

Short Paper

Pulsed-Latch Replacement Using Concurrent Time Borrowing and Clock Gating

Chih-Long Chang and Iris Hui-Ru Jiang, *Member, IEEE*

Abstract—Flip-flops are the most common form of sequencing elements; however, they have a significantly higher sequencing overhead than latches in terms of delay, power, and area. Hence, pulsed latches are a promising option to reduce power for high-performance circuits. In this paper, to save power and compensate for timing violations, we fully utilize the intrinsic time borrowing property of pulsed latches and consider clock gating during pulsed-latch replacement. Experimental results show that our approach can generate very power efficient results.

Index Terms—Clock power, pulsed latches, time borrowing.

I. INTRODUCTION

Clock power has become the major contributor of total chip power consumption, and a large portion of it is consumed by sequencing elements [1], [2]. Flip-flops are the most common form of sequencing elements because of its simple timing model. A conventional flip-flop has a significantly higher sequencing overhead than a latch in terms of delay, power, and area. A pulsed latch is a latch synchronized by a pulse clock [3]. Since the pulse is short, a pulsed latch can be approximated as a fast, low-power, and small flip-flop taking advantages of both flip-flops and latches. Therefore, migration from a flip-flop-based design to a pulsed-latch-based counterpart can effectively reduce the sequencing overhead.

Recent research endeavors have been devoted to electronic design automation (EDA) solutions for pulsed-latch-based circuits, e.g., [4]–[10]. Most of these works adopt the generic pulsed-latch structure illustrated in Fig. 1(a) and flip-flop-like timing analysis. Since the pulse generator (PG) and latches are placed apart in the generic form, pulses can easily be distorted. In some implementation, clock gating is integrated into the PG, e.g., [2], [10], [11], but this integration may increase the hardware cost. In contrast, in a multibit pulsed-latch (MBPL) design [10], [12], [13] [see Fig. 1(b)], the PG and latches are placed and hard-wired together in a compact and symmetric form. The pulse distortion and clock skew can thus be well controlled. Moreover, clock gating is not included to keep the cell small and to preserve the flexibility to clock gating cloning.

Under flip-flop-like timing analysis, prior works [4]–[6] use aggressive time borrowing techniques; however, various pulse widths, clock skew scheduling, and retiming may induce some

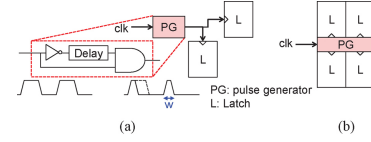


Fig. 1. (a) Generic pulsed latch. (b) Multibit pulsed latch.

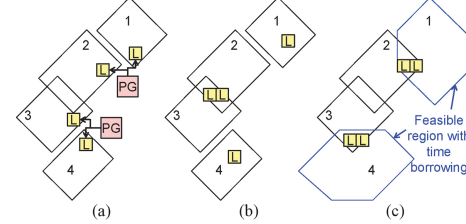


Fig. 2. Pulsed-latch replacement. (a) Generic pulsed latches without time borrowing. (b) MBPL without time borrowing. (c) MBPL with time borrowing.

difficulties on timing closure and functional verification. In fact, we can utilize only the intrinsic time borrowing of latches to provide flexibility to relocate pulsed latches as well as tolerate timing violations [3].

Hence, based on the MBPL structure and time borrowing offered by the pulse width, we apply postplacement pulsed-latch replacement to reduce power subject to timing constraints. According to the timing slacks obtained from a placed flip-flop-based design, we extract feasible regions with time borrowing consideration to replace flip-flops with pulsed latches. Interestingly, the feasible regions with time borrowing consideration are very irregular. Directly manipulating these shapes is complicated, and we show that they can be efficiently handled by four interval graphs. We further consider clock gating during forming MBPLs. Fig. 2 demonstrates with time borrowing, flip-flops can be replaced by MBPLs to reduce power and satisfy timing requirements. Experimental results show that using concurrent time borrowing and clock gating, we can effectively reduce clock power consumed by sequencing elements and tolerate timing violations.

