CLOCK DISTRIBUTION IN SYNCHRONOUS SYSTEMS

In a synchronous digital system, the clock signal is used to define a time reference for the movement of data within that system. Because 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 greatest distances, and operate at the highest speeds of any signal, either control or data, within the entire system. Because 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 increasing significance 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 and 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 (1–27). 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 (28) awaiting the incoming clock signal, which determines when the data signal leaves the register. When the enabling clock signal reaches the register, the data signal leaves the bistable register, propagates through the combinatorial network, 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 (29–31):

1. the memory storage elements,
2. the logic elements, and
3. the clocking circuitry and distribution network.

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 article, bears separate focus.

The article is organized as follows. In the first section, an overview of the operation of a synchronous system is provided. In the next section, 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 then described. The interplay among the aforementioned three subsystems making up a synchronous digital system is described next, 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, as are the automated layout and synthesis of clock distribution networks. Then techniques for making clock distribution networks less sensitive to process parameter variations are discussed. Localized scheduling of the clock delays is useful in optimizing the performance of high-speed synchronous circuits. Determining the optimal timing characteristics of a clock distribution network is reviewed next. 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 Very Large Scale Integration (VLSI)-based systems has been described in the literature, and some examples of these circuits are described. 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 the next section. Finally, a summary of this article with some concluding remarks is provided.

SYNCHRONOUS SYSTEMS

A digital synchronous circuit is 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 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.*

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

$$\frac{1}{f_{\text{clkMAX}}} = T_{CP}(\min) \geq T_{PD(\max)} + T_{\text{Skew}} \tag{1}$$

where

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

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 after the clock signal C_i arrives $T_{\text{C-Q}}$, 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 $T_{\text{Logic}(\min)}$ because data are latched into each register within the same clock period.

The sum of the delay components in Eq. (2) must satisfy the timing constraint of Eq. (1) in order to attain the clock period $T_{CP}(\min)$, which is the inverse of the maximum possible clock frequency f_{clkMAX}. The clock skew $T_{\text{Skew}ij}$ can be positive or negative depending on whether C_j leads or lags C_i, respectively. The clock period is chosen such that the latest data generated by the initial register is latched by the final register with 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 in the output of the initial register arrives. The waveforms in Fig. 1 show the timing requirement of Eq. (1) being barely satisfied (i.e., the data signal arrives at R_f just before the clock signal arrives).

An example of a local data path $R_i \Rightarrow R_f$ is shown in Fig. 2. 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_{if}, 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 $d(i, f)$ and $D(i, f)$, respectively. Note that both $d(i, f)$ and $D(i, f)$ are caused by 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 in-

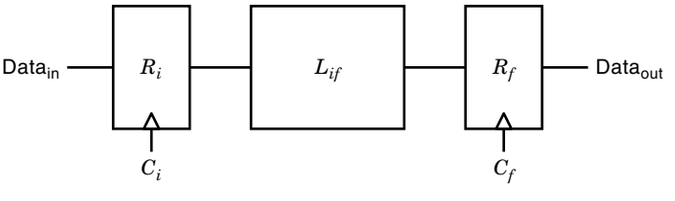

Figure 1. Timing diagram of clocked local data path.

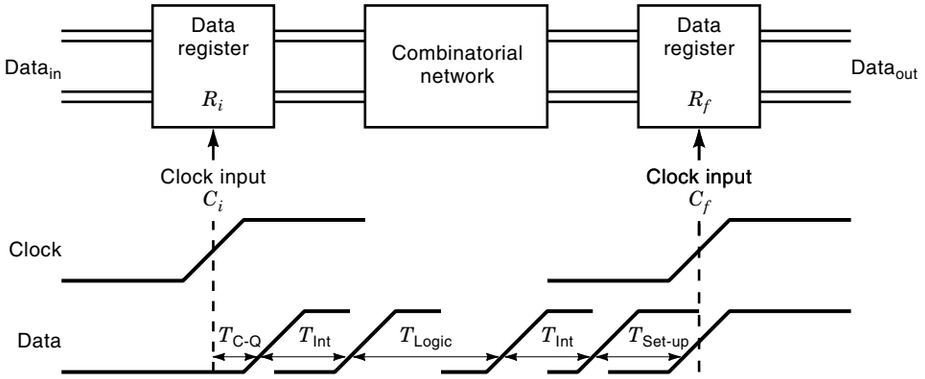

Figure 2. A local data path. Composed of an initial and final register with a logic block between the two registers.

terconnect delay caused by the presence of wires on the signal path, $R_i \Rightarrow R_f$.

A synchronous digital system can be modeled (32,33) as a *directed graph* G with *vertex set* $V = \{v_1, \ldots, V_{Nr}\}$ and *edge set* $E = \{e_1, \ldots, e_{Np}\}$ *CVXV*. 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 Fig. 3, an input or an output of the circuit is indicated by an edge that is incident to only one vertex.

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, where both clock signals originate from the same clock signal source and 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 jth clocked register is the *clock delay,* T_{Cj}. The clock delays of the initial clock signal T_{Ci} and the final clock signal T_{Cf} define the time reference when the data signals begin to leave their respective registers. These clock signals originate from a clock distribution network that is designed to generate a specific clock signal waveform used to synchronize each register. This standard clock distribution network structure is based on *equipotential clocking,* where the entire network is considered a surface that 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.

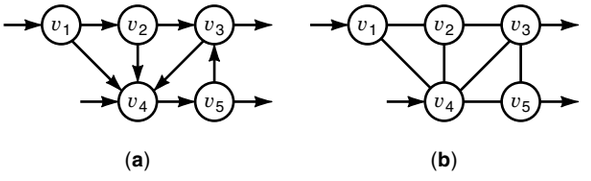

Figure 3. Graph G of a circuit with $N_R = 5$ registers. (a) The directed graph G. (b) The underlying graph G_u corresponding to the graph G in (a).

Definition of Clock Skew

The difference in clock signal arrival time between *two sequentially adjacent registers,* as shown in Eq. (3), 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 follows.

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

$$T_{Skewij} = T_{Ci} - T_{Cj} \tag{3}$$

where T_{Ci} and T_{Cj} 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 relevant only to sequentially adjacent registers making up a single data path, as shown in Fig. 2. Thus, systemwide (or chipwide) 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, systemwide global clock skew places constraints on the permissible local clock skew. It should be noted that in Refs. 11 and 34, 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 (caused by 3 and 4),

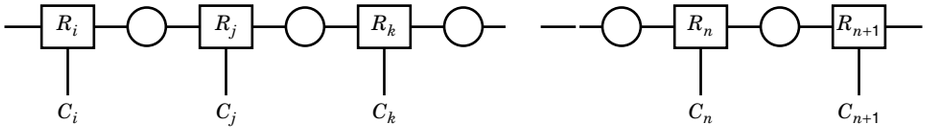

Figure 4. Global data path composed of multiple local data paths.

3. differences in passive interconnect parameters, such as line resistivity, dielectric constant and thickness, via/contact resistance, line and fringing capacitance, and line dimensions, and
4. differences in active device parameters, such as MOS threshold voltages and channel mobilities, which affect the delay of the active buffers.

It should be noted that for a well-designed and well-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 article 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 (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 here.

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 that 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 that do not belong to the same global data path has no physical meaning because no functional data transfer between these registers occurs.

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 in Theorem 2.

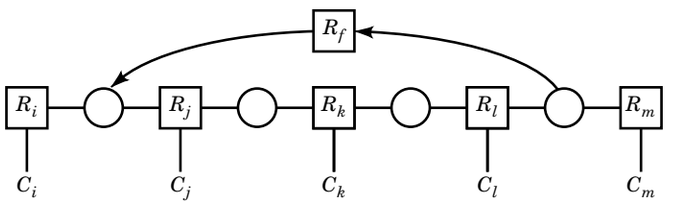

Figure 5. Global path with feedback path.

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

$$T_{\text{Skewfeedback},lj} = -T_{\text{Skewforward},jl} \tag{4}$$

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 the section entitled "Specification of the Optimal Timing Characteristics of Clock Distribution Networks."

TIMING CONSTRAINTS CAUSED BY 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 next.

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 Eq. (1) (7,8,11,16,31,34–37). 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 arrival of the clock signal at the initial register of the same sequential data path T_{Ci} [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 that 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 affects only 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 Eqs. (1) and (2), the maximum

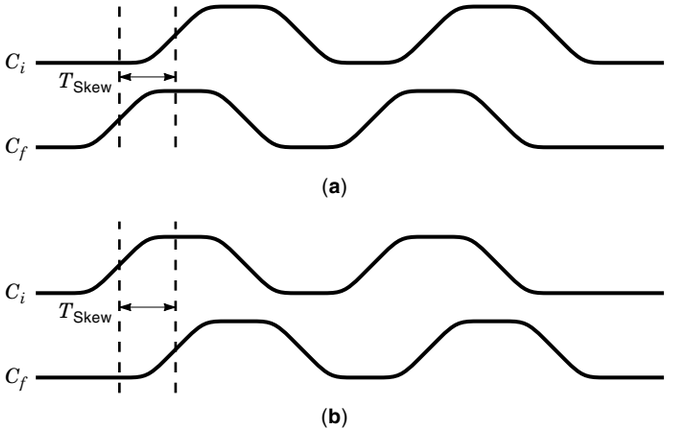

Figure 6. Clock timing diagrams: (a) positive clock skew, (b) negative clock skew.

permissible positive clock skew can be expressed as (7,8,11,16,31,34–37)

$$T_{\text{Skew}} \le T_{\text{CP}} - T_{\text{PD(max)}} = T_{\text{CP}} - (T_{\text{C-Q}} + T_{\text{Logic(max)}} + T_{\text{Int}} + T_{\text{Set-up}}) \quad \text{for } T_{\text{C}i} > T_{\text{C}f} \quad (5)$$

where $T_{\text{PD(max)}}$ 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 Eq. (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 Eq. (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.

