

Latch Clustering for Minimizing Detection-to-Boosting Latency Toward Low-Power Resilient Circuits

Chih-Cheng Hsu[†], Mark Po-Hung Lin[†], and Masanori Hashimoto[‡]

[†]Department of Electrical Engineering, National Chung Cheng University, Chiayi, Taiwan

[‡]Department of Information Systems Engineering, Osaka University, Suita-shi, Japan

ABSTRACT

Dynamic voltage scaling (DVS) has become one of the most effective approaches to achieve ultra-low-power SoC. To eliminate timing errors arising from DVS, several error-resilient circuit design techniques were proposed to detect and/or correct timing violations. The most recently proposed time-borrowing-and-local-boosting (TBLB) technique has the advantage of lower power consumption and less performance degradation due to the needlessness of pipeline stalls. On the other hand, to make the best use of the TBLB technique, a special timing requirement for TBLB latches must be considered in the physical design process. To address this issue, a novel reliability-aware latch clustering method for low-power TBLB resilient circuits is introduced. Experimental results show that the proposed approach is very effective in reducing the delay of both combinational and error-detection circuits, which indicates better circuit reliability.

1. INTRODUCTION

Dynamic voltage scaling (DVS) has become one of the most effective approaches to achieve ultra-low power SoC design. To eliminate the timing errors arising from DVS, several timing error resilient circuits or error detection flip-flop/latch design techniques, such as canary flip-flops [1], Razor flip-flops [2, 3], and time-borrowing-and-local-boosting (TBLB) latches [4], had been proposed to dynamically detect timing violations and to control the supply voltage based on in-situ circuit operations. Among these techniques, the TBLB latch based design has the advantages of both error tolerance and correction with less power, performance, and area overhead.

According to [4], a TBLB resilient circuit with TBLB latches, as shown in Figure 1(a), consists of three major components: a transition detector (TD), a level-converting pulse-latch (PL) driven by a pulse generator (PG), and a boost controller. During the normal operation, the system is expected to operate at a lower supply voltage, VDD_{DVS} , for lower power consumption. If the delay of the n^{th}

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](http://permissions.acm.org).

SLIP '16, June 04 2016, Austin, TX, USA

© 2016 ACM. ISBN 978-1-4503-4430-2/16/06...\$15.00

DOI: <http://dx.doi.org/10.1145/2947357.2947364>

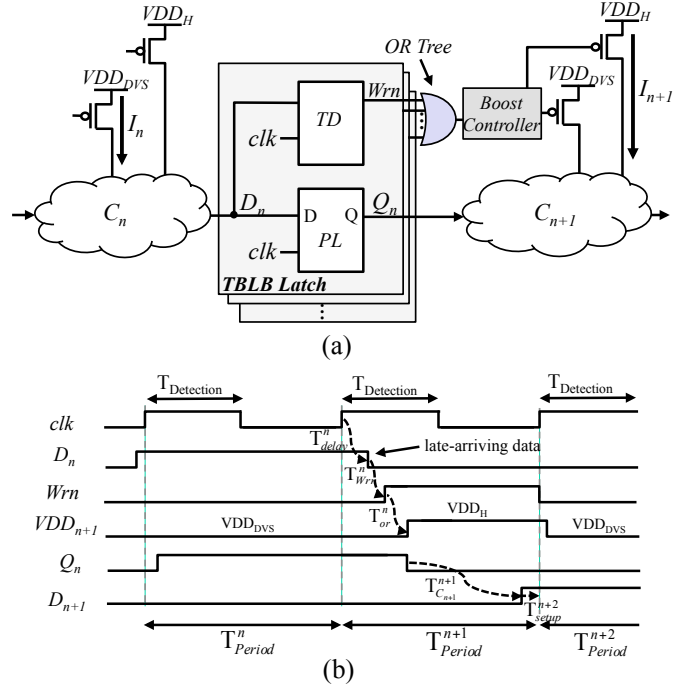


Figure 1: The TBLB resilient circuit and its timing diagram [4].