II. PRELIMINARIES AND PROBLEM FORMULATION

A. Timing Model

Let $t_{su}(t'_{su})$ denote the setup time, $t_{hd}(t'_{hd})$ denote the hold time, $t_{cq}(t'_{cq})$ denote clock-to-Q delay for flip-flops (pulsed latches), t'_{dq} denote data-to-Q delay for pulsed latches, T denote cycle period, and w denote pulse width. $t_{fo}(i)$ (respectively $t_{fi}(j)$) means the wire delay contributed by the wirelength between sequencing element i (respectively j) and its fanout (respectively fanin). $(t_{fo}(i) + D_{ij} + t_{fi}(j))$ is the maximum delay of combinational circuit between sequencing elements i and j , and $(t_{fo}(i) + d_{ij} + t_{fi}(j))$ is the minimum. The setup and hold constraints between flip-flops i, j are

$$t_{cq} + t_{fo}(i) + D_{ij} + t_{fi}(j) + t_{su} \leq T \quad (1)$$

$$t_{cq} + t_{fo}(i) + d_{ij} + t_{fi}(j) - t_{hd} \geq M \quad (2)$$

where M is referred to as the hold time margin.

Manuscript received June 12, 2012; revised August 28, 2012; accepted December 1, 2012. Date of current version January 18, 2013. This work was supported in part by Faraday and NSC of Taiwan, under Grant NSC 100-2220-E-009-047 and Grant NSC 101-2628-E-009-012-MY2. An earlier version is published at ISPD 2012 [16]. This paper was recommended by Associate Editor C.-K. Koh.

C.-L. Chang and I. H.-R. Jiang are with the Department of Electronics Engineering and Institute of Electronics, National Chiao Tung University, Hsinchu 30010, Taiwan (e-mail: huiru.jiang@gmail.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCAD.2012.2234828

When flip-flops are replaced with pulsed latches, we may utilize time borrowing. To guarantee successful time borrowing, in this paper, time borrowing is allowed between two adjacent timing windows. The setup and hold constraints are

$$t'_{cq} + t_{fo}(i) + D_{ij} + t_{fi}(j) + t'_{su} \leq T + w \quad (3)$$

$$2t'_{cq} + t_{fo}(i) + D_{ij} + t_{fi}(j) + t_{fo}(j) + D_{jk} + t_{fi}(k) + 2t'_{su} \leq 2T \quad (4)$$

$$t'_{cq} + t_{fo}(i) + d_{ij} + t_{fi}(j) - t'_{hd} \geq M = M_0 + w. \quad (5)$$

M used at flip-flop-based synthesis, placement, and timing analysis is set to $M_0 + w$, where M_0 is the original margin.

We then convert the timing slacks for $t_{fo}(i)$ and $t_{fi}(j)$ obtained by flip-flop-based timing analysis into pulsed-latch-based slacks without time borrowing as follows:

$$S_{fo}(i) = S_{fi}(j) = \frac{T - (D_{ij} + t_{cq} + t_{su}) + \epsilon_1}{2} = \frac{T - D'_{ij}}{2} \quad (6)$$

$$H_{fo}(i) = H_{fi}(j) = \frac{d_{ij} + t_{cq} - t_{hd} - \epsilon_2 - M_0 - w}{2} = \frac{d'_{ij} - w}{2} \quad (7)$$

where $S_{fo}(i)$, $S_{fi}(j)$, $H_{fo}(i)$, $H_{fi}(j)$ are setup and hold slacks for $t_{fo}(i)$ and $t_{fi}(j)$, and $\epsilon_1 = (t_{cq} + t_{su}) - (t'_{cq} + t'_{su})$, $\epsilon_2 = (t_{cq} - t_{hd}) - (t'_{cq} - t'_{hd})$.

B. Problem Formulation

We formulate the following two problems.

The Multibit Pulsed-Latch Replacement (mPLR) problem: Given an MBPL library, the setup and hold slacks of flip-flop inputs and outputs, and clock gating patterns of flip-flops in a placed design, replace flip-flops by MBPLs with time borrowing to minimize power on pulsed latches subject to timing and placement density constraints.

The Generic Pulsed-Latch Replacement (gPLR) problem: Given a pulsed-latch library, pulse-distortion constraints, the setup and hold slacks of flip-flop inputs and outputs, and clock gating patterns of flip-flops in a placed design, replace flip-flops by pulsed latches with time borrowing to minimize power on pulsed latches subject to timing, pulse distortion and placement density constraints.

A clock gating pattern describes when a sequential element is active, and the pattern can be obtained by gate-level simulations. To consider the pulse distortion issue and facilitate the slew/skew/latency control, pulse-distortion constraints prevent PGs from driving numerous or distant pulsed latches: N_{pg} is the maximum number of latches driven by a PG, and L_{pg} is the maximum distance between each latch and its PG.

