Minimizing Sensitivity to Delay Variations in High-Performance Synchronous Circuits

Xun Liu Marios C. Papaefthymiou
Department of Electrical Engineering
and Computer Science
University of Michigan
Ann Arbor, Michigan 48109

Eby G. Friedman
Department of Electrical Engineering
University of Rochester
Rochester, New York 14627

Abstract

This paper investigates retiming and clock skew scheduling for improving the tolerance of synchronous circuits to delay variations. It is shown that when both long and short paths are considered, circuits optimized by the combined application of the two techniques are more tolerant to delay variations than when optimized by either of the two techniques separately. A novel mixed-integer linear programming formulation is given for simultaneous retiming and clock scheduling with a target clock period and tolerance under setup and hold constraints. Experiments with LGSynth93 and ISCAS89 benchmark circuits demonstrate the effectiveness of the combined optimization. For half of the test circuits, tolerance to delay variations increased by at least 23% over the separate application of retiming and clock scheduling. Moreover, for two thirds of the test circuits, maximum tolerance improved by at least 11%.

1. Introduction

Retiming is an architectural-level transformation that optimizes digital circuits by relocating their storage elements. Clock scheduling adjusts the delays of the clock signals in a circuit and can be used as an alternative to retiming. Significant research has been devoted to each of the two optimizations separately. The investigation of the combined application of these techniques has been limited, however.

This paper investigates the simultaneous application of retiming and clock scheduling for increasing the tolerance of a digital circuit's timing to delay variations. These variations often present a fundamental constraint in the design of high-performance circuits. Typically, there are three sources of delay variations: process parameter variations, temperature or environmental variations, and power supply variations. The creation of new design techniques and methodologies that minimize the sensitivity of circuit timing to delay variations is of paramount importance for high-performance design.

Two main analytical contributions are contained in this paper. First, we give a set of $O(E^2)$ constraints for the problem of simultaneous retiming and clock scheduling to achieve a target clock period and delay tolerance. Second, we formulate the problem of simultaneous retiming and clock scheduling under setup and hold constraints as a mixed-integer linear program (MILP). A circuit with maximum tolerance to delay variations can be computed by performing a binary search over the range of possible tolerance values.

In experiments with benchmark circuits from the LGSynth93 and ISCAS89 suites, simultaneous retiming and clock scheduling resulted in significantly more tolerant circuits than the independent application of the two optimization techniques. For half of the circuits in our test suite, maximum tolerance to delay variations improved by at least 23% over separate retiming or clock skew scheduling. For about two thirds of the test circuits, maximum tolerance to delay variations improved by at least 11%.

Retiming has been investigated for a variety of clocking disciplines [7, 9, 10, 15], delay models [8, 16], and optimization objectives [1, 4, 12, 14]. A linear programming formulation of the clock scheduling problem was first described in [5]. The combined application of retiming and clock scheduling was discussed in [11]. A two-step procedure for maximizing the operating frequency of a synchronous circuit by combining retiming with clock scheduling was proposed in [2]. That work is concerned only with setup violations, however, and does not explore the expanded solution space resulting when both setup and hold constraints are considered.

The main challenge with the integration of retiming and clock scheduling is the formulation of the problem as a conjunction of linear constraints. As is the case with other

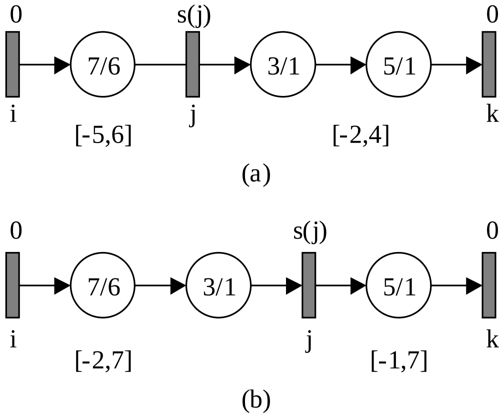

Figure 1. (a) Original and (b) retimed circuit.

retiming problems [8, 16], the co-existence of setup and hold constraints introduces disjunctions among constraints. Thus, the resulting solution space precludes the application of powerful convex programming techniques. This paper presents a mixed-integer linear program for simultaneous retiming and clock scheduling that is derived by a combination of upper bounding and graph-theoretic techniques.