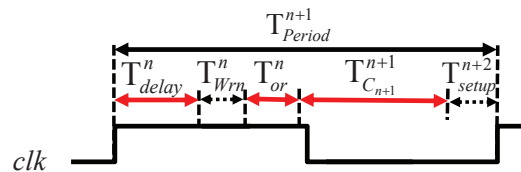


Figure 2: Detailed timing information at the stage, C_{n+1} , when it is borrowed by the previous stage, C_n , because the combinational logic delay at C_n exceeds the clock period.

combinational logic, C_n , is longer than the clock period, T_{period} , as seen in Figure 1(b), where D_n arrives late, due to circuit aging and other reliability effects, the transition detector will flag the warning signal, Wrn . The warning signals are then transferred to the boost controller through a large or multi-level OR gate (i.e. an OR tree). Since the combinational logic delay at C_n exceeds its cycle limit and requires time borrowing from the next stage, C_{n+1} , to ensure correct data, C_{n+1} is required to speed up immediately by

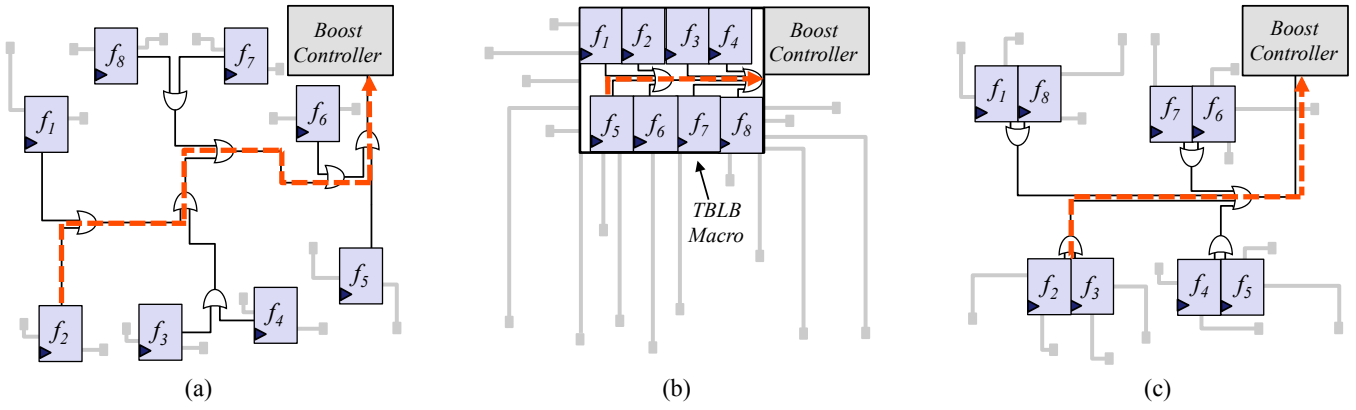


Figure 3: Different design styles for TBLB latches. (a) Discrete 1-bit TBLB latches resulting in longer propagation delay of an OR tree. (b) An integrated TBLB Macro leading to many critical paths in combination circuits. (c) Distributed multi-bit TBLB latches achieving better tradeoff between the propagation delay of OR trees and that of combination circuits.

boosting the local voltage to a higher supply voltage, VDD_H , to prevent timing error propagation. Note that similar error correction can also be implemented with body biasing, where speed boosting is achieved by forward body bias. Consequently, timing violations can be rescued without extra-cycle or performance overhead. Figure 2 collapses the timing diagram in Figure 1(b), and details the timing information at the stage, C_{n+1} , where T_{delay}^n is late-arriving delay from C_n , T_{Wrn}^n is warning detection delay of transition detector, T_{or}^n is the propagation delay through the OR-tree, $T_{C_{n+1}}^{n+1}$ is the combinational logic delay of C_{n+1} , and T_{setup}^{n+2} is the setup time for C_{n+2} .

Because of the elimination of extra-cycle overhead with TBLB resilient circuits, the delay margin of both error detection and correction must be strictly limited within a clock period. If a timing delay occurs at C_n , the timing constraint at C_{n+1} , as seen in Equation (1), must have to be satisfied such that the timing violation at C_n can be rescued without data error and performance overhead.

$$T_{delay}^n + T_{Wrn}^n + T_{or}^n + T_{C_{n+1}}^{n+1} + T_{setup}^{n+2} \leq T_{Period}^{n+1}. \quad (1)$$

In Equation (1), both T_{Wrn}^n and T_{setup}^{n+2} are constants, which were determined when a TBLB latch is designed. We shall minimize T_{delay}^n , T_{or}^n and $T_{C_{n+1}}^{n+1}$ during logic and physical synthesis such that the timing constraint is satisfied. With circuit aging, the delay of combinational logic cells becomes much longer, and hence the delay margin for error-correction is even more stringent.