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 (11,12,31,34,36,38–41). In this case, when C_f *lags* C_i, the clock skew must be less than the time required for the data to leave the initial register, propagate through the interconnect combinatorial logic, and set up 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 because of fluctuations in ambient temperature or power supply voltage (36). Correct operation requires that R_f latches data that correspond to the data R_i latched during the previous clock period. This constraint on clock skew is

$$|T_{\text{Skew}}| \le T_{\text{PD(min)}} = (T_{\text{C-Q}} + T_{\text{Logic(min)}} + T_{\text{Int}} + T_{\text{Hold}}) \quad \text{for } T_{\text{C}f} > T_{\text{C}i} \quad (6)$$

where $T_{\text{PD(min)}}$ 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 after the clock signal changes state.

An important example in which this minimum constraint can occur is in those designs that 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, $T_{\text{LOGIC(min)}}$ is zero and T_{Int} approaches zero (because cascaded registers are typically designed, at the geometric level, to abut). If $T_{\text{C}f} > T_{\text{C}i}$ (i.e., negative clock skew), then the minimum constraint becomes

$$|T_{\text{Skew}}| \le T_{\text{C-Q}} + T_{\text{Hold}} \quad \text{for } T_{\text{C}f} > T_{\text{C}i} \quad (7)$$

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 (42) 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 can be negative and greater in magnitude than the local register delays. Therefore, with increased negative clock skew, Eq. (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 $S > 1$ (16,43–45). Device-dependent delays, such as $T_{\text{C-Q}}$, $T_{\text{Set-up}}$, and T_{Logic}, scale as $1/S$, whereas 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, Eqs. (6) and (7) should be considered carefully (46). One straightforward method to avoid the effect of technology scaling on those

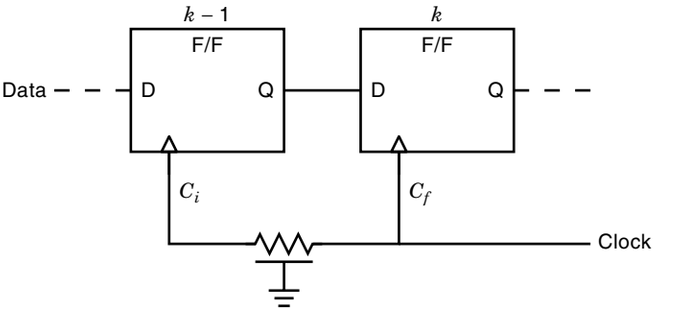

Figure 7. k-bit shift register with positive clock skew.

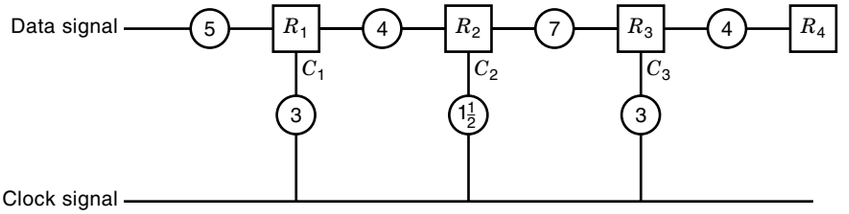

Figure 8. Example of applying localized negative clock skew to synchronous circuit.

data paths particularly susceptible to negative clock skew is to not scale the clock distribution lines. Svensson and Afghahi (47) show that by using courser than ordinary lines for global clock distribution, 20 mm wide chip sizes with CMOS circuits scaled to 0.3 μm 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.

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 (29–31,36,41,48). 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 (36), deskewing data pulses (38), cycle stealing (48,49), useful clock skew (45), and prescribed skew (50).

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. Because a particular clock signal synchronizes a register that 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 ith path results in a positive clock skew for the preceding path, which may then establish the new upper limit for the system clock frequency.

An Example of Applying Localized Negative Clock Skew to Synchronous Circuits. Consider the nonrecursive synchronous circuit shown in Fig. 8 where the horizontal circles represent logic elements with logic delays and the vertical circles represent clock delays. Because 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, then 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 $|T_{Skew}| < T_{PD}$ ($|-1.5$ ns$| < 9$ 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 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 caused by 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.

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 VLSI circuits. The requirement of distributing a tightly controlled clock signal to each synchronous register on a large nonredundant hierarchically structured VLSI circuit (an example floorplan is shown in Fig. 9) within specific temporal bounds is 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 de-

Table 1. Performance Characteristics of Circuit of Fig. 8 without and with Localized Clock Skew

Local Data Path	T_{PD} (min) with Zero Skew	T_{Ci}	T_{Cf}	T_{Skew}	T_{PD} (min) with Nonzero 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.

veloped. The most common and general approach to equipotential clock distribution is the use of buffered trees, which are discussed in the subsection entitled "Buffered Clock Distribution Trees." 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 the subsection entitled "Symmetric H-Tree Clock Distribution Networks." In developing structured custom VLSI circuits, such as the floorplan pictured in Fig. 9, specific circuit design techniques are used to control the delays within the clock distribution network. One important compensation technique is described in the subsection entitled "Compensation Techniques for Controlling Clock Skew." Low power design techniques are an area of significant currency and importance. Some recent efforts to reduce the power dissipated within the clock distribution network are reviewed in the subsection entitled "Design of Low-Power Clock Distribution Networks."

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; it is illustrated in Fig. 10. Occasionally, a mesh version of the clock tree structure is used. As such, shunt paths are placed farther down the clock distribution network to minimize the interconnect resistance within the clock tree. This mesh structure effectively places the branch resistances in parallel, minimizing both the clock delay and the clock skew. An example of this mesh structure is described and illustrated in the subsection entitled "The DEC 64 Bit Alpha Microprocessor." The mesh version of the clock tree is considered in this article as an extended version of the standard, more commonly used clock tree depicted in Fig. 10. The clock distribution network is typically organized as a rooted tree structure (31,51), as illustrated in Figs. 8 and 9, and is often called a clock tree (31).

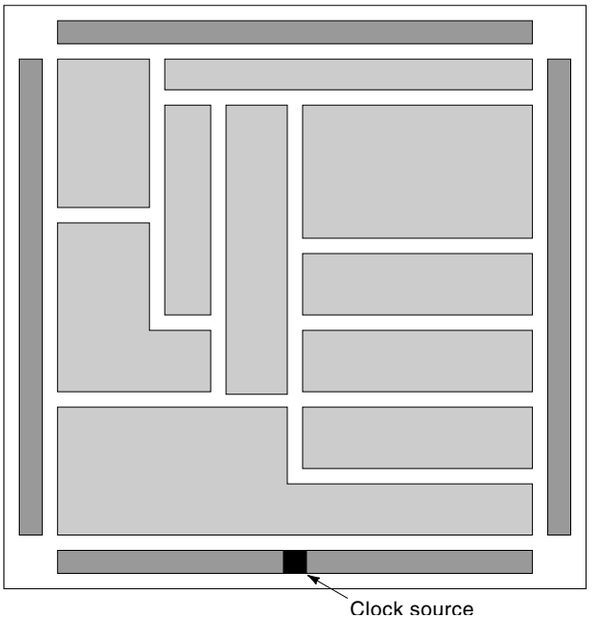

Figure 9. Floorplan of structured custom VLSI circuit requiring synchronous clock distribution.

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 enough 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 (35). A three-level buffer clock distribution network using this strategy is shown in Fig. 11. In this approach, a single buffer drives multiple clock paths (and buffers). The number of buffer stages between the clock source and each clocked register depends on the total capacitive loading, in the form of registers and interconnect, and the permissible clock skew (52). It is worth noting that the buffers are a primary source of the total clock skew within a well-balanced clock distribution network because 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)

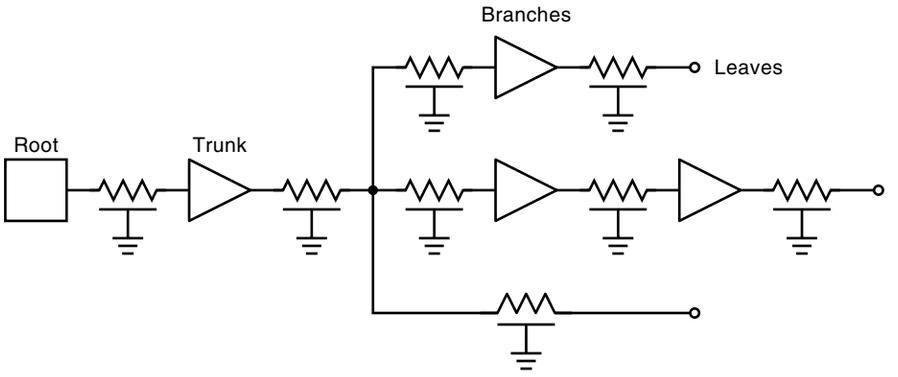

Figure 10. Attributes of tree structure of a clock distribution network.

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 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 the section on timing constraints caused by clock skew, 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.

Symmetric H-Tree Clock Distribution Networks

Another approach for distributing clock signals, a subset of the distributed buffer approach depicted in Fig. 10, uses a hierarchy of planar symmetric H-tree or X-tree structures (see Fig. 12) (45,53,54) to ensure zero clock skew 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 then continues through several layers 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 that 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 caused by 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 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 (45,53–55) and four times the impedance for an X-tree structure.