We may convert a gPLR instance to an mPLR one. First, a PG and m single-bit pulsed latches driven by this PG is viewed as a virtual m -bit MBPL cell. We construct a virtual MBPL library, including 1-bit, $\dots$, N_{pg} -bit pulsed latches. Second, we convert the L_{pg} constraint by enlarging the feasible region of each latch by L_{pg} . The gPLR instance is thus converted to an mPLR one. We apply our mPLR approach, and the locations of MBPLs are the locations of PGs. Each latch is relocated to the point which is nearest to its PG and inside its feasible region.

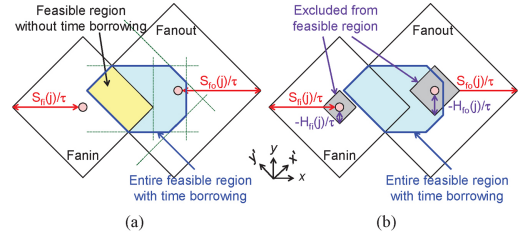


Fig. 3. Feasible region analysis. (a) Time borrowing. (b) Hold violation.

III. FEASIBLE REGION ANALYSIS

We derive the feasible region with time borrowing consideration. All circumstances are classified into four types.

At postplacement, wire delays $t_{fo}(i)$ and $t_{fi}(j)$ can be approximated by piece-wise linear functions:

$$t_{fo}(i) = \tau L_{fo}(i) \text{ and } t_{fi}(j) = \tau L_{fi}(j)$$

where τ is calibrated by the delay table of the pulsed-latch library. If i has multiple fanouts, then each fanout corresponds to one constraint on $L_{fo}(i)$. For a range which covers several linear functions, we use the line connecting the two ends. If τ is different for $t_{fo}(j)$ and $t_{fi}(j)$, then $\tau = \max(\tau_{fo}, \tau_{fi})$.

A. Type I: Setup Satisfied and Hold Satisfied

Here we consider the input design satisfies setup and hold constraints: $S_{fi}(j), S_{fo}(j) \geq 0$; $H_{fi}(j), H_{fo}(j) \geq 0$. The hold constraint is still met after including $t_{fo}(i)$ and $t_{fi}(j)$. Based on Inequalities (3), (4), (6), and (7), we have

$$t_{fi}(j) = \tau L_{fi}(j) \leq S_{fi}(j) + t_b \quad (8)$$

$$t_{fo}(j) = \tau L_{fo}(j) \leq S_{fo}(j) - t_b \quad (9)$$

$$0 \leq t_b \leq w \quad (10)$$

where t_b is the time borrowed from window $j-k$ to window $i-j$. Initially, the fanin and fanout setup slacks define two diamonds centered at the fanin and fanout gates of pulsed-latch j . The overlap area is the initial feasible region without time borrowing. When we borrow time t_b , the fanin diamond is expanded by t_b/τ , while the fanout diamond is shrunk by t_b/τ . The overlap area slides horizontally or vertically. When we keep borrowing, the fanin/fanout diamond would reach the centerlines in between the boundaries of fanin/fanout diamonds, and the overlap area is truncated. Consequently, the entire feasible region is irregular as shown in Fig. 3(a). In the worst case, the feasible region could be an octagon.

B. Type II: Setup Satisfied and Hold Violated

In this case, we consider $S_{fi}(j), S_{fo}(j) \geq 0$; $H_{fi}(j) < 0$ and/or $H_{fo}(j) < 0$. The feasible regions derived for type I are reused for setup time handling. Since we reserve a hold time margin, $M = M_0 + w$, at flip-flop-based logic synthesis and placement, hold violations at postplacement are not severe. Wire delay $t_{fo}(i)$ and $t_{fo}(j)$ can be used to fix hold. Then

$$\tau L_{fi}(j) + H_{fi}(j) \geq 0, \text{ or } \tau L_{fo}(j) + H_{fo}(j) \geq 0. \quad (11)$$

Hold time fixing is done by setting the lower bound on the fanin wirelength, $L_{fi}(j) \geq -H_{fi}(j)/\tau$. Similarly, we can fix

$H_{fo}(j)$ as well. As shown in Fig. 3(b), the shaded area around fanin and fanout gates are excluded from the feasible region.