Due to the aforementioned critical and stringent timing constraint, the physical design styles for TBLB latches may have great impact on the circuit performance. Figure 3 shows three different physical design styles for TBLB latches, including *discrete 1-bit TBLB latches*, *integrated TBLB macros*, and *distributed multi-bit TBLB latches*. The design style with discrete 1-bit TBLB latches may have more gates in the OR-tree, which results in much larger T_{or} . Although the design style with integrated TBLB macros will result in the smallest T_{or} , it may introduce more critical paths in the combinational circuits among different pipeline stages because of longer interconnections. Compared with discrete 1-bit TBLB latches and integrated TBLB macros, the design style with distributed multi-bit TBLB latches is expected to achieve the best tradeoff among T_{delay} , T_{or} , and $T_{C_{n+1}}$ during physical synthesis.

1.1 Previous Work

Recent physical synthesis approaches [5, 6] dealt with timing error resilient circuits. However, they did not consider the special timing requirement for TBLB error resilient circuits. These works mainly focused on reducing hold buffer penalties arising from short paths instead of shortening the error-detection delay, or the OR-tree delay, for larger delay margin of error correction.

Although flip-flop/latch merging and multi-bit flip-flop/latch generation methods [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17] during physical synthesis were extensively studied in the literature, all of them tried to merge as many flip-flops as possible while satisfying timing and other physical design constraints. None of them consider the delay of combinational and error-detection circuits as the first-order design objective, whereas it is essential in TBLB resilient circuits.

1.2 Our Contributions

This paper presents a novel timing-driven multi-bit latch replacement method for TBLB resilient circuits. The contributions of this paper can be summarized in the following:

- Different from the previous works which do not consider the special timing requirement for TBLB latches, we proposed a novel physical synthesis flow and algorithms for TBLB resilient circuits. Our approach can simultaneously minimize the delay of error-detection circuits and that of ordinary data paths.
- In order to reduce the delay of TBLB error-detection circuits and consequently increase the margin for TBLB error-correction, we propose a novel OR-tree-latency-aware TBLB latches clustering to minimize both OR-tree wirelength and latency with Hamiltonian path and dynamic-programming (DP) formulations.
- Experimental results based on the IWLS-2005 benchmark show that the proposed approach applying multi-bit TBLB latches is very effective in reducing the delay of both combinational and error-detection circuits compared with TBLB macro based approach.

The remainder of this paper is organized as follows. Section 2 introduces the problem formulation. Section 3 details the proposed placement flow and the corresponding algorithms. Section 4 reports the experimental results. Finally, section 5 concludes this paper.

2. PROBLEM FORMULATION

Given a TBLB error-resilient circuit, which contains combinational logic cells, sequential logic cells including TBLB latches and their pipeline stages, maximum capacitance loading of a pulse-generator, and multi-bit TBLB latches with different bit numbers, we want to generate a legalized non-overlapped placement for the TBLB resilient circuit with multi-bit TBLB latches such that the delay of combinational and error-detection circuits, T_{delay} , $T_{C_{n+1}}$, and T_{or} , is minimized while satisfying the maximum loading constraint of all pulse generators (i.e. the maximum bit number of multi-bit TBLB latches), and other common physical design rules and/or constraints.

3. THE PROPOSED TBLB PHYSICAL SYNTHESIS FLOW AND ALGORITHMS

Based on the problem formulation, we propose a novel physical synthesis flow for TBLB error-resilient circuits, which consists of five major steps: (1) Initial placement, (2) OR-tree-latency-aware TBLB latch clustering, (3) PG-group-aware incremental placement, (4) Multi-bit TBLB latch replacement, and (5) OR-Tree synthesis. At the beginning, all TBLB latches are one-bit. The initial placement produces a good solution in terms of wirelength, density, and other placement constraints. Based on the initial placement, the TBLB latches is then clustered according to the construction of the OR-tree with latency minimization, which is followed by PG group extraction for all TBLB latches. The incremental placement is further performed according to PG groups of TBLB latches for timing optimization. The multi-bit TBLB latches are finally generated, and the OR-trees are re-synthesized to achieve the shortest OR-tree delay with multi-bit TBLB latches.