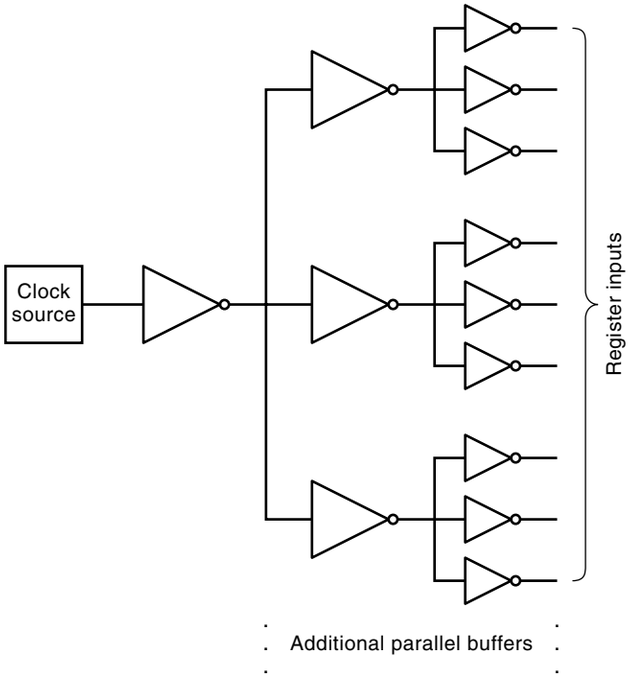

Figure 11. Three-level buffer tree-structured clock distribution network.

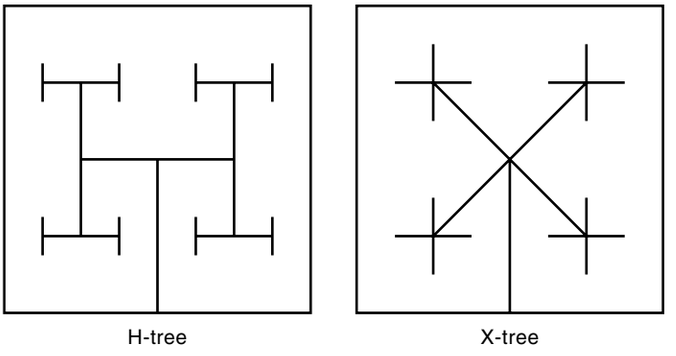

Figure 12. Symmetric H-tree and X-tree clock distribution networks.

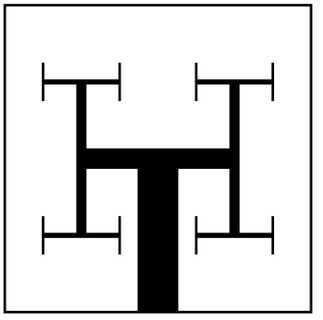

Figure 13. Tapered H-tree clock distribution network.

This tapered H-tree structure is illustrated in Fig. 13:

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

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 be routed in both the vertical and horizontal directions. For a standard two-level metal CMOS process, this creates added difficulty in routing the clock lines without using either resistive interconnect or multiple high resistance vias between the two metal lines. This 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 because the total wire length tends to be much greater (56). 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, because clock skew affects only 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.

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 that are irregular in nature, such as those pictured in Figs. 9 and 14. In these types of systems, buffered tree topologies

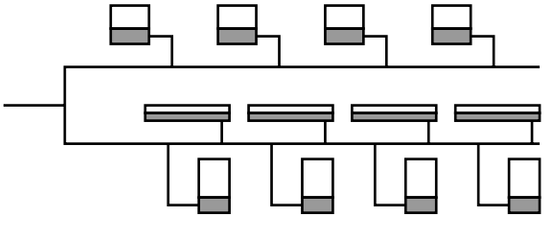

Figure 14. Trunk-structured clock distribution network for structured custom VLSI circuit.

integrated with structured custom design methodologies (57) 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.

Compensation Techniques for Controlling Clock Skew

One structured custom approach, oriented for hierarchical VLSI-based circuits, uses compensation techniques to minimize the variation of interconnect impedances and capacitive loads between clock signal paths (35,38,58,59). A general schematic of a clock distribution network is shown in Fig. 15, 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 comparable to or greater than the interconnect resistance seen at the buffer output. This 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 (45,60). 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 (38) or geometrically sized transistor widths (35) 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, whereas 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 intercon-

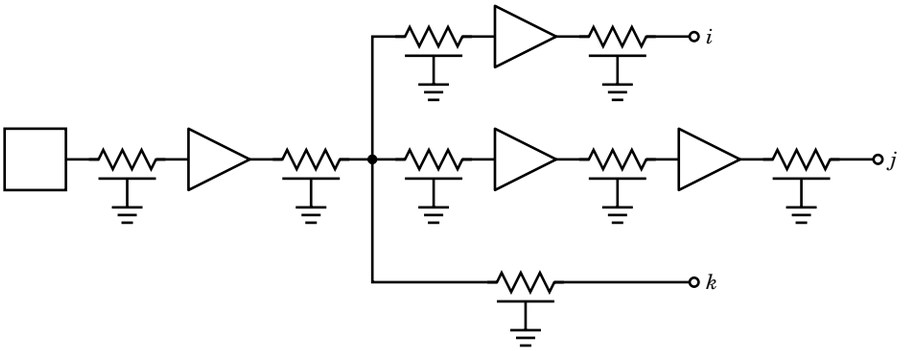

Figure 15. Tree-structured clock distribution network with cascaded buffers and distributed *RC* interconnect impedances.

nect 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 between 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. 16). 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 and 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 internal functional block clock skew should be specified and controlled using the same hierarchical clock distribution strategy as is used at the global VLSI system level (35).

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 is a result of improved partitioning of the *RC* loads. Because 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 are 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

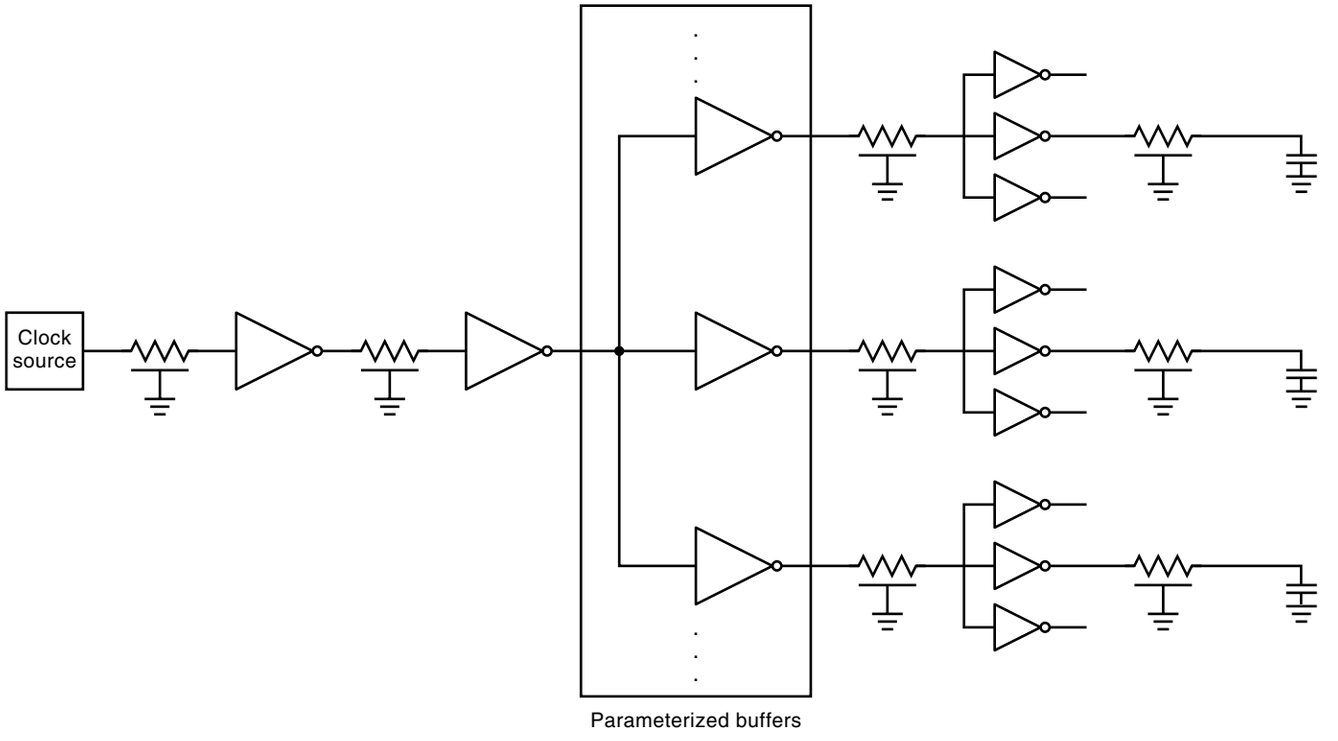

Figure 16. Parameterized buffers integrated into a global clock distribution network to control local clock skew (35).

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 very 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 (35).

Several specific examples of clock distribution networks are discussed in the literature (e.g., 1,11,35,52,59,61,62). Some examples of clock distribution networks applied to high-speed circuits are described in the section entitled "Directions for Future Research in the Design of Clock Distribution Networks." 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 determine 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 (63). Minimal work exists, however, in developing circuit procedures and algorithms for automating the circuit design of clock distribution networks in structured custom VLSI circuits (64–67). 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 that required when calculating absolute delay values (as in the delay of a clock path).

Design of Low-Power Clock Distribution Networks (68–70)

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 (69,71).

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 (68) 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. (69) describe a similar strategy in which the clock signals operate also 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 because, 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% to 25% in power dissipated within the clock tree are reported with no degradation in clock frequency (70). The development of design strategies for minimizing the power 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.

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 (72–77) to the automated layout of clock distribution nets for application to gate array and standard cell-based integrated circuits (45,50,61,78–108). The area of automated layout and the integration of the effects of clock distribution into behavioral synthesis methodologies are described in this section.

Automated Layout of Clock Distribution Networks

A second research path in the area of automated layout has been the development of algorithms that carefully control the variations in delay between clock signal net length so as to minimize clock skew (e.g., 50,82,85–90,100). 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 so 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

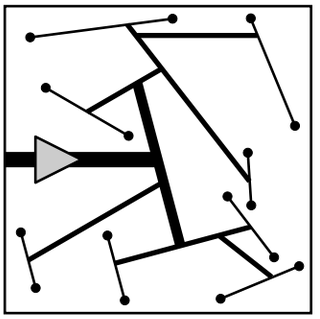

Figure 17. Geometric matching to create zero clock skew.