C. Type III: Setup Violated and Hold Satisfied

Time borrowing may slightly compensate setup violations

$$0 \leq D'_{ij} \leq T + w, \text{ and } 0 \leq D'_{ij} + D'_{jk} \leq 2T. \quad (12)$$

If $D'_{ij} > T$, since only the fanin side can use time borrowing, we redistribute the setup slacks defined in (6) as follows:

$$S_{fo}(i) = 0, S_{fi}(j) = T - D'_{ij}. \quad (13)$$

In this case, we consider $S_{fi}(j) < 0$, $S_{fo}(j) \geq 0$; $H_{fi}(j)$, $H_{fo}(j) \geq 0$. The maximum amount of time borrowing is w , and $-S_{fi}(j)$ has been consumed by D'_{ij} already. Hence

$$0 \leq t_b \leq w + S_{fi}(j). \quad (14)$$

The value of w used in feasible region extraction [see (16) and (17)] should be replaced by $w + S_{fi}(j)$. On the other hand, the fanout diamond should be shrunk by $-S_{fi}(j)/\tau$ at the beginning. Initially, the fanin diamond contains only one point—the fanin gate. Then, the fanin diamond expands, while the fanout diamond shrinks.

D. Type IV: Setup Violated and Hold Violated

Here we consider the input design violates setup and hold constraints. The feasible regions derived for Type III are used for setup time handling, while the lower bound on fanin/fanout wirelength for Type II is used for hold time fixing.

IV. OUR APPROACH

In this section, we present our postplacement pulsed-latch replacement framework. First, we extract feasible regions and represent them by four interval graphs. Second, we use some clustering method to form MBPLs; meanwhile, we consider clock gating during forming MBPLs. Finally, we relocate the newly formed MBPLs.

A. Feasible Region Extraction

The feasible regions are determined by the pulse width and the differences between boundaries of fanin/fanout diamonds. If some fanout boundary is outside the corresponding fanin one, we can find a fence to constrain the feasible region sliding. The entire feasible region with time borrowing can be extracted by finding eight fences. Using these eight fences, we can handle all irregular feasible regions.

To facilitate our feasible region extraction, we adopt a simple and fast coordinate transformation proposed in [14]. The fanin/fanout diamonds in Cartesian coordinate system C become squares in C' , obtained by rotating by 45° . As shown in Fig. 4(a), we define the four boundaries of a fanin/fanout diamond as right, bottom, left, and top boundaries.

Let δ_{rr} , δ_{bb} , δ_{ll} , and δ_{tt} denote the differences between fanin and fanout boundaries in right, bottom, left, and top directions, respectively

$$\begin{aligned} \delta_{rr}(j) &= \max(0, x'_{r,fo}(j) - x'_{r,fi}(j)) \\ \delta_{bb}(j) &= \max(0, y'_{b,fo}(j) - y'_{b,fi}(j)) \\ \delta_{ll}(j) &= \max(0, x'_{l,fi}(j) - x'_{l,fo}(j)) \\ \delta_{tt}(j) &= \max(0, y'_{t,fo}(j) - y'_{t,fi}(j)) \end{aligned} \quad (15)$$

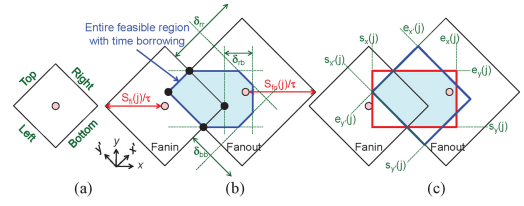


Fig. 4. Feasible region extraction. Any feasible region can be represented by four intervals. (a) Diamond boundary. (b) Fence finding. (c) Four intervals.

TABLE I

MBPL LIBRARY: POWER VERSUS AREA

	Bit number	Normalized power	Normalized area
Flip-flop	1	0.99	1.01
Pulsed-latch	1	1.00	1.00
	2	1.48	1.92
	4	2.45	3.85
	8	4.60	7.58

TABLE II

STATISTICS OF THE BENCHMARK

Circuit	#FFs	#Bins	#Grids	Clock period (ns)	Avg. activity
Industry1	120	6 × 6	600 × 600	1.0	0.25
Industry2	120	6 × 6	600 × 600	1.0	0.13
Industry3	60 000	100 × 300	2000 × 3000	2.0	0.69
Industry4	5524	100 × 200	2000 × 2000	2.0	0.44
Industry5	953	30 × 160	600 × 1600	1.0	0.25

Avg. activity: the average active rate of clock gating functions.

where $x'_{r,fo}(j)$ means the x' coordinate of the right fanout boundary of j . We define the rest of coordinates similarly.

