A Practical Algorithm for Retiming Level-Clocked Circuits[1]

Naresh Maheshwari Sachin S. Sapatnekar
Department of Electrical and Computer Engineering
Iowa State University, Ames, IA 50011.
naresh@iastate.edu, sachin@iastate.edu

Abstract

A new approach for fast retiming of level-clocked circuits is presented here. The method relies on the relation between clock skew and retiming, and computes the optimal skew solution to translate it to a retiming. Since clock skew optimization operates on the latches (rather than the gates as in conventional retiming), it is much faster because of a smaller problem size; the translation to the retiming solution is computationally cheap. The minimum period retiming for each of the ISCAS89 circuits was obtained within minutes by this algorithm.

1 Introduction

Timing optimization plays a vital role in the synthesis of VLSI circuits. One method that is of great interest to the design and CAD community is the procedure of retiming [5], which takes an unoptimized circuit and relocates the memory elements (such as latches, flip-flops or registers) to achieve a specified or the minimum clock period.

Much work has been done in retiming circuits with edge-triggered flip-flops (FF's), and fast algorithm like [9, 2] are now available. However the current methods for the much harder problem of retiming circuits with level-triggered latches like [4, 6, 7] do not report results on large circuits and may be unable to handle them due to the large computational complexity. Level-clocked circuits have a potential to run at a faster clock than edge-triggered circuits. Hence there is a need for fast automation tools to handle retiming of level-clocked circuits. This work is motivated by such a need.

This approach is based on the relation between clock skew optimization [3] and retiming. This relation was utilized in [2] for fast retiming of circuits with edge-triggered FF's. We show here that a similar relation is valid for level-clocked circuits, and that moving a latch across a gate is equivalent to applying a skew at that latch. We use this relation to develop an algorithm that translates the calculated optimum skew values into a retiming solution for level-clocked circuits. As will be shown in our experimental results, the computational expense of this algorithm is very low.

The objective here is to retime a level-clocked circuit to achieve a specified period or the minimum possible period. The solution is divided into two phases:

Phase A : The clock skew optimization problem for a level-clocked circuit is solved with the objective of minimizing the clock period.

Phase B : The target clock period is known, and the skew solution is translated to a retimed circuit by relocating latches across gates in an attempt to set the values of all latch skews to be as close to zero as possible.

To find a retiming solution for a specified (not necessarily minimum) clock period, we simply execute Phase B.

After Phase B has been performed, the designer may choose to achieve the optimal clock period by using a combination of clock skew and retiming. Alternatively, any skews that could not be set exactly to zero could now be forced to zero. This may cause the clock period to increase. An upper bound on this increase will be derived in Theorem 4; and it will be seen that this increase is generally small.

The paper is organized as follows: Section 2 presents the clock model, while Section 3 presents the timing constraints for level-clocked circuits. Section 4 presents the equivalence between retiming and clock skew. Phases A and B of the minimum period retiming algorithm follow in Sections 5 and 6, followed by theoretical results on the optimality in Section 7. Finally, we present experimental results in Section 8 and conclude the paper in Section 9.While we have attempted to present a complete description of the algorithm, some details have been omitted due to lack of space.

[1] This work was supported in part by the National Science Foundation under award MIP-9502556.

2 Clock model

In this work, we have adopted the clock model of Sakallah, Mudge and Olukotun [8], and we describe it here for completeness. A k-phase clock is a set of k periodic signals, $\Phi = \{\phi_1 \ldots \phi_k\}$ where ϕ_i is referred to as phase i of the clock Φ. All of the ϕ_i's have the same clock period T_Φ, and each phase i has an active interval of duration T_{ϕ_i} and a passive interval of duration $(T_\Phi - T_{\phi_i})$. The latches controlled by a clock phase are enabled during the active interval and disabled during the passive interval. When the clock period, T_Φ, is changed, the relative ratios of the active intervals of each phase are scaled proportionately. Associated with each phase is a *local time zone*, shown in Figure 1, such that the passive interval starts at time $t = 0$, the enabling edge occurs at time $(T_\Phi - T_{\phi_i})$ and the latching edge occurs at time T_Φ. There is also a global time reference and values e_i denote the time, relative to this global time reference, when the phase ϕ_i ends. Phases are ordered so that $e_1 \leq e_2 \ldots \leq e_{k-1} \leq e_k = T_\Phi$. The phases are numbered modulo-k, i.e., $\phi_{k+1} = \phi_1$ and $\phi_{1-1} = \phi_k$.

