

# REDUCED SENSITIVITY OF CLOCK SKEW SCHEDULING TO TECHNOLOGY VARIATIONS

José Luis Neves and Eby G. Friedman

University of Rochester  
 Department of Electrical Engineering  
 Rochester, New York 14627  
 email: [neves.friedman@ee.rochester.edu](mailto:neves.friedman@ee.rochester.edu)

**Abstract** - A methodology is presented in this paper for determining an optimal set of clock path delays for designing high performance VLSI/ULSI-based clock distribution networks. This methodology emphasizes the use of non-zero clock skew to reduce the system-wide minimum clock period. Although choosing (or scheduling) clock skew values has been previously recognized as an optimization technique for reducing the minimum clock period, the difficulty in controlling the delays of the clock paths due to process parameter variations has limited its effectiveness. In this paper the minimum clock period is reduced using intentional clock skew by calculating a permissible clock skew range for each local data path while incorporating process dependent delay values of the clock signal paths.

Graph-based algorithms are presented for determining the minimum clock period and for selecting a range of process-tolerant clock skews for each local data path in the circuit. These algorithms have been demonstrated on the ISCAS-89 suite of circuits. Furthermore, examples of clock distribution networks with intentional clock skew are shown to tolerate worst case clock skew variations of up to 30% without causing circuit failure while increasing the system-wide maximum clock frequency by up to 20% over zero skew-based systems.

## 1. INTRODUCTION

Clock skew occurs when the clock signals arrive at sequentially-adjacent storage elements at different times. Although it has been shown that intentional clock skew can be used to improve the clock frequency of a synchronous circuit [1, 2, 3, 4, 5, 6], clock skew is typically minimized when designing the clock distribution network, since unintentional clock skew due to process parameter variations may limit the maximum frequency of operation, as well as cause circuit failure independent of the clock frequency (*i.e.*, race conditions). In this paper, the clock skew of a local data path  $L_{ij}$  is defined as  $T_{Skewij}(L_{ij}) = T_{CDi} - T_{CDj}$ , where  $T_{CDi}$  and  $T_{CDj}$  are the clock signal delays of registers  $R_i$  and  $R_j$ . The clock skew is described as *negative* if  $T_{CDi}$  precedes  $T_{CDj}$  ( $T_{CDi} < T_{CDj}$ ) and as *positive* if  $T_{CDi}$  follows  $T_{CDj}$  ( $T_{CDi} > T_{CDj}$ ).

It is shown in [1,2] that *double clocking* (the same clock pulse triggers the same data into two adjacent storage elements) can be prevented when the clock skew between these storage elements satisfies  $T_{Skewij} \geq -T_{PDmin}$ , where  $T_{PDmin}$  is the minimum propagation delay of the path connecting both storage elements. Furthermore, it is also shown in [1,2] that *zero clocking* (the data reaches a storage element too late relative to the following clock pulse) is prevented when  $T_{Skewij} \leq T_{CP} - T_{PDmax}$ , where  $T_{CP}$  is the clock period and  $T_{PDmax}$  is the maximum propagation delay of the data path connecting both storage elements. The limits of both inequalities,  $T_{Skewij(min)} = -T_{PDmin}$  and  $T_{Skewij(max)} = T_{CP} - T_{PDmax}$ , define a region of valid clock skew for each pair of adjacent

storage elements, called the *permissible range* [7] or *certainty region* [8], as shown in Figure 1. A violation of the lower bound leads to circuit failure while a violation of the upper bound limits the clock frequency of the circuit. Based on these observations, the *process variation tolerant optimal clock skew scheduling problem* can be divided into two sub-problems: determining a minimum clock period that defines a valid permissible range for any two storage elements in the circuit, and determining a minimum width for each permissible range such that unacceptable variations in the target clock skew remain within the bounds of a permissible range. In this paper, a solution for this problem is presented.

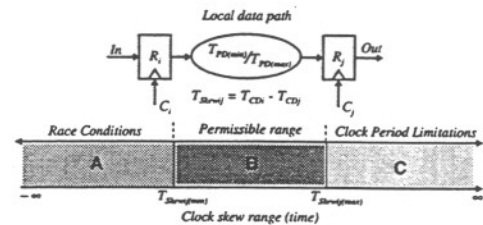


Figure 1: Permissible range of a local data path.

The problem of determining a minimum clock period has been previously solved [1, 3-6] in which a set of timing equations is used to determine the optimal clock period and the clock delay to each register in the circuit, thereby defining the local clock skews. However, in order to better control the effects of process parameter variations, it is advantageous to determine the permissible range of each local data path, select a clock skew value that permits the greatest variation of skew within the permissible range and, finally, determine the clock delays to each register.

This paper is organized as follows: in Section 2, a localized clock skew schedule is derived from the effective permissible range of the clock skew for each local data path considering any global clock skew constraints and process parameter variations. In Section 3, techniques for determining the set of clock skew values that are tolerant to process parameter variations are presented. In Section 4, these results are evaluated on a series of benchmark circuits, demonstrating performance improvements while tolerating process parameter variations. Finally, some conclusions are drawn in Section 5.

## 2. OPTIMAL CLOCK SKEW SCHEDULING

A synchronous digital circuit  $C$  can be modeled as a finite directed multi-graph  $G(V,E)$ , as illustrated in Figure 2. Each vertex in the graph,  $v_j \in V$ , is associated with a register, circuit input, or circuit output. Each edge in the graph,  $e_{ij} \in E$ , represents a physical connection between vertices  $v_i$  and  $v_j$ , with an optional combinational logic path between the two vertices. An edge is a bi-weighted connection representing the maximum (minimum) propagation delay  $T_{PDmax}$  ( $T_{PDmin}$ ) between two sequentially-adjacent storage elements, where  $T_{PD}$  includes the register, logic, and interconnect delays of a local data path [7],

This research was supported by Grant 200484/89.3 from CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) - Brazil, the National Science Foundation under Grant No. MIP-9208165 and Grant No. MIP-9423886, the Army Research Office under Grant No. DAAH04-93-G-0323, and by a grant from the Xerox Corporation

$T_{PD} = T_{C-Q} + T_{Logic} + T_{Int} + T_{Set-up}$ . A local data path  $L_{ij}$  is a set of two vertices connected by an edge,  $L_{ij} = \{v_i, e_{ij}, v_j\}$  for any  $v_i, v_j \in V$ , as shown in Figure 2. A global data path,  $P_{kl} = v_k \xrightarrow{P} v_l$ , is a set of alternating edges and vertices  $\{v_k, e_{kl}, v_l, e_{l2}, \dots, e_{n-1n}, v_n\}$ , representing a physical connection between vertices  $v_k$  and  $v_l$ , respectively (see Figure 2). A multi-input circuit can be modeled as a single input graph, where each input is connected to vertex  $v_0$  by a zero-weighted edge.  $Pl(L_{ij})$  is defined as the permissible range of a local data path and  $Pg(P_{kl})$  is the permissible range of a global data path.