The remainder of this paper has eight sections. Section 2 demonstrates the performance advantage of simultaneous retiming and clock scheduling. Background material is given in Section 3. In Section 4, we give a shortest-paths formulation for the problem of clock scheduling with a target tolerance, a target clock period, and fixed register locations. Section 5 presents necessary and sufficient conditions for achieving correct timing when a circuit is optimized by simultaneous retiming and clock scheduling under setup and hold constraints. An alternative formulation of these conditions in terms of an auxiliary graph is given in Section 6. This formulation is used in Section 7 to derive an equivalent mixed-integer linear program. Section 8 compares the results obtained by the separate application of retiming and clock scheduling with those obtained by the simultaneous application of the two optimizations. Our contributions are summarized in Section 9.

2. Motivation

The effectiveness of simultaneous retiming and clock scheduling is demonstrated by the circuit in Figure 1. Each vertex represents a block of combinational logic, and each rectangle represents an edge-triggered register. Each pair x/y denotes the maximum and minimum propagation delay of the signals through the corresponding node. The clock skew between the input/output registers i and k is assumed to be zero. The setup and hold constraints along each combinational path yield a range $[x, y]$ of permissible clock skews [13] for register j. The permissible skew range of j is obtained by intersecting all these possible ranges.

Consider the original circuit in Figure 1(a). For a target clock period of 12 time units, the intersection of the two ranges is [-2,4]. When clock skew is zero, the permissible range of j is [-2,2], assuming symmetric clock delay variations. Thus the tolerance of this circuit is 4. When clock signals arrive at j with a delay $s(j) = 1$, however, the permissible range is [-2,4], and delay tolerance increases to 6.

Figure 1(b) shows a retimed version of the original circuit that is obtained by shifting j forward. In this case, the intersection of the two skew ranges is [-1,7]. When clock skew is zero, the permissible range of j is [-1,1], and the tolerance drops to 2. When the arrival of the clock signals at j is delayed by $s(j) = 3$, however, the permissible range becomes [-1,7], and tolerance increases to 8. This value is the maximum tolerance that can be achieved by simultaneous retiming and clock scheduling. Moreover, it cannot be achieved by the separate application of the optimizations on the original circuit.

An interesting observation in this example is that the delay tolerance of the retimed circuit is smaller than that of the original circuit when skews are zero. Nevertheless, the retimed circuit exhibits maximum tolerance to delay variations when clock skews are nonzero.

3. Background

3.1. Circuit and Delay Model

An edge-triggered circuit is modeled as a directed multigraph $G = \langle V, E, d, w \rangle$. The vertices V correspond to the combinational logic elements in the circuit. Each vertex $v \in V$ is associated with a nonnegative weight $d(v)$ which describes the propagation delay through the corresponding logic block. Our results can be extended to include the case where each logic block has a maximum propagation delay $d_{\max}(v)$ and a minimum propagation delay $d_{\min}(v)$.

The directed edges E of the graph model the interconnections between the combinational blocks. Each edge $e \in E$ corresponds to a wire that connects an output of a combinational block to the input of another combinational block, possibly through one or more globally clocked, edge-triggered registers. For each edge $e \in E$, the register count of the corresponding wire is given by an integer, nonnegative edge-weight $w(e)$. In every directed cycle of G, there is an edge with a strictly positive register count.

3.2. Retiming

A *retiming* of an edge-triggered circuit $G = \langle V, E, d, w \rangle$ is an integer-valued vertex-labeling $r : V \rightarrow \mathbf{Z}$ that denotes a transformation of the original circuit G into a functionally

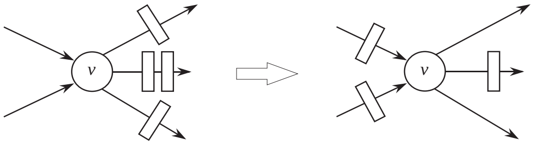

Figure 2. Retiming a vertex v by $r(v) = 1$.

