FF-Bond: Multi-bit Flip-flop Bonding at Placement

Chang-Cheng Tsai[1], Yiyu Shi[2], Guojie Luo[3], and Iris Hui-Ru Jiang[1]
[1]Dept. of Electronics Engineering and Inst. of Electronics, National Chiao Tung University, Hsinchu 30010, Taiwan
[2]Dept. of ECE, Missouri University of Science and Technology, Rolla, MO 65409, US
[3]Center for Energy-Efficient Computing and Applications (CECA), Peking University, Beijing 100871, P.R. China
shrimpcct@gmail.com; yshi@mst.edu; gluo@pku.edu.cn; huiru.jiang@gmail.com

ABSTRACT
Clock power contributes a significant portion of chip power in modern IC design. Applying multi-bit flip-flops can effectively reduce clock power. State-of-the-art work performs multi-bit flip-flop clustering at the post-placement stage. However, the solution quality may be limited because the combinational gates are immovable during the clustering process. To overcome the deficiency, in this paper, we propose multi-bit flip-flop bonding at placement. Inspired by ionic bonding in Chemistry, we direct flip-flops to merging friendly locations thus facilitating flip-flop merging. Experimental results show that our algorithm, called FF-Bond, can save 27% clock power on average. Compared with state-of-the-art post-placement multi-bit flip-flop clustering, FF-Bond can further reduce 14% clock power.

Categories and Subject Descriptors
B.7.2 [**INTEGRATED CIRCUITS**]: Design Aids – *placement and routing*

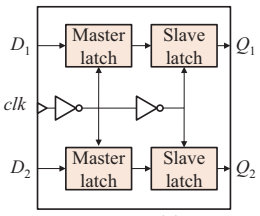
General Terms
Algorithms, Performance, Design.

Keywords
Multi-bit flip-flops, placement, clock power, timing.

1. INTRODUCTION
Clock power has become the main source of chip power in modern IC design [1]. As revealed by [2][3][4], relocating flip-flops benefits clock network synthesis. As shown in Figure 1, compared with single-bit flip-flops, multi-bit flip-flops (MBFFs) present a smaller load on the clock network due to the shared clock logic in the cell [5]. Thus, replacing flip-flops with MBFFs can effectively reduce both the clock network power and the MBFF power consumption. However, the signal wirelength may somewhat increase which may not be acceptable or lead to an increase of power consumption on timing critical paths. Thus, use of MBFFs requires ensuring sufficient timing slacks to avoid impacting timing critical paths.

Due to the lack of physical information before the placement stage, state-of-the-art work handles MBFF clustering at the post-placement stage, e.g., [6][7][8][9][10]. In order not to sacrifice timing, most of these works model the movable regions of flip-flops by an intersection graph. A clique of a proper size in the intersection graph corresponds to an MBFF. Yan and Chen form MBFFs from largest maximal cliques in [6]. Chang *et al.* present a progressive window-based clustering method in [7]. Wang *et al.* allocate MBFFs extracted from a randomly sampled subset of maximal cliques in [8]. Jiang *et al.* encode the intersection graph by interval graphs to identify mergeable flip-flops in [9]. Liu *et al.* propose a bottom-up merging method in [10]. Among these works, [9] delivers the most power efficient result.

ISPD'13, March 24–27, 2013, Stateline, Nevada, USA.



(a)

Bit number	Normalized power per bit	Normalized area per bit
1	1.00	1.00
2	0.86	0.96
4	0.78	0.71

(b)

Figure 1. Multi-bit flip-flop. (a) A dual-bit flip-flop, where the inverter chain is shared. (b) Power and area of the MBFF library.

However, the combinational gates are immovable during the post-placement MBFF clustering scheme. The clustering flexibility and quality are thus limited. To break this limitation, in this paper, we perform MBFF bonding at placement.

A possible solution is to directly integrate placement and post-placement MBFF clustering together. These two tasks are sequentially applied at each iteration. Nevertheless, if doing so, the movement of flip-flops is constrained by the placement at the current iteration and may oscillate among iterations.

In contrast, inspired by ionic bonding in Chemistry [11], we guide flip-flops to move towards merging friendly locations at the global placement stage without sacrificing timing. An ionic bond is formed when the atom of an element releases some of its electron(s) and the atom of another element then captures the electron(s) to attain a stable electron configuration. (see Figure 2(a)) We devise a flip-flop bonding scheme so that flip-flops are