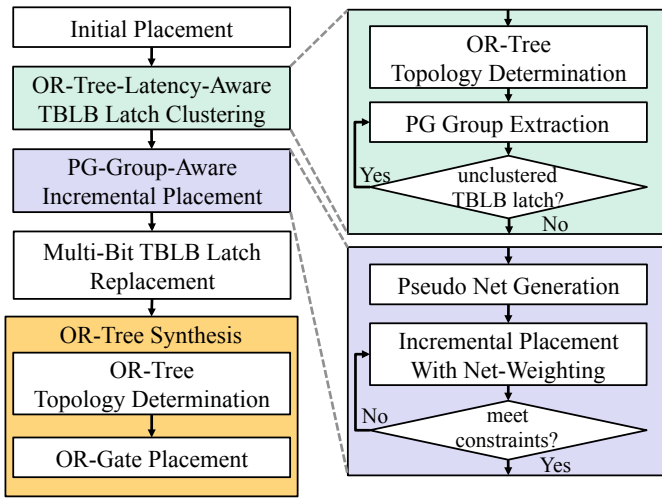


Figure 4: The proposed physical synthesis flow for TBLB error-resilient circuits.

3.1 Initial Placement

Since minimizing signal net wirelength and placement density are the most important objectives for a general global placement problem, we consider both objectives and try to find the best trade-off between the two objectives at the beginning stage. Inputting a design netlist, we first perform initial placement based on the analytical placer [18], to obtain the initial locations of all cells. The initial placement is formulated with an unconstrained minimization problem as follows:

$$\min \hat{W}(\mathbf{x}, \mathbf{y}) + \lambda_d \sum (\hat{D}_{b_i}(\mathbf{x}, \mathbf{y}) - D_{MAX})^2, \quad (2)$$

where $\hat{W}(\mathbf{x}, \mathbf{y})$ is the log-sum-exponential (LSE) wirelength function for all signal nets, $\hat{D}_{b_i}(\mathbf{x}, \mathbf{y})$ is a smoothed density function for each bin, D_{MAX} is the maximum allowable placement density, and λ_d is a Lagrange multiplier, which controls the weighting of the density. We solve a series of the unconstrained optimization problem in Equation (2) based on the conjugate gradient method with increasing λ_d until the cells are evenly distributed throughout the chip area. Similar to [15], we integrate our analytical placer with a timer, and apply a net-weighting method to enlarge the wirelength costs of the timing critical nets in the objective function during the last few iterations.

After performing the initial placement, we can capture more accurate physical information to optimize the locations of all TBLB error-detection latches for reducing the delay of error-detection circuits in the following steps.

3.2 OR-Tree-Latency-aware TBLB Flip-Flop Clustering

Once the cells are evenly distributed with minimized wirelength, we then perform OR-tree-latency-aware TBLB latch clustering to reduce the delay of error-detection circuits and clock sinks without degrading circuit performance. The proposed OR-tree-latency-aware TBLB latch clustering consists of two major steps: (1) OR-tree topology determination, and (2) PG group extraction.

3.2.1 OR-Tree Topology Determination

Since we want to construct an OR-tree topology with minimized wirelength, we first construct a TBLB latch chain to represent the adjacency relationship among different TBLB latches with respect to their physical locations. In order to minimize the total distance of the TBLB latch chain, we model the TBLB latch chain construction problem as a Hamiltonian path problem, and find an optimal TBLB latch chain by searching the shortest Hamiltonian path [19]. The closer TBLB latches in a TBLB latch chain will have higher opportunity to be clustered into the same branch or neighboring branches of an OR-tree topology. In addition, minimizing the total distance of a TBLB latch chain can help to reduce total wirelength of the OR-tree when performing OR-tree synthesis.

After obtaining the TBLB latch chain by searching the shortest Hamiltonian path, we formulate the problem of OR-tree topology determination as a dynamic programming problem by inputting the TBLB latch chain. The objective, $D[i, j]$, is to parenthesize the sub-chain of TBLB latches, $f_i \dots f_j$, in order to minimize the latency of OR-tree, which can be defined in Equation (3). d_{f_i} is the negative slack of f_i from C_n . d_{OR} is the intrinsic delay of OR-gate, and d_{wire} is the estimated delay of the wire. By using d_{f_i}

$$D[i, j] = \begin{cases} d_{f_i} & \text{if } i = j, \\ \min_{i \leq k < j} \{ \max(D[i, k], D[k + 1, j]) + d_{OR} + d_{wire} \} & \text{if } i < j. \end{cases} \quad (3)$$

of each f_i as the weight, we can estimate the locations of all OR-gates based on the force-directed method and calculate the latency of each sub-path of OR-tree as its solutions during our algorithms.