equivalent circuit $G_r = \langle V, E, d, w_r \rangle$. For each edge $u \xrightarrow{e} v$ in G_r, w_r is defined by the equation

$$w_r(e) = w(e) + r(v) - r(u) \ . \tag{1}$$

The retiming transformation for a vertex v in V is shown in Figure 2. The output of v's computation in G_r is generated $r(v)$ clock cycles later than in G. The retimed circuit G_r is *well-formed* if for all edges $e \in E$, we have

$$w_r(e) \geq 0 \ . \tag{2}$$

Equation (1) implies that for every vertex pair u, v in V, the change in the register count along *any* path $u \overset{p}{\leadsto} v$ depends solely on its two endpoints:

$$w_r(p) = w(p) + r(v) - r(u) \ , \tag{3}$$

where $w(p) = \sum_{e \in p} w(e)$. Thus, the maximum decrease in the register count of any path $u \overset{p}{\leadsto} v$ is

$$W(u, v) = \min \left\{ w(p) : u \overset{p}{\leadsto} v \right\} \ . \tag{4}$$

The only paths $u \overset{p}{\leadsto} v$ that can become combinational (and possibly lead to a timing violation) in G_r are those for which $w(p) = W(u, v)$ in G. For each of the $O(V^2)$ vertex pairs u, v in V, the quantities

$$D(u, v) = \max \left\{ d(p) : u \overset{p}{\leadsto} v, w(p) = W(u, v) \right\} \ , \tag{5}$$

$$\Delta(u, v) = \min \left\{ d(p) : u \overset{p}{\leadsto} v, w(p) = W(u, v) \right\} \ , \tag{6}$$

where $d(p) = \sum_{x \in p} d(x)$, represent the longest and shortest propagation delays from u to v, respectively, whenever the retimed circuit includes a combinational path between the two vertices. Therefore, the clock period of any retimed circuit G_r is always some element in the $O(V^2)$-size set of $D(u, v)$.

When only long paths are considered, a retimed circuit that achieves a given clock period c can be computed in $O(VE)$. A retimed circuit that achieves the minimum possible clock period can be computed in $O(VE + V^2 \lg V)$ steps [9].

3.3. Clock Skew Scheduling

In synchronous circuits, clock signals provide a global time reference that synchronizes the flow of data between storage elements. These signals are delivered by a distribution network [6]. A variety of factors such as differences in interconnect delay, parasitic impedances, and process parameters variations affect their arrival times at the storage elements of the circuit. The difference between the arrival times at two sequentially-adjacent registers is known as the clock skew between these registers [6].

A *clock schedule* of a circuit $G = \langle V, E, d, w \rangle$ is a real-valued edge-labeling $s : E \longrightarrow$**R** that gives the propagation delay from the global clock source to each wire e in the circuit. By adjusting these delays, timing violations can be fixed (or created). For example, consider a combinational path $u \overset{p}{\leadsto} v$ which is bounded by registers on $? \xrightarrow{e} u$ and $v \xrightarrow{e'} ?$. If $s(e) \geq s(e')$, then the time available for the propagation of signals from e to e' decreases by $s(e) - s(e')$. Conversely, if $s(e) \leq s(e')$, then the available time increases by $s(e') - s(e)$. These changes may introduce new critical paths or eliminate existing ones. They may also introduce or eliminate hold violations.

A linear programming framework for clock scheduling was first presented in [5]. A graph-theoretic approach to clock scheduling was subsequently described in [3]. In both papers, the placement of the storage elements was assumed to be fixed. Algorithms for scheduling local clocks to improve the tolerance of a circuit to process parameter variations were presented in [13].

4. Clock Scheduling Constraints

This section gives a precise statement of the clock scheduling problem with a given tolerance as a single-source shortest-paths problem with $O(E^2)$ constraints.

The following theorem captures the timing conditions that must be satisfied by a clock schedule that achieves a target clock period. These conditions can be extended to include nonzero setup and hold times. The proof of the theorem follows from [5].

Theorem 1 *Let $G = \langle V, E, d, w \rangle$ be an edge-triggered circuit and c a given constant. Moreover, let $s_m : E \longrightarrow$ **R** and $s_M : E \longrightarrow$ **R** be assignments of minimum and maximum clock delays, respectively. Then, G is timed correctly if and only if for every pair $? \xrightarrow{e} u$, $v \xrightarrow{e'} ?$ in E such that $w(e) \geq 1$, $w(e') \geq 1$, and $W(u, v) = 0$, we have*

$$\Delta(u, v) + s_m(e) - s_M(e') \geq 0 \ , \tag{7}$$

$$D(u, v) + s_M(e) - s_m(e') \leq c \ . \tag{8}$$

□

We can now express the clock scheduling problem with a target clock period and tolerance as a shortest-paths problem with $O(E^2)$ inequalities that can be computed in $O(E^2)$ time and can be solved in $O(E^3)$ steps using the Bellman-Ford single-source shortest-paths algorithm.