Fig. 17. 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 (108–110).

Some early clock-routing algorithms (82,88,89) 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 Refs. 108–110, 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 that 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 (108). This occurs because 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 (111–113), 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

$$T_{Di} = \sum_k R_{ki} C_k \tag{9}$$

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 because 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. Other, more subtle considerations, such as bias-dependent loading and varying waveform shapes, must also be considered. The primary focus of the existing research into the automatic layout of clock distribution networks has been placed on minimizing total wire length, metal-to-metal contacts, and crossovers, as well as attaining zero system-wide clock skew (assuming nominal conditions).

Integration of Clock Distribution into Behavioral Synthesis

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 Refs. 64–67, 72–77, 114. 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 Refs. 64–66 and 115–123, 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 in more detail later. 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 Refs. 64 and 116. Algorithms are developed for converting the clock skew information into path-specific clock delays. With this information and some 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 that satisfy the individual branch delays are synthesized. Circuit techniques to implement the clock tree delay elements are discussed further in Refs. 65, 66, 116, and 123. Finally, a variety of techniques exist to lay out the clock distribution trees. This work represents early research in the development of a systematic methodology for synthesizing tree-structured clock distribution networks that contain distributed cascaded buffers and, furthermore, exploit nonzero localized clock skew.

In Refs. 73–75 and 124, a delay model characterizing the timing components of a local data path, similar to Eqs. (1) and (2), 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 Refs. 74, 75, and 124, 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. Because 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 (36). As Fishburn mentions in Ref. 36, because both methods are linear, "it is likely that efficient procedures could be given for optimizing systems by jointly considering both sets of variables."

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 the subsection entitled "Design of Process-Insensitive Clock Distribution Networks." Techniques for calculating clock skew are summarized in the section entitled "Timing Constraints Caused by Clock Skew" as well as in the subsection entitled "Deterministic Models for Estimating Clock Skew."

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 Eqs. (5) and (6) are satisfied. Previous research (116,121) 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 Ref. 121 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. 18. When $T_{\text{Skew}}(i, f) \in [d(i, f), T_{\text{CP}} - D(i, f)]$—as shown in Fig. 18—Eqs. (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 an important opportunity to improve the tolerance of a synchronous digital system to process parameter variations. By selecting a value of clock skew toward the center of the permissible range, the tolerance of the local data path to process parameter and environmental delay variations is improved. 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 the following subsection.

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 can limit 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 that mitigate the effect of process tolerances on the design of clock distribution networks while maintaining an effective methodology for designing these networks have been developed.

Threshold Tracking to Control Clock Skew. An important circuit design technique for making clock distribution networks less process sensitive is described by Shoji (125). 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 because the cause of a positive threshold voltage shift is one type of Metal Oxide Semiconductor (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

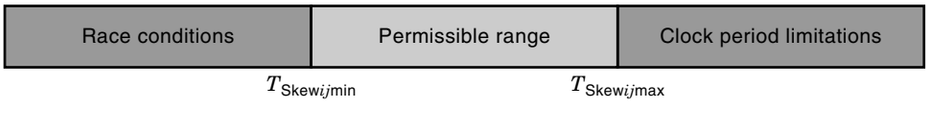

Figure 18. Permissible range of the clock skew of a local data path, $R_i \Rightarrow R_f$. A clock hazard exists if $T_{\text{Skew}}(i, f) \notin [d(i, f), T_{\text{CP}} - D(i, f)]$.

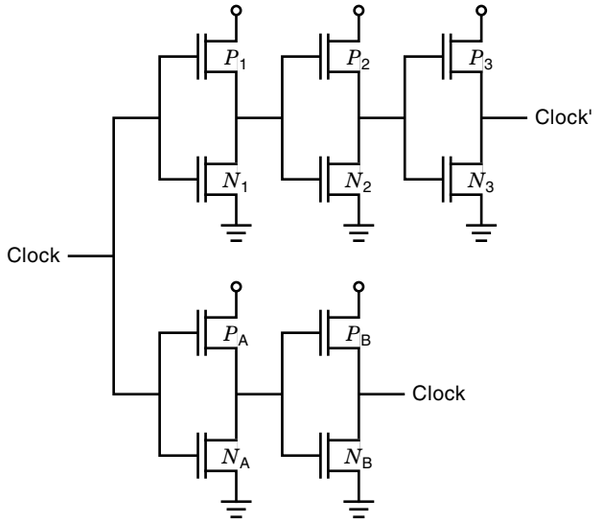

Figure 19. Elimination of process-induced clock skew by scaling matched transistor types.

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.

1. Match the sum of the pull-up delays of the *p*-channel Metal Oxide Semiconductor Field-Effect Transistor (MOSFET) with the pull-up delays of any related clock signal paths.
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 using this technique is shown in Fig. 16. 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 Complementary Metal Oxide Semiconductor (CMOS) circuit design, the transistor widths are adjusted to satisfy

$$T_A + T_B = T_1 + T_2 + T_3 \tag{10}$$

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, Eq. (10) will no longer be satisfied, and a large skew will develop between the outputs, clock and clock′, shown in Fig. 19. Instead of designing the circuit in Fig. 19 to satisfy Eq. (10), the circuit is designed to satisfy Eqs. (11) and (12). Thus, the primary objective of Eq. (10) (zero clock skew) is maintained while the added more stringent design constraint makes the entire clock distribution circuit more tolerant to process variations. This increased tolerance occurs because by satisfying both Eqs. (11) and (12), 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:

$$T_A = T_1 + T_3 \tag{11}$$

$$T_B = T_2 \tag{12}$$

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 (125).

Interconnect Widening to Minimize Clock Skew Sensitivity. As described in the subsection entitled "Automated Layout of Clock Distribution Networks," 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 line widths vary. In order to design these clock nets to be less sensitive to process variations, Pullela et al. (108–110) have developed an automated layout algorithm that widens the clock nets rather than lengthens them while equalizing the line delays. These nets are therefore less sensitive to both under- and over-etching 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 line width 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 line width affects the total path delay more significantly than decreasing the resistance at the leaf node because 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 line width 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 (111–113). 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.

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. 15. 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 Ref. 126, 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 (45,60) 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 (44,45,126)

$$T_{R/F} = 1.02R_{Int}C_{Int} + 2.21(R_{Tr}C_{Int} + R_{Tr}C_{Tr} + R_{Int}C_{Tr}) \quad (13)$$

where R_{Int} and C_{Int} are the resistance and capacitance of the interconnect section, respectively, and R_{Tr} and C_{Tr} are the output on-resistance of the driving buffer and the input load capacitance ($=C_{OX}WL$) 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 (44,45,126)

$$R_O \approx \frac{L/W}{\mu C_{ox}(V_{DD} - V_T)} \quad (14)$$

where μ is the channel mobility, V_{DD} is the power supply voltage, and V_T is the device threshold voltage. Equation (14) is derived from the large signal I–V equation of a MOSFET operating in the saturation region close to the linear region and is accurate for small channel geometries because velocity saturation decreases the quadratic behavior of the MOS device operating in the saturation region. The physical delay model represented by Eqs. (13) and (14) 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 (127). Tsay and Lin (48,49) continue this approach by describing an innovative timing analyzer that 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. Dagenais and Rumin (128,129) present a timing analysis system that 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.

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 Refs. 29, 30, 36, 130–138, 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 (36,134).

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 (36), although the application of localized clock skew to increase the clock frequency and to eliminate race conditions was known previously (34). 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 the section entitled "Timing Constraints Caused by Clock Skew," are eliminated by solving the set of linear inequalities derived from Eqs. (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 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 (36). Syzmanski 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 (133).

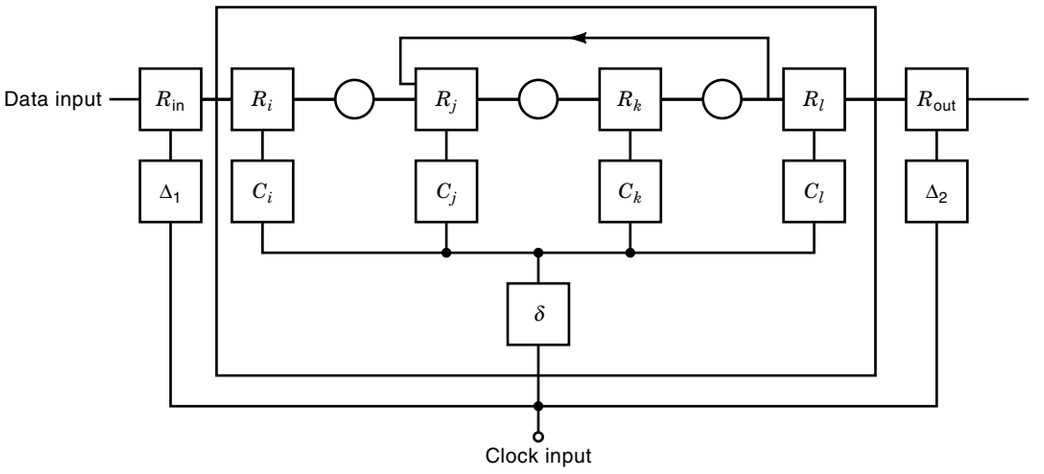

Figure 20. Multistage block diagram of a synchronous digital system.

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. 20. 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, $T_{\text{Logic(max)}}$ and $T_{\text{Logic(min)}}$. 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. 20 considers only 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.

Off-Chip Clock Skew. The circuit model in Fig. 20 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, because the circuit is assumed to be a fully synchronous system. This relationship is represented by

$$T_{\text{Skewin,out}} = T_{\text{Skewin},i} + T_{\text{Skew}i,j} + \cdots + T_{\text{Skew}l,\text{out}} = 0 \qquad (15)$$

Therefore, to satisfy Eq. (15), in Fig. 20, $\Delta_1 = \Delta_2 = \Delta$.