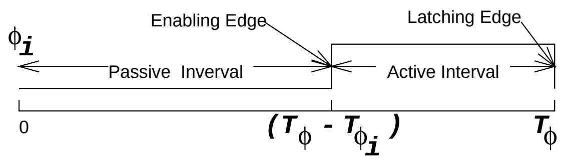

Figure 1. Phase i of a k-phase clock (all times in local time zone).

A phase shift operator $E_{i,j}$, shown in Figure 2, is defined as follows:

$$E_{i,j} = \begin{cases} (e_j - e_i) & \text{for } i < j \\ (T_\Phi + e_j - e_i) & \text{for } i \geq j \end{cases} \tag{1}$$

Note that $E_{i,j}$ takes on positive values. When subtracted from a time point in the current time zone of ϕ_i, it changes the frame of reference to the next local time zone of ϕ_j.

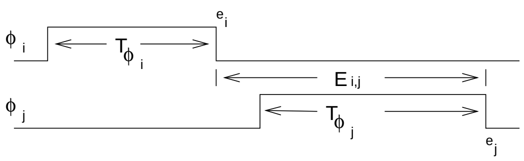

Figure 2. The phase shift operator.

We now augment the Sakallah-Mudge-Olukotun model with our own notation. Let the circuit have l latches numbered $1, \cdots, l$, and let us associate a skew S_i with latch i. S_i represents the time by which the clock is delayed in arriving at the latch i, relative to a fixed reference (typically the primary inputs and outputs of the circuit) which is set to zero. Note that the skew values here are not physical skews that will be applied to the final circuit, but conceptual ideas that will eventually help us to achieve a retiming solution. No restrictions are placed on the value of S_i, i.e. $-\infty \leq S_i \leq \infty$. Each latch i is clocked by exactly one phase of the clock Φ, which is denoted by $p(i)$.

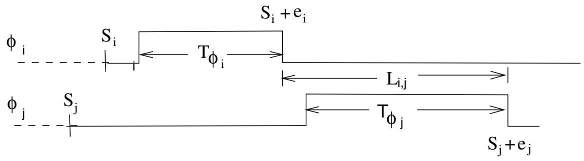

Figure 3. The latch shift operator.

We define a *latch shift* operator $L_{i,j}$, shown in Figure 3, much like the phase shift operator. This operator converts time from the local time zone of latch i to the local time zone of latch j and is defined as

$$L_{i,j} = \begin{cases} (S_j + e_{p(j)}) - (S_i + e_{p(i)}) & \text{for } i < j \\ (T_\Phi + (S_j + e_{p(j)}) - (S_i + e_{p(i)}) & \text{for } i \geq j \end{cases} \tag{2}$$

which can be rewritten in terms of the phase shift operator as

$$L_{i,j} = (S_j - S_i) + E_{p(i),p(j)} \tag{3}$$

Each latch i also has an associated latest arrival time, A_i, and a latest departure time, D_i, in its local time zone.

In this paper, we consider k-phase level-clocked circuits with no overlaps or underlaps. We expect that it would not be difficult to generalize this algorithm to handle overlaps or underlaps exist [4]. The circuits are assumed to be well-formed [6]. We neglect to consider latch setup and hold times, since handled by including the setup times in the path delays and the hold time in the clock periods.