Theorem 2 *Let $G = \langle V, E, d, w\rangle$ be an edge-triggered circuit. Moreover, let c and t be given real constants. Then, G achieves a clock period c with tolerance t if and only if there exist nonnegative functions $s_m : E \rightarrow \mathbf{R}$ and $s_M : E \rightarrow \mathbf{R}$ such that for each edge $u \xrightarrow{e} v$,*

$$s_m(e) \leq s_M(e) - t \ , \tag{9}$$

and for every edge pair $? \xrightarrow{e} u$, $v \xrightarrow{e'} ?$ in E such that $w(e) \geq 1$, $w(e') \geq 1$, and $W(u, v) = 0$,

$$s_M(e') \leq s_m(e) + \Delta(u, v) \ , \tag{10}$$

$$s_M(e) \leq s_m(e') + c - D(u, v) \ . \tag{11}$$

□

For a target clock period c, the maximum tolerance τ_s can be determined by a binary search in t. Given s_m and s_M, the corresponding schedule s with maximum tolerance to symmetric delay variations is obtained by setting $s(e) = (s_m(e) + s_M(e))/2$ for all e in E.

5. Clock Scheduling and Retiming

The following theorem gives a set of $O(E^2)$ constraints for correct timing when clock scheduling and retiming are applied simultaneously. Its correctness follows from Theorem 2.

Theorem 3 *Let $G = \langle V, E, d, w\rangle$ be a synchronous circuit, and let c and t be given constants. Moreover, let $r : V \rightarrow \mathbf{Z}$ be a retiming function, let $s_M : E \rightarrow \mathbf{R}$ be an assignment of maximum clock delays, and let $s_m : E \rightarrow \mathbf{R}$ be an assignment of minimum clock delays. Then the retimed circuit G_r is well-formed and achieves a clock period c with tolerance t if and only if for every edge $u \xrightarrow{e} v \in E$,*

$$s_m(e) \leq s_M(e) - t \ , \tag{12}$$

$$w(e) + r(v) - r(u) \geq 0 \ , \tag{13}$$

and for every pair of edges $? \xrightarrow{e} u, v \xrightarrow{e'} ? \in E$,

$$E(e, e') > 0 \ \Rightarrow \ W_r(u, v) \geq 1 \text{ or } (w_r(e) = 0 \text{ or } w_r(e') = 0) \ , \tag{14}$$

where $E(e, e') = D(u, v) + s_M(e) - s_m(e') - c$ and $E(e, e') = -(\Delta(u, v) + s_m(e) - s_M(e'))$ for the setup and hold constraints, respectively. □

For simplicity, the constraints of Theorem 3 assume zero setup and hold times. Non-zero times T_{setup} and T_{hold} can be included in a straightforward manner by setting $E(e, e') > -T_{setup}$ or $E(e, e') > -T_{hold}$, as appropriate, in the left-hand side of the implication in Relation (14). For a target clock period c, the maximum tolerance τ_{rs} over all retimings and clock schedules can be determined by a binary search in t.

6. Companion Graph

A *companion graph* $G' = \langle V', E', w'\rangle$ can be used to transform the timing constraints from Theorem 3 into a mixed-integer linear program. The construction of G' from the circuit graph G is identical to that in [8]. Each edge $u \xrightarrow{e} v \in E$ is segmented into two edges, $u \xrightarrow{e_1} x_{uv}$ and $x_{uv} \xrightarrow{e_2} v$, where x_{uv} is a dummy vertex. The edge e_1 has exactly one register when the corresponding edge $e \in E$ has a positive register count and zero registers otherwise. Thus, the register count of e_1 serves as an *index function* for the register count of the corresponding generating edge $e \in E$. The edge e_2 carries the balance of the registers up to $w(e)$.

In mathematical terms, the companion graph $G' = \langle V', E', w'\rangle$ is defined as

$$\begin{aligned} V' &= V \cup \left\{ x_{uv} : u \xrightarrow{e} v \in E \right\} \ , \\ E' &= \left\{ u \xrightarrow{e_1} x_{uv}, x_{uv} \xrightarrow{e_2} v : u \xrightarrow{e} v \in E \right\} \ , \end{aligned}$$

where for each edge $u \xrightarrow{e} v \in E$,

$$\begin{aligned} w'(e_1) &= \min\{1, w(e)\} \ , \text{ and} \\ w'(e_2) &= w(e) - \min\{1, w(e)\} \ . \end{aligned}$$

The following lemma recasts Theorem 3 in terms of G' and a corresponding retiming function r'. Given r', $r(u)$ can be obtained for every $u \in V$ by setting $r(u) = r'(u)$.

