Cell-Aware MBFF Utilization for Clock Power Reduction

Jin-Tai Yan and Meng-Tian Chen
Department of Computer Science and Information Engineering,
Chung-Hua University
Hsinchu, Taiwan, R. O. C.

Chia-Heng Yen
Department of Computer Science and Engineering,
National Taiwan Ocean University
Keelung, Taiwan, R.O.C.

Abstract -- ***Utilization of multi-bit flip-flops(MBFFs) in a synchronous design has been becoming a significant methodology for clock power reduction. In this paper, given a synchronous system with a set of 1-bit flip-flops in a placement plane, the timing constraints of the associated signals on the flip-flops and the available MBFFs in a cell library, firstly, based on the timing constraints of the signals on the flip-flops, a timing-constrained merging graph(TCMG) can be constructed. Furthermore, based on the available MBFFs in the given cell library, an ILP(Integer-Linear-Programming) formulation can be proposed to merge 1-bit flip-flops into the available MBFFs for clock power reduction. Compared with the original design, the experimental results show that our proposed ILP-based approach can reduce 20.05% of the clock power for five tested examples on the average.***

I. INTRODUCTION

In a synchronous design, utilization of multi-bit flip-flops (MBFFs) for a set of 1-bit flip-flops can eliminate redundant inverters to achieve an effective power-saving implementation. In addition to the reduction of the power consumption and the total area on flip-flops, utilization of MBFFs can bring some other benefits for the construction of a clock tree. The more the 1-bit flip-flops are merged into MBFFs, the fewer the clock sinks in a clock tree are. Hence, the less routing cost and the less sink capacitance can be used to construct a clock tree. Recently, numerous studies[1-9] had attempted to reduce the clock power by utilizing MBFFs during post-placement. The problem of replacing1-bit flip-flops with MBFFs can be formulated to minimize the power consumption on flip-flop with satisfying both the given timing and placement density constraints.

For utilization of MBFFs during post-placement, Yan et al.[1] firstly proposed an efficient approach to utilize MBFFs to reduce the clock power for a placement result. In this work, a greedy merging process is adopted for replacement of MBFFs under the given timing and density constraints. Chen et al.[2] further proposed an efficient approach to merge and replace some 1-bit flip-flops with MBFFs for the same problem with additional routability consideration. However, the replacement results may be impractical because it is assumed that MBFFs with any bit number are available in a cell library. In the consideration of limited MBFFs in a cell library, Lin et al.[3-4] presented a progressive window-based approach to merge and replace some 1-bit flip-flops with available MBFFs under the given timing and density constraints. In this work, the proposed approach obtained more clock power saving and used less routing wirelength. By transforming the problem of utilizing MBFFs into the problem of finding cliques in an adjacent graph, Wang et al.[5], Jiang et al.[6], Shyu et al[7] and Liu et al.[8] also proposed a clique-based approach, a three-phase approach, an effective and efficient approach and an agglomerative-based approach to merge and replace some 1-bit flip-flops with available MBFFs under the given timing and density constraints, respectively. However, the works neglects the important features of the available MBFFs in a practical cell library such that the published approaches cannot obtain optimal power-saving result. Recently, based on the construction of an extraction tree for the available MBFFs in a cell library, Chen et al.[9] proposed an efficient library-aware search-based approach to find the best utilization of the available MBFFs to maximize the reduction of the total consumed power. However, the proposed heuristic search cannot lead to an optimal solution.

In this paper, given a synchronous system with a set of 1-bit flip-flops in a placement plane, the timing constraints of the associated signals on the flip-flops and the available MBFFs in a cell library, an ILP(Integer-Linear-Programming)-based approach is proposed to merge 1-bit flip-flops into the available MBFFs for clock power reduction. Compared with the original design, the experimental results show that our proposed ILP-based approach can reduce 20.05% of the clock power for five tested examples on the average. Compared with Yan's approach[9], the experimental results show that our proposed approach can use less CPU time to further reduce 1.74% of the clock power for five tested examples on the average.