Once the optimal parenthesization of the TBLB latch chain is obtained, we then construct the corresponding OR-tree topology for PG grouping extraction. The input for the OR-tree topology construction is a set of nodes, which represent the corresponding TBLB latches, respectively, and the initial weight of each node is set to 1. We first trace the TBLB latch chain according to the parentheses from inner to outer, and add the nodes to the sub-chain of TBLB latches when there is a pair of parentheses. The weight of each node is then assigned by summing up the weight of its child nodes. Figure 5 shows an example of eleven TBLB latches, $f_1, f_2 \dots f_{11}$ in chain, E , with optimal parenthesization and the corresponding weighted OR-tree topology. Based on the weighted OR-tree topology, the nodes, whose weights are more than two, can be replaced by multi-input (e.g. 2-input, 3-input, etc.) OR gates in the cell library according to the best total weight.

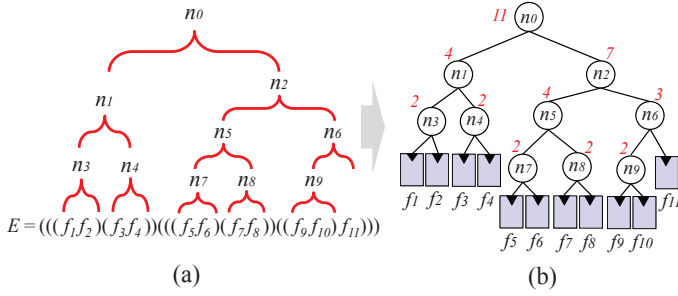


Figure 5: (a) An example for OR-tree topology determination after dynamic programming. (b) The corresponding OR-tree topology with node weights.

3.2.2 PG Group Extraction

After constructing the weighted OR-tree topology, we further extract the PG groups from root to leaves of the OR-tree topology according to the maximum capacitance loading constraint. Intuitively, grouping the TBLB latches having the same branch or the nearest branches in the OR-tree topology can help to reduce the total wirelength of the OR-tree as well as the OR-tree latency. In addition to the capacitance loading constraint, we estimate the total signal net wirelength of each candidate of PG group, g_i , and select the candidate of PG groups which contain total signal net wirelength of g_i within $3X$ of $W_{New}^{g_i}/W_{Ori}^{g_i}$, where $W_{Ori}^{g_i}$ and $W_{New}^{g_i}$ are the estimated total signal net wirelengths of g_i before and after PG grouping. With this constraint, the total signal net wirelength of selected g_i can be prevented from large increase and timing quality can be maintained when grouping g_i during PG-group-aware incremental placement. The algorithm iteratively clusters the TBLB latches based on the result of weighted OR-tree topology until all the nodes of the weighted OR-tree topology is traced or all the

TBLB latches are grouped.

3.3 PG-Group-Aware Incremental Placement

After applying OR-tree-latency-aware TBLB latch clustering according to OR-tree latency and physical locations of TBLB latches, PG-group-aware incremental placement is performed to progressively place TBLB latches of the same group close to each other for reducing the delay of error-detection circuits. In addition to the placement adjustment among TBLB latches, the locations of all the other cells can also be refined such that the placement density constraints can be met without degrading circuit performance.

To achieve this, we first calculate the target locations of all PG groups by the force-directed method. The value of each force of a TBLB latch, f_i , is set to d_{f_i} . Since the larger d_{f_i} implies that the data path related to f_i is more critical than other data paths, we would like to locate the target location of the corresponding PG group closer to f_i such that the wirelength of the path related to f_i after moving f_i to a new location does not increase too much during the PG-group-aware incremental placement. Figure 6 gives an example of a PG group with four TBLB latches. In Figure 6(a), the four TBLB latches, f_1, f_2, f_3 , and f_4 have different force values, $3ns$, $4ns$, $2ns$, and $4ns$, respectively. Since the force values of the TBLB latches, f_2 and f_4 , are larger than the other two, the target location of this PG group is located closer to f_2 and f_4 , but farther from f_1 and f_3 .