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 the subsection entitled "Minimum Data Path/Clock Skew Constraint Relationship," 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 in 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 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 the subsection entitled "Symmetric H-Tree Clock Distribution Networks"). Observe that restricting the off-chip clock skew to zero does not preclude the circuit from being optimized with localized clock skew. The primary effect is that the performance improvement is less than that obtained without this constraint.

Observe that Eq. (15) 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).

Global and Local Timing Constraints. As described earlier, in order to avoid either type 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, Eq. (5) must be satisfied. To avoid race conditions between two sequentially adjacent registers, R_i and R_j, Eq. (6) must be satisfied.

The system-wide clock period is minimized by finding a set of clock skew values that satisfy Eqs. (5) and (6) for each local data path and Eq. (15) 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. Consider the following.

Minimize T_{CP} subject to the local and global timing constraints:

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

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

$$T_{CP} \ge T_{PD} + T_{Skew_{ij}}$$

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

$$T_{Skewin,i} + T_{Skew_{i,j}} + T_{Skew_{j,k}} + \cdots + T_{Skew_{n,out}} = 0$$

An Example of Determining the Optimal Clock Schedule of a Pipelined System. An example of determining the minimum clock period of a multistage system with feedback paths is the circuit illustrated in Fig. 21. This example is similar to that used in Ref. 23, 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 Ref. 23, all the register timing parameters are assumed to be zero for simplicity.

The linear program that gives the minimum clock period and the optimal clock skew schedule for the circuit shown in Fig. 21 follows.

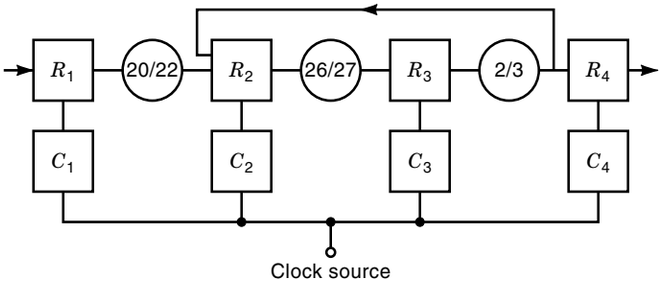

Figure 21. Example of synchronous circuit with feedback.

Minimize T_{CP} subject to

$$\begin{aligned}
R_1 - R_2: &\quad C_1 - C_2 = T_{Skew12} \ge -20\,\text{ns} \\
&\quad T_{Skew12} - T_{CP} \le -22\,\text{ns} \\
R_2 - R_2: &\quad C_2 - C_3 = T_{Skew23} \ge -26\,\text{ns} \\
&\quad T_{Skew23} - T_{CP} \le -27\,\text{ns} \\
R_3 - R_4: &\quad C_3 - C_4 = T_{Skew34} \ge -2\,\text{ns} \\
&\quad T_{Skew34} - T_{CP} \le -3\,\text{ns} \\
R_3 - R_2: &\quad C_3 - C_2 = T_{Skew32} \ge -2\,\text{ns} \\
&\quad T_{Skew32} - T_{CP} \le -3\,\text{ns} \\
R_1 - R_4: &\quad T_{Skew12} + T_{Skew23} + T_{Skew34} = 0
\end{aligned}$$

where the optimal clock schedule is

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

If zero clock skew between off-chip registers is not considered, the minimum clock period is T_{CP} = 15 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_{CP} = 27 ns as a result of the worst case path delay of the local data path between registers R_2 and R_3.

EXAMPLE IMPLEMENTATIONS OF CLOCK DISTRIBUTION NETWORKS

A number of interesting and innovative examples of high-performance fully synchronous clock distribution networks have been developed for highly specialized and high-performance commercial processors and have been described in the literature. These VLSI-based systems required an unusual combination of methodologies and practices commensurate with large design teams while maintaining the localized circuit optimization requirements important to high-speed VLSI circuit design. The design of the clock distribution network used to synchronize some well-known industrial circuit examples are discussed in this section.

The Bell Telephone WE32100 32 Bit Microprocessor (139,140)

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. Because performance was of fundamental importance, significant attention was placed on synchronization, particularly the design of the global clock distribution network. In 1982, Shoji (139) described the clock distribution of the BELLMAC-32A, a 146,000 transistor central processing unit (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 ns $\pm$ 3.5 ns defining the maximum permissible clock skew. A four-phase clocking strategy is used; 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 liens 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 ±3.5 ns. It is worth noting that because of 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 (140) 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 uses 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 the subsection entitled "Compensation Techniques for Controlling Clock Skew" (35) and depicted in Fig. 14 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 the subsection entitled "Design of Process-Insensitive Clock Distribution Networks." 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 (125).

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. Because individual integrated circuits are processed in different wafer lots, the device parameters may differ; 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 (140).

The DEC 64 Bit Alpha Microprocessor (71,141)

An important application area for high-speed clock distribution networks is the development of high speed microprocessors. The performance of these circuits is often limited by the clocking strategy used in their implementation. The DEC Alpha chip currently represents a significant milestone in microprocessor technology. The VLSI circuit operates above 200 MHz with a 3.3 V power supply implemented in 0.75 μm CMOS three-level metal technology. A clock period of 5 ns must be satisfied for each local data path. Therefore, the clock skew should be of very small magnitude (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 circuits and interconnects is required in order to design and analyze this type of high-speed system. The Alpha microprocessor contains 1.68 million transistors and supports a fully pipelined 64 bit data structure. The functional attributes of the microprocessor are described in greater detail in Ref. 71 because the focus herein is on the clocking strategy used within the circuit.

In designing this high-speed 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 trilevel metal process, as is the power distribution, because the third layer of metal is thicker and wider (7.5 μm pitch with contacts 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 chip. 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 caused by the 63,000 transistor gates being driven by the clock distribution system. The distribution of the loads is nonsymmetric, 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 a tree as shown in Fig. 22 (142). 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 chip is to permit only positive clock skew, thereby assuring that no catastrophic race conditions induced by neg-

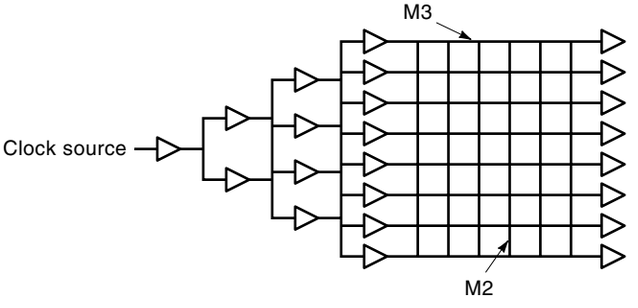

Figure 22. Clock distribution network of DEC Alpha microprocessor. Note the mesh structure within the clock tree.

ative 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 its periphery. By carefully monitoring the design of this clock distribution methodology, the likelihood of developing a catastrophic amount of negative clock skew (i.e., $|T_{\text{Skew}}| > T_{\text{PD(min)}}$) is minimized.

8 Bit × 8 Bit Pipelined Multiplier (11,56,143–146)

Another application area that requires sophisticated clock distribution is heavily pipelined digital signal processors (DSP), such as Finite Impulse Response (FIR)/Infinite Impulse Response (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 system clock frequency (29–31).

The multiplier function is a good example of a complex circuit element capable of significant improvement in clock rate with high levels of pipelining. Because the data flow is nonrecursive, fewer pipeline registers are required as compared to those structures that 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 subsection, 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 ($T_{\text{C-Q}}$ and $T_{\text{Set-up}}$), permitting very high multiplication throughput. Specialized architectures, which are beyond the scope of this article, such as carry save addition, are used to improve the throughput of these VLSI-based multipliers (11,56,143–146).

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 data flow. As described by Hatamian and Cash (143,144) and pictured in Fig. 23, assuming that 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 the subsection entitled "Minimum Data Path/Clock Skew Constraint Relationship" and Eq. (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.

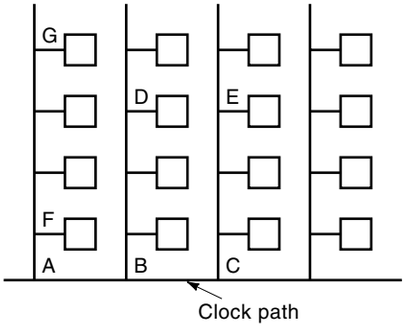

Figure 23. Clock distribution network of pipelined multiplier array (143,144).

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 that 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 × 8 bit multiplier described in Refs. 143 and 144, which is implemented in a 2.5 μm 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 Ref. 145. Also an 8 bit × 8 bit multiplier, the multiplier described in Ref. 145 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 1 bit addition, the multiplier architecture is very similar to that described in Ref. 143. 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 because 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 × 8 bit multiplier (56). 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 Refs. 143–145. The circuit uses a single-phase clocking scheme (147). 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 10 of these buffers drive the registers (i.e., the leaves of the tree). The maximum skew measured between two clock liens is 150 ps (56).

These three multiplier examples are intended to provide some insight into distributing clock signals within highly ar-

rayed 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 MHz) resulting from heavy pipelining, make the multiplier an important class of VLSI system that requires highly specialized clock distribution networks.

DIRECTIONS FOR FUTURE RESEARCH IN THE DESIGN OF CLOCK DISTRIBUTION NETWORKS

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

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, improved delay models that account for nonlinear active transistor behavior, the use of localized clock skew to increase circuit speed, and the integration of *RC* 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.

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. As functional clock periods approach and exceed one ns (i.e., one GHz clock frequency), variations in delay of tens of picoseconds could significantly degrade the performance and reliability of these high-speed synchronous systems. Variations of this magnitude are quite common in modern semiconductor technologies. This topic is of immediate importance to the existing high-speed processor community.