II. PROBLEM FORMULATION

In general, a 1-bit flip-flop can be composed by a clock driver, a master latch and a slave latch as illustrated in Fig. 1(a). As technology advances into 65nm and beyond, a minimum-sized inverter is capable of driving multiple latches. As a result, a clock driver can be used to drive multiple 1-bit flip-flops such that multiple 1-bit flip-flops can be replaced with a MBFF in a synchronous design. As illustrated in Fig. 1(b), two 1-bit flip-flops can be merged into a 2-bit flip-flop and 2-bit flip-flop can be utilized in a synchronous design. By sharing one common clock driver to eliminate the unnecessary inverters in the utilization of an

978-1-5090-6113-6/16/$31.00 ©2016 IEEE

available MBFF, the consumed power of a MBFF is less than that of the original multiple 1-bit flip-flops. Besides that, the occupied area of a MBFF is less than that of the original multiple 1-bit flop-flops.

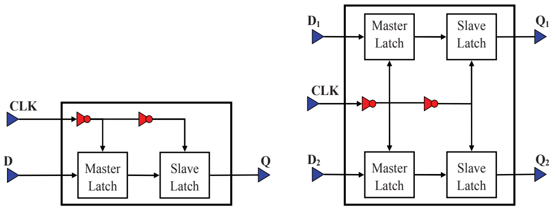


Fig.1 Design diagrams of 1-bit and 2-bit flip-flops

Based on the master-slave structure inside a MBFF, a minimum-sized inverter can be only capable of driving finite latches under technology consideration. Hence, the bit number on a MBFF in a cell library must be restricted by an upper value. On the other hand, if MBFFs with all the possible bit numbers are considered in a cell library, the number of the MBFFs in a cell library may lead to take more undesirable time in logic synthesis. Hence, the bit number on the MBFFs in a practical cell library must be discrete and bounded in a small range. As a result, there are few available MBFFs in a practical cell library.

For cell-aware MBFF utilization in a synchronous system, it is known that the input and output signals on the original 1-bit flip-flops must be rerouted and the delays of the input and output signals on the original 1-bit flip-flops may be modified due to the locations of the new MBFFs. To ensure that the delays of the input and output signals in the original1-bit flip-flops are satisfied, the *timing constraint* of any input or output signal in a 1-bit flip-flop can be defined as the maximum required delay of the input or output signal in the 1-bit flip-flop. On the other hand, a placement plane is generally partitioned into an array of bins and each bin has a restricted available area. Because a MBFF needs a larger area than a 1-bit flip-flop, the available area for the MBFFs inside the placed bin must be evaluated. To make sure whether any bin accommodates a desired multi-bit flip-flop, the *area constraint* of any bin in a placement plane can be defined as the maximal available area inside the bin.

Given a synchronous system with a set of n 1-bit flip-flops, $F=\{FF_1, FF_2,\ldots,FF_n\}$, in a placement plane and the available MBFFs in a cell library, L, any 1-bit flip-flop, FF_k, $1\leq k\leq n$, has the input timing constraint, $T^I_k \leq t^I_{k,max}$ from its input pin, p^i_k, to the flip-flop, FF_k, and the output timing constraint, $T^O_k \leq t^O_{k,max}$, from the flip-flop, FF_k, to its output pin, p^o_k. In a placement plane, the plane contains an array of mxm bins, $B=\{B_{1,1},\ldots,B_{1,m},\ldots,B_{m,1},\ldots,B_{m,m}\}$, and any bin, $B_{i,j}$, $1\leq i, j\leq m$, has its area constraint, $a_{i,j}$. In the given library, L, any MBFF has some circuit specifications including delay, area and power consumption. Under the given timing constraints of the input and output signals on the given 1-bit flip-flops and the area constraints of the bins in a placement plane, the cell-aware utilization of the available MBFFs in a given cell library is to merge some 1-bit flip-flops into MBFFs in a synchronous design such that the total power consumption is minimized.

Given a synchronous system with 10 1-bit flip-flops, FF_1, $FF_2,\ldots,FF_{10}$, inside an array of 4x4 bins, the timing constraints for the input and output signals on 10 flip-flops, the area constraints for 16 partitioned bins in a placement plane in Fig. 2(a), it is assumed that 2-bit flip-flop, *2FF*, and 4-bit flip-flop, *4FF*, are available in the given cell library. For the cell-aware utilization of the available MBFFs in a synchronous system, the maximal power-saving result contains two 1-bit flip-flops, FF_6 and FF_{10}, two 2-bit flip-flops, $2FF_1\{FF_1, FF_2\}$and $2FF_2\{FF_4, FF_5\}$, and one 4-bit flip-flop, $4FF_1\{FF_3, FF_7, FF_8, FF_9\}$, in a synchronous system as illustrated in Fig. 2(b).