Lemma 4 *Let $G = \langle V, E, d, w\rangle$ be a circuit graph, let $G' = \langle V', E', w'\rangle$ be its corresponding companion graph, and let c and t be given constants. Moreover, let $r' : V' \rightarrow \mathbf{Z}$ be a retiming function, let $s_M : E \rightarrow \mathbf{R}$ be an assignment of maximum clock delays, and let $s_m : E \rightarrow \mathbf{R}$ be an assignment of minimum clock delays. Then the retimed circuit G_r is well-formed and achieves a clock period c with tolerance t if and only if for every edge $u \xrightarrow{e} v \in E$, we have*

$$s_m(e) \leq s_M(e) - t \ , \tag{15}$$

for every edge $u \xrightarrow{e} v \in E'$,

$$w'(e) + r'(v) - r'(u) \geq 0 \ , \tag{16}$$

for every edge $u \xrightarrow{e_1} x_{uv} \in E'$,

$$w'(e_1) + r'(x_{uv}) - r'(u) \leq 1 \ , \tag{17}$$

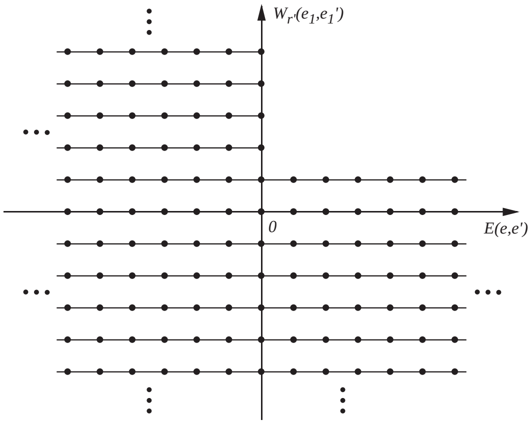

Figure 3. Solution space for Relation (20).

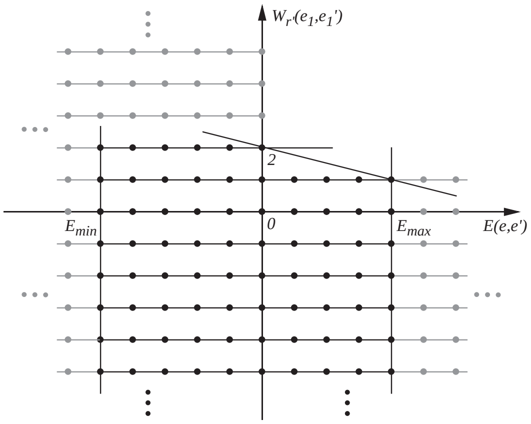

Figure 4. Equivalent convex solution space.

for every pair of edges $u \xrightarrow{e_1} x_{uv}, x_{uv} \xrightarrow{e_2} v \in E$,

$$w'(e_2) + r'(v) - r'(x_{uv}) \le F \cdot (w'(e_1) + r'(x_{uv}) - r'(u)) \ , \qquad (18)$$

where $F = \max\{W(u,v) + W(v,u) : u, v \in V\}$, and for every pair of edges $? \xrightarrow{e} u, v \xrightarrow{e'} ? \in E$,

$$E(e,e') > 0 \quad \Rightarrow \qquad (19)$$
$$W_{r'}(u,v) \ge 1 \text{ or } (w_{r'}(e_1) = 0 \text{ or } w_{r'}(e_1') = 0)$$

where $E(e,e') = D(u,v) + s_M(e) - s_m(e') - c$ and $E(e,e') = -(\Delta(u,v) + s_m(e) - s_M(e'))$ for the setup and hold constraints, respectively. □

Relation (19) is recast as an equivalent disjunction in the following lemma.