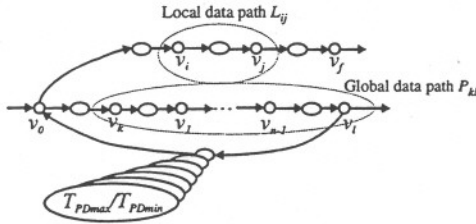


Figure 2: Graph model of a synchronous circuit in terms of local and global data paths

## 2.1 Timing Constraints

The timing behavior of a circuit  $C$  can be described in terms of two sets of timing constraints, local constraints and global constraints. The local constraints ensure that the data signal is correctly latched into the registers of a local data path in order to prevent double and zero clocking. The local timing constraints are represented by the following equation [1-6] to prevent zero clocking,

$$T_{Skew}(L_{ij}) \geq T_{Holdj} - T_{PD(\min)} + \zeta_{ij} \quad (1)$$

and the following equation to prevent double clocking,

$$T_{Skew}(L_{ij}) \leq T_{CP} - T_{PD(\max)} \quad (2)$$

where  $\zeta_{ij}$  is a safety term introduced in [7] to prevent race conditions due to process parameter variations, as described in Section 3.

Besides satisfying the permissible range of a local data path  $Pl(L_{ij})$ , it is also necessary to ensure the existence of a permissible range for each global data path  $Pg(P_{kl})$  to guarantee a race-free circuit, particularly when there are multiple feedback and parallel paths between the two vertices  $v_k$  and  $v_l$ . Two paths with common vertices are said to be in parallel when the signal data flows in the same direction in both paths. Likewise, a path is a feedback path when the data signal flows in a direction that is the reverse direction of the data signal flowing from the input of the circuit to the output of the circuit.

To illustrate a circuit configuration where it is necessary to provide a permissible range for each global data path, consider a circuit composed of several global data paths connecting two common vertices  $v_k$  and  $v_l$ . The vertices  $v_k$  and  $v_l$  represent two registers, each register driven by a single clock signal. The two clock signals define a unique clock skew value between  $v_k$  and  $v_l$  independent of the path connecting  $v_k$  to  $v_l$ . Therefore, a valid clock skew between  $v_k$  and  $v_l$  only exists if the clock skew is common to all the global data paths connecting  $v_k$  and  $v_l$ . Since the clock skew between vertices  $v_k$  and  $v_l$  is also the sum of the clock skew of each cascaded local data path connecting  $v_k$  to  $v_l$  [9], the resulting sum is independent of the global path between  $v_k$  to  $v_l$ . Alternatively, the permissible range of each of the paths connecting the vertices  $v_k$  and  $v_l$  is the sum of the permissible

range of each cascaded local data path between  $v_k$  and  $v_l$ , independent of the global path between  $v_k$  and  $v_l$ . Therefore, a clock skew between the vertices  $v_k$  and  $v_l$  exists if the intersection of the permissible ranges of the paths connecting  $v_k$  and  $v_l$  form a non-empty set, where the intersection of the permissible ranges is determined by the recursive application of the intersection operation applied to a set [10] and a set is the collection of clock skew values within a permissible range. The following example illustrates two circuits, one with two parallel paths and one with one forward path in parallel with a feedback path. Note that determining the permissible range of each local data path is not a sufficient condition for both circuits to work.

**Example 1:** An example of applying the concept of a permissible range of a clocked system to a circuit composed of multiple paths is illustrated in Figure 3, where the numbers assigned to the edges are the maximum and minimum propagation delay of each  $L_{ij}$ , and the register set-up and hold times are arbitrarily assumed to be zero. Furthermore, the pair of clock skew values associated with a vertex (in bold and *italic*) are the minimum and maximum clock skew calculated with respect to the origin vertex  $v_0$  for a given clock period. The pairs in *italic* are determined with  $T_{CP} = 6$  tu (time units) while the pairs in bold are determined with  $T_{CP} = 8$  tu.

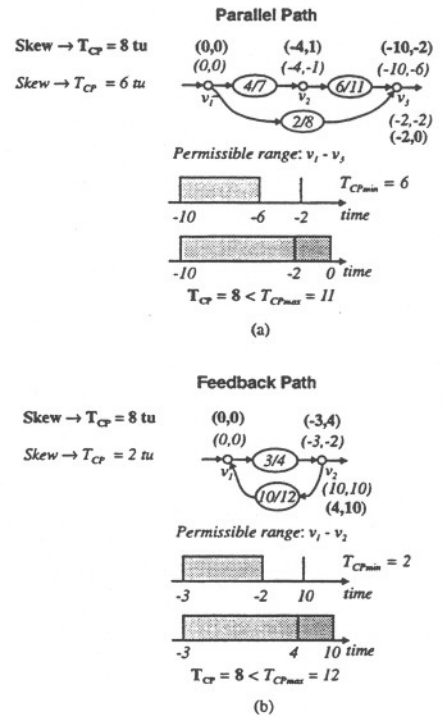


Figure 3: Example circuits describing the process for matching permissible clock skew ranges by adjusting the clock period  $T_{CP}$ . (a) System composed of two forward flowing parallel paths; (b) System composed of a single forward path and a single feedback path.

