Timing Optimization by Replacing Flip-Flops to Latches

Ko Yoshikawa Keisuke Kanamaru Shigeto Inui[§]
Yasuhiko Hagihara[§] Yuichi Nakamura[†] Takeshi Yoshimura[‡]

CAD Engineering Department, Computers Division, NEC Corp., Fuchu, Tokyo 183-8501
[§]Silicon Systems Research Laboratories, NEC Corp., Sagamihara, Kanagawa 229-1198
[†]Multimedia Research Laboratories, NEC Corp., Kawasaki, Kanagawa 216-8555
[‡]Graduate School of Information, Production and Systems, Waseda University, Kita-Kyushu, Fukuoka 808-0135

Latch circuits have advantage for timing and are widely used for high-speed custom circuits. However, ASIC design flows are based on the circuits with flip-flops. Then, ASIC designers don't use latches. This paper describes a new timing optimization algorithm for ASIC by replacing flip-flops to latches without changing the functionality of the circuits. After latch replacement, restricted retiming called fixed-phase retiming is carried out for timing optimization by minimizing the impact of clock skew and jitter. The experimental results show that 17% delay improvement of the benchmark circuits is achieved by proposed algorithms.

I. Introduction

LSI design is classified into semi-custom design style and custom design style. In semi-custom design style, since primitive blocks in a library are prepared by semiconductor vendors or IP vendors and lots of design automation programs such as logic synthesis and automatic layout tools are supported, LSI design effort is minimized. On the contrary, in custom design style, circuits are designed by full use of manual design to realize the highest performance or the minimum area.

ASICs are designed with semi-custom design style. In [1], applying some custom design techniques to ASIC to improve ASIC performance is proposed. They compared the difference between ASIC and custom design. Their conclusion is that custom design circuit is faster than best practice ASIC by using dynamic logic, tuning the process, reducing clock overhead, and sizing of transistors and wires. Among them, we think that reducing clock overhead is very important, since the ratio of skew and jitter in clock cycle time becomes large in future high-speed circuits.

We use latches (level-sensitive latches) to overcome this clock overhead problem. A latch has two superior points to a flip-flop (edge-triggered flip-flop) in timing. One is skew and jitter tolerance and the other is slack passing and time borrowing. Since the ratio of skew and jitter in clock cycle time becomes large in high-speed circuits, skew and jitter tolerance is especially important for high-end microprocessors [2][3].

In ASIC design, flip-flops are widely used but latched are rarely used. In [1], a latch-based design is recommended for ASIC since timing could be 25% improved by using lathes instead of flip-flops. However, ASIC designers are familiar with flip-flops and are still using only flip-flops.

In this paper, we propose new timing optimization algorithm by replacing flip-flops to latches. The objective is to transform a given circuit with flip-flops to faster one with latches without changing the functionality of the circuit.

Automatic replacement of flip-flops to latches is proposed in [4]. Their method is based on retiming of flip-flops [5]. An algorithm in [4] replaces each flip-flop to two flip-flops, retimes the circuit with timing constraint that is the same as the STLDS (Skew Tolerant Latch Design Scheme) circuits in [2], and then replaces flip-flops by low active or high active latches alternately. This algorithm improves timing by moving latches to the positions where clock skew and jitter are canceled. There are also several papers about retiming of latches [6][7][8][9].

Although these algorithms are quite effective, there are some drawbacks. First, formal verification between an original flip-flop based circuit and retimed latch-based one is very difficult [10][11][12]. Without formal verification, ASIC designers don't want to use automatic latch replacement. Secondly, only limited optimization techniques, such as buffer insertion or gate resizing, can be applied after latch replacement. This limitation is also arisen from the lack of formal verification. Without formal verification, effective logic optimization techniques that change the structure of the circuit can't be applied to the circuit.

New type of retiming called fixed-phase retiming is proposed in [13]. Fixed-phase retiming preserves the original flip-flop's position by fixing one latch and retimes only the other one. In [13], fixed-phase retiming is used for power optimization.

