Retiming and Clock Scheduling for High-Performance Synchronous Circuits

Marios C. Papaefthymiou[1] Eby G. Friedman[2] Xun Liu[2]

[1]Department of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, Michigan 48109

[2]Department of Electrical and Computer Engineering
University of Rochester
Rochester, New York 14627

Abstract

This paper investigates retiming and clock skew scheduling for improving the performance of synchronous circuits. It is shown that when both long and short paths are considered, circuits optimized by the simultaneous application of retiming and clock scheduling can achieve shorter clock periods than optimized circuits generated by applying either of the two techniques separately. A novel mixed-integer linear program is given for the problem of simultaneous retiming and clock scheduling with a target clock period and tolerance to delay variations under setup and hold constraints. Experiments with LGSynth93 and ISCAS89 benchmark circuits demonstrate the effectiveness of simultaneous retiming and clock scheduling. For one third of the test circuits, the operating frequency increased by at least 14% over the optimized circuits obtained by applying retiming or clock scheduling separately.

1 Introduction

Retiming improves the speed of a digital circuit by relocating its storage elements while preserving the functionality of the original design. Clock scheduling achieves the same effect as retiming by introducing skew between the clock signals that control the timing of a circuit's storage elements. Significant research has been devoted to each of the two optimizations separately. The investigation of the simultaneous application of these two optimization techniques has been limited, however.

This paper explores the simultaneous application of retiming and clock scheduling under setup and hold constraints for increasing the operating frequency of synchronous digital circuits. This combined application adds flexibility during circuit synthesis and, as we demonstrate by a simple example, is more effective than if either of the two optimizations is applied separately.

Two main analytical contributions are described in this paper. First, a set of $O(E^2)$ constraints is given for the problem of simultaneous retiming and clock scheduling with a target clock period and delay tolerance. This problem is subsequently formulated as a mixed-integer linear program (MILP) that can be solved using branch-and-bound or MILP solvers. When combined with binary search, these solvers can maximize tolerance to delay variations or speed of operation.

Preliminary simulations with benchmark circuits from the LGSynth93 and ISCAS89 suite show that simultaneous retiming and clock scheduling can yield significantly faster circuits than the independent application of the two optimization techniques. For one third of the test circuits, operating frequency improved by at least 14% over separate retiming or clock skew scheduling.

Due to their complementary nature, retiming and clock scheduling have been usually investigated separately. Retiming has been investigated for a variety of clocking disciplines [7, 9, 10, 14], delay models [8, 15], and optimization objectives [1, 4, 11, 13]. A linear programming formulation of the clock scheduling problem was first described by Fishburn in [5]. 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 considers only setup violations, however.

The remainder of this paper has six sections. Section 2 demonstrates the performance gains achievable by combining retiming and clock scheduling. Background material is presented in Section 3. Section 4 gives a shortest-paths formulation for the clock skew scheduling problem. A set of nonconvex conditions for simultaneous retiming and clock scheduling under setup and hold constraints is presented in Section 5. An equivalent set of constraints that can be solved using general mixed-integer linear programming techniques is given in Section 6. Simulation results on benchmark circuits are presented in Section 7.

2 Motivation

The example circuit in Figure 1 illustrates the effectiveness of simultaneous retiming and clock scheduling. Each node represents a block of combinational logic. Each pair x/y denotes the minimum and maximum propagation delay of the signals through the corresponding node. The pair $[x, y]$ next to each register represents the permissible clock skew range [12] associated with each combinational data path that terminates at that register.

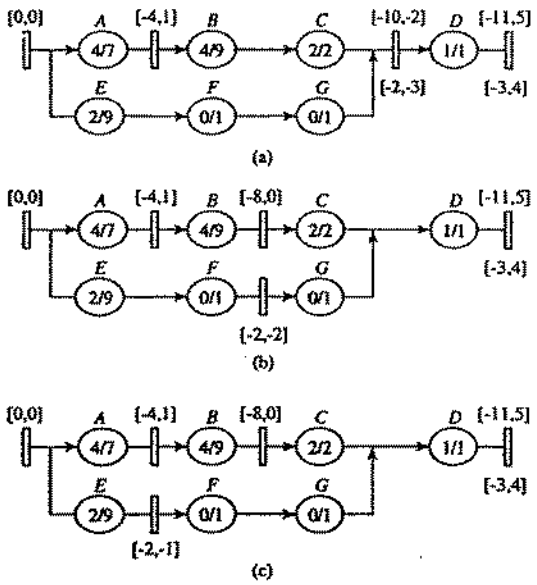

Figure 1: (a) Original circuit. (b) Fastest retimed circuit with zero skew. The register at F's output has no tolerance to variations in the clock delay. (c) Fastest retimed circuit with nonzero skew.