3 Level-clocked timing constraints

We now enumerate the set of timing constraints, that dictate the correct operation of a level-clocked circuit in the presence of skews,. The long path constraint[2] for any path from latch i to latch j with a delay of d_{ij} is then given by

$$\begin{aligned} D_i + d_{ij} - L_{i,j} &\leq A_j \\ T_\Phi - T_{p(i)} &\leq D_i \leq T_\Phi \\ A_i &\leq D_i \end{aligned} \tag{4}$$

which can be rewritten as

$$\begin{aligned} (S_i + D_i) + d_{ij} - E_{p(i),p(j)} &\leq (S_j + D_j) \\ T_\Phi - T_{p(i)} &\leq D_i \leq T_\Phi \end{aligned} \tag{5}$$

[2] We do not consider short path constraints here, and rely on Theorem 1 in [6], which assures us that in our final retimed circuit with zero skew, there will be no short path violations.

To make the discussion simpler we subtract T_Φ from both sides of the equation, leaving it unchanged, and substitute

$$X_i = (S_i + D_i - T_\Phi) \tag{6}$$

to get

$$X_i + d_{ij} - E_{p(i),P(j)} \leq X_j \tag{7}$$
$$-\infty \leq X_i \leq \infty \tag{8}$$

We refer to X_i as the *Global Departure Time* (GDT). Equation (7) can be written as the following set of difference constraints and solved efficiently.

$$X_i - X_j \leq E_{p(i),p(j)} - d_{ij} \tag{9}$$

For a given circuit d_{ij} is constant and for a given clocking scheme Φ (with a given clock period) $E_{p(i),p(j)}$ is also constant. Even though the expressions for X_i contain T_Φ, the difference $X_i - X_j$ is independent of T_Φ.

At this time, we also note the relation between the GDT, X_i at a latch i, and the corresponding minimum magnitude skew, S_i:

$$S_i = \begin{cases} X_i & \text{if } X_i \geq 0 \\ 0 & \text{if } -T_{\phi_i} \leq X_i \leq 0 \\ X_i + T_{\phi_i} & \text{if } -T_{\phi_i} > X_i \end{cases} \tag{10}$$

4 Equivalence between skew and retiming

A formal presentation of the equivalence between clock skew and retiming for edge triggered FF's is presented in [2]. We suggest a similar relation between retiming and skew for level-clocked circuits.

An FF can be conceptualized as a level sensitive latch with a very small active interval. If we were allowed to apply arbitrary skews at each latch, we could adjust the skew, S_i, of a latch so as to force $D_i = T_\Phi$, which is same as a negative edge triggered FF. Since $X_i = S_i + D_i - T_\Phi$, this gives us $S_i = X_i$. Hence, for Phase A we can think of X_i for latches as skews for FF's and thus get the optimal clock period in a manner similar to [2], with the difference that instead of the clock period, we use the phase shift operator, $E_{i,j}$.

Note that in reality, we are not restricted to setting $D_i = T_\Phi$, and that we can reduce D_i by as much as T_{ϕ_i} and increase S_i by the same amount, keeping X_i constant, as described in Equation (10). Therefore, we can absorb a skew of up to T_{ϕ_i} in the D_i without violating the long path constraint. Thus, in our model, level sensitive latches can be conceptualized as FF's that have a capacity to absorb some skew. Using this rationalization, we state the following theorem.

Theorem 1 For a circuit that operates at a clock scheme Φ satisfying the long path delay constraints,

(a) retiming a latch by moving it from the output of a single-input gate of delay d_1 to its input(s) is equivalent to decreasing its GDT by d_1.

(b) retiming a latch by moving it from the input(s) of a single-input gate of delay d_2 to its output is equivalent to increasing its GDT by d_2.

This result will be generalized to multi input gates later in Theorem 2.

Therefore, if one were to calculate the optimal clock GDT's, one could retime the circuit by moving latches with positive (negative) GDT's from the output to the input (input to the output) until the GDT's at the latches are nearly equal to zero. It must be noted that since gate delays take on discrete values, it can not be guaranteed that the GDT at a latch can always be reduced to zero through retiming operations. However, unlike FF's, latches have the ability to absorb some skew, and it is therefore possible to reduce the real skew S_i to a magnitude smaller than that of X_i, without changing the GDT; simply by changing D_i. Since D_i's can vary from $T_\Phi - T_{\phi_i}$ to T_Φ we have a "freedom" of T_{ϕ_i} in setting the skew. As will be shown in Section 7, if the maximum gate delay is less than the least T_{ϕ_i}, we can always achieve zero skews because of this freedom, thus achieving the optimal period.