In this paper, we introduce new latch replacement technique based on fixed-phase retiming for timing optimization. By using fixed-phase retiming, our timing optimization method overcomes the previous work's drawbacks about formal equivalence verification that is required in ASIC design flows. In addition, since fixed-phase retiming preserves the original flip-flop's position, SCAN test [14][15] patterns for original flip-flop based circuit can be used after latch replacement by using SCAN latches [3]. Even retiming is restricted in fixed-phase retiming, our method optimizes retiming positions, so that the impact of clock skew and jitter that causes large performance loss in the high-speed circuits are minimized and significant timing improvement is achieved.

While [13] used fixed-phase retiming for power optimization, our method uses fixed-phase retiming for timing optimization. In our method, an original circuit based on flip-flops is optimized with the proposed constraints that don't have clock skew and jitter in the large portion of the circuit, and then the flip-flops are replaced to latches and fixed-phase retiming is carried out. The retimed circuit works correctly at the target clock cycle even with skew and jitter. In other words, an original circuit based on flip-flops is optimized for timing by canceling the effect of skew and jitter.

By these features, our method is very useful for speeding up ASICs, since the method is not only easily adopted to current ASIC design flow including formal equivalence verification and SCAN test method, but also effectively canceling the effect of skew and jitter for timing optimization.

The remainder of the paper is organized as follows. Preliminaries such as latch's properties and timing constraints are described in section 2. The new timing optimization technique is described in section 3. The experimental results are discussed in section 4 and finally conclusion is in section 5.

II. Preliminaries

A. Flip-flop and Latch

Sequential elements are classified as flip-flops and latches. Flip-flops store input data value at the rising (or falling) edge of a clock signal but don't change the value at the other timings. Latches have an active state when the input data value goes through output data and hold the value at the end of the active state. There are two types of latches. One is a high active latch that is active state when a clock signal is high, and the other is a low active latch that is active state when a clock signal is low. When a latch is an active state, it is said that the latch is transparent.

Important properties for latch replacement are as follows.

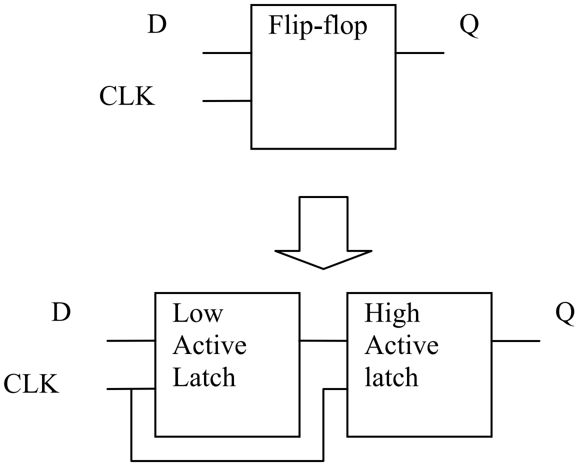


Fig. 1 Flip-flop to Latches Replacement

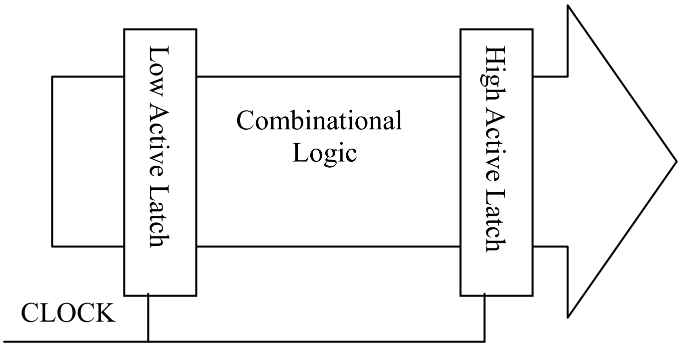


Fig. 2 Single Phase Latches Scheme

(1) A flip-flop is replaced with a pair of low active latch and high active latch without changing the functionality of the circuit. (Figure 1)
(2) Positions of latches can be moved by retiming [5] without changing the functionality of the circuit.