Lemma 5 *For every pair of edges $? \xrightarrow{e} u, v \xrightarrow{e'} ? \in E$, Relation (19) is equivalent to the disjunction*

$$E(e,e') \le 0 \ \text{ or } \ w'_{r'}(e_1) + w'_{r'}(e_1') - W_{r'}(u,v) \le 1 \ , \qquad (20)$$

where $E(e,e') = D(u,v) + s_M(e) - s_m(e') - c$ and $E(e,e') = -(\Delta(u,v) + s_m(e) - s_M(e'))$ for the setup and hold constraints, respectively. □

The solution space of Relation (20) is described by the solid lines in Figure 3. This space is not convex and precludes the use of convex programming techniques.

7. Mixed-Integer Linear Program

This section presents a set of $O(E^2)$ constraints that ensure correct timing under simultaneous retiming and clock skew scheduling. These constraints are obtained by restricting the solution space of the constraints in Lemma 4 while maintaining their feasibility. The final constraints set comprises linear inequalities with integer and real unknowns.

The following lemma gives upper bounds on the quantity $w'_{r'}(e_1) + w'_{r'}(e_1') - W_{r'}(u,v)$ from Relation (20) that restrict the solution space in the first and second quadrant.

Lemma 6 *Let $r' : V' \longrightarrow$ **Z** be a retiming function that satisfies the conditions in Lemma 4. Then, for every pair of edges $? \xrightarrow{e} u, v \xrightarrow{e'} ? \in E$,*

$$w'_{r'}(e_1) + w'_{r'}(e_1') - W_{r'}(u,v) \le 2 \ , \qquad (21)$$

$$w'_{r'}(e_1) + w'_{r'}(e_1') - W_{r'}(u,v) \le 2 - \frac{E(e,e')}{E_{\max}(e,e')} \ , \qquad (22)$$

where $E_{\max}(e,e')$ is an upper bound of $E(e,e')$ that depends on the maximum possible clock skew values, as they are determined by the largest realizable chip die size. □

The convex solution space derived from the bounds in Lemma 6 is illustrated in Figure 4. The bold line segments represent possible solutions. The shaded lines and points denote the points of the original solution space that are now excluded. The horizontal line in the second quadrant arises from Inequality (21), and the sloped upper bound in the first quadrant arises from Inequality (22). The two vertical lines correspond to the bounds on $E(e,e')$.

Based on Lemmas 5 and 6, the simultaneous retiming and clock scheduling problem can now be recast as a mixed-integer linear program with $O(E^2)$ constraints.

Theorem 7 *Let $G = \langle V, E, d, w\rangle$ be a synchronous circuit, and let c and t be given constants. Moreover, let $r : V \longrightarrow$ **Z** be a retiming function, let $s_M : E \longrightarrow$ **R** be an assignment of maximum clock delays, and let $s_m : E \longrightarrow$ **R** be an assignment of minimum clock delays. Then the retimed circuit G_r is well-formed and achieves a clock period $\Phi(G_r) \le c$ with tolerance t if and only if for every edge $u \xrightarrow{e} v \in E$,*

$$s_m(e) \le s_M(e) - t \ , \qquad (23)$$

for every edge $u \xrightarrow{e} v \in E'$*,*

$$w'(e) + r'(v) - r'(u) \geq 0 \;, \qquad (24)$$

for every edge $u \xrightarrow{e_1} x_{uv} \in E'$*,*

$$w'(e_1) + r'(x_{uv}) - r'(u) \leq 1 \;, \qquad (25)$$

for every pair of edges $u \xrightarrow{e_1} x_{uv}, x_{uv} \xrightarrow{e_2} v \in E$*,*

$$\begin{aligned} & w'(e_2) + r'(v) - r'(x_{uv}) \leq \\ & \quad F \cdot (w'(e_1) + r'(x_{uv}) - r'(u)) \;, \end{aligned} \qquad (26)$$

where $F = \max\{W(u,v) + W(v,u) : u, v \in V\}$*, and for every pair of edges* $? \xrightarrow{e} u, v \xrightarrow{e'} ? \in E$*,*

$$E(e,e') \leq E_{\max}(e,e') \;, \qquad (27)$$

$$E(e,e') \geq E_{\min}(e,e') \;, \qquad (28)$$

$$w'_{r'}(e_1) + w'_{r'}(e'_1) - W_{r'}(u,v) \leq 2 \;, \qquad (29)$$

$$w'_{r'}(e_1) + w'_{r'}(e'_1) - W_{r'}(u,v) \leq 2 - \frac{E(e,e')}{E_{\max}(e,e')} \;, \qquad (30)$$

where $E(e,e') = D(u,v) + s_M(e) - s_m(e') - c$ *and* $E(e,e') = -(\Delta(u,v) + s_m(e) - s_M(e'))$ *for the setup and hold constraints, respectively.*