5 Phase A: Optimizing clock skews

Consider a combinational circuit segment that lies between two latches i and j, with phases $p(i)$ and $p(j)$, respectively. As stated earlier, if X_i and X_j are the GDT's at the two latches, then the following inequality must be satisfied:

$$X_i - X_j \leq E_{p(i),p(j)} - d_{ij} \tag{11}$$

where d_{ij} is the maximum delay of any combinational path between latches i and j.

The GDT problem for minimizing the clock period, T_Φ, for a given clocking scheme can be solved by solving the following linear program:

$$\begin{array}{ll} \text{minimize} & T_\Phi \\ \text{subject to} & X_i - X_j \leq E_{p(i),p(j)} - d_{ij} \end{array} \tag{12}$$

for every pair, (i, j) of latches such that there is at least one purely combinational path from latch i to latch j.

For a given circuit and clocking scheme, $E_{p(i),p(j)}$ depends only on the period T_Φ. Therefore, for a constant value of T_Φ, the constraint matrix reduces to a system of difference constraints. The above linear program is similar to that in [1], and can similarly be solved with a binary search on T_Φ, applying efficient graph-theoretic methods at each value of T_Φ to check for feasibility.

6 Phase B: Clock skew minimization

6.1 Introduction

In Phase B, the magnitudes of the clock skew component of GDT's obtained from Phase A are brought as close to zero as possible , by applying retiming transformations. This employs relocation of the latches with nonzero skews across logic gates while maintaining the optimal clock period previously found. Because of the "freedom" provided to D_i by the active interval of clock phase ϕ_i (which allows D_i to be set to any value between $T_\Phi - T_{\phi_i}$ and T_Φ), $S_i = 0$ can be achieved if

$$-T_{\phi_i} \leq X_i \leq 0.$$

Thus, if S_i cannot be set to zero, we try to bring X_i as close to 0 or $-T_{\phi_i}$ as possible so as to minimize the magnitude of the final skew S_i (refer to Equation (10)). After the skew magnitudes have been reduced by as much as possible, the retimed circuit may be implemented either by applying the requisite skews at a latch (to get the minimum achievable clock period), or by setting all skews to zero to get a clock period that is, as will be shown in Section 7, no more than a fixed bound above the optimum. Note that the word "optimum" here refers to the optimum period achievable using skews, and may not be achievable by retiming, which is a discrete optimization.

We will describe the procedure for relocating latches with positive GDT values; the procedure for negative GDT is analogous. Before we proceed, we will state the following result, which is a generalization of Theorem 1.

Theorem 2:
(a) Retiming transformations may be used to move latches from all of the inputs of any combinational segment to all of its outputs, provided all such latches are of the same phase. The equivalent GDT of the relocated latch at output j is given by

$$X_j = \max_{1 \leq i \leq n} (X_i + d_{ij}) \tag{13}$$

where the $X_i, 1 \leq i \leq n$ are the GDT's at the input latches, X_j is the equivalent GDT's at output j, and d_{ij} is the worst-case delay of any path from i to j.
(b) Similarly, if latches are moved from outputs to all inputs, the equivalent GDT at input k is given by

$$X_k = \min_{1 \leq j \leq m} (X_j - d_{kj}) \tag{14}$$

where the $X_j, 1 \leq j \leq m$ are the GDT's at the input latches, X_k is the equivalent GDT at input k, and d_{kj} is the worst-case delay of any path from k to j.