The circuit with low active latches and high active latches that have the same clock signal and low active latches and high active latches are placed one after the other is called single phase latches scheme [16]. This is used for high-speed circuits [2][4] and is shown in figure 2. In this paper, latches are always used in single phase latches scheme.

B. Timing Constraints

ASIC is usually designed with synchronous style that has clock signals to synchronize sequential elements in the circuit. We assume the duty of the clock signal is 50%.

The clock signal rises and falls periodically and the ideal time between two rising edge is called cycle time. However, the clock signal has uncertainties called skew and jitter [17].

Skew is the time difference of the equivalent edge of clock signal at the different positions. The time difference is caused mainly by a difference of the clock signal's delay time through a clock distribution network.

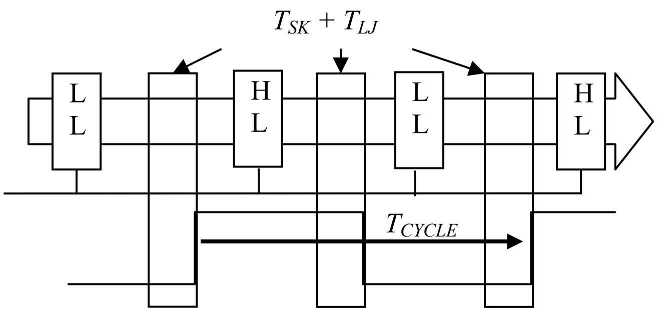
Jitter is the time difference of the clock edges between expected clock signal and real clock signal. Jitter is classified into short-term jitter T_{SJ} and long-term jitter T_{LJ}. Short term jitter is also called cycle-to-cycle jitter and is the time difference measured by consecutive rising or falling edges of the same clock signal. Long-term jitter is measured by the time difference between the ideal clock signal and the actual clock signal over the many cycles [17].

A circuit should satisfy timing constraints to work correctly with a given clock cycle. There are two types of timing constraints. One is hold time constraint and the other is worst case constraint.

Hold time constraint is to avoid races by short paths. The constraint is: [17]

$$D_{CL,MIN} = T_{SK} + T_H - T_{CQ} \qquad (1)$$

$D_{CL,MIN}$ is minimum case combinational delay between sequential element, T_{SK} is clock skew, T_H is hold time, and



LL : Low active Latch
HL : High active Latch

Fig. 3 Balanced Latches Timing

T_{CQ} is clock-to-output delay of a sequential element.

Worst case constraint is to assure the circuit works correctly with the target clock cycle. Worst case constraint for flip-flops is

$$D_{CL,MAX} = T_{CYCLE} – T_{SETUP} – T_{SK} – T_{SJ} – T_{CQ} \quad (2)$$

$D_{CL,MAX}$ is the worst case combinational delay between flip-flops, T_{CYCLE} is the target clock cycle time, T_{SETUP} is setup time of flip-flop, T_{SK} is clock skew, T_{SJ} is short-term jitter, and T_{CQ} is clock-to-output delay of flip-flop.

For balanced latch circuits, if all latches are placed the outside position where the clock skew and jitter effect, the effect of clock skew and jitter are canceled. This is illustrated in figure 3. In this case, worst case constraint for balanced latches is: [17]

$$D_{CL,MAX} = T_{CYCLE} – 2 * D_{TP} \quad (3)$$

$D_{CL,MAX}$ is worst case combinational delay between low active latches, T_{CYCLE} is target clock cycle time, and D_{TP} is combinational delay time of latches in transparent state.