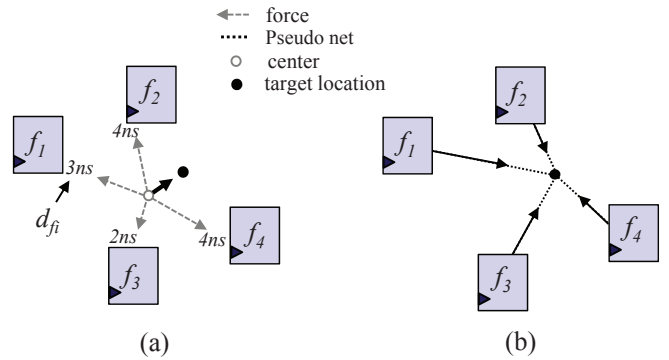


Figure 6: An example of a PG-group-aware incremental placement for a PG group containing four TBLB latches. (a) Determination of the target location of the PG group based on the force-directed method. (b) Incremental placement of the four latches in the same PG group with pseudo nets and attracting force.

In order to place all TBLB latches of the same group closer to each other, the pseudo nets are introduced. Each pseudo net connects the target location and one of the TBLB flip-flops in the same group such that the delay of error-detection circuits can be reduced. To strengthen the attractions, the weight of each pseudo net should be greater than the weight of the ordinary signal nets, which is about 10X according to our experimental study. Figure 6(b) shows

the four TBLB latches in the same group are connected and attracted to the target location by the generated pseudo nets with strengthened attractions.

3.4 Multi-Bit TBLB Flip-Flop Replacement and OR-Tree Synthesis

Once all TBLB latches of the same PG group are closed enough to the target location, the TBLB latches in each PG group are replaced with a multi-bit TBLB latch. We reconstruct the OR-tree topology because the original OR-tree topology might be slightly changed after multi-bit TBLB latch replacement. Based on the reconstructed OR-tree topology, we can calculate the optimal OR gate locations. Consequently, the resulting OR-tree with optimized wirelength and latency can be obtained.

4. EXPERIMENTAL RESULTS

We implemented our algorithms in C/C++ programming languages on a 2.26GHz Intel Xeon machine under the Linux operating system, and integrated with the placer based on NTUplace3 [18]. We experimentally tested our algorithms on the five OpenCores [20] circuits in the IWLS-2005 benchmark suite [21] with the Nangate 45nm Open cell Library [22]. Based on the library, a pulse generator can drive at most 10 TBLB latches, and the available multi-bit TBLB latches range from 1 to 10 bits. Table 1 lists the names of the circuits (“Circuit”), the numbers of combinational logic cells (“# of Comb. Logic Cells”), the numbers of sequential logic cells (“# of Seq. Logic Cells”), and the numbers of nets (“# of Nets”). We compared the design style with TBLB macros, as seen in Figure 3(b), resulting from the analytical placer [18], and the design style with multi-bit TBLB latches, as seen in Figure 3(c), resulting from the proposed approach. After obtaining a legal placement with either TBLB macros or multi-bit TBLB latches, hold buffer insertion/short path padding [6] should be further performed to fix hold violations.

Table 1: Five OpenCores circuits [20] in IWLS-2005 benchmark [21].

Circuit	# of Comb. Logic Cells	# of Seq. Logic Cells	# of Nets
<i>ac97_ctrl</i>	9656	2199	11637
<i>aes_core</i>	20265	530	20626
<i>mem_ctrl</i>	10357	1083	11280
<i>pci_bridge32</i>	13457	3359	16726
<i>wb_conmax</i>	28264	770	29675

Table 2 lists the names of the benchmark circuits (“Circuit”), clock cycle time (“ T_{Period} ”), total signal net wirelength (“WL”), OR-tree delay (“ T_{or} ”), clock wirelength (“CWL”), worst negative slack (“WNS”), total negative slack (“TNS”), and runtime (“Time”) for the two approaches based on different design styles of TBLB resilient circuits. The clock wirelength was obtained based on [23], while the worst negative slack and total negative slack were obtained based on [24].

The total signal net wirelength resulting from the design style with multi-bit TBLB latches is 48% shorter than that resulting from the design style with TBLB macros. Since the design style with

TBLB macros compacts all the TBLB latches without considering any physical information of the combinational circuits, the interconnections from TBLB latches to combinational circuits are substantially increased.

The OR-tree latency resulting from the design style with multi-bit TBLB latches is 39% larger than that resulting from the design style with TBLB macros. It is because the design style with TBLB macros has the advantages of integrated TBLB resilient circuits, including all TBLB latches, in each pipeline stage. The OR-tree latency can be minimized due to the compacted layout of TBLB resilient circuits.