The minimum clock skew of each local data path  $L_{ij}$  is obtained by applying the maximum permissible negative clock skew to the local data path, or  $T_{Skewij} = T_{PDmin}$  from (1). The maximum bound is obtained directly from (2), given that the clock period  $T_{CP}$  is known. Adding the minimum (maximum) clock skew of each cascaded local data path, the permissible range of each global data path connecting  $v_l$  to  $v_3$  is obtained, as illustrated in Figure 3. Observe that in Figure 3a, for a clock

period  $T_{CP} = 6$  tu, no value of clock skew exists that is common to the two paths connecting vertices  $v_1$  and  $v_3$ , since the permissible range of the path  $v_1-v_2-v_3$  is  $[-10,-6]$  and the permissible range of the path  $v_1-v_3$  is  $[-2,-2]$ , and these permissible ranges do not intersect (or overlap) in time.

A common value of clock skew is only obtained when the clock period is increased to 8 tu. Note that  $T_{CP} = 8$  tu is less than the minimum clock period determined with zero clock skew, which is 11 tu. Therefore, a reduction of the minimum clock period from 11 tu to 8 tu is obtained with the application of negative clock skew.

From the example in Figure 3, in order to prevent circuit failures at the global level, circuits with parallel and feedback paths must have a non-empty permissible range composed of the intersection or overlap among the permissible ranges of each individual parallel and feedback path. Therefore, a new set of global timing constraints are required and formalized below. The concept of permissible range overlap of a global data path  $P_{kl}$  can be stated as follows:

**Theorem 1:** Let  $P_{kl} \in V$  be a global data path within a circuit  $C$  with  $m$  parallel and  $n$  feedback paths. Let the two vertices,  $v_k$  and  $v_l \in P_{kl}$ , which are not necessarily sequentially-adjacent, be the origin and destination of the  $m$  parallel and  $n$  feedback paths, respectively. Also, let  $P_g(P_{kl})$  be the permissible range of the global data path composed of vertices  $v_k$  and  $v_l$ .  $P_g(P_{kl})$  is a non-empty set of values iff the intersection of the permissible ranges of each individual parallel and feedback path is a non-empty set, or

$$P_g(P_{kl}) = \left( \bigcap_{i=1}^m P_g(P_{kl}^i) \right) \cap \left( \bigcap_{j=1}^n P_g(P_{kl}^j) \right) \neq \emptyset \quad (3)$$

**Proof**  $\Rightarrow$ : The clock skew between vertices  $v_k$  and  $v_l$ ,  $T_{Skewkl}$ , is unique and independent of the number of paths connecting the two vertices. Also, the clock skew  $T_{Skewkl}$  of a single path that connects both vertices is the sum of the clock skew of each local data path along the path. Assuming that a value of clock skew exists between vertices  $v_k$  and  $v_l$ , this value is always the same independent of the path connecting  $v_k$  and  $v_l$ . Furthermore, for each path connecting vertices  $v_k$  and  $v_l$ , the minimum (maximum) clock skew value is the sum of the minimum (maximum) clock skews of each local data path along the path defining the permissible range of the global path. Therefore, a valid clock skew between vertices  $v_k$  and  $v_l$  must be within the permissible range of clock skew of each and every path connecting both vertices. In other words, the intersection of permissible ranges must be a non-empty set.

$\Leftarrow$ : Assume that  $P_g(P_{kl}) = \emptyset$  and there exists a valid clock skew value between vertices  $v_k$  and  $v_l$ . If this value of clock skew exists, it must be contained within the permissible range of all the paths connecting the vertices  $v_k$  and  $v_l$ . If a clock skew value exists for all the paths, the result of the intersection of all the permissible ranges cannot be an empty set. Therefore the valid value of clock skew contradicts the initial assumption. ■

Similar to the permissible range of a local data path, the permissible range of a global data path is bounded by a minimum and maximum clock skew value. These values, the upper and lower bounds of the permissible range  $P_g(P_{kl})$ , can be determined as a function of the upper and lower bounds of the permissible ranges of each independent parallel or feedback path connecting vertices  $v_k$  and  $v_l$ .

**Lemma 1:** Let the two vertices,  $v_k$  and  $v_l \in P_{kl}$ , be the origin and destination of a global data path with  $m$  forward and  $n$  feedback paths. If  $P_g(P_{kl}) \neq \emptyset$ , the upper bound of  $P_g(P_{kl})$  is given by

$$T_{Skew}(P_{kl})_{\max} = \text{MAX} \left\{ \min_{1 \leq i \leq m} \left[ T_{Skew}(P_{kl}^i)_{\max} \right], \min_{1 \leq j \leq n} \left[ T_{Skew}(P_{kl}^j)_{\min} \right] \right\}, \quad (4)$$

and the lower bound of  $P_g(P_{kl})$  is given by

$$T_{Skew}(P_{kl})_{\min} = \text{MAX} \left\{ \max_{1 \leq i \leq m} \left[ T_{Skew}(P_{kl}^i)_{\min} \right], \max_{1 \leq j \leq n} \left[ T_{Skew}(P_{kl}^j)_{\max} \right] \right\}. \quad (5)$$

Observe that both bounds of a clock skew region given by (3) are dependent on the clock period in the presence of feedback paths between vertices  $v_k$  and  $v_l$ . This recursive characteristic is used to increase the tolerance of the clock distribution network to process parameter variations, as explained in Section 3. For a non-recursive data path (either local or global), the lower clock skew bound is independent of the clock period, as shown in (1).

## 2.2 Optimal Clock Period

Without exploiting intentional clock skew, the minimum clock period is determined from (2) for the local data path with the maximum propagation delay. However, applying intentional negative clock skew to a local data path permits the circuit to operate at higher clock frequencies. The minimum clock period of a circuit operating with intentional clock skew must simultaneously satisfy (1), (2), and (3) for every local data path.