Since our method replaces flip-flops to latches, the input interface data of the latch replaced portion are also clocked by the same clock signal of the latches. Taking into consideration skew and long term jitter, set the latest rising edge of the clock signal at the input side interface of latch replaced portion to the standard time 0.

The open edge is the clock edge at which the latch becomes active state and the closing edge is the clock edge at which the latch becomes non-active state. We assume the duty of the clock signal is 50% as described in earlier. Then, the closing edge and the open edge of the i-th low active latch and high active latch from input interfaces of the latch replaced portion based on the standard time are as follows.

The latest open edge T_{OPEN} of the i-th low active latch from input interfaces:

$$T_{OPEN} = i * T_{CYCLE} – T_{CYCLE}/2 \quad (4)$$

The earliest closing edge $T_{CLOSING}$ of the i-th low active latch from input interfaces:

$$T_{CLOSING} = i * T_{CYCLE} – T_{SK} –T_{LJ} \quad (5)$$

The latest open edge T_{OPEN} of tbe i-th high active latch from input interfaces:

$$T_{OPEN} = i * T_{CYCLE} \quad (6)$$

The earliest closing edge $T_{CLOSING}$ of the i-th high active latch from input interfaces:

$$T_{CLOSING} = i * T_{CYCLE} + T_{CYCLE}/2 – T_{SK} –T_{LJ} \quad (7)$$

Input data of the latch replaced portion goes through each latch. The latch works correctly as transparent, if the following equations are satisfied.

$$T_{OPEN} < A \quad (8)$$
$$A + D_{TP} < T_{CLOSING} \quad (9)$$

A is an arrival time of the latch based on the standard time, and D_{TP} is the delay time of the latch when it is transparent, respectively.

From (4),(5),(6),(7),(8),and(9), each latch works correctly as transparent, if the arrival time of latch satisfy following conditions.

for the i-th low active latch

$$A_{Li} + D_{TP} < i * T_{CYCLE} – T_{SK} –T_{LJ} \quad (10)$$
$$A_{Li} > i * T_{CYCLE} – T_{CYCLE}/2 \quad (11)$$

for the i-th high active latch

$$A_{Hi} + D_{TP} < i * T_{CYCLE} + T_{CYCLE}/2 – T_{SK} –T_{LJ} \quad (12)$$
$$A_{Hi} > i * T_{CYCLE} \quad (13)$$

The equation (10)'s right hand side is also the worst case departure time of the i-th low active latches.

C. Circuit Model

The circuit is modeled as a directed graph. The vertices V correspond to combinational elements. The edges E correspond to the interconnections of combinational elements from input side to output side. Sequential elements exist on the edges. The number of sequential elements existing on the edge e is expressed as $W(e)$.

D. Retiming

Retiming [5] is a optimization algorithm that move sequential elements over the combinational elements without changing the functionality of the circuits. Figure 4 shows a simple example of retiming.

The circuit optimized by retiming always holds the following equation.

$$Wr(e) = W(e) + r(v) – r(u) \quad (14)$$

, where the edge e connects the node v to node u, $Wr(e)$ is the number of sequential elements after retiming, and $r(v)$ is

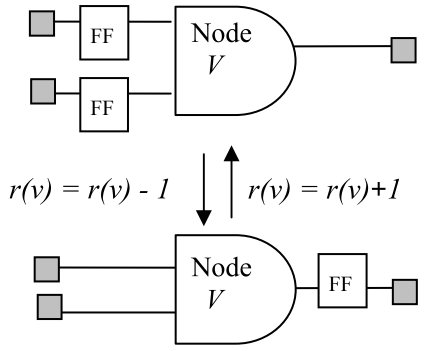


Fig. 4 Retiming

number of sequential elements that move over the node *v* from output side to input side. This equation assures that retiming doesn't change the functionality of the circuit.

III. Timing Optimization Method

A. Inputs and Outputs

The input is a net-list designed using flip-flops. The output is the timing optimized one in which flip-flops are replaced to latches.