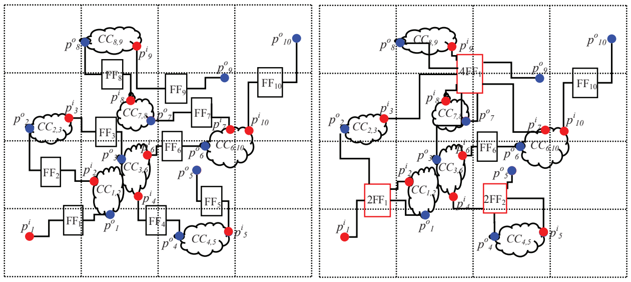


Fig. 2 Cell-aware MBFF utilization in a synchronous system with 10 1-bit flip-flops

III. Cell-Aware MBFF Utilization for Clock Power Reduction

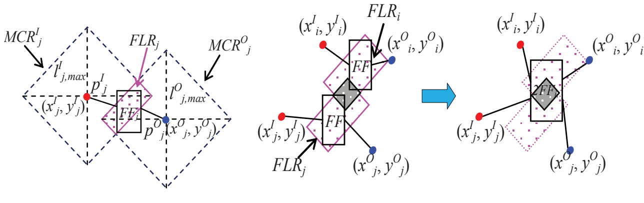
For cell-aware MBFF utilization in a synchronous system, an ILP-based approach can be proposed and the proposed approach can be divided into two steps: *Construction of a TCMG* and *ILP formulation for cell-aware MBFF utilization*.

3.1 Construction of a TCMG

Given a set of timing constraints on all the flip-flops in a synchronous design, for a pair of two timing constraints, $T^I_j \leq t^I_{j,max}$ and $T^O_j \leq t^O_{j,max}$, on flip-flop, FF_j, the maximal routing lengths, $l^I_{j,max}$ and $l^O_{j,max}$, of the input and output connections in a Manhattan routing plane on the flip-flop, FF_j, can be computed. Furthermore, based on the maximum routing lengths of the input and output connections on all the flip-flops in a synchronous design, the corresponding possible locations of all the 1-bit flip-flops can be obtained.

In a Manhattan routing model, the collection of points within a fixed distance of a given point is called as a *Manhattan circular region*(MCR) whose boundary is composed of two line segments with slope +1 and two line segments with slope -1. Clearly, the radius of a Manhattan circular region is the Manhattan distance between the given point and the MCR boundary. For any 1-bit flip-flop, FF_j, $1 \leq j \leq n$, its *feasible location region(FLR)*, FLR_j, can be defined as the overlapping region of the Manhattan circular region, MCR^I_j, with the given pin, p^I_j, on the coordinate, $(x^I_j$,

y^I_j), and its radius, $l^I_{j,\max}$, for the input connection on FF_j and the Manhattan circular region, MCR^O_j, with the given pin, p^O_j, on the coordinate, (x^O_j, y^O_j), and its radius, $l^O_{j,max}$, for the output connection on FF_j as illustrated in Fig. 3(a), that is, if the flip-flop, FF_j, is assigned inside the region, FLR_j, the timing constraint, $T^I_j \le t^I_{j,max}$, and the timing constraint, $T^O_j \le t^O_{j,max}$, for the input and output connections on FF_j, can be satisfied on the same time. If the FLRs of the k given 1-bit flip-flops overlap each other, i.e., there is a common overlapping region for the k corresponding FLRs, the k 1-bit flip-flops can be merged into a k-bit flip-flop and the k-bit flip-flop can be utilized with no timing violation on the k given 1-bit flip-flops. As illustrated in Fig. 3(b), it is assumed that there are two 1-bit flip-flops, FF_i and FF_j, inside a placement plane. Based on the timing constraints on the flip-flops, FF_i and FF_j, the FLRs, FLR_i and FLR_j, of the two flip-flops, FF_i and FF_j, can be constructed. Because the FLRs, FLR_i and FLR_j, of two 1-bit flip-flops, FF_i and FF_j, overlap each other, the two 1-bit flip-flops, FF_i and FF_j, can be merged into a 2-bit flip-flop, $2FF_{i,j}$, and the flip-flop, $2FF_{i,j}$, can be utilized with no timing violation.