The minimum clock period to safely latch data through a local data path  $L_{ij}$  can be determined by the differences in propagation delay of the combinational logic block within  $L_{ij}$ , assuming that the timing parameters of the registers ( $T_{Set-up}$ ,  $T_{Hold}$ , and  $T_{C-Q}$ ) are constant. When the maximum possible negative clock skew [2] is applied to  $L_{ij}$ , the clock period is the difference between the propagation delays, since the maximum negative clock skew is the minimum propagation delay within  $L_{ij}$ . The maximum negative clock skew defines the lower bound of the clock period of  $L_{ij}$ . The upper bound of the clock skew can be any value defined by the minimum clock period. Similarly, the clock period of a circuit is bounded by two values,  $T_{CPmin}$  and  $T_{CPmax}$ , determined from the differences in propagation delay within the local data paths of the circuit, as shown below and independently demonstrated by Deokar and Sapatnekar [6]. The lower bound of the clock period,  $T_{CPmin}$ , is the greatest difference in propagation delay of any local data path  $L_{ij} \in C$ ,

$$T_{CPmin} = \text{MAX} \left[ \min_{v_{ij} \in V} (T_{PDmaxij} - T_{PDminij}), \max_{v_i \in V} (T_{PDmaxii}) \right], \quad (6)$$

and the upper bound of the clock period,  $T_{CPmax}$ , is the greatest propagation delay of any local data path  $L_{ij} \in G$ ,

$$T_{CPmax} = \text{MAX} \left[ \max_{v_{ij} \in V} (T_{PDmaxij}), \max_{v_i \in V} (T_{PDmaxii}) \right]. \quad (7)$$

The second term in (6) and (7) accounts for the self-loop where the output of a register is connected to its input through an optional logic block. Since the initial and final registers are the same, the clock skew in a self-loop is zero and the clock period is determined by the maximum propagation delay of the path connecting the output of the register to its input. Observe that a clock period is equal to the lower bound in circuits without parallel and/or feedback paths. Furthermore, the permissible ranges determined with a clock period equal to the upper bound  $T_{CPmax}$  will always satisfy (3) since the permissible range of any local data path in the circuit contains zero clock skew. Although (7) satisfies any local and global timing constraints of circuit  $C$ , it is possible to determine a minimum clock period that satisfies (3) while including intentional clock skew. This transformation leads to the optimal clock period problem which is stated in the following theorem:



**Theorem 2:** Given a synchronous circuit  $C$  modeled by a graph  $G(V,E)$ , there exists a clock period  $T_{CP}$  satisfying (3) and bounded by  $T_{CPmin} \leq T_{CP} \leq T_{CPmax}$ . The clock period is a minimum if the permissible range resulting from (3) contains only a single value of clock skew.

**Proof:** For a local data path, if the clock period increases (decreases) monotonically, the upper bound of the permissible range always increases (decreases) monotonically due to the linear dependency between the clock skew and the clock period. The lower bound does not change since it is independent of the clock period. Therefore, starting with  $T_{CP} = T_{CPmax}$  and progressively reducing the clock period is equivalent to constraining the permissible ranges to narrower regions. In the limit, the minimum clock period is determined when a single value of clock skew within the permissible range is reached, since, due to monotonicity, a further reduction in the clock period would result in an empty permissible range, violating (3). ■

A graph-based algorithm is presented in Figure 4 to determine the minimum clock period that ensures that each of the permissible ranges in the circuit satisfy (3). The initial clock period is given by (6) and, for each pair of registers in the circuit  $C$ , the local and global permissible ranges are calculated, as illustrated in Figure 4 in the lines 4-13. The content of the permissible range is evaluated (line 14) and if empty, the clock period is increased (line 25), otherwise the clock period is decreased (line 26). A binary search is performed on each new clock period within the algorithm *Intercept* until the minimum clock period has been reached.

1. *Intercept*(  $G(V,E), T_{CP}$  )
2. for each  $v_x \in V$  do
3. for each  $v_y \in V$  and  $v_y \neq v_x$  do
4. for  $i \leftarrow 1$  to  $m$  do (intersection of  $m$  parallel paths)
5. calculate the bounds of the permissible range  $Pg(P_{xy}^i)$
6. if  $Set_{parallel} = \emptyset$  then  $Set_{parallel} = Pg(P_{xy}^i)$
7. else  $Set_{parallel} = Set_{parallel} \cap Pg(P_{xy}^i)$
8. for  $j \leftarrow 1$  to  $n$  do (intersection of  $n$  feedback paths)
9. calculate the bounds of the permissible range  $Pg(P_{xy}^j)$
10. if  $Set_{feedback} = \emptyset$  then  $Set_{feedback} = Pg(P_{xy}^j)$
11. else  $Set_{feedback} = Set_{feedback} \cap Pg(P_{xy}^j)$
12.  $Pg(P_{xy}) = Set_{parallel} \cap Set_{feedback}$
13. if  $Pg(P_{xy}) = \emptyset$  then
14. return "permissible ranges do not intercept"
15. else if  $|T_{Skew}[Pg(P_{xy})_{max}] - T_{Skew}[Pg(P_{xy})_{min}]| < C_I$
16. then return "permissible range too small"
17. else return "success"
18. end *Intercept*
19. end *Intercept*
20. *Optimal\_TCP*(  $C$  )
21.  $lower = T_{CPmin}$ ;  $upper = T_{CPmax}$ ;
22. while (  $upper - lower$  ) >  $\epsilon$
23.  $T_{CP} = (upper + lower)/2$ ;
24. *Intercept*(  $C, T_{CP}$  );
25. if "no success" then  $lower = T_{CP}$ ;
26. else  $upper = T_{CP}$ ;
27. end

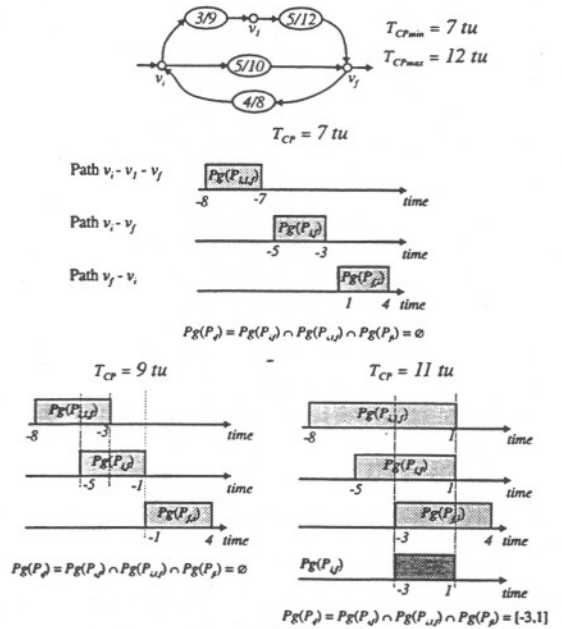
**Figure 4: Pseudo-code of algorithm for determining the minimum clock period based on permissible range overlap**

The order of the algorithm in Figure 4 is  $O(V^2)$ . This order is similar to other clock scheduling algorithms referenced in the literature [5,6], since the number of edges  $E$  is approximately of the same order as the number of vertices  $V$  by a linear transformation, or  $E = O(V)$ .