B. Outline of the Algorithms

The outline of the timing optimization algorithm is shown below.

(1) Logic optimization is carried out on the input circuit. The difference from traditional logic optimization is that the clock skew and jitter is assigned only to the paths starting from flip-flops that are not converted to latches. For other paths between flip-flops, clock skew and jitter are not assigned.
(2) Each flip-flop is converted to a pair of low active latch and high active latch.
(3) Fixed-phase retiming that is explained in the following sub-section is carried out.
(4) Technology dependent, and if necessary technology independent logic re-optimizations are carried out. Any kind of logic optimization algorithms could be applied.

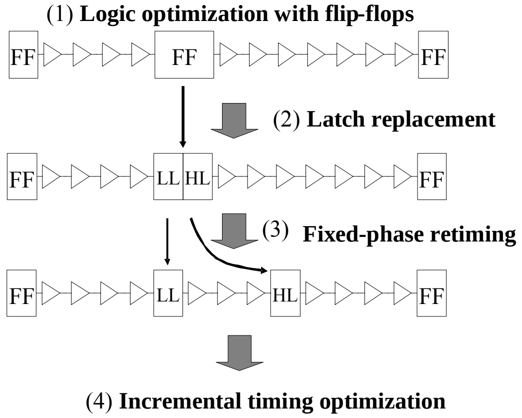

(1) Logic optimization with flip-flops
FF
FF
FF
(2) Latch replacement
FF
LL
HL
FF
(3) Fixed-phase retiming
FF
LL
HL
FF
(4) Incremental timing optimization

Fig.5 An Outline of the Algorithm

Above procedure is illustrated in figure 5.

C. Fixed-Phase Retiming

After replacing flip-flops to latches, edges of the circuit have zero or even number of latches in the circuit model. This is expressed as

$W(e) = 2 * n$ (15)

For the simplicity, we assume that each edge has at most one flip-flops in the original circuit. This simplicity is appropriate in practical circuits by the hold time constraint.

$W(e) = 0 \textit{ or } 2$ (16)

A retiming method that move only one of pair of latches and fixes the others at original flip-flop's positions is useful since the flip-flop's position still remains in the circuit after latch replacement.

This retiming has the following conditions other than traditional retiming equation (14).

$r(v) = -1, 0, \textit{or } 1$ (17)

$\textit{if} \ (W(e) \mathrel{!=} 0) \ \textit{then} \ Wr(e) \mathrel{!=} 0$ (18)

Condition (17) assures only one latch can move over the combinational nodes. Condition (18) assures one of the pair of latches is fixed at original flip-flops position.

Retiming restricting the condition (17) to move only one direction is proposed as fixed-phase retiming in [13]. They used fixed-phase retiming for power optimization. We classify fixed-phase retiming into fixed-phase forward retiming and fixed-phase backward retiming. Fixed-phase forward retiming moves at most one latch over combinational nodes from input side to output side. This implies that in single phase latches scheme, only high active latches are forward retimed. Fixed-phase forward retiming has the following condition other than original retiming (14).

$r(v) = 0 \textit{ or } -1$ (19)

Fixed-phase backward retiming moves at most one latch over combinational nodes from output side to input side. The condition is:

$r(v) = 0 \textit{ or } 1$ (20)

The paper [13] is classified as fixed-phase backward retiming. In this paper, fixed-phase forward retiming is described. However, extension to fixed-phase backward retiming is easily realized. Fixed-phase forward retiming is especially useful, if SCAN low active latch exists in cell library [3]. In this case, SCAN test patterns for flip-flop based circuit can be used after latch replacement without any modifications. Formal equivalence verification is relatively easy for fixed-phase forward retiming, since the original flip-flop's positions are preserved by the low active latches.