First of all, we have four fences in C' located at

$$\begin{aligned} x' &= x'_{r,fr}(j) + \min(w, \delta_{rr}(j)/2) = e_x(j) \\ y' &= y'_{b,fr}(j) - \min(w, \delta_{bb}(j)/2) = s_y(j) \\ x' &= x'_{l,fr}(j) - \min(w, \delta_{ll}(j)/2) = s_x(j) \\ y' &= y'_{t,fr}(j) + \min(w, \delta_{tt}(j)/2) = e_y(j) \end{aligned} \quad (16)$$

where $x'_{r,fr}(j)$ means the x' coordinate of the right boundary of j 's initial feasible region. We define the rest of coordinates similarly. Let $\delta_{rb}(j) = \min(\delta_{rr}(j), \delta_{bb}(j))$, $\delta_{lb}(j) = \min(\delta_{ll}(j), \delta_{bb}(j))$, $\delta_{lt}(j) = \min(\delta_{ll}(j), \delta_{tt}(j))$, $\delta_{rt}(j) = \min(\delta_{rr}(j), \delta_{tt}(j))$. We also have four fences in C

$$\begin{aligned} x &= x_{rb,fr}(j) + \min(w, \delta_{rb}(j)/2) = e_x(j) \\ y &= y_{lb,fr}(j) - \min(w, \delta_{lb}(j)/2) = s_y(j) \\ x &= x_{lt,fr}(j) - \min(w, \delta_{lt}(j)/2) = s_x(j) \\ y &= y_{rt,fr}(j) + \min(w, \delta_{rt}(j)/2) = e_y(j) \end{aligned} \quad (17)$$

where $x_{rb,fr}(j)$ means the x coordinate of the right-bottom corner of j 's initial feasible region. We define the rest of coordinates similarly. Based on successful time borrowing described in Section II-A, (15)–(17) are applicable to loops and can be generalized to multiple fanouts.

As shown in Fig. 4(b) and (c), the four fences in C' defined by (16) form a rectangle, while the four fences in C defined by (17) form another. The entire feasible region is the overlap area of these two rectangles. Projecting the two rectangles formed by eight fences to x' -, y' -, x -, y -axes, we obtain four intervals, e.g., $[s_x(j), e_x(j)]$ is j 's x' interval.

B. Clustering

Several pulsed latches can form an MBPL if their feasible regions overlap, i.e., forming a clique in x' , y' , x , y intervals simultaneously. We use some clustering method to form MBPLs, e.g., one-way clustering finds maximal cliques along the

TABLE III
TIME BORROWING (IGNORING CLOCK GATING SIGNALS)

Circuit	One-Way Clustering [18] without Time Borrowing			One-Way Clustering [18] with Time Borrowing $w = 100$ ps			Spiral Clustering [20] with Time Borrowing $w = 100$ ps		
	Power Ratio	#Sinks	Time (s)	Power Ratio	#Sinks	Time (s)	Power Ratio	#Sinks	Time (s)
Industry1	74.93%	62	< 0.01	70.63%	52	< 0.01	68.92%	48	< 0.01
Industry2	75.78%	64	< 0.01	72.37%	56	< 0.01	71.49%	54	< 0.01
Industry3	57.54%	7,558	3.36	57.52%	7,558	3.94	57.50%	7,504	3.07
Industry4	62.98%	1,520	0.41	60.92%	1,243	0.40	60.82%	1,233	0.39
Industry5	65.36%	311	0.04	62.73%	255	0.05	62.50%	250	0.05

Power Ratio: the total power contributed from MBPL library cell (considering only α in Equation (18)) over the total power from single-bit pulsed-latch cells.
#Sinks: the number of clock sinks. Time: runtime.

TABLE IV
DIFFERENT PULSE WIDTHS (IGNORING CLOCK GATING SIGNALS)

Circuit	Clustering with Time Borrowing $w = 100$ ps			Clustering with Time Borrowing $w = 150$ ps			Clustering with Time Borrowing $w = 200$ ps		
	Power Ratio	#Sinks	Time (s)	Power Ratio	#Sinks	Time (s)	Power Ratio	#Sinks	Time (s)
Industry1	68.92%	48	< 0.01	68.07%	46	< 0.01	67.64%	45	< 0.01
Industry2	71.49%	54	< 0.01	70.22%	51	< 0.01	69.79%	50	< 0.01
Industry3	57.50%	7,504	3.07	57.50%	7,500	3.20	57.50%	7,500	3.23
Industry4	60.82%	1,233	0.39	60.52%	1,184	0.41	60.46%	1,170	0.40
Industry5	62.50%	250	0.05	62.00%	239	0.05	62.12%	240	0.04

diagonal of the chip [14], or spiral clustering finds maximal cliques from four corners toward the center [16].