8. Experimental Results

This section presents results from the application of simultaneous retiming and clock scheduling on LGSynth93 and ISCAS89 benchmark circuits. Each test circuit was optimized to achieve maximum delay tolerance with a clock period $1.1 \times c_{\min}$, where $c_{\min}$ was the shortest clock period of the original circuit. The following experimental procedure was applied. Each circuit was optimized using retiming, clock scheduling, and simultaneous retiming and clock scheduling. An additional optimization heuristic was applied, in which the original circuit was first retimed for maximum tolerance with zero skew, and clock skews were subsequently scheduled to increase tolerance further.

Our results are listed in Table 1. The first three columns give the name and size of each test circuit. The fourth column gives the target clock period. The fifth column gives the maximum tolerance τ of the original circuit with zero skew. The sixth column gives the maximum tolerance τ_s of the original circuit after clock scheduling. (Retiming results are omitted, since clock scheduling always resulted in circuits with greater tolerance.) The seventh column gives the maximum tolerance $\tau_{r;s}$ achieved by applying the two optimizations in sequence. The eighth column gives the relative improvement achieved over separate scheduling, and the ninth column gives the runtime of the heuristic. The tenth column gives the maximum tolerance τ_{rs} that was achieved by simultaneous retiming and clock scheduling. The relative improvements achieved over separate scheduling are given in the eleventh column. The runtimes of the combined optimization are listed in the last column.

Simultaneous retiming and clock scheduling improved the tolerance of all test circuits and resulted in significant improvements for most of them. For half of the circuits in our test suite, relative improvements over scheduling were at least 23%. For about two thirds of the circuits, improvements exceeded 11%. Our sequential retiming and clock scheduling heuristic improved the maximum tolerance of most test circuits. For one quarter of the circuits, relative improvements exceeded 10%. The runtime of this optimization was comparable to scheduling. Our experiments were performed on an Intel Pentium II with 128MB of main memory. Our simultaneous retiming and clock scheduling algorithm was terminated if no further improvements were achieved for 10 hours of execution.

Gate delays were calculated using the formula $a + b \cdot (fanout + rand)$. The parameters a and b denote the intrinsic gate delay and the delay increment of a single gate load, respectively. Their values were obtained from the library `iwls93.mis2lib` in the LGSynth93 benchmark. The parameter $rand$ was a uniformly distributed random number that introduced variation to gate delays. The range of $rand$ was [-1,0] and [0,1] for minimum and maximum propagation delays, respectively.

Our simultaneous retiming and clock scheduling algorithm explores the solution space using a branch and bound approach. During its execution, it maintains a permissible range for the retiming value of each vertex. Once the retiming function is fixed, the clock delays are computed using a Bellman-Ford single-source shortest-paths algorithm. The optimal tolerance is determined by iterating this algorithm in a binary search. The overall complexity of our algorithm is exponential in the worst case. When register mobility is constrained by considering loops, however, the permissible region of most vertices becomes very small.

9. Conclusion

This paper explores the application of retiming and clock scheduling for maximizing the tolerance of synchronous circuits to delay variations. When both long and short paths are considered, we show that the combined optimization can result in more delay tolerant circuits than if either of the two optimizations is applied separately. Moreover, we give a MILP formulation of the simultaneous retiming and clock scheduling problem. Our experiments show that retiming and clock scheduling can significantly increase the maximum tolerance of benchmark circuits to delay variations.