For simplicity, we assume setup time of flip-flop, clock-to-output delay of flip-flop, setup time of latch, and transparent combinational delay time of latch are all same.

$$T_{SETUP} = T_{CQ} = D_{SETUP} = D_{TP} = \Delta \quad (21)$$

In practical cases, these values' difference is relatively small and the re-optimization process after fixed-phase retiming can cancel it. If these values' difference is large, extension for using these values as they are is possible.

Before fixed-phase retiming, flip-flop based circuit is optimized by following conditions.

For paths from input interfaces of the latch replaced portion to flip-flops that are replaced to latches:

$$D_{CL} + \Delta + T_{SK} + T_{LJ} < T_{CYCLE} \quad (22)$$

Other paths start within the latch replaced portion:

$$D_{CL} + 2*\Delta < T_{CYCLE} \quad (23)$$

These conditions are shown in figure 6 (A).

After optimizing with the conditions, flip-flops within the latch replaced portion are replaced to pairs of high active latch and low active latch as shown in figure 6 (B). After that, as shown in figure 6 (C), high active latches are retimed to the output edge of nodes whose delay time D_{CL} satisfy following conditions. Then, the retimed circuit correctly works at the target clock cycle time T_{CYCLE}.

$$T_{SK} + T_{LJ} < D_{CL} \quad (24)$$
$$D_{CL} < T_{CYCLE} / 2 - \Delta \quad (25)$$

The remarkable point is that there is no skew and jitter in condition (23). In other words, for optimization of latch replaced portion's flip-flops, skew and jitter is not necessary to take into consideration. Even it, fixed-phase retiming transforms the circuit to work correctly within the target clock cycle time with skew and jitter. This is very useful for high-speed circuits, since skew and jitter are neglected in large portion of the circuit. This optimization method is illustrated in figure 6.

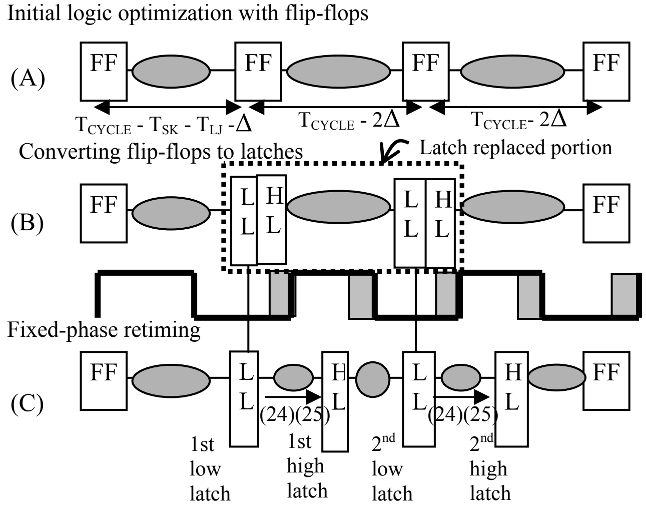


Fig.6 Optimization Constraints

Proposition 1. A circuit based on flip-flops optimized by conditions (22) and (23), followed by latch replacement and fix-phase forward retiming by conditions (24) and (25) works at the target clock cycle time T_{CYCLE} with skew and jitter.

Proof: For the 1st low active latches, latches are optimized by condition (22). Then, arrival time of 1st low active latches A_{L1} is same as D_{CL} of condition (22):

$$A_{L1} < T_{CYCLE} - T_{SK} - T_{LJ} - \Delta \quad (26)$$
$$A_{L1} + \Delta < T_{CYCLE} - T_{SK} - T_{LJ} \quad (27)$$

The 1st low active latches are transparent and work correctly on the next equation from (10):

$$A_{L1} + \Delta < T_{CYCLE} - T_{SK} - T_{LJ} \quad (28)$$

Equation (27) equals to equation (28). This ensures the 1st low active latches are transparent even in the worst cases.