C. MBPL Extraction Considering Clock Gating

If the found maximal clique does not fit any MBPL cell, a subset containing the essential pulsed latch is extracted from this clique [14], [16]. The remaining pulsed latches will be considered in the subsequent clustering.

Since the latches inside one MBPL cell share the pulse clock, their clock gating functions are integrated by logic OR operations. To reduce power, we consider clock gating during MBPL extraction. The effective power ratio of an MBPL is

$$P_r = \alpha\beta \quad (18)$$

where α is the power ratio of the MBPL cell with regard to separate single-bit pulsed latches, while β is the activity ratio of the shared clock gating pattern with regard to the average of original ones. We consider the following three strategies.

- 1) We choose the largest feasible MBPL cell and greedily collect pulsed latches with the most similar patterns (in terms of their Hamming distance).
- 2) We collect pulsed latches with the identical clock gating pattern and fit into the largest feasible MBPL cell.
- 3) We extract a feasible-sized subset with minimum $P_r \leq 1$.

Fig. 5 gives an example of MBPL extraction using different clock gating strategies: 1) strategy 1 tries to keep MBPLs as large as possible; 2) strategy 2 is suitable for very different patterns; and 3) strategy 3 keeps a good balance between the number of clock sinks and power saving.

D. Pulsed-Latch Relocation

If an MBPL is located within the bounding box formed by the median x - and y -coordinates of its fanin/fanout gates, the total fanin/fanout wirelength is minimum.



Fig. 5. Clock gating strategies. (a) Clock gating patterns of a maximal clique {1, 2, 3, 4, 5, 6}, where pulsed-latch 1 is essential. (b) The results of Strategies 1, 2, 3 using the MBPL library given in Table I.

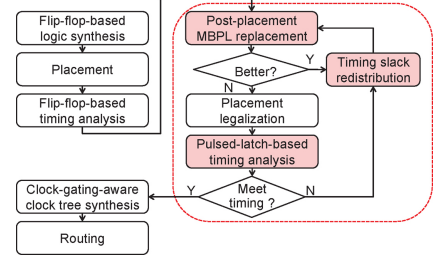


Fig. 6. Pulsed-latch replacement with timing slack redistribution.

We consider two cases. If the feasible region overlaps with the bounding box, we pick a point within the overlapped region. Otherwise, we find the point inside the feasible region which is nearest to the bounding box [15].

If the selected point is occupied, is located inside the forbidden region due to hold violations, or conflicts with the placement density constraint, we check the vacancy around this point. If we still cannot relocate this MBPL, we generate a smaller MBPL from its originating clique instead.

E. Extension to Timing Slack Redistribution

Two types of flip-flops can be used to redistribute slacks to form larger MBPLs. The first type is a flip-flop that cannot be replaced by a pulsed-latch, e.g., a flip-flop directly connected to some primary input/output for timing model characterization, or a negative-edge-triggered flip-flop. The second type is a flip-flop in some formed MBPL that still has unconsumed setup slacks.

As shown in Fig. 6, we iteratively do pulsed-latch replacement and timing slack redistribution until no better MBPL clustering can be found. At each iteration, we freeze the formed MBPLs of the maximum bit number and redistribute their timing slacks to other pulsed latches. The released timing slacks facilitate MBPL forming in the subsequent iteration. For a timing window $i-j$, if i is a forbidden flip-flop (the first type flip-flop) or belongs to a frozen MBPL (the second type flip-flop), we have the redistributed slacks for j as follows:

$$S_{fi}(j) = T - D'_{ij} - t_{fo}(i) \text{ and} \quad (19)$$

$$H_{fi}(j) = t_{fo}(i) + d'_{ij} - w. (\text{or } H_{fi}(j) = t_{fo}(i) + d'_{ij}).$$

Similarly, if j is a forbidden flip-flop or belongs to a frozen MBPL, we have the redistributed slacks for i as follows:

$$S_{fo}(i) = T - D'_{ij} - t_{fi}(j) \text{ and} \quad (20)$$

$$H_{fo}(i) = t_{fi}(j) + d'_{ij} - w.$$

Slack redistribution enlarges the feasible regions of some pulsed latches thus possibly forming larger MBPLs in the subsequent iteration. If a more power efficient clustering result is found, the MBPLs are updated. In case legalization