**Example 2:** An example circuit illustrating how the clock period is determined is presented in Figure 5. The circuit is composed of

three registers symbolized by  $v_i$ ,  $v_j$ , and  $v_f$  with combinational logic within each local data path. It is assumed for simplicity in this example that the timing parameters of each register ( $T_{Set-up}$ ,  $T_{Hold}$ , and  $T_{C-Q}$ ) are zero.

The minimum clock period  $T_{CPmin}$  is determined from (6) and is 7 tu, which is the difference in propagation delay within the logic block of the local data path  $v_j-v_f$ . The maximum clock period  $T_{CPmax}$  is the maximum propagation delay through a logic block in the circuit, which is 12 tu. Starting with  $T_{CPmin}$ , the permissible ranges of each local data path are used to calculate the permissible range of each global data path connecting vertices  $v_i$  to  $v_f$ . Since a unique clock skew must exist between vertices  $v_i$  and  $v_f$ , this value of clock skew must exist within the permissible range of each global data path connecting both vertices.



**Figure 5: Example of selecting the clock period  $T_{CP}$**

From Figure 5 with  $T_{CP} = 7$  tu, the permissible ranges do not intersect, thus no value of clock skew exists that will permit the circuit to function correctly. Increasing  $T_{CP}$  to 9 tu permits the permissible ranges of the global data paths  $v_i-v_j-v_f$  and  $v_j-v_i$  to intersect, but the permissible range of the path  $v_i-v_j-v_f$  does not intersect with the permissible range of the path  $v_j-v_i$ . Therefore, the clock period is again increased. In the example shown in Figure 5, the clock period is increased beyond the optimal clock period to 11 tu to illustrate the existence of a permissible range for vertices  $v_i$  and  $v_j$  that permits choosing more than one value of clock skew between vertices  $v_i$  and  $v_j$ . A single value permissible range is obtained using the algorithm shown in Figure 4, which for this example determines a minimum clock period of 9.67 tu.

The difference between the algorithm described here and other algorithms described in the literature [4-6] is the process for verifying whether a timing violation exists. In the approach offered by Szymanski [4], the existence of positive cycles, indicating a violation of the timing relationships, is checked with Lawler's algorithm [11], where Szymanski also indicates that the Bellman-Ford algorithm is a more efficient strategy for testing for positive cycles. This approach is adopted by Shenoy and Brayton [5] and Deokar and Sapatnekar [6]. Each of these algorithms run in  $O(VE)$  time, where  $V$  is the number of registers and  $E$  is the number of edges. Linear programming solutions to this problem have also been developed by Fishburn [1] and by Sakallah *et al.* [3]. The solution of these algorithms produces the clock delay

from the clock source to each register in the circuit, thereby defining the clock skew of each local data path. However, in order to better control the effects of process parameter variations, it is advantageous to determine the permissible range of each local data path, select a value of clock skew that allows a maximum variation of skew within the permissible range and, with the clock skews selected, determine the clock delay to each register.

### 2.3 Selecting Clock Skew Values

The permissible range of a local data path  $Pl(L_{ij})$  bounded by (1) and (2) defines the set of valid clock skews for a single local data path. However, for a circuit composed of multiple local data paths connected to form parallel and/or feedback paths, not all of the clock skew values that are valid for a local data path can be used to satisfy the permissible range of a global data path. Consider, for example, the path  $P_{i,l,f}$  shown in Figure 5 with  $T_{CP} = 11$  tu. A clock skew of  $T_{skewi,l} + T_{skewl,f} = -3 + -5 = -8$  tu is a value of clock skew that is not within  $Pg(P_{ij})$ , although the individual clock skews are within the respective permissible ranges,  $Pl(L_{il}) = [-3,2]$  and  $Pl(L_{lf}) = [-5,-1]$ . This example indicates that only a sub-set of the permissible range of each local data path can be used to obtain the permissible range of the global data paths of the circuit.

**Lemma 2:** Let  $L_{ij}$  be a local data path within a global data path  $P_k$ . Given a clock period  $T_{CP}$  that satisfies (3), the sub-set of values within  $Pl(L_{ij})$  used to determine  $Pg(P_k)$  is called the *effective permissible range* of a local data path  $\rho(L_{ij})$ , such that  $\rho(L_{ij}) \subseteq Pl(L_{ij})$ .

Lemma 2 does not define the actual position of an effective permissible range within each  $Pl(L_{ij})$ , since several solutions are possible, as illustrated in the example shown in Figure 5. Considering the path  $P_{i,l,f}$ ,  $\rho(L_{il}) + \rho(L_{lf}) = [-2,2] + [-1,-1] = [-3,1]$  and  $\rho(L_{il}) + \rho(L_{lf}) = [0,2] + [-3,-1] = [-3,1]$ , two valid choices exist for the effective permissible range of  $L_{il}$  and  $L_{lf}$ , respectively, since both choices result in  $Pg(P_{ij}) = [-3,1]$ . The actual choice of the effective permissible range is constrained by additional criteria, such as reducing the absolute value of the clock skew [6], or ensuring the largest possible effective permissible range for each local data path so as to maximize the tolerance to process parameter variations. Observe that the possibility of multiple solutions is consistent with the existence of multiple solutions to the problem of indirectly choosing non-zero clock skews by calculating a set of clock path delays to satisfy a valid clock period [1,6]. Therefore, the selection of a specific value of clock skew for each local data path is performed in two steps. In the first step, the effective permissible range is determined for each local data path, while during the second step, the specific local clock skews are chosen to maximize the tolerance to process parameter variations. The assignment of the largest possible effective permissible range to a local data path begins with determining the unique solution to the permissible range of each global data path, as formulated below:

**Theorem 3:** Given a synchronous circuit  $C$  modeled by a graph  $G(V,E)$ , let the two vertices,  $v_k$  and  $v_l \in V$ , be the origin and destination of a global data path  $P_k$  with  $m$  forward and  $n$  feedback paths. Let  $Pg(P_k)$  be determined by (3). If  $Pg(P_k) \neq \emptyset$ , the width of  $Pg(P_k)$  is greatest when the bounds of  $Pg(P_k)$  are determined by (4) and (5), respectively.