6.2 Positive GDT reduction

6.2.1 Moving a latch across a single gate

In the case of a latch j that has a positive GDT at the end of Phase A, as shown in Figure 4(a). Note that all these latches must be of the same phase because the circuit was well-formed to begin with. Through retiming operations, it is possible to transform the circuit in Figure 4(a) to the one in Figure 4(b); the equivalent GDT at each latch in Figure 4(b) are calculated using Theorem 2(b); the precise procedure will be described later. Therefore, at the output of the gate p, there now exists a set of n "virtual" latches[3] as shown in Figure 4(b), with effective GDT's $X_1, X_2 \ldots X_n$. The GDT's at these latches need to satisfy the constraints:

$$X_i + d(i,p) \quad \leq \quad X_k + E_{p(i),p(k)} \; \forall \; 1 \leq k \leq n \tag{15}$$

where X_i is the GDT at an input latch i of the combinational block to which latch $1 \cdots n$ are output latches, and $d(i,p)$ is the largest combinational delay from latch i to the output of gate p. The above constraints reduce to

$$X_i + d(i,p) \leq \min_{1 \leq k \leq n} (X_k) + E_{p(i),p(k)} \tag{16}$$

We may have one of two scenarios:

(1) If all of the n latches in the array have positive GDT's, the minimum of all the positive GDT latches is positive and hence the set of latches may be moved across the gate p, as illustrated in Figure 4(c). If the sign of the GDT were to change after the relocation, the relocation would not be carried out unless it reduced the magnitude of the skew S_i (calculated using Equation 10). One may also take advantage of slacks in the combinational paths to reduce the GDT's at latches. If input r to gate p has a slack, $slack(r)$ (i.e., the worst-case delay at input r could have been increased by $slack(r)$ without changing the worst-case delay to the output of gate p), then the GDT may be further reduced by $slack(r)$.

(2) If one or more of the "virtual" latches has a negative GDT value, then the GDT at the latch j under consideration is set to zero. This violates no timing constraints, since it leaves the minimum skew at an output of gate p unchanged.

6.2.2 Outline of the minimization procedure

The steps involved in minimizing the GDT's at latches with positive GDT's are outlined below:

[3] We refer to these latches as "virtual" latches because we do not physically move them to the input of gate p at this point.

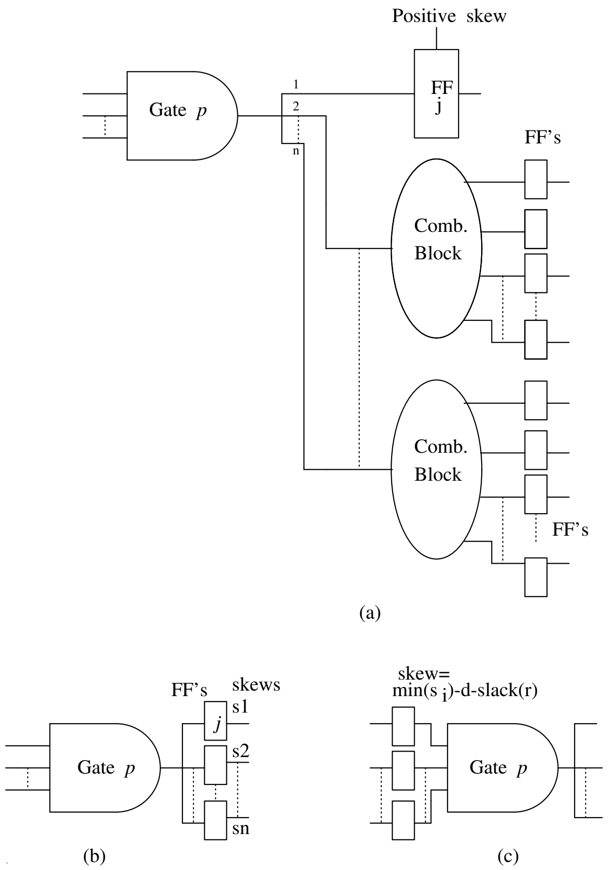

Figure 4. Retiming for a positive skew latch.

Step 1 All latches in the circuit with positive GDT's are placed on a queue, Q.