TABLE V
COMPARISON ON DIFFERENT CLOCK GATING STRATEGIES (W = 100 PS)

Circuit	Clustering (Ignoring Clock Gating Signals)			Clustering with Clock Gating Strategy 1			Clustering with Clock Gating Strategy 2			Clustering with Clock Gating Strategy 3		
	Power Ratio	Pattern-Aware Power Ratio	#Sinks	Power Ratio	Pattern-Aware Power Ratio	#Sinks	Power Ratio	Pattern-Aware Power Ratio	#Sinks	Power Ratio	Pattern-Aware Power Ratio	#Sinks
Industry1	68.92%	140.66%	48	70.20%	137.25%	51	95.68%	95.68%	110	95.68%	95.68%	110
Industry2	71.49%	102.20%	54	71.93%	99.78%	55	77.96%	77.96%	69	77.96%	77.96%	69
Industry3	57.50%	79.58%	7,504	57.50%	70.05%	7,507	70.22%	70.22%	23,223	63.62%	68.74%	15,067
Industry4	60.82%	99.20%	1,233	61.69%	94.18%	1,349	73.92%	73.92%	2,701	73.30%	74.01%	2,630
Industry5	62.50%	121.02%	250	63.26%	113.90%	267	77.27%	77.27%	531	77.14%	77.26%	529

Pattern-Aware Power Ratio: the total effective power of MBPLs (considering both α and β in Equation (18)) over the total effective power of single-bit pulsed-latches.

TABLE VI
COMPARISON ON WIRELENGTH, POWER, AND TIMING VIOLATIONS UNDER DIFFERENT AMOUNT OF TIME BORROWING (INDUSTRY2)

Pulse width	Fanin/Fanout Wirelength Ratio	Pattern-Aware Power Ratio	#Sinks	Initial #bit of setup vio	Initial TNS (ns)	Initial #bit of hold vio	Initial THS (ns)	After #bit of setup vio	After TNS (ns)	After #bit of hold vio	After THS (ns)
w = 0 ps	100.00%	100.00%	120 (120/0/0/0)	49	2.709	0	0	49	2.709	0	0
w = 100 ps	103.17%	91.77%	101 (82/19/0/0)	49	2.709	0	0	10	0.179	0	0
w = 150 ps	106.98%	83.13%	81 (48/30/3/0)	49	2.709	0	0	0	0	0	0
w = 200 ps	109.55%	80.09%	74 (34/37/3/0)	49	2.709	120	6.010	0	0	4	0.030

w: the maximum allowable amount of time borrowing; w = 0 ps means no time borrowing. TNS: total negative setup slack. THS: total negative hold slack.

#bit of setup (hold) vio: the total number of bits of pulsed-latches whose setup (hold) constraint is violated.

Initial (After): the timing results reported by flip-flop-based (pulsed-latch-based) timing analysis before (after) pulsed-latch replacement.

induces timing violations based on the pulsed-latch-based timing analysis, the current pulse width cannot afford sufficient time borrowing. If the MBPL library provides variant pulse widths, we may adopt a larger pulse width and redo pulsed-latch replacement. Moreover, because a single-bit pulsed-latch and a single-bit flip-flop consume almost the same power/area (see Table I), the single-bit pulsed latches formed by pulsed-latch replacement remain flip-flops if its timing is satisfied without time borrowing.

V. EXPERIMENTAL RESULTS

Our algorithm was implemented in the C++ programming language and evaluated on a platform with an Intel Xeon 3.8GHz CPU and with 16GB memory. Table I lists the normalized power and area ratio of a discrete MBPL library based on 55-nm technology. Experiments are conducted on five industrial cases provided by [17] as listed in Table II.

In the first set of experiments, the setup slacks for these circuits range from 10 to 200 ps, and there are no hold violations. In Table III, we focus on power reduction contributed from the MBPL library with time borrowing consideration. It can be seen that time borrowing indeed helps power reduction (see Power Ratio). Table IV compares power reduction under different amount of time borrowing based on spiral clustering. In general, the larger pulse width, the more power saving. Table V shows even if power reduction from the MBPL library is significant, the effective power saving may be not. Strategy 1 may consume more power under very random clock gating patterns. Strategy 2 is suitable for pulsed latches with very different patterns. Strategy 3 keeps the balance between the effective power reduction and the number of sinks. Using time borrowing and clock gating consideration, we can achieve power efficient results.