(a) (b)

Fig. 3 Construction of feasible location region for two given flip-flops, FF_i and FF_j,

Given a set of timing constraints, $T^O_1 \le t^O_{1,max}$, $T^I_1 \le t^I_{1,max}$, $T^O_2 \le t^O_{2,max}$, $T^I_2 \le t^I_{2,max}$,…, and $T^O_n \le t^O_{n,max}$, $T^I_n \le t^I_{n,max}$, on n 1-bit flip-flops, FF_1, FF_2,..., and FF_n, in a synchronous design, the FLRs of all the given flip-flops in a synchronous design can be firstly constructed. Based on the constructed FLRs of all the given flip-flops and the overlapping relations of the constructed FLRs, an undirected timing-constrained merging graph(TCMG), $G(V, E)$, can be further constructed as follows: each vertex, v_i, in V represents the FLR, FLR_i, of a flip-flop, FF_i, and each undirected edge, $e_{i,j}$, in E represents the overlapping relation between two FLRs, FLR_i and FLR_j. Refer to the 10 given 1-bit flip-flops, FF_1, FF_2,..., and FF_{10}, in a synchronous design in Fig 2, based on the 10 constructed FLRs of the 10 flip-flops and the overlapping relations of the 10 constructed FLRs, a TCMG, $G(V, E)$, for a given set of timing constraints on the flip-flops can be constructed as illustrated in Fig. 4, where V=$\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$ and E=$\{e_{1,2}, e_{1,4}, e_{2,3}, e_{3,6}, e_{3,7}, e_{3,8}, e_{3,9}, e_{4,5}, e_{4,6}, e_{5,6}, e_{6,7}, e_{6,8}, e_{7,8}, e_{7,9}, e_{7,10}, e_{8,9}\}$.

3.2 ILP Formulation for Cell-aware MBFF utilization

Based on the construction of a TCMG for a set of timing constraints on the flip-flops in a synchronous design, the extraction of a k-clique in a TCMG implies that a set of k 1-bit flip-flops may be merged into a k-bit flip-flop in a synchronous design. Furthermore, if the merged k-bit flip-flop is an available MBFF in a given cell library, the k-bit flip-flop can be used for utilization of MBFFs in a synchronous design and the corresponding k-clique in a TCMG can be defined as a *k-utilized pattern* in a TCMG. For cell-aware MBFF utilization in a synchronous design, it implies that a k-utilized pattern in a TCMG can be directly extracted from the TCMG and a set of k 1-bit flip-flops can be further replaced with an available k-bit MBFF in the cell library.

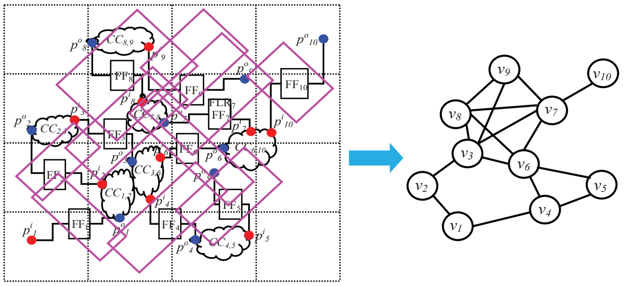


Fig. 4 Construction of a TCMG for 10 given flip-flops in a synchronous design

In general, a 1-bit flip-flop can be treated as an available MBFF in a given cell library. Hence, the smallest pattern in a TCMG is a 1-utilized pattern, that is, an isolated vertex. According to the information of the available MBFFs in a given cell library, it is known that a TCMG can be covered by a set of utilized patterns. Refer to the TCMG in Fig. 4, given 1-bit flip-flop, 2-bit flip-flop and 4-bit flip-flop in a cell library, the TCMG can be covered by 10 1-utilized prime-patterns, 16 2-utilized prime-patterns, $\{v_1, v_2\}$, $\{v_1, v_4\}$, $\{v_2, v_3\}$, $\{v_3, v_6\}$, $\{v_3, v_7\}$, $\{v_3, v_8\}$, $\{v_3, v_9\}$, $\{v_4, v_5\}$, $\{v_4, v_6\}$, $\{v_5, v_6\}$, $\{v_6, v_7\}$, $\{v_6, v_8\}$, $\{v_7, v_8\}$, $\{v_7, v_9\}$, $\{v_7, v_{10}\}$ and $\{v_8, v_9\}$, and 2 4-utilized patterns, $\{v_3, v_6, v_7, v_8\}$ and $\{v_3, v_7, v_8, v_9\}$, as illustrated in Fig. 5.