The original circuit depicted in Figure 1(a) achieves a clock period of $\Phi = 11$ tu (time units) when clock skew is zero. When the clock signal at the input of D is delayed by 2 tu, the circuit achieves a shorter clock period, $\Phi = 9$ tu. No schedule of clock skews exists that can result in $\Phi = 8$ tu, because the difference between the maximum and the minimum propagation delay along the path EFG is $(9 + 1 + 1) - 2 = 9 > 8$ tu. Thus, the associated clock skew range $[-2, -3]$ is empty. Intuitively, as the amount of time available for propagating a slow signal along EFG is increased, this signal can be affected by a subsequent fast signal.

Figure 1(b) shows a retimed version of the original circuit, obtained by shifting the register at the outputs of C and G to the outputs of B and F. With zero skews, this circuit achieves $\Phi = 10$ tu. With nonzero skews, it achieves $\Phi = 8$ tu. In the latter case, the clock signal at the output of F is delayed by 2 tu and has zero tolerance to clock delay variations. With zero skews, no retimed circuit can achieve a clock period shorter than 9 tu, the maximum propagation delay of B.

Another retimed version of the original circuit that can achieve $\Phi = 8$ tu with nonzero clock skew is shown in Figure 1(c). In this case, the clock signal at the output of E can be delayed by 1 tu to 2 tu, and thus the tolerance in the delay variation of the clock line is 1 tu.

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. For each vertex $v \in V$, a nonnegative weight $d(v)$ denotes the propagation delay through the corresponding logic block. (It is straightforward to extend our results to encompass maximum and minimum propagation delays $d_{\max}(v)$ and $d_{\min}(v)$, respectively.)

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, the total register count is strictly positive.

3.2 Retiming

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

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

When a vertex v is retimed by $r(v)$, 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

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

An important implication of Equation (1) is that for every vertex pair u, v in V, the register count $w_r(p)$ along *any* path $u \overset{p}{\rightsquigarrow} v$ depends solely on its two endpoints:

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

where $w(p) = \sum_{e \in p} w(e)$. Therefore, for any given vertices u and v in V, the maximum change in the register count along any path $u \overset{p}{\rightsquigarrow} v$ is given by

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

Thus, the only paths $u \overset{p}{\rightsquigarrow} v$ 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

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

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

where $\sum_{x \in p} d(x)$, give 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 circuit G_r is always some element in the $O(V^2)$-size set of $D(u,v)$'s.

When only long paths are considered, a retimed circuit that achieves a given clock period c can be computed in $O(VE)$ steps by an iterative relaxation algorithm akin to the Bellman-Ford algorithm. 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

The arrival times of the clock signal at a circuit's registers differ due to various factors such as differences in interconnect delay, parasitic impedances, and process parameters variations. The difference between the arrival times of a clock signal at two sequentially-adjacent registers in a circuit is known as the clock skew between these two registers [6].

A *clock schedule* of an edge-triggered circuit $G = \langle V, E, d, w\rangle$ is a real-valued edge-labeling $s : E \rightarrow \mathrm{R}$. This labeling describes the propagation delay from the global clock source to each wire e in a circuit. By adjusting (or tuning) these local clock skews, timing violations can be fixed (or created).

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 relative placement of the registers was assumed to be fixed. Algorithms for scheduling local clocks to improve the tolerance of a circuit to process parameter variations are described in [12].

4 Clock scheduling constraints

The following theorem captures the timing conditions that must be satisfied by a clock schedule in order to achieve a target clock period with a specified tolerance.

Its proof follows from the linear program in [5].

Theorem 1 *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 $s_m : E \to R$ and $s_M : E \to R$ such that for each edge $u \xrightarrow{e} v$,*

$$s_m(e) \leq s_M(e) - t \ , \tag{7}$$

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

$$s_M(e') \leq s_m(e) + \Delta(u,v) \ , \tag{8}$$

$$s_M(e) \leq s_m(e') + c - D(u,v) \ . \tag{9}$$

The $O(E^2)$ inequalities in Theorem 1 can be computed in $O(E^2)$ time using the Bellman-Ford single-source shortest-paths algorithm. The same algorithm can be used to solve them in $O(E^3)$ steps. For a given clock period c, a schedule with maximum tolerance can be determined by a binary search in the clock skew range.