Step 2 Let j be the latch that is currently at the head of the queue, and p the gate that fans into it. The procedure for finding the equivalent GDT's is as follows. The gate p is added to the tail of a queue R.[4] A PERT/CPM evaluation is employed to trace the transitive fanouts of gate p, up to the point where latches are encountered. When a gate is encountered, it is added to the queue. During the process, we keep track of the worst-case delay, d, from gate p. As a consequence of Theorem 2, if the optimal GDT at a latch is t units, then its equivalent GDT at the output of gate p is $t - d$ units.

Step 3 If any equivalent GDT at a "virtual" latch is negative, then the GDT at j is set to zero and it is not relocated; if not, the GDT after relocation is found as in Step 2. If the magnitude of the skew for this GDT is smaller than that for the current GDT at j, then j and all of the "virtual" latches at the input to p are retimed across p and the new GDT's are found.

Step 4 If the relocated latch has a positive GDT, it is placed at the tail of Q.

Step 5 If Q is not empty, go to Step 2.

7 Properties of the retiming procedure

In this section we give the bound on the optimal clock period achievable by retiming. Proofs are omitted due to lack of space.

Lemma 3 At the end of the retiming procedure in Phase B, the magnitude of skew at each latch is no more than

$$\delta = \max_{1 \leq i \leq k} \left[0, \left| \frac{G - T_{\phi_{p(i)}}}{2} \right| \right] \tag{17}$$

where G is the maximum gate delay.

Theorem 4 If, at the end of the retiming procedure, all skews are set to zero, then the optimal clock period for this circuit is no more than

$$P_{skew} + \max \left[0, k \cdot \left| G - T_{\phi_p(i)} \right| \right] \; \forall i = 1, \cdots, k,$$

where P_{skew} is the optimal clock period found in Phase A, and G is the maximum delay of any gate in the circuit.

8 Experimental Results

The algorithm was implemented as a C program, and could easily handle the entire ISCAS89 benchmark suite. However due to limited space only a representative set of the results is presented.

We present results for a one-phase and a symmetric two-phase clocking scheme [6] (with a 50% duty cycle). The one-phase circuits were obtained from the ISCAS89 circuits (which contain edge-triggered FF's only) by replacing each FF by a level sensitive latch. As in [7] the two-phase circuits were obtained by replacing each FF by a pair of latches. For simplicity the delays, setup time and hold times of latches where taken to be zero, although nonzero values can easily be handled.

The results for unit gate delays were as expected, and the optimal skew period could be achieved after retiming for each circuit (unlike in the edge-triggered case [1]). We will not specifically show those results here; instead, in Tables 1 and 2, we show results for the case where the gate delays were chosen randomly. In this case, it will be seen that the optimal skew period is not always achievable. For each circuit, the tables provide the number of latches $|L|$, the initial

[4] Note that R is distinct from the queue Q.

Table 1. Results for One-Phase Circuits

Circuit	$\lvert L \rvert$	P_{init}	P_{ret}	$Change$	T_{exec}	G
s526n	21	26.0	16.0	38.46%	0.06s	13
s967	29	30.0	28.0	6.67%	0.06s	11
s635	32	189.0	97.0	48.68%	0.07s	2
s938	32	43.0	25.5	40.70%	0.09s	5
s1269	37	74.0	38.7	47.70%	0.15s	24
s4863	104	114.0	59.0	48.25%	0.48s	16
s3271	116	44.7	28.3	36.69%	0.23s	28
s3330	132	46.7	44.0	5.78%	0.32s	32
prolog	136	67.0	47.5	29.10%	0.33s	48
s3384	183	84.0	39.0	53.57%	0.53s	26
s9234.1	211	81.0	81.0	0.00%	0.00s	32
s6669	239	118.7	48.5	59.14%	1.41s	34
s13207	669	95.3	81.5	14.48%	4.42s	37
s38584.1	1426	191.0	183.0	4.19%	2.83s	88
s35932	1728	137.0	110.0	19.70%	25.53s	128