In the following experiment, we demonstrate the ability to compensate timing violations based on time borrowing. Industry 2 is resynthesized based on a tighter clock period. Table VI shows the tradeoffs between wirelength overhead, power reduction, and timing improvement under different amount of timing borrowing. A proper value of w also can compensate timing violations well, e.g., $w = 150$ ps.

VI. CONCLUSION

Pulsed latches are promising to reduce power for high performance circuits. In this paper, we proposed a novel pulsed-latch replacement approach to save power using concurrent time borrowing and clock gating. Our results showed that our approach generates very power efficient results.

REFERENCES

- [1] R. S. Shelar, "An efficient clustering algorithm for low power clock tree synthesis," in *Proc. ISPD*, Mar. 2007, pp. 181–188.
- [2] S. D. Naffziger, G. Colon-Bonet, T. Fischer, R. Riedlinger, T. J. Sullivan, and T. Grutkowski, "The implementation of the Itanium 2 microprocessor," *IEEE JSSC*, vol. 37, no. 11, pp. 1448–1460, Nov. 2002.
- [3] N. H. E. Weste and D. M. Harris, *Integrated Circuit Design*. 4th ed., Reading, MA: Addison Wesley, 2011.
- [4] H. Lee, S. Paik, and Y. Shin, "Pulse width allocation with clock skew scheduling for optimizing pulsed-latch-based sequential circuits," in *Proc. ICCAD*, Nov. 2008, pp. 224–229.
- [5] S. Lee, S. Paik, and Y. Shin, "Retiming and time borrowing: Optimizing high-performance pulsed-latch-based circuits," in *Proc. ICCAD*, Nov. 2009, pp. 375–380.
- [6] S. Paik, L. Yu, and Y. Shin, "Statistical time borrowing for pulsed-latch circuit designs," in *Proc. ASP-DAC*, Jan. 2010, pp. 675–680.
- [7] Y.-L. Chuang, S. Kim, Y. Shin, and Y.-W. Chang, "Pulsed-latch-aware placement for timing-integrity optimization," in *Proc. DAC*, Jun. 2010, pp. 280–285.
- [8] S. Shibatani and A. H.C. Li. (2006, Jul.). Pulse-latch approach reduces dynamic power. *EE Times* [Online]. Available: <http://www.eetimes.com>
- [9] H.-T. Lin, Y.-L. Chuang, and T.-Y. Ho, "Pulsed-latch-based clock tree migration for dynamic power reduction," in *Proc. ISLPED*, Aug. 2011, pp. 39–44.
- [10] S. Paik, G.-J. Nam, and Y. Shin, "Implementation of pulsed latch and pulsed register circuits to minimize clocking power," in *Proc. ICCAD*, Nov. 2011, pp. 640–646.
- [11] S. Kozu, M. Daito, Y. Sugiyama, H. Suzuki, H. Morita, M. Nomura, K. Nadehara, S. Ishibuchi, M. Tokuda, Y. Inoue, T. Nakayama, H. Harigai, and Y. Yano, "A 100MHz 0.4W RISC processor with 200MHz multiply-adder, using pulse-register technique," in *Proc. ISSCC*, Feb. 1996, pp. 140–141.
- [12] A. Venkatraman, R. Garg, and S. P. Khatri, "A robust, fast pulsed flip-flop design," in *Proc. GLSVLSI*, May 2008, pp. 119–122.
- [13] J. Warnock, L. Sigal, D. Wendel, K. P. Muller, J. Friedrich, V. Zyuban, E. Cannon, and A. J. Kleinosowski, "POWER7™ local clocking and clocked storage elements," in *Proc. ISSCC*, Feb. 2010, pp. 178–179.
- [14] I. H.-R. Jiang, C.-L. Chang, and Y.-M. Yang, "INTEGRA: Fast multi-bit flip-flop clustering for clock power saving," *IEEE TCAD*, vol. 31, no. 2, pp. 192–204, Feb. 2012.
- [15] H. Edelsbrunner, "Computing the extreme distances between two convex polygons," *J. Algorithms*, vol. 6, no. 2, pp. 213–224, Jun. 1985.
- [16] C.-L. Chang, I. H.-R. Jiang, Y.-M. Yang, E. Y.-W. Tsai, and A. S.-H. Chen, "Novel pulsed-latch replacement based on time borrowing and spiral clustering," in *Proc. ISPD*, Mar. 2012, pp. 121–128.
- [17] Faraday Technology Corporation. (2011) [Online]. Available: <http://www.faraday-tech.com>.