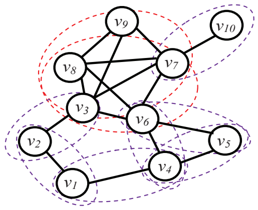


Fig. 5 Covering of 10 1-utilized patterns, 16 2-utilized patterns and 2 4-utilized patterns for a TCMG

To formulate the cell-aware utilization of the available MBFFs in a given cell library as a 0-1 ILP formulation, the k-utilized pattern, $\{v_{r(1)},\ldots, v_{r(k)}\}$, $r(i)<r(i+1)$, $1 \le i \le k-1$, inside a given TCMG can be associated with a binary variable, $x_{r(1),\ldots,r(k)}$. If $x_{r(1),\ldots,r(k)}$ =1 in the 0-1 ILP solution, the corresponding k-utilized pattern, $\{v_{r(1)},\ldots, v_{r(k)}\}$, inside a given TCMG will be assigned for cell-aware utilization of a k-bit flip-flop. In contrast, if $x_{r(1),\ldots,r(k)}$=0 in the 0-1 ILP solution, the corresponding the corresponding k-utilized pattern, $\{v_{r(1)},\ldots, v_{r(k)}\}$, inside a given TCMG will be not assigned for cell-aware utilization of a k-bit flip-flop.

For the cell-aware utilization of the available MBFFs, 2-bit FF, 4-bit FF,..., *u*-bit FF in a given cell library, the total clock power, P_{total}, of the cell-aware utilization for a given TMG can be computed as

$$P_{total} = (\sum_{i=1}^{n} x_i)P_1 + (\sum_{Pattern\{v_i,v_j\}} \sum x_{i,j})P_2 + \ldots + (\sum_{Pattern\{v_{r(1)},\ldots v_{r(u)}\}} \sum\sum x_{r(1),\ldots r(u)})P_u \quad (1)$$

and treated as the objective function in the 0-1 ILP formulation. Based on the power consumption, P_k, of the utilization of the corresponding flip-flops in a given library, the proposed 0-1 ILP formulation can be formally defined as follows:

Minimize $(\sum_{i=1}^{n} x_i)P_1 + (\sum_{Pattern\ \{v_i,v_j\}} \sum x_{i,j})P_2 + (\sum_{Pattern\ \{v_i,v_j,v_{pi},v_q\}} \sum\sum x_{i,j,p,q})P_4$,

subject to the unique-covering constraint on any vertex in a given TCMG, $x_{r(1),\ldots,r(k)} \in \{0, 1\}$.

Refer to the covering of the 1-utilized prime-patterns, the 2-utilized patterns and the 4-utilized patterns for a TCMG in Fig. 5, the 0-1 ILP formulation of the cell-aware MBFF utilization for a given TMG with 10 vertices can be defined as follows:

Minimize

$$(\sum_{i=1}^{10} x_i)P_1 + (x_{1,2} + x_{1,4} + x_{2,3} + x_{3,6} + x_{3,7} + x_{3,8} + x_{3,9} + x_{4,5} + x_{4,6}$$
$$+ x_{5,6} + x_{6,7} + x_{6,8} + x_{7,8} + x_{7,9} + x_{7,10} + x_{8,9})P_2 + (x_{3,6,7,8} + x_{3,7,8,9})P_4$$

subject to $x_1 + x_{1,2} + x_{1,4} = 1$,
$x_2 + x_{1,2} + x_{2,3} = 1$,
$x_3 + x_{2,3} + x_{3,6} + x_{3,7} + x_{3,8} + x_{3,9} + x_{3,6,7,8} + x_{3,7,8,9} = 1$,
$x_4 + x_{1,4} + x_{4,5} + x_{4,6} = 1$,
$x_5 + x_{4,5} + x_{5,6} = 1$,
$x_6 + x_{3,6} + x_{4,6} + x_{5,6} + x_{6,7} + x_{6,8} + x_{3,6,7,8} = 1$,
$x_7 + x_{3,7} + x_{6,7} + x_{7,8} + x_{7,9} + x_{7,10} + x_{3,6,7,8} + x_{3,7,8,9} = 1$,
$x_8 + x_{6,8} + x_{7,8} + x_{8,9} + x_{3,6,7,8} + x_{3,7,8,9} = 1$,
$x_9 + x_{3,9} + x_{7,9} + x_{8,9} + x_{3,7,8,9} = 1$, and
$x_{10} + x_{7,10} = 1$.