The clock wirelength resulting from the design style with multi-bit TBLB latches is 12% larger than that resulting from the design style with TBLB macros. Similar to OR-tree latency, the design style with TBLB macros has the advantages of integrated all TBLB latches. The clock wirelength can be reduced due to much less clock sinks.

The worst negative slack and total negative slack resulting from the design style with multi-bit TBLB latches are 39% and 50% smaller than those resulting from the design style with TBLB macros. Since the design style with TBLB macros may introduce longer signal net wirelength and more critical paths in the combinational circuits among different pipeline stages. The circuit performance may also be degraded.

The runtime resulting from the design style with multi-bit TBLB latches is 10% larger than that resulting from the design style with TBLB macros because the design style with multi-bit TBLB latches resulting from the proposed approach additionally performs TBLB latch clustering, incremental placement, and OR-tree synthesis, which require more sophisticated computations.

To sum up, the design style with multi-bit TBLB latches resulting from the proposed physical synthesis flow and the corresponding algorithms based on the IWLS-2005 benchmark are very effective in reducing the delay of both combinational and error-detection circuits, which indicates better circuit reliability due to circuit aging.

5. CONCLUSIONS

In this paper, we have introduced the problem of multi-bit TBLB latch replacement for the state-of-the art TBLB resilient circuits. We have also proposed a novel timing-driven multi-bit latch replacement method for low-power TBLB resilient circuits, which simultaneously minimizes the delay of error-detection circuits and that of ordinary data paths. Experimental results based on the IWLS-2005 benchmark have shown that the proposed approach is very effective in reducing the delay of both combinational and error-detection circuits without degrading circuit performance, which indicates better circuit reliability due to circuit aging.

6. REFERENCES

- [1] H. Fuketa, M. Hashimoto, Y. Mitsuyama, and T. Onoyo, “Adaptive performance compensation with in-situ timing error predictive sensors for subthreshold circuits,” *IEEE TVLSI*, vol. 20, no. 2, pp. 333–343, Feb. 2012.
- [2] S. Das, D. Roberts, S. Lee, S. Pant, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, “A self-tuning DVS processor

Table 2: Comparisons of total signal net wirelength (“WL”), OR-tree latency (“ T_{or} ”), clock wirelength (“CWL”), worst negative slack (“WNS”), total negative slack (“TNS”), and runtime (“Time”) based on both design styles, TBLB macros and multi-bit TBLB latches.

Circuit	T_{Period} (ns)	The design style with TBLB macros						The design style with multi-bit TBLB latches resulting from the proposed approach					
		WL $\times 10^8$ (nm)	T_{or} (ns)	CWL $\times 10^7$ (nm)	WNS (ns)	TNS (ns)	Time (s)	WL $\times 10^8$ (nm)	T_{or} (ns)	CWL $\times 10^7$ (nm)	WNS (ns)	TNS (ns)	Time (s)
<i>ac97_ctrl</i>	0.44	8.10	0.31	1.91	1.28	598.65	787	2.98	0.33	2.38	0.92	266.14	836
<i>aes_core</i>	1.21	5.90	0.15	2.84	1.32	72.99	127	4.15	0.25	2.89	0.86	37.56	127
<i>mem_ctrl</i>	1.55	6.32	0.22	1.44	1.68	170.45	216	2.67	0.26	1.60	0.84	89.54	231
<i>pci_bridge32</i>	1.01	15.05	0.36	3.23	2.10	609.73	1905	4.80	0.46	3.90	0.66	113.26	2395
<i>wb_conmax</i>	0.92	11.38	0.19	3.03	1.80	357.11	514	9.10	0.33	3.18	1.58	303.60	588
Comp.	-	1	1	1	1	1	1	0.522	1.392	1.125	0.612	0.504	1.107