**Proof:** This theorem is proved by observing that the bounds of  $Pg(P_k)$  depend directly on the bounds of the permissible range of each global data path connecting vertices  $v_k$  and  $v_l$ . Assuming that the two vertices  $v_k$  and  $v_l$  are connected by two parallel paths and the minimum [maximum] clock skew of the permissible range between the two vertices is a value smaller than the value given

by (4) [(5)], a permissible range is produced with a width larger than the permissible range obtained with (4) and (5). However, from Lemma 1 and the property of monotonicity, this assumption is a contradiction since the larger width can only result from the interception of larger permissible ranges. Therefore, a smaller bound indicates that the upper and lower bounds of a particular global data path have not been constrained by (4) or (5). ■

The pseudo-code to determine the clock skew of each local data path is presented in Figure 6. The algorithm *Intercept* is first used to determine the permissible range of each global data path in the circuit, given a clock period  $T_{CP}$  that satisfies (3).

Determining the effective permissible range and selecting the clock skew value for each local data path are performed as follows: 1) the permissible range of a global data path  $Pg(P_k)$  is divided equally among each local data path connecting the vertices  $v_k$  and  $v_l$  (line 5); 2) each effective permissible range  $\rho(L_{ij})$  is placed as close as possible to the upper bound of the original permissible range  $Pl(L_{ij})$  (lines 6 and 7), thereby minimizing the likelihood of creating any race conditions; and 3) the clock skew is chosen in the middle of the effective permissible range, since no prior information of the variation of a particular clock skew value may exist (line 8). The minimum clock path delays are determined from this clock skew schedule [9].

1. *Select\_Skew*( $G(V,E), T_{CP}$ )
2. *Intercept*( $G(V,E), T_{CP}$ )
3. **for each**  $P_k^n \in G(V,E)$  **do**
4.     **for**  $i \leftarrow k$  **to**  $l \in P_k^n$  **do**
5.          $Width[\rho(L_{ij})] = MAX[Pg(P_k^i)] - MIN[Pg(P_k^l)] / \# L_{ij} \in P_k^n$
6.         Upper bound of  $\rho(L_{ij}) = MAX[Pl(L_{ij})]$ ;
7.         Lower bound of  $\rho(L_{ij}) = MAX[Pl(L_{ij})] - Width[\rho(L_{ij})]$ ;
8.          $T_{skewij} = MAX[\rho(L_{ij})] - MIN[\rho(L_{ij})] / 2$ ;
9.     **end** *Select\_Skew*

Figure 6: Pseudo-code of algorithm for selecting the non-zero clock skew of a local data path

### 3. REDUCED SENSITIVITY TO PROCESS PARAMETER VARIATIONS

A top-down design system has been developed for synthesizing intentionally skewed clock distribution networks from the timing constraints of the circuit without prior layout information [7,9], as illustrated in Figure 7. The top-down synthesis system is integrated with a bottom-up verification phase (darker-shaded region in Figure 7) to ensure that the effects of process parameter variations on the selected clock skew values do not violate the bounds of the effective permissible range of each local data path.

The clock distribution network is primarily composed of active devices (CMOS inverters) that accurately implement the clock path delays that enforce the non-zero clock skew. The circuit model of the clock tree with active devices is based on the alpha-power law model [12]. Due to the active devices within the clock tree, the clock path delay variations are primarily due to the effects of process parameter variations on the active devices rather than variations of the interconnect lines within the clock tree [13].

Once the clock distribution network has been designed, each clock path delay is re-calculated assuming that the cumulative effects of device parameter variations, such as threshold voltage and channel mobility, can be collected into a single parameter characterizing the gain of a CMOS inverter, specifically the output current  $I_{DO}$  [12]. The worst case variation of each clock skew is determined from calculating the minimum and maximum clock path delays considering the minimum and maximum  $I_{DO}$  of each inverter within each branch of the clock distribution network. If a single worst case clock skew value is outside the

effective permissible range of the corresponding local data path,  $T_{Skewij} \propto \rho(L_{ij})$ , a timing constraint is violated and the circuit will not function properly.

This violation is passed to the top-down synthesis system, indicating which bound of the effective permissible range is violated. The assigned clock skew of at least one local data path  $L_{ij}$  within the system may violate the upper bound of  $\rho(L_{ij})$ , i.e.,  $T_{Skewij} > MAX[\rho(L_{ij})]$ . This violation is corrected by increasing the clock period  $T_{CP}$ , since due to monotonicity the effective permissible clock skew range for each local data path is also increased ( $T_{Skewij(max)}$  is increased). The new clock skew value may also violate the lower bound of a local data path, i.e.,  $T_{Skewij} < T_{Skewij(min)}$  where  $T_{Skewij(min)} \subset \rho(L_{ij})$ .

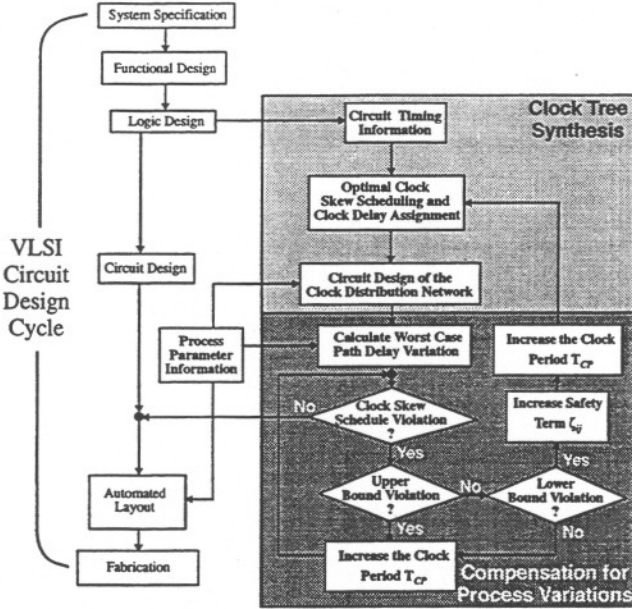


Figure 7: Synthesis methodology of clock distribution networks tolerant to process variations

Two compensation techniques are used to prevent lower bound violations, depending on where the effective permissible range of a local data path  $\rho(L_{ij})$  is located within the permissible range of the local data path,  $PI(L_{ij})$ . If the lower bound of  $\rho(L_{ij})$  is greater than the lower bound of  $PI(L_{ij})$ , the clock period  $T_{CP}$  is increased until the race condition is eliminated, since the effective permissible range will increase due to monotonicity. However, if after increasing the clock period, the clock skew violation still exists and the lower bound of the effective permissible range is equal to the lower bound of the local data path  $\{MIN[\rho(L_{ij})] = MIN[PI(L_{ij})]\}$ , any further increase of the clock period will not eliminate the violation caused by not satisfying (1).