Design of Microwave Frequency Clock Distribution Networks

As system-wide clock frequencies increase beyond gigahertz levels, transmission line effects will begin to influence the performance characteristics of clock distribution networks. Models of interconnect sections will require the inclusion of accurate inductive elements. These microwave effects will become more apparent as ultra-high-speed digital technologies, such as Heterostructure Bipolar Transistors (HBT) and the superconductive digital electronic technology, Single Flux Quantum (SFQ) logic (148–153), which operate well into the 10 GHz to 100 GHz frequency range, become more readily available.

System Issues on 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 multi-chip modules (154), and wafer scale integration all require specialized synchronization strategies. Related systems issues, such as ultra-low power circuits which require ultra-low power clock distribution networks (68–70), are becoming increasingly important. Improving the power dissipation characteristics of clock distribution networks is particularly important because these networks dissipate a large portion of the total system-wide power budget (as much as 40% of the total power dissipated in large microprocessors (71)). Topics, such as distributing small differential voltage signals and task monitoring (or power management) strategies, are important areas of research in low-power and low-voltage clock distribution network design.

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. There is little research that describes how best to debug high-speed clock distribution networks as well as how to best test these networks in a production environment.

SUMMARY AND CONCLUSIONS

All electronic systems are fundamentally asynchronous in nature; by carefully inserting 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 systems can be applied, easing the timing constraints on data flow, albeit requiring a clock distribution network to provide the synchronizing reference signal. By synchronizing with distributed clock signals, 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 that operate at performance levels that would be otherwise unattainable without requiring significant architectural or technological improvements can be developed.

It is often noted that the design of the clock distribution network represents the fundamental circuit limitation to performance in high-speed synchronous digital systems. The local data path-dependent nature of clock skew, rather than its 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 emphasis placed on asynchronous systems. 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 applied to the design of clock distribution networks to further enhance system performance.

In this article on synchronous clock distribution, timing relationships are examined and are used to constrain the timing characteristics of the overall system. Various architectures and applications are considered, and circuit strategies for distributing the clock signals are offered. It is the intention of this article to integrate these various subtopics and to provide some sense of cohesiveness to the field of clocking and, specifically, clock distribution networks.

BIBLIOGRAPHY

1. F. Anceau, A synchronous approach for clocking VLSI systems, *IEEE J. Solid-State Circuits,* **SC-17**: 51–56, 1982.
2. M. A. Franklin and D. F. Wann, Asynchronous and clocked control structures for VLSI based interconnection networks, *Proc. 9th Annu. Symp. Comput. Archit.,* 1982, pp. 50–59.
3. D. Wann and M. Franklin, Asynchronous and clocked control structures for VLSI based interconnection networks, *IEEE Trans. Comput.,* **C-32**: 284–293, 1983.
4. S. Dhar, M. Franklin, and D. Wann, Reduction of clock delays in VLSI structures, *Proc. IEEE Int. Conf. Comput. Des.,* 1984, pp. 778–783.
5. S. Unger and C.-J. Tan, Optimal clocking schemes for high speed digital systems, *Proc. IEEE Int. Conf. Comput. Des.,* 1983, pp. 366–369.
6. J. Beausang and A. Albicki, A method to obtain an optimal clocking scheme for a digital system, *Proc. IEEE Int. Conf. Comput. Des.,* 1983, pp. 68–72.
7. S. H. Unger and C.-J. Tan, Clocking schemes for high-speed digital systems, *IEEE Trans. Comput.,* **C-35**: 880–895, 1986.
8. D. Noice, R. Mathews, and J. Newkirk, A clocking discipline for two-phase digital systems, *Proc. IEEE Int. Conf. Circuits Comput.,* 1982, pp. 108–111.
9. C. Svensson, Signal resynchronization in VLSI system, *Integr. VLSI J.,* **4** (1): 75–80, 1986.
10. M. S. McGregor, P. B. Denyer, and A. F. Murray, A single-phase clocking scheme for CMOS VLSI, *Proc. Stanford Conf. Adv. Res. Very Large Scale Integr.,* 1987, pp. 257–271.
11. M. Hatamian and G L. Cash, Parallel bit-level pipelined VLSI designs for high-speed signal processing, *Proc. IEEE,* **75**: 1192–1202, 1987.
12. M. Hatamian et al., Fundamental interconnection issues, *AT& T Tech. J.,* **66** (4): 13–30, 1987.
13. K. D. Wagner, Clock system design, *IEEE Des. Test Comput.,* **5** (5): 9–27, 1988.
14. A. F. Champernowne et al., Latch-to-latch timing rules, *IEEE Trans. Comput.,* **C-39**: 798–808, 1990.
15. D. G. Messerschmitt, Synchronization in digital system design, *IEEE J. Sel. Areas Commun.,* **8**: 1404–1419, 1990.
16. M. Afghahi and C. Svensson, Performance of synchronous and asynchronous schemes for VLSI systems, *IEEE Trans. Comput.,* **C-41**: 858–872, 1992.
17. V. Chi, Designing salphasic clock distribution systems, *Proc. Symp. Integr. Syst.,* 1993, pp. 219–233.
18. V. L. Chi, Salphasic distribution of clock signals for synchronous systems, *IEEE Trans. Comput.,* **C-43**: 597–602, 1994.
19. M. C. Papaefthymiou and K. H. Randall, Edge-triggering vs. two-phase level-clocking, *Proc. Symp. Integr. Syst.,* 1993, pp. 201–218.
20. W. K. C. Lam, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, Valid clocking in wavepipelined circuits, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1992, pp. 124–131.
21. D. A. Joy and M. J. Ciesielski, Clock period minimization with wave pipelining, *IEEE Trans. Comput.-Aided Des.,* **CAD-12**: 461–472, 1993.
22. C. T. Gray, W. Liu, and R. K. Cavin, III, Timing constraints for wave-pipelined systems, *IEEE Trans. Comput.-Aided Des.,* **CAD-13**: 987–1004, 1994.
23. X. Zhang and R. Sridhar, Synchronization of wave-pipelined circuits, *Proc. IEEE Int. Conf. Comput. Des.,* 1994, pp. 164–167.
24. E. G. Friedman, Clock distribution design in VLSI circuits—an overview, *Proc. IEEE Int. Symp. Circuits Syst.,* 1993, pp. 1475–1478.
25. K. A. Sakallah et al., Synchronization of pipelines, *IEEE Trans. Comput.-Aided Des.,* **CAD-12**: 1132–1146, 1993.
26. K. Wagner and E. McCluskey, *Tuning, Clock Distribution, and Communication in VLSI High-Speed Chips,* Stanford, CA: Stanford Univ., 1984, CRC Tech. Rep. 85-5.
27. K. D. Wagner, *A Survey of Clock Distribution Techniques in High-Speed Computer Systems,* Stanford, CA: Stanford Univ., 1986, CRC Tech. Rep. No. 86-20.
28. E. G. Friedman, Latching characteristics of a CMOS bistable register, *IEEE Trans. Circuits and Systems I, Fundam. Theory Appl.,* **CAS-40**: 902–908, 1993.
29. E. G. Friedman and J. H. Mulligan, Jr., Clock frequency and latency in synchronous digital systems, *IEEE Trans. Signal Process.,* **SP-39**: 930–934, 1991.
30. E. G. Friedman and J. H. Mulligan, Jr., Pipelining of high performance synchronous digital systems, *Int. J. Electron.,* **70** (5): 917–935, 1991.
31. E. G. Friedman and J. H. Mulligan, Jr., Pipelining and clocking of high performance synchronous digital systems, in M. A. Bayoumi and E. E. Swartzlander, Jr., (eds.), *VLSI Signal Processing Technology,* Norwell, MA: Kluwer, 1994, pp. 97–133.
32. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms,* Cambridge, MA: MIT Press, 1989.
33. D. B. West, *Introduction to Graph Theory,* Upper Saddle River, NJ: Prentice-Hall, 1996.
34. M. Hatamian, Understanding clock skew in synchronous systems, in S. K. Tewksbury, B. W. Dickinson, and S. C. Schwartz (eds.), *Concurrent Computations Algorithms, Architecture and Technology,* New York: Plenum, 1988, pp. 87–96.
35. 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,* **SC-21**: 240–246, 1986.
36. J. P. Fishburn, Clock skew optimization, *IEEE Trans. Comput.,* **C-39**: 945–951, 1990.
37. J. Alves Marques and A. Cunha, Clocking of VLSI circuits, in Randell and Treleaven (eds.), *VLSI Architecture,* Englewood Cliffs, NJ: Prentice-Hall, 1983, pp. 165–178.
38. Method of deskewing data pulses, *IBM Tech. Disclosure Bull.,* **28** (6): 2658–2659, 1985.
39. 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, *Proc. 1st Int. Conf. Syst. Integr.,* 1990, pp. 550–558.