Circuit	nodes	edges	c	τ	τ_s	$\tau_{r;s}$	$\tau_{r;s}/\tau_s - 1$	CPU ($\tau_{r;s}$)	τ_{rs}	$\tau_{rs}/\tau_s - 1$	CPU (τ_{rs})
							(%)	(sec)		(%)	(sec)
daio	17	30	2.91	0.00	0.76	0.76	0	0.1	0.79	4	1
dk27	24	254	3.74	0.21	0.59	0.64	8	0.4	0.64	8	265
tav	26	59	3.36	0.00	1.05	1.08	2	0.4	1.22	15	2
bbtas	31	87	3.78	0.20	0.46	0.63	39	0.9	0.63	39	6914
s208	37	112	4.67	0.00	1.64	1.67	2	1.5	1.67	2	29086
dk512	39	107	4.11	0.23	0.63	0.70	11	1.4	0.70	11	77313
dk17	40	114	4.69	0.26	0.52	0.58	0	1.4	0.60	3	6505
s420	46	177	5.67	0.00	1.67	1.76	5	4.4	2.28	36	53360
dk15	49	154	5.26	0.29	1.55	1.55	0	2.2	2.05	32	202
dk14	69	238	5.64	0.35	1.27	1.40	11	7.7	1.62	28	116412
ex4	70	207	5.83	0.00	0.66	0.70	7	7.2	0.83	26	38346
opus	71	242	8.53	0.47	2.11	2.11	0	10.1	2.69	27	7370
ex6	102	379	7.28	0.41	1.02	1.06	4	27.6	1.16	14	76091
dk16	120	567	8.89	0.49	1.38	1.38	0	133.0	1.45	5	12030
ex1	193	887	14.63	0.00	2.16	2.16	0	367.0	2.94	36	49821
s713	377	590	41.30	0.00	3.42	3.78	10	546.0	4.21	23	48871

Table 1. Tolerance to delay variations for original and optimized circuits.

Acknowledgments

This research was supported in part by the National Science Foundation under Grant No. MIP-9423886 and Grant No. MIP-9610108, a grant from the New York State Science and Technology Foundation, and by grants from the Xerox, IBM, and Intel Corporations.

References

[1] S. Chakradhar and S. Dey. Resynthesis and retiming for optimum partial scan. In *Proc. 31st ACM/IEEE Design Automation Conf.*, pages 87–93, June 1994.

[2] L.-F. Chao and E. H.-M. Sha. Retiming and clock skew for synchronous systems. In *Proc. International Symp. on Circuits and Systems*, pages 283–286, June 1994.

[3] R. B. Deokar and S. S. Sapatnekar. A graph-theoretic approach to clock skew optimization. In *Proc. International Symp. on Circuits and Systems*, pages 407–410, May 1995.

[4] S. Dey and S. Chakradhar. Retiming sequential circuits to enhance testability. In *Proc. 12th IEEE VLSI Test Symp.*, pages 28–33, April 1994.

[5] J. P. Fishburn. Clock skew optimization. *IEEE Trans. on Computers*, 39(7):945–951, July 1990.

[6] E. G. Friedman. *Clock Distribution Networks in VLSI Circuits and Systems*. IEEE Press, 1995.

[7] A. T. Ishii, C. E. Leiserson, and M. C. Papaefthymiou. Optimizing two-phase, level-clocked circuitry. *J. ACM*, 41(1):148–199, Jan. 1997.

[8] K. N. Lalgudi and M. C. Papaefthymiou. DELAY: an efficient tool for retiming with realistic delay modeling. In *Proc. 32nd ACM/IEEE Design Automation Conf.*, June 1995.

[9] C. E. Leiserson and J. B. Saxe. Retiming synchronous circuitry. *Algorithmica*, 6(1), 1991.

[10] B. Lockyear and C. Ebeling. Optimal retiming of multi-phase, level-clocked circuits. In *Advanced Research in VLSI and Parallel Systems: Proc. 1992 Brown/MIT Conf.* MIT Press, March 1992.

[11] H.-G. Martin. Retiming by combination of relocation and clock delay adjustment. In *Proc. European Design Automation Conf.*, pages 384–389, September 1993.

[12] J. Monteiro, S. Devadas, and A. Ghosh. Retiming sequential circuits for low power. In *Digest of Technical Papers of the 1993 IEEE International Conf. on CAD*, pages 398–402, Nov. 1993.

[13] J. L. Neves and E. G. Friedman. Optimal clock skew scheduling tolerant to process variations. In *Proc. 33rd ACM/IEEE Design Automation Conf.*, pages 623–628, June 1996.

[14] M. C. Papaefthymiou and K. H. Randall. TIM: A timing package for two-phase, level-clocked circuitry. In *Proc. 30th ACM/IEEE Design Automation Conf.*, June 1993.

[15] N. Shenoy, R. K. Brayton, and A. Sangiovanni-Vincentelli. Retiming of circuits with single phase level-sensitive latches. In *International Conf. on Computer Design*, Oct. 1991.

[16] T. Soyata, E. Friedman, and J. Mulligan. Incorporating interconnect, register, and clock distribution delays into the retiming process. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 16(1):105–120, Jan. 1997.