Rather, if the lower bound of  $\rho(L_{ij})$  is equal to the lower bound of  $PI(L_{ij})$ , a safety term  $\zeta_{ij} > 0$  is added to the local timing constraint that defines the lower bound of  $PI(L_{ij})$  [see (1)]. The clock period is increased and a new clock skew schedule is calculated for this value of the clock period. The increased clock period is required to obtain a set of effective permissible ranges with widths equal to or greater than the set of effective permissible ranges that existed before the clock skew violation. Observe that by including the safety term  $\zeta_{ij}$ , the lower bound of the clock skew of the local data path containing the race condition is shifted to the right, moving the new clock skew schedule of the entire circuit away from the bound violation and minimizing the likelihood of any race conditions. This iterative process continues until the worst case variations of the selected clock skews no longer violate the corresponding effective permissible ranges.

**Example 3:** An example of a synchronous circuit with upper and lower bound violations is presented in Figure 8. This circuit is composed of two parallel paths. The initial clock period is  $T_{CP} = 9$  tu. The valid permissible range between vertices  $v_1$  and  $v_3$  is initially  $Pg(P_{13}) = [-2,0]$ , as indicated by the dark-shaded area shown in Figure 8.

**Upper bound violation:** Assume, for example, that the worst case variation of the clock skew results in an upper bound violation. For example, the nominal clock skew is  $T_{Skew13} = -1$  tu while the clock skew caused by the worst case variation is 0.5 tu. By increasing the clock period to 9.5 tu, the width of the permissible range between vertices  $v_1$  and  $v_3$  is increased to  $Pg(P_{13}) = [-2,1]$ . The new upper bound is greater than the worst case variation of clock skew and the circuit will now operate correctly.

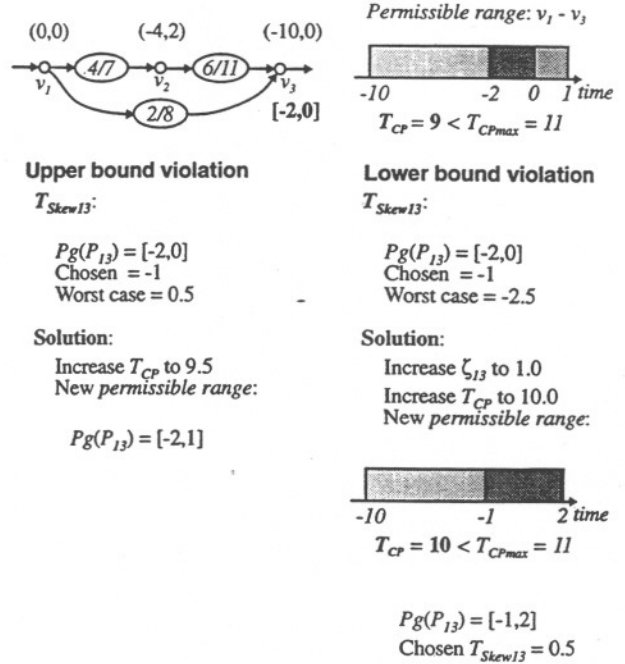


Figure 8: Example of upper and lower bound clock skew violations and strategy to remove these violations

**Lower bound violation:** Now assume that for the same circuit shown in Figure 8, the worst case variation causes the clock skew to be -2.5 tu, rather than a nominal value of the clock skew of -1 tu, violating the lower bound of the permissible range  $Pg(P_{13})$ . In this case, increasing the clock period does not remove the violation since the lower bound of  $Pg(P_{13})$  is independent of the clock period. Rather, the safety term  $\zeta_{13}$  is increased arbitrarily from 0 to 1 tu and the clock period  $T_{CP}$  is increased to 10 tu. As discussed previously, a revised schedule of clock skew values is calculated by applying the top-down design system to re-design the clock distribution network. The topology of the clock distribution does not change, although the delay values assigned to the branches must change to reflect the new clock skew schedule. In Figure 8, increasing  $T_{CP}$  to 10 tu will create a new permissible range  $Pg(P_{13}) = [-1,2]$ . A new clock skew is selected from this permissible range and the worst case clock skew variations are calculated. Assuming that the worst case variation is still 1.5 tu of the nominal value (see Figure 8), this new target clock skew is within  $Pg(P_{13})$  and the violation is eliminated.



The simulation results presented in this section illustrate the performance improvements obtained by exploiting non-zero clock skew while considering the effects of process parameter variations. In order to demonstrate these performance improvements, the suite of ISCAS-89 sequential circuits is chosen as benchmark circuits [14]. A unit fanout delay model (one unit delay per gate plus 0.2 units for each fanout of the gate) is used to estimate the minimum and maximum propagation delay of the logic blocks. The set-up and hold times are set to zero. The performance results are illustrated in Table 1. The number of registers and gates within the circuit including the I/O registers are shown in Column 2. The clock period assuming zero clock skew is shown in Column 3. The clock period obtained with intentional clock skew is shown in Column 4. The resulting performance gain is shown in Column 5. The clock period obtained with the constraint of zero clock skew imposed among the I/O registers is shown in Column 6 while the performance gain with respect to a zero skew implementation is shown in Column 7.