40. K. A. Sakallah, T. N. Mudge, and O. A. Olukoton, Analysis and design of latch-controlled synchronous digital circuits, *Proc. ACM / IEEE Des. Autom. Conf.,* 1990, pp. 111–117.
41. E. G. Friedman, *Performance limitations in synchronous digital systems,* Ph.D. dissertation, Irvine: Univ. California, 1989.
42. T. W. Williams and K. P. Parker, Design for testability—A survey, *Proc. IEEE,* **71**: 98–112, 1983.
43. R. H. Dennard et al., Design of ion-implanted MOSFET's with very small physical dimensions, *IEEE J. Solid-State Circuits,* **SC-9**: 256–268, 1974.
44. H. B. Bakoglu and J. D. Meindl, Optimal interconnection circuits for VLSI, *IEEE Trans. Electron Devices,* **ED-32**: 903–909, 1985.
45. H. B. Bakoglu, *Circuits, Interconnections, and Packaging for VLSI,* Reading, MA: Addison-Wesley, 1990.
46. C. V. Gura, *Analysis of Clock Skew in Distributed Resistive-Capacitive Interconnects,* Urbana: Univ. Illinois, 1987, SRC Tech. Rep. No. T87053.
47. C. Svensson and M. Afghahi, On *RC* line delays and scaling in VLSI systems, *Electron. Lett.,* **24** (9): 562–563, 1988.
48. I. Lin, J. A. Ludwig, and K. Eng, Analyzing cycle stealing on synchronous circuits with level-sensitive latches, *Proc. ACM / IEEE Des. Autom. Conf.,* 1992, pp. 393–398.
49. R.-S. Tsay and I. Lin, Robin Hood: A system timing verifier for multi-phase level-sensitive clock designs, *Proc. IEEE Int. Conf. ASICs,* 1992, pp. 516–519.
50. T.-H. Chao et al., Zero skew clock routing with minimum wirelength, *IEEE Trans. Circuits Syst. II, Analog Digital Signal Process.,* **CAS-39**: 799–814, 1992.
51. S. Y. Kung, *VLSI Array Processors,* Englewood Cliffs, NJ: Prentice-Hall, 1988.
52. D. Mijuskovic, Clock distribution in application specific integrated circuits, *Microelectron. J.,* **18** (4): 15–27, 1987.
53. 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, *Proc. IEEE Int. Conf. Comput. Des.,* 1986, pp. 118–122.
54. M. Nekili et al., Logic-based H-trees for large VLSI processor arrays: A novel skew modeling and high-speed clocking method, *Proc. 5th Int. Conf. Microelectron.,* 1993, pp. 1–4.
55. D C. Keezer and V. K. Jain, Clock distribution strategies for WSI: A critical survey, *Proc. IEEE Int. Conf. Wafer Scale Integr.,* 1991, pp. 277–283.
56. D. Somasekhar and V. Visvanathan, A 230-MHz half-bit level pipelined multiplier using true single-phase clocking, *IEEE Trans. Very Large Scale Integr. VLSI Syst.,* **VLSI-1**: 415–422, 1993.
57. D. D. Gajski, *Silicon Compilation,* Reading, MA: Addison-Wesley, 1988.
58. B. Wu and N. A. Sherwani, Effective buffer insertion of clock tree for high-speed VLSI circuits, *Microelectron. J.,* **23** (4): 291–300, 1992.
59. S. Padin, Scheme for distributing high-speed clock signals in a large digital system, *Electron. Lett.,* **25** (2): 92–93, 1989.
60. S. Dhar and M. A. Franklin, Optimum buffer circuits for driving long uniform lines, *IEEE J. Solid-State Circuits,* **SC-26** (1): 32–40, 1991.
61. G. M. Blair, Skew-free clock distribution for standard-cell VLSI designs, *IEE Proc. G,* **139** (2): 265–267, 1992.
62. R. Maini, J. McDonald, and L. Spangler, A clock distribution circuit with a 100 PS skew window, *Proc. IEEE Bipolar Circuits Technol. Meet.,* 1987, pp. 41–43.
63. G. Yacoub et al., A system for critical path analysis based on back annotation and distributed interconnect impedance models, *Microelectron. J.,* **19** (3): 21–30, 1988.
64. J. L. Neves and E. G. Friedman, Topological design of clock distribution networks based on non-zero clock skew specifications, *Proc. 36th Midwest Symp. Circuits Syst.,* 1993, pp. 468–471.
65. J. L. Neves and E. G. Friedman, Circuit synthesis of clock distribution networks based on non-zero clock skew, *Proc. IEEE Int. Symp. Circuits Syst.,* 1994, pp. 4.175–4.178.
66. J. L. Neves and E. G. Friedman, Synthesizing distributed buffer clock trees for high performance ASICs, *Proc. IEEE Int. Conf. ASICs,* 1994, pp. 126–129.
67. J. Chung and C.-K. Cheng, Optimal buffered clock tree synthesis, *Proc. IEEE Int. Confer. ASICs,* 1994, pp. 130–133.
68. E. De Man and M. Schobinger, Power dissipation in the clock system of highly pipelined ULSI CMOS circuits, *Proc. Int. Workshop Low Power Des.,* 1994, pp. 133–138.
69. H. Kojima, S. Tanaka, and K. Sasaki, High-swing clocking scheme for 75% power saving in clocking circuitry, *Proc. IEEE Symp. Very Large Scale Integr. Circuits,* 1994, pp. 23–24.
70. J. L. Neves and E. G. Friedman, Minimizing power dissipation in non-zero skew-based clock distribution networks, *Proc. IEEE Int. Symp. Circuits Syst.,* 1995, pp. 1576–1579.
71. D. W. Dobberpuhl et al., A 200-MHz 64-b dual-issue CMOS microprocessor, *IEEE J. Solid-State Circuits,* **SC-27**: 1555–1565, 1992.
72. N. Park and A. Parker, Synthesis of optimal clocking schemes, *Proc. 22nd ACM / IEEE Des. Autom. Conf.,* 1985, pp. 489–495.
73. E. G. Friedman, The application of localized clock distribution design to improving the performance of retimed sequential circuits, *Proc. IEEE Asia-Pac. Conf. Circuits Syst.,* 1992, pp. 12–17.
74. T. Soyata, E. G. Friedman, and J. H. Mulligan, Jr., Integration of clock skew and register delays into a retiming algorithm, *Proc. IEEE Int. Symp. Circuits Syst.,* 1993, pp. 1483–1486.
75. T. Soyata and E. G. Friedman, Retiming with non-zero clock skew, variable register, and interconnect delay, *Proc. IEEE Int. Conf. Comput -Aided Des.,* 1994, pp. 234–241.
76. L.-F. Chao and E. H.-M. Sha, Retiming and clock skew for synchronous systems, *Proc. IEEE Int. Symp. Circuits Syst.,* 1994, pp. 1.283–1.286.
77. B. Lockyear and C. Ebeling, The practical application of retiming to the design of high-performance systems, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1993, pp. 288–295.
78. Y. Ogawa et al., Efficient placement algorithms optimizing delay for high-speed ECL masterslice LSI's, *Proc. 23rd ACM / IEEE Des. Autom. Conf.,* 1986, pp. 404–410.
79. S. Boon et al., High performance clock distribution for CMOS ASICs, *Proc. IEEE Custom Integr. Circuits Conf.,* 1989, pp. 15.4.1–15.4.5.
80. A. Chao, Clock tree synthesis for large gate arrays, *High Perform. Syst.,* **10**: 32, 1989.
81. D. Y. Montuno and R. C. S. Ma, A layout methodology for the synthesis of high speed global clock nets, *Proc. IEEE Custom Integr. Circuits Conf.,* 1992, pp. 28.4.1–28.4.4.
82. P. Ramanathan and K. G. Shin, A clock distribution scheme for non-symmetric VLSI circuits, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1989, pp. 398–401.
83. K. D. Boese and A. B. Kahng, Zero-skew clock routing trees with minimum wirelength, *Proc. IEEE Int. Conf. ASICs,* 1992, pp. 1.1.1–1.1.5.
84. J. Burkis, Clock tree synthesis for high performance ASICs, *Proc. IEEE Int. Conf. ASICs,* 1991, pp. 9.8.1–9.8.4.
85. T.-H. Chao, Y.-C. Hsu, and J.-M. Ho, Zero skew clock net routing, *Proc. ACM / IEEE Des. Autom. Conf.,* 1992, pp. 518–523.