Assume that for the n-th low active latches are transparent in the worst cases. In this case, from equation (10), the departure time of the n-th low active latches are:

$$A_{Ln} + \Delta = n * T_{CYCLE} - T_{SK} - T_{LJ} \quad (29)$$

For the (n+1)-th low active latches, if the n-th high active latch is transparent, the arrival time of the (n+1)-th low active latches are:

$$A_{L(n+1)} = A_{Ln} + \Delta + D_{CL} + D_{TP} \quad (30)$$

From (29) and (23), equation (30) is:

$$\begin{aligned} A_{L(n+1)} &= (n * T_{CYCLE} - T_{SK} - T_{LJ}) + D_{CL} + \Delta \\ &< n * T_{CYCLE} - T_{SK} - T_{LJ} + T_{CYCLE} - 2*\Delta + \Delta \\ &= (n+1) * T_{CYCLE} - T_{SK} - T_{LJ} - \Delta \quad (31) \end{aligned}$$

Then, Δ is transported:

$$A_{L(n+1)} + \Delta < (n+1) * T_{CYCLE} - T_{SK} - T_{LJ} \quad (32)$$

This shows equation (10) is satisfied at the (n+1)-th low active latches.

For the n-th high active latches, we assume the n-th low active latches are transparent even in the worst cases. The n-th high active latches are retimed within the conditions (24) and (25). The n-th low active latches' worst case departure time is equation (10). The arrival time of the n-th high active latches retimed by condition (24) is:

$$\begin{aligned} A_{Hn} &= n * T_{CYCLE} - T_{SK} - T_{LJ} + D_{CL} \\ &> n * T_{CYCLE} - T_{SK} - T_{LJ} + T_{SK} + T_{LJ} \\ &= n * T_{CYCLE} \qquad (33) \end{aligned}$$

Equation (33) satisfies the equation (13). For condition (25), the arrival time of the i-th high active latches is:

$$\begin{aligned} A_{Hn} &= n * T_{CYCLE} - T_{SK} - T_{LJ} + D_{CL} \\ &< n * T_{CYCLE} - T_{SK} - T_{LJ} + T_{CYCLE} / 2 - \Delta \qquad (34) \end{aligned}$$

Equation (34) satisfies equation (12). From (12), (13), (33), and (34), the n-th high active latches are transparent in the worst cases.

From equation (27) for the 1st low active latches, (32) for the n+1-th low active latches, and (33) and (34) for the n-th high active latches, all latches are proved to be transparent with the target clock cycle even in the worst cases.

Intuitively, the circuit optimized by Proposition 1 works correctly, since the all latches are placed at the positions where skew and jitter don't effect as shown in figure 6 (C).

IV. Experimental Results

The above techniques have been implemented in a program running on UNIX workstations. We experimented on large ISCAS'87 sequential benchmark data. In the experiments, each benchmark data written in Verilog-HDL was synthesized and timing optimized by commercial logic synthesis tool to get the fastest results. Then, constraints are changed for latch replacement and re-synthesized followed by latch replacement, fixed-phase retiming, and timing re-optimization to get the faster circuits.

Table 1 shows the results of area increase and delay improvements of latch replacement and fixed-phase retiming compared to original flip-flop based fastest circuits. On the average, delay was improved by 12.9% while the area increase was 15.7%. If latch replacement isn't used, these timing improvements aren't realized. This shows the effectiveness of the proposed latch replacement methods.

For s15850 and s35932 that aren't shown on table 1, we evaluated them and found that the critical paths were ended at the flip-flops that are replaced to 1st low active latches. In these cases, no timing optimization is possible by fixed-phase forward retiming. Then, these data are excluded for experiments.

Table 1: Experimental Results

Circuit	Area Inc. (%)	Delay Dec. (%)
s1488	17.2	14.1
s5378	28.1	17.2
s9234	19.6	13.9
s13207	12.3	10.6
s38417	8.2	11.2
s38584	9.1	10.4
Average	15.7	12.9

V. Summary and Conclusions

We have developed a timing optimization algorithm by replacing flip-flops to latches. Our experiments show that the new algorithm can improve the timing of the circuits maximum 17.2% from the original flip-flop based ones without changing the functionality of the circuit. We see that the new timing optimization algorithm is very useful for speeding up ASIC designs.