Table 1: Performance improvement with non-zero clock skew

circ.	size #reg./#gates	$T_{CP0}$		gain (%)	$T_{CP}$		gain (%)
		$T_{Skewij} = 0$	$T_{Skewij} \neq 0$		$T_{SkewIO} = 0$		
ex1	20/-	11.0	6.3	43	7.2	35	
s27	7/10	9.2	5.4	41	6.2	33	
s298	23/119	16.2	11.6	28	11.6	28	
s386	20/159	19.8	19.8	0	19.8	0	
s444	30/181	18.6	11.1	41	11.1	41	
s510	32/211	19.8	17.3	13	17.3	13	
s838	67/446	27.0	13.5	50	15.6	42	

The results shown in Table 1 demonstrate reductions of the minimum clock period of up to 50% when intentional clock skew is exploited. The amount of reduction is dependent on the characteristics of each circuit, particularly the differences in propagation delay between each local data path. Note also that by constraining the clock skew of the I/O registers to zero, circuit speed can be improved, although less than without this constraint.

Examples of clock distribution networks which exploit intentional clock skew and are less sensitive to the effects of process parameter variations are listed in Table 2. The clock trees are synthesized with the system presented in [7,9]. The clock skew values are derived from a circuit simulation of the clock path delays of a clock tree using SPICE Level-3 assuming the MOSIS SCMOS 1.2  $\mu\text{m}$  fabrication technology. The minimum clock period assuming zero clock skew  $T_{CP0}$  and intentional clock skew  $T_{CPi}$  is shown in Column 2, respectively. The permissible range most susceptible to process parameter variations is illustrated in Column 3. The target clock skew value is shown in Column 4. In Columns 5 and 6, respectively, the nominal and maximum clock skew are depicted, assuming a 15% variation of the drain current  $I_{DO}$  of each inverter. Note that both the nominal and the worst case value of the clock skew are within the permissible range. The per cent variation of clock skew due to the effects of process parameter variations is shown in column 7. A 20% improvement in speed with up to a 30% variation in the nominal clock skew, and a 33% improvement in speed with up to an 18% variation in the nominal clock skew are observed for the example circuits listed in Table 2.

Table 2: Worst case variations in clock skew due to process parameter variations,  $I_{DO} = 15\%$

circuit	$T_{CP0}/T_{CPi}$	permissible range	selected clock skew	Simulated skew (ns)		Error (%)	
				nom	worst case	nom	worst case
cdn 1	11/9	[-8,-2]	-3.0	-3.0	-2.10	0.0	30.0
cdn 2	18/15	[-6.8, -1.4]	-4.2	-4.1	-3.3	2.4	21.4
cdn 3	27/18	[-14, 2.3]	1.1	1.14	1.3	3.6	18.2

## 5. CONCLUSIONS

The problem of scheduling clock path delays such that intentional localized clock skew is used to improve performance and reliability while considering the effects of process parameter variations is examined in this paper. A graph-based approach is presented for determining the minimum clock period and the permissible ranges of each local data path. The process of determining the bounds of these ranges and selecting the clock skew value for each local data path so as to minimize the effects of process parameter variations is described. Rather than placing limits or bounds on the clock skew variations, this approach guarantees that each selected clock skew value is within the permissible range despite worst case variations of the clock skew.

The clock skew scheduling algorithms for compensating for process variations have been incorporated into a top-down, bottom-up clock tree synthesis environment. In the top-down phase, the clock skew schedule and permissible ranges of each local data path are determined to allow the maximum variation of the clock skew. In the bottom-up phase, possible clock skew violations due to process parameter variations are compensated by the proper choice of clock skew for each local data path and the controlled increase of the clock period  $T_{CP}$ . The clock period of a number of ISCAS-89 benchmark circuits are minimized with this clock scheduling algorithm. Scheduling the clock skews to make a clock distribution network more tolerant to process parameter variations is presented for several example networks. The results listed in Table 2 confirm the aforementioned claim that variations in clock skew due to process parameter variations can be both tolerated and compensated.

## 6. REFERENCES

- [1] J. P. Fishburn, "Clock Skew Optimization," *IEEE Transactions on Computers*, Vol. C-39, No. 7, pp. 945-951, July 1990.
- [2] E. G. Friedman, *Clock Distribution Networks in VLSI Circuits and System*, IEEE Press, 1995.
- [3] K. A. Sakallah, T. N. Mudge, and O. A. Olukotun, "checkTc and minTc: Timing Verification and Optimal Clocking of Synchronous Digital Circuits," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 111-117, June 1990.
- [4] T. G. Szymanski, "Computing Optimal Clock Schedules," *Proceedings of the IEEE/ACM Design Automation Conference*, pp. 399-404, June 1992.
- [5] N. Shenoy and R. K. Brayton, "Graph Algorithms for Clock Schedule Optimization," *Proceedings of the IEEE International Conference on Computer-Aided Design*, pp. 132-136, November 1992.
- [6] R. B. Deokar and S. Sapatnekar, "A Graph-theoretic Approach to Clock Skew Optimization," *Proceedings of the IEEE International Symposium on Circuits and Systems*, pp. 407-410, May 1994.

- [7] J. L. Neves and E. G. Friedman, "Design Methodology for Synthesizing Clock Distribution Networks Exploiting Non-Zero Localized Clock Skew," *IEEE Transactions on VLSI Systems*, Vol. VLSI-4, No. 2, June 1996.
- [8] D. G. Messerschmitt, "Synchronization in Digital System Design," *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 6, pp. 1404-1419, October 1990.
- [9] J. L. Neves, *Synthesis of Clock Distribution Networks for High Performance VLSI/ULSI-Based Synchronous Digital Systems*, Ph.D. Dissertation, University of Rochester, December 1995.
- [10] D. F. Stanat and D. F. McAllister, *Discrete Mathematics in Computer Science*, Prentice Hall, 1977.
- [11] E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Rinehart and Winston, 1976.
- [12] T. Sakurai and A. R. Newton, "Alpha-Power Law MOSFET Model and its Applications to CMOS Inverter Delay and Other Formulas," *IEEE Journal of Solid State Circuits*, Vol. SC-25, No. 2, pp. 584-594, April 1990.
- [13] M. Shoji, "Elimination of Process-Dependent Clock Skew in CMOS VLSI," *IEEE Journal of Solid State Circuits*, Vol. SC-21, No. 5, pp. 875-880, October 1986.
- [14] S. Yang, "Logic Synthesis and Optimization Benchmarks User Guide: Version 3.0," *Technical Report, Microelectronics Center of North Carolina*, January 1991.