86. J. Cong, A. B. Kahng, and G. Robins, Matching-based methods for high-performance clock routing, *IEEE Trans. Comput.-Aided Des.,* **CAD-12**: 1157–1169, 1993.
87. M. Edahiro, A clock net reassignment algorithm using Voronoi diagrams, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1990, pp. 420–423.
88. M. A. B. Jackson, A. Srinivasan, and E. S. Kuh, Clock routing for high performance ICs, *Proc. ACM / IEEE Des. Autom. Conf.,* 1990, pp. 573–579.
89. A. B. Kahng, J. Cong, and G. Robins, High-performance clock routing based on recursive geometric matching, *Proc. ACM / IEEE Des. Autom. Conf.,* 1991, pp. 322–327.
90. R. S. Tsay, Exact zero skew, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1991, pp. 336–339.
91. T. Saigo et al., Clock skew reduction approach for standard cell, *Proc. IEEE Custom Integr. Circuits Conf.,* 1990, pp. 16.4.1–16.4.4.
92. P. D. Ta and K. Do, A low power clock distribution scheme for complex IC system, *Proc. IEEE Int. Conf. ASICs,* 1991, pp. 1-5.1–1-5.4.
93. P. R. Mukund and D. W. Bouldin, A graph theoretic approach to the clock distribution problem, *Proc. IEEE Int. Conf. ASICs,* 1991, pp. 7-5.1–7-5.4.
94. J. Cong, A. Kahng, and G. Robins, On clock routing for general cell layouts, *Proc. IEEE Int. Conf. ASICs,* 1991, pp. 14-5.1–14-5.4.
95. N. A. Sherwani and B. Wu, Clock layout for high-performance ASIC based on weighted center algorithm, *Proc. IEEE Int. Conf. ASICs,* 1991, pp. 15-5.1–15-5.4.
96. D. A. Joy and M. J. Ciesielski, Placement for clock period minimization with multiple wave propagation, *Proc. 28th ACM / IEEE Des. Autom. Conf.,* 1991, pp. 640–643.
97. Y-M. Li and M. A. Jabri, A zero-skew clock routing scheme for VLSI circuits, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1992, pp. 458–463.
98. W. Khan, M. Hossain, and N. Sherwani, Zero skew clock routing in multiple-clock synchronous systems, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1992, pp. 464–467.
99. Q. Zhu and W. W-M. Dai, Perfect-balance planar clock routing with minimal path-length, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1992, pp. 473–476.
100. R-S. Tsay, An exact zero-skew clock routing algorithm, *IEEE Trans. Comput.-Aided Des.,* **CAD-12**: 242–249, 1993.
101. W. Khan and N. Sherwani, Zero skew clock routing algorithm for high performance ASIC systems, *Proc. IEEE Int. Conf. ASICs,* 1993, pp. 79–82.
102. W. Khan, S. Madhwapathy, and N. Sherwani, An hierarchical approach to clock routing in high performance systems, *Proc. IEEE Int. Symp. Circuits Syst.,* 1994, pp. 1.467–1.470.
103. S. Lin and C. K. Wong, Process-variation-tolerant zero skew clock routing, *Proc. IEEE Int. Conf. ASICs,* 1993, pp. 83–86.
104. M. Edahiro, A clustering-based optimization algorithm in zero-skew routings, *Proc. ACM / IEEE Des. Autom. Conf.,* 1993, pp. 612–616.
105. N-C. Chou and C-K. Cheng, Wire length and delay minimization in general clock net routings, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1993, pp. 552–555.
106. M. Edahiro, Delay minimization for zero-skew routing, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1993, pp. 563–566.
107. M. Edahiro, An efficient zero-skew routing algorithm, *Proc. 31st ACM / IEEE Des. Autom. Conf.,* 1994.
108. S. Pullela, N. Menezes, and L. T. Pillage, Reliable non-zero clock trees using wire width optimization, *Proc. ACM / IEEE Des. Autom. Conf.,* 1993, pp. 165–170.
109. N. Menezes et al., Skew reduction in clock trees using wire width optimization, *Proc. Custom Integr. Circuits Conf.,* 1993, pp. 9.6.1–9.6.4.
110. S. Pullela et al., Skew and delay optimization for reliable buffered clock trees, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1993, pp. 556–562.
111. W. C. Elmore, The transient response of damped linear networks with particular regard to wideband amplifiers, *J. Appl. Phys.,* **19**: 55–63, 1948.
112. J. Rubinstein, P. Penfield, Jr., and M. A. Horowitz, Signal delay in RC tree networks, *IEEE Trans. Comput.-Aided Des.,* **CAD-2**: 202–211, 1983.
113. J. Vlach et al., Group delay as an estimate of delay in logic, *IEEE Trans. Comput.-Aided Des.,* **CAD-10**: 949–953, 1991.
114. J. Chung and C.-K. Cheng, Skew sensitivity minimization of buffered clock tree, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1994, pp. 280–283.
115. J. L. Neves and E. G. Friedman, Buffered clock tree synthesis with non-zero clock skew scheduling for increased tolerance to process parameter variations, *J. VLSI Signal Process.,* **16** (2/3): 149–161, 1997.
116. J. L. Neves and E. G. Friedman, Design methodology for synthesizing clock distribution networks exploiting non-zero clock skew, *IEEE Trans. Very Large Scale Integr. VLSI Syst.,* **VLSI-4**: 286–291, 1996.
117. I. S. Kourtev and E. G. Friedman, Topological synthesis of clock trees with non-zero clock skew, *Proc. ACM / IEEE Int. Workshop Timing Issues Specif. Synth. Digital Syst.,* 1997, pp. 158–163.
118. I. S. Kourtev and E. G. Friedman, Topological synthesis of clock trees for VLSI-based DSP systems, *Proc. IEEE Workshop Signal Process. Syst.,* 1997, pp. 151–162.
119. I. S. Kourtev and E. G. Friedman, The automated synthesis of high performance clock distribution networks, *Proc. IEEE Int. Workshop Clock Distrib. Netw. Des., Synth., Anal.,* 1997, pp. 11–12.
120. I. S. Kourtev and E. G. Friedman, Simultaneous clock scheduling and buffered clock tree synthesis, *Proc. IEEE Int. Symp. Circuits Syst.,* 1997, pp. 1812–1815.
121. J. L. Neves and E. G. Friedman, Optimal clock skew scheduling tolerant to process variations, *Proc. ACM / IEEE Des. Autom. Conf.,* 1996, pp. 623–628.
122. J. L. Neves and E. G. Friedman, Reduced sensitivity of clock skew scheduling to technology variations, *Proc. 5th ACM / SIGDA Phys. Des. Workshop,* 1996, pp. 241–248.
123. J. L. Neves and E. G. Friedman, Buffered clock tree synthesis with optimal clock skew scheduling for reduced sensitivity to process parameter variations, *Proc. TAU ACM / SIGDA Int. Workshop Timing Issues Specif. Synth. Digital Syst.,* 1995, pp. 131–141.
124. T. Soyata, E. G. Friedman, and J. H. Mulligan, Jr., Incorporating interconnect, register, and clock distribution delays into the retiming process, *IEEE Trans. Comput.-Aided Des.,* **CAD-16**: 105–120, 1997.
125. M. Shoji, Elimination of process-dependent clock skew in CMOS VLSI, *IEEE J. Solid-State Circuits,* **SC-21**: 875–880, 1986.
126. T. Sakurai, Approximation of wiring delay in MOSFET LSI, *IEEE J. Solid-State Circuits,* **SC-18**: 418–426, 1983.
127. E. Vanden Meersch, L. Claesen, and H. De Man, Automated analysis of timing faults in synchronous MOS circuits, *Proc. IEEE Int. Symp. Circuits Syst.,* 1988, pp. 487–490.
128. M. R. Dagenais and N. C. Rumin, Automatic determination of optimal clocking parameters in synchronous MOS VLSI circuits, *Proc. 1988 Stanford Conf. Adv. Res. Very Large Scale Integr.,* 1988, pp. 19–33.

129. M. R. Dagenais and N. C. Rumin, On the calculation of optimal clocking parameters in synchronous circuits with level-sensitive latches, *IEEE Trans. Comput.-Aided Des.,* **CAD-8**: 268–278, 1989.

130. K. A. Sakallah et al., Optimal clocking of circular pipelines, *Proc. IEEE Int. Conf. Comput. Des.,* 1991, pp. 642–646.

131. K. A. Sakallah, T. N. Mudge, and O. A. Olukoton, *checkTc* and *minTc*: Timing verification and optimal clocking of synchronous digital circuits, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1990, pp. 552–555.

132. K. A. Sakallah, T. N. Mudge, and O. A. Olukoton, Optimal clocking of synchronous systems, *Proc. TAU90-ACM Int. Workshop Timing Issues Specif. Synth. Digital Syst.,* 1990.

133. T. G. Szymanski, Computing optimal clock schedules, *Proc. ACM / IEEE Des. Autom. Conf.,* 1992, pp. 399–404.

134. T. G. Szymanski and N. Shenoy, Verifying clock schedules, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1992, pp. 124–131.

135. N. Shenoy and R. K. Brayton, Graph algorithms for clock schedule optimization, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1992, pp. 132–136.

136. R. B. Deokar and S. S. Sapatnekar, A graph-theoretic approach to clock skew optimization, *Proc. IEEE Int. Symp. Circuits Syst.,* 1994, pp. 1.407–1.410.

137. W. Chuang, S. S. Sapatnekar, and I. N. Hajj, A unified algorithm for gate sizing and clock skew optimization to minimize sequential circuit area, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1993, pp. 220–223.

138. T. M. Burks, K. A. Sakallah, and T. N. Mudge, Identification of critical paths in circuits with level-sensitive latches, *Proc. IEEE Int. Conf. Comput.-Aided Des.,* 1992, pp. 137–141.

139. M. Shoji, Electrical design of BELLMAC-32A microprocessor, *Proc. IEEE Int. Conf. Circuits Comput.,* 1982, pp. 112–115.

140. M. Shoji, Reliable chip design method in high performance CMOS VLSI, *Proc. IEEE Int. Confer. Comput. Des.,* 1986, pp. 389–392.

141. J. Montanaro et al., A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor, *Digital Tech. J.,* **9** (1): 49–62, 1997.

142. M. Horowitz, Clocking strategies in high performance processors, *Proc. IEEE Symp. Very Large Scale Integr. Circuits,* 1992, pp. 50–53.

143. M. Hatamian and G. Cash, A 70-MHz 8-bit $\times$ 8-bit parallel pipelined multiplier in 2.5μm CMOS, *IEEE J. Solid-State Circuits,* **SC-21**: 505–513, 1986.

144. M. Hatamian and G. L. Cash, High speed signal processing, pipelining, and VLSI, *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.,* 1986, pp. 1173–1176.

145. T. G. Noll et al., A pipelined 330-MHz multiplier, *IEEE J. Solid-State Circuits,* **SC-21**: 411–416, 1986.

146. M. R. Santoro, Design and clocking of VLSI multipliers, Ph.D. dissertation, Palo Alto, CA: Stanford Univ., 1989.

147. Y. Jiren, I. Karlsson, and G. Svensson, A true single-phase-clock dynamic CMOS circuit technique, *IEEE J. Solid-State Circuits,* **SC-22**: 899–901, 1987.

148. K. Gaj et al., A clock distribution scheme for large RSFQ circuits, *IEEE Trans. Appl. Supercond.,* **AS-5**: 3320–3324, 1995.

149. 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. Supercond.,* **1**: 3–28, 1991.

150. K. Gaj, E. G. Friedman, and M. J. Feldman, Timing of multi-Gigahertz rapid single flux quantum digital circuits, *J. VLSI Signal Process.,* **16** (2/3): 247–276, 1997.

151. K. Gaj, E. G. Friedman, and M. J. Feldman, Timing of Large RSFQ Digital Circuits, *Proc. 6th Int. Supercond. Electron. Conf.,* 1997, pp. 299–301.

152. K. Gaj, E. G. Friedman, and M. J. Feldman, Two-phase clocking for medium to large RSFQ circuits, *Proc. 6th Int. Supercond. Electron. Conf.,* 1997, pp. 302–304.

153. K. Gaj, E. G. Friedman, and M. J. Feldman, Choice of the optimum timing scheme for RSFQ digital circuits, *Proc. 5th Int. Workshop High-Temp. Supercond. Electron Devices,* 1997, pp. 39–40.

154. R. Reinschmidt and D. Leuthold, Clocking considerations for a pentium based CPU module with a 512K byte secondary cache, *Proc. IEEE Multi-Chip Module Conf.,* 1994, pp. 26–31.

EBY G. FRIEDMAN
University of Rochester