- using delay-error detection and correction,” *IEEE JSSC*, vol. 41, no. 4, pp. 792–804, 2006.
- [3] S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw, “RazorII: In situ error detection and correction for PVT and SER tolerance,” *IEEE JSSC*, vol. 44, no. 1, pp. 32–48, 2009.
- [4] J.-S. Wang, K.-J. Chang, T.-J. Lin, R. W. Prasojo, and C. Yeh, “A 0.36V, 33.3uW 18-band ANSI S1.11 1/3-octave filter bank for digital hearing aids in 40nm CMOS,” in *Proc. VLSIC*, 2013, pp. C254–C255.
- [5] M. Kurimoto, H. Suzuki, R. Akiyama, T. Yamanaka, H. Ohkuma, H. Takata, and H. Shinohara, “Phase-adjustable error detection flip-flops with 2-stage hold-driven optimization, slack-based grouping scheme and slack distribution control for dynamic voltage scaling,” *ACM TODAES*, vol. 15, no. 2, 2010.
- [6] Y.-M. Yang, I. H.-R. Jiang, and S.-T. Ho, “PushPull: Short-path padding for timing error resilient circuits,” *IEEE TCAD*, vol. 33, no. 4, pp. 558–570, 2014.
- [7] C.-L. Chang and I. H.-R. Jiang, “Pulsed-latch replacement using concurrent time borrowing and clock gating,” *IEEE TCAD*, vol. 32, no. 2, pp. 242–246, 2013.
- [8] Y.-T. Chang, C.-C. Hsu, M. P.-H. Lin, Y.-W. Tsai, and S.-F. Chen, “Post-placement power optimization with multi-bit flip-flops,” in *Proc. ICCAD*, 2010, pp. 218–223.
- [9] M. P.-H. Lin, C.-C. Hsu, and Y.-T. Chang, “Post-placement power optimization with multi-bit flip-flops,” *IEEE TCAD*, vol. 30, no. 12, pp. 1870–1882, 2011.
- [10] S.-H. Wang, Y.-Y. Liang, T.-Y. Kuo, and W.-K. Mak, “Power-driven flip-flop merging and relocation,” *IEEE TCAD*, vol. 31, no. 2, pp. 180–191, 2012.
- [11] 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, 2012.
- [12] Y.-T. Shyu, J.-M. Lin, C.-P. Huang, C.-W. Lin, Y.-Z. Lin, and S.-J. Chang, “Effective and efficient approach for power reduction by using multi-bit flip-flops,” *IEEE TVLSI*, vol. 21, no. 4, pp. 624–635, 2013.
- [13] S.-Y. S. Liu, W.-T. Lo, C.-J. Lee, and H.-M. Chen, “Agglomerative-based flip-flop merging and relocation for signal wirelength and clock tree optimization,” *ACM TODAES*, vol. 18, no. 3, p. 40, 2013.
- [14] Z.-W. Chen and J.-T. Yan, “Routability-constrained multi-bit flip-flop construction for clock power reduction,” *Integration*, vol. 46, no. 3, pp. 290–300, 2013.
- [15] M. P.-H. Lin, C.-C. Hsu, and Y.-C. Chen, “Clock-tree aware multibit flip-flop generation during placement for power optimization,” *IEEE TCAD*, vol. 34, no. 2, pp. 280–292, 2015.
- [16] C.-C. Hsu, M. P. Lin, and Y.-T. Chang, “Crosstalk-aware multi-bit flip-flop generation for power optimization,” *Integration*, vol. 48, pp. 146–157, 2015.
- [17] C. Xu, P. Li, G. Luo, Y. Shi, and I. H.-R. Jiang, “Analytical clustering score with application to post-placement multi-bit flip-flop merging,” in *Proc. ISPD*, 2015, pp. 93–100.
- [18] T.-C. Chen, Z.-W. Jiang, T.-C. Hsu, H.-C. Chen, and Y.-W. Chang, “NTUplace3: An analytical placer for large-scale mixed-size designs with preplaced blocks and density constraints,” *IEEE TCAD*, vol. 27, no. 7, pp. 1228–1240, 2008.
- [19] K. Helsgaun, “General k -opt submoves for the lin-kernighan TSP heuristic,” *Math. Program. Comput.*, vol. 1, no. 2-3, pp. 119–163, 2009.
- [20] OpenCores. [Online]. Available: <http://www.opencores.org>
- [21] IWLS 2005 Benchmarks. [Online]. Available: <http://iwls.org/iwls2005/benchmarks.html>
- [22] Nangate 45nm Open cell Library. [Online]. Available: <http://www.nangate.com>
- [23] D. J.-H. Huang, A. B. Kahng, and C.-W. A. Tsao, “On the bounded-skew clock and steiner routing problems,” in *Proc. DAC*, 1995, pp. 508–513.
- [24] Cadence, Inc. [Online]. Available: <http://www.cadence.com>