5 Clock scheduling and retiming constraints

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

Theorem 2 *Let $G = \langle V, E, d, w \rangle$ be a synchronous circuit, and let c and t be given constants. Moreover, let $r : V \to Z$ be a retiming function, let $s_M : E \to R$ be an assignment of maximum clock delays, and let $s_m : E \to R$ 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 $u \xrightarrow{e} v \in E$,*

$$s_m(e) \leq s_M(e) - t \ , \tag{10}$$

$$w(e) + r(v) - r(u) \geq 0 \ , \tag{11}$$

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

$$E(e,e') > 0 \ \Rightarrow \ W_r(u,v) \geq 1 \text{ or } (w_r(e) = 0 \text{ or } w_r(e') = 0) \ , \tag{12}$$

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 2 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 (12).

A *companion graph* $G' = (V', E', w')$ can be used to transform the timing constraints from Theorem 2 into a mixed-integer linear program. The companion graph G' is obtained from the circuit graph G as described in [8]. In mathematical terms, $G' = (V', E', w')$ is defined as

$$V' = V \cup \left\{x_{uv} : u \xrightarrow{e} v \in E\right\} ,$$

$$E' = \left\{u \xrightarrow{e_1} x_{uv}, x_{uv} \xrightarrow{e_2} v : u \xrightarrow{e} v \in E\right\} ,$$

where for each edge $u \xrightarrow{e} v \in E$, $w'(e_1) = \min\{1, w(e)\}$ and $w'(e_2) = w(e) - \min\{1, w(e)\}$. The following lemma recasts Theorem 2 in terms of the companion graph G' and a corresponding retiming function r'. If r' is known, $r(u)$ is obtained for every $u \in V$ by setting $r(u) = r'(u)$.

Lemma 3 *Let $G = (V, E, d, w)$ be a synchronous circuit graph, let $G' = (V', E', w')$ be its corresponding companion graph, and let c and t be given constants. Moreover, let $r' : V' \rightarrow Z$ be a retiming function, let $s_M : E \rightarrow R$ be an assignment of maximum clock delays, and let $s_m : E \rightarrow R$ 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*

$$s_m(e) \leq s_M(e) - t , \tag{13}$$

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

$$w'(e) + r'(v) - r'(u) \geq 0 , \tag{14}$$

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

$$w'(e_1) + r'(x_{uv}) - r'(u) \leq 1 , \tag{15}$$

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

$$w'(e_2) + r'(v) - r'(x_{uv}) \leq F \cdot (w'(e_1) + r'(x_{uv}) - r'(u)) , \tag{16}$$

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

$$E(e,e') > 0 \;\Rightarrow\; W_{r'}(u,v) \geq 1 \text{ or } (w_{r'}(e_1) = 0 \text{ or } w_{r'}(e_1') = 0) . \tag{17}$$

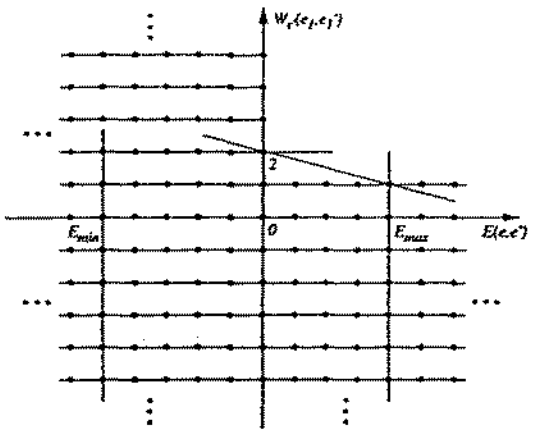

Figure 2: Original and equivalent convex solution space.

The solution space of Lemma 3 can be simplified by reducing the number of disjunctions in Relation (17).

Lemma 4 *Relation (17) is equivalent to the single disjunction*

$$E(e,e') \leq 0 \quad or \quad w'_r(e_1) + w'_r(e'_1) - W_r(u,v) \leq 1 \ . \tag{18}$$

The solution space of the constraints imposed by Relation (18) is described by the horizontal lines in Figure 2. In its current form, this space precludes the use of convex programming solution techniques.

