships, an IBM University Research Award, an Outstanding IEEE Chapter Chairman Award, and a University of Rochester College of Engineering Teaching Excellence Award. He is also a Fulbright scholar.