References

[1] D Chinnery, B. Nikolic, and K. Keutzer, "Achieving 550 MHz in an ASIC methodology" Proceedings of 38th Design Automation Conference, 2001, pp.420-425.

[2] Y. Hagihara, S. Inui, A. Yoshikawa, T. Uesugi, T. Osada, S.Nakazato, M. Ikeda, M. Okada, and S. Yamada, "A Skew-Tolerant Design Scheme for Over 1-GHz LSIs," Proceedings of ESSCIRC 2000, pp.416-419.

[3] J. D. Warnock, J. M. Keaty, J. Petrovick, J. G. Clabes, C. J.Kircher, B. L. Krauter, P. J. Restle, B. A. Zoric, and C. J.Anderson, "The circuit and physical design of the POWER4 microprocessor," IBM J. Res. & Dev. 46 No.1 January 2002, pp.27-51.

[4] D. Chinnery, K. Keutzer, J. Sanghavi, E. Killian, and K. Sheth, "Automatic Replacement of Flip-Flops by Latches in ASICs," Closing the Gap Between ASIC and Custom, chapter 7, pp.187-208, Kluwer Academic Publishers, 2002.

[5] C.E. Leiserson and J.B. Saxe, "Optimization Synchronous Systems," Journal of VLSI and Computer Systems, 1(1): pp.41-67, 1983.

[6] A.T.Ishii, C.E.Leiserson, and M.C.Papaefthymiou, "Optimizing Two-Phase, Level-Clocked Circuitry", Journal of the ACM, Vol.44, No.1, pp148-199, 1997.

[7] N.Maheshwari and S.S.Sapatnekar, "A Practical Algorithm for Retiming Level-Clocked Circuits", Proceedings of the ICCD, 1996, pp440-445.

[8] N.Maheshwari and S.S.Sapatnekar, "Optimizing large Multiphase Level-Clocked Circuits", IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, Vol.18, No.9, 1999, pp.1249-1264.

[9] N.Shenoy, R.K.Brayton, and A.Sangiovanni-Vincentelli, "Retiming of Circuits with Single Phase Transparent Latches", International Conference of Computer Design, 1991, pp.86-89.

[10] C.A.J.van Eijik, "Sequential Equivalence Checking without State Traversal", Proceedings of DATE, 1998, pp618-623.

[11] G.Hasteer, A.Mathur, and P.Banerjee, "Efficient Equivalence Checking of Multi-Phase Designs Using Retiming", Proceedings of ICCAD, 1998, pp557-562.

[12] D.Anastasakis, R.Damiano, H.T.ma, and T.Stanion, "A Practical and Efficient Method for Compare-Point matching", Proceeding of DAC, 2002, pp305-310.

[13] K.N.Lalgudi and M.C.Papaefthymiou, "Fixed-Phase Retiming for Low Power", Proceeding of IWLPED, 1996, pp.259-264.

[14] A. Kobayashi, S. Matsue, and H. Shiba, "Flip-flop Circuit with FLT Capability", Proceedings of IECEO conference, 1968, p.962. (in Japanese)

[15] S. Funatsu, N. Wakatsuki, and A. Yamada, "Designing Digital Circuits with Easily Testable Consideration", Proceedings of Test Conference, 1978, pp.98-102.

[16] M.S.McGregor, P.B.Denyer, and A.F.Murray, "A Single-Phase Clocking Scheme for CMOS VLSI", 1987 Stanford Conference on Very Large Scale Integration, MIT Press: Cambridge, MA, 1987, pp. 257-271.

[17] H. Partovi, "Clocked Storage Elements", Design of High-performance Microprocessor Circuits", IEEE Press, 2000, pp.207-233.