Table 2. Results for Two-Phase Circuits

Circuit	$\lvert L \rvert$	P_{init}	P_{ret}	$Change$	T_{exec}	G
s526n	42	26.0	16.0	38.46%	0.08s	13
s967	58	34.0	28.0	17.65%	0.15s	11
s635	64	189.0	97.0	48.68%	0.09s	2
s938	64	47.0	25.5	45.74%	0.14s	5
s1269	74	81.0	38.7	52.22%	0.24s	24
s4863	208	117.0	59.0	49.57%	0.61s	16
s3271	232	67.0	30.5	54.48%	0.85s	28
s3330	264	70.0	44.0	37.14%	0.89s	32
prolog	272	80.0	51.0	36.25%	0.96s	48
s3384	366	126.0	39.0	69.05%	1.92s	26
s9234.1	422	89.0	81.0	8.99%	1.24s	32
s6669	478	178.0	48.5	72.75%	3.12s	34
s13207	1338	143.0	81.5	43.01%	51.25s	37
s38584.1	2852	191.0	183.0	4.19%	43.47s	88
s35932	3456	137.0	128.0	6.57%	250.85s	128

clock period P_{init}, the final retimed period P_{ret}, percentage improvement in clock period $Change$, the execution time T_{exec} and the maximum gate delay G. For purposes of reference, the minimum gate delay in each circuit was 1 unit. The CPU times are on an HP 735 workstation, and do not include the time spent in reading in the circuit.

For all cases the final clock period is always within $2 \cdot k \cdot \delta$ of the optimal (skew) period, as predicted by the bound in Theorem 4; in fact, in most cases, the optimal Phase A period is achieved. The execution time for most circuits is less than a few seconds, and even the largest circuits run in only a few minutes. In most cases, the optimal skew period was achieved through retiming; the only case where there were significant differences between the skew and retiming periods was s35932, where the granularity of gate sizes is large, as can be seen from the value of G.

9 Conclusion

An approach that takes advantage of the equivalence between retiming and clock skew is presented, and is used for gate-level retiming. The method is shown to be practical and capable of handling large circuits. All of the circuits in the ISCAS89 benchmark suite where easily handled. The use of skew optimization enables handling level sensitive latches like edge triggered FFs, thus avoiding a complicated formulation that is forced to handle critical path propagation over several latches.

The chief reason for the efficiency of this algorithm is that it first takes a global view of retiming by solving the clock skew problem; the number of variables for this problem is the number of *latches*, rather than the number of *gates*, as in a Leiserson-Saxe based approach. In the second phase, local transformations are used to perform the retiming. The logic behind this approach is that in realistic circuits, latches do not have to be moved across large numbers of gates during retiming. Therefore, in practical cases, the latter phase takes only a small amount of computation; this is borne out by our experimental results.

References

[1] R. B. Deokar and S. S. Sapatnekar. A graph-theoretic approach to clock skew optimization. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 1.407–1.410, 1994.

[2] R. B. Deokar and S. S. Sapatnekar. A fresh look at retiming via clock skew optimization. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 310–315, 1995.

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

[4] A. Ishii, C. E. Leiserson, and M. C. Papaefthymiou. Optimizing two-phase, level-clocked circuitry. In *Advanced Research in VLSI and Parallel Systems: Proceedings of the 1992 Brown/MIT Conference*, pages 246–264, 1992.

[5] C. E. Leiserson and J. B. Saxe. Retiming synchronous circuitry. *Algorithmica*, 6:5–35, 1991.

[6] B. Lockyear and C. Ebeling. Optimal retiming of level-clocked circuits using symmetric clock schedules. *IEEE Transactions on Computer-Aided Design*, pages 1097–1109, Sept. 1994.

[7] M. C. Papaefthymiou and K. H. Randall. TIM: A timing package for two-phase, level-clocked circuitry. In *Proceedings of the ACM/IEEE Design Automation Conference*, pages 497–502, 1993.

[8] K. A. Sakallah, T. N. Mudge, and O. A. Olukotun. Analysis and design of latch-controlled synchronous digital circuits. *IEEE Transactions on Computer-Aided Design*, 11(3):322–333, Mar. 1992.

[9] N. Shenoy and R. Rudell. Efficient implementation of retiming. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pages 226–233, 1994.