6 Mixed-integer linear program

The following lemma gives two upper bounds on the quantity $w'_r(e_1) + w'_r(e'_1) - W_r(u,v)$ in Relation (18). These bounds restrict the solution space of the constraints in Lemma 3 while maintaining their feasibility.

Lemma 5 *Let $r' : V' \rightarrow Z$ be a retiming function that satisfies the conditions in Lemma 3. Then, for every pair of edges $? \xrightarrow{e} u, v \xrightarrow{e'} ? \in E$,*

$$w'_r(e_1) + w'_r(e'_1) - W_r(u,v) \leq 2 \ , \tag{19}$$

$$w'_r(e_1) + w'_r(e'_1) - W_r(u,v) \leq 2 - \frac{E(e,e')}{E_{\max}(e,e')} \ , \tag{20}$$

where $E_{\max}(e,e')$ is an upper bound of $E(e,e')$ that depends on the maximum possible value of the clock skews.

Circuit	nodes	edges	clock cycle (orig.)	clock cycle (sched.)	clock cycle (ret.)	clock cycle (ret. & sched.)	1-(7)/(5) (%)	CPU (sec)
lion	16	34	28.11	14.49	27.43	14.3	1	51
daio	17	30	26.58	15.58	12.5	6.91	81	6
tav	26	59	14.87	9.87	10.14	7.6	30	14
bbtas	31	87	52.24	47.75	46.08	44.05	5	36152
s208	37	112	31.04	20.37	30.85	17.93	14	24598
dk15	49	154	35.52	20.81	34.36	20.61	1	5512
ex4	70	207	130.6	126.5	130.6	125.5	1	25973
s208.1	104	181	4.58	1.57	3.79	1.54	2	36027

Table 1: Performace improvements by retiming and clock scheduling.

The solution space derived from Lemma 5 is shown in Figure 2. The bold line segments represent possible solutions of the problem. The shaded lines and points denote the parts of the original solution space that can be safely excluded.

From Lemmas 4 and 5, it follows that simultaneous retiming and clock scheduling can be expressed as a mixed-integer linear program with $O(E^2)$ constraints.

7 Experimental results

The combined retiming and clock scheduling optimization has been applied to a collection of LGSynth93 and ISCAS89 benchmark circuits. First, the shortest clock period of each original circuit was computed. Subsequently, each test circuit was optimized by retiming, clock scheduling, and simultaneous retiming and clock scheduling to achieve minimum cycle time.

Our experimental results are summarized in Table 1. For one third of the test circuits, relative improvements were at least 14%. The combined optimization improved the operating speed of `daio` and `tav` by 81% and 30%, respectively, over the best result obtained by applying either of the two techniques separately. Our programs executed on a SUN Ultra I SparcStation with 64MB of main memory. Although many CPU times are in tens of hours, more than 10% improvements can be achieved within one CPU hour for most test circuits.

Gate delays were calculated using the linear delay formula $a+b\cdot(fanout+rand)$, where a and b denote the intrinsic delay and the delay increment parameter of a single-gate load, respectively. The parameters a and b were obtained using the library `iwls93.mis2lib`. The parameter $rand$ is a random number that introduces delay variations and is distributed uniformly in the range [-1, 1].

Acknowledgments

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

References

[1] S. Chakradhar and S. Dey. Resynthesis and retiming for optimum partial scan. In *Proceedings of the 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 *Proceedings of the International Symposium 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 *Proceedings of the International Symposium on Circuits and Systems*, pages 407–410, May 1995.

[4] S. Dey and S. Chakradhar. Retiming sequential circuits to enhance testability. In *Proceedings of the 12th IEEE VLSI Test Symposium*, 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. *Journal of the ACM*, 41(1):148–199, January 1997.

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

[9] C. E. Leiserson and J. B. Saxe. Retiming synchronous circuitry. *Algorithmica*, 6(1), 1991. Also available as MIT/LCS/TM-372.

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

[11] 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, November 1993.

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

[13] M. C. Papaefthymiou and K. H. Randall. TIM: a timing package for two-phase, level-clocked circuitry. In *Proceedings of the 30th ACM/IEEE Design Automation Conf.*, June 1993. Also available as an MIT VLSI Memo 92–693, October 1992.

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

[15] 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, January 1997.