IV. Experimental Results

For the cell-aware utilization of the available flip-flops in a cell library, our proposed ILP-based approach has been implemented by using standard C++ language and the open source linear solver, lp_solve[10] and run on an Intel Core i7-3770 CPU 3.40GHz machine with 8GB memory. Five tested examples, Ex01, Ex02, Ex03, Ex04 and Ex05, are from [2] with strict timing constraints of input and output signals. The parameters in 0.18μm process from NTRS'97 are used to compute the timing delay of the input and output signals for all the flip-flops. It is assumed that there are three kinds of MBFFs, 1FF, 2FF and 4FF, in a given library. From the information in Faraday library[9], the values of the normalized AC power per bit for 1FF, 2FF and 4FF are set as 1, 0.86 and 0.78, respectively. The experimental results in Yan's approach[9] and our proposed ILP-based approach are shown in Table I. Compared with the original design, the experimental results show that our proposed ILP-based approach can reduce 20.05% of the clock power for five tested examples on the average. Compared with Yan's approach[9], the experimental results show our proposed approach can use less CPU time to further reduce 1.74% of the clock power for five tested examples on the average.

V. Conclusions

Based on the elimination feature of redundant inverters on the utilization of the available MBFFs in a cell library, an ILP-based approach is proposed to reduce total clock power consumption on flip-flops in a synchronous design.

References

[1] J. T. Yan and Z. W. Chen, "Construction of constrained multi-bit flip-flops for clock power reduction," *International Conference on Green Circuits and Systems*, pp.675–678, 2010.

[2] Z. W. Chen and J. T. Yan, "Routability-driven flip-flop merging process for clock power reduction," *International Conference on Computer Design*, pp.203–208, 2010.

[3] Y. T. Chang, C. C. Hsu, P. H. Lin, Y. W. Tsai and S. F. Chen, "Post-placement power optimization with multi-bit flip-flops," *International Conference on Computer-Aided Design*, pp.218–223, 2010.

[4] P. H. Lin, C. C. Hsu and Y. T. Chang, "Post-placement power optimization with multi-bit flip-flops," *IEEE transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no.12, pp.1870–1882, 2011.

[5] S. H. Wang, Y. Y. Liang, T. Y. Kuo and W. K. Mak, "Power-driven flip-flop merging and relocation," *IEEE transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no.2, pp.180–191, 2012.

[6] I. H. R. Jiang, C. L. Chang and Y. M. Yang, "INTEGRA: Fast multi-bit flip-flop clustering for clock power saving," *IEEE transaction on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no.2, pp.192–204, 2012.

[7] Y. T. Shyu, J. M. Lin, C. P. Hung, C. W. Lin, Y. Z. Lin and S. J. Chang, "Effective and efficient approach for power reduction by using multi-bit flip-flops," *IEEE Transaction on Very Large Scale Integration Systems*, vol. 21, no. 4, pp.624–635, 2013.

[8] 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 Transaction on Design Automation of Electronic Systems*, vol. 18, no. 3, Article 40, 2013.

[9] Z.W. Chen and J. T. Yan, "Library-aware utilization of multi-bit flip-flops for clock power reduction," *International Conference on Electronics, Circuits and Systems*, pp.677-680, 2012.

[10] lp_solve: an open source linear programming solver. [Online]. Available: http://sourceforge.net/projects/lpsolve

Table I Experimental Results of Cell-Aware MBFF Utilization for Clock Power Reduction

Examples	#FFs	Original Power	Yan's Approach[9]			Our ILP-based Approach		
			#1FF/#2FF/#4FF	Final Power	CPU Time(s)	#1FF/#2FF/#4FF	Final Power	CPU Time(s)
Ex01	120	120(100%)	12/22/16	99.76(83.13%)	0.46	6/9/24	96.36(80.30%)	0.23
Ex02	120	120(100%)	10/13/21	97.88(81.57%)	0.42	2/3/28	94.52(78.77%)	0.29
Ex03	800	800(100%)	28/96/145	645.52(80.69%)	26.73	12/72/161	638.16(79.77%)	6.74
Ex04	900	900(100%)	36/152/140	734.24(81.58%)	38.68	24/108/165	724.56(80.51%)	7.21
Ex05	1000	1000(100%)	50/147/164	814.52(81.45%)	47.62	34/103/190	803.96(80.40%)	8.26