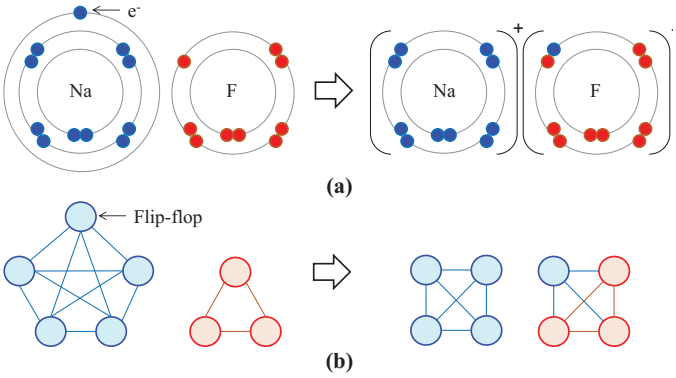
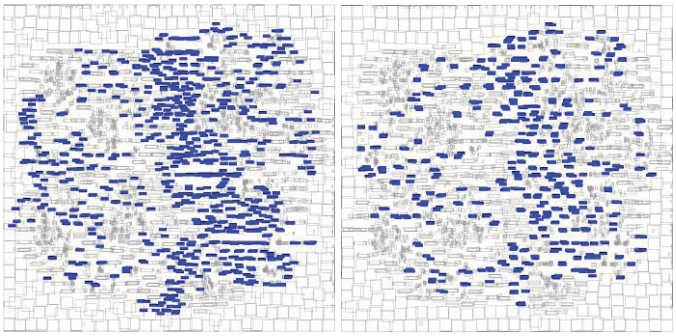


Figure 2. (a) Ionic bonding: Na + F→ NaF. (b) Flip-flop bonding.



(a) (b)

Figure 3. The snapshot right after timing-driven global placement of s38417. (The solid squares indicate single-bit flip-flops.) (a) Without flip-flop bonding. (Number of resulting 4-/2-/1-bit flip-flops: 35/252/237.) (b) With flip-flop bonding. (Number of resulting 4-/2-/1-bit flip-flops: 159/105/35.)

moved to merging friendly locations. For example, according to the MBFF library given in Figure 1(b), a four-bit flip-flop is most power efficient. Thus, a clique of size 4 in the intersection graph is considered perfect. Via flip-flop bonding, we release flip-flops from an oversized clique (larger than 4) to an undersized clique (less than 4). (see Figure 2(b))

In this paper, we propose an MBFF bonding at placement algorithm, called FF-Bond. To demonstrate our flow, we develop a net-based timing-driven placer [12]. The wirelength-driven placement kernel is based on an analytical placement method proposed in [13]. Rather than incorporating an approximate delay model into the placer, we tune the net weights by the timing slacks computed by a signoff timing engine for more accurate timing information. By introducing a flip-flop bonding force, we guide each flip-flop to a merging friendly location. (see Figure 3) Consequently, after timing-driven global placement with flip-flop bonding, flip-flops can easily be merged together thus reducing power. Legalization and detailed placement are then performed to remove overlap and incrementally refine the placement result.

Experimental results show that FF-Bond can save 27% clock power on average. Compared with state-of-the-art post-placement MBFF clustering, FF-Bond can further reduce 14% clock power. The remainder of this paper is organized as follows. Section 2 introduces post-placement MBFF clustering and gives the problem formulation. Section 3 details our MBFF bonding at placement algorithm, FF-Bond. Section 4 lists experimental results. Finally, Section 5 gives a conclusion.

2. PRELIMINARIES

In this section, we introduce post-placement MBFF clustering and give the problem formulation.

2.1 Post-Placement MBFF Clustering

The post-placement MBFF clustering problem is that given a placed design, an MBFF library, timing slacks, and placement density constraints, replace flip-flops with MBFFs such that the power is minimized and the timing and placement density constraints are satisfied.

As mentioned in Section 1, INTEGRA proposed in [9] delivers the most power efficient result among prior works. We take INTEGRA as an example to demonstrate post-placement MBFF clustering.

First of all, timing analysis reports the timing slacks of the fanin/fanout pin of each flip-flop. Based on a delay-wirelength conversion, the movable region of each flip-flop without hurting timing is obtained. As shown in Figure 4(a), the fanin and fanout slacks are converted to diamonds. The overlap region of these diamonds is the feasible region of a flip-flop. Figure 4(b) illustrates the extracted feasible regions of flip-flops for a sample design. The corresponding intersection graph is constructed as shown in Figure 4(c). If the feasible regions of several flip-flops overlap (i.e., a clique in the intersection graph), these flip-flops can form an MBFF.

As shown in Figure 4(d), INTEGRA applies coordinate transformation and encodes the intersection graph by two interval graphs. Two sequences are used to record the starting (type s) and ending (type e) *x'*-/*y'*-coordinates of feasible regions in ascending order. It is shown that all maximal cliques can be extracted at decision points (the 'se' patterns in the sorted *x'*-sequence). {1, 2, 4} and {1, 3, 4} are found at the first decision point; {3, 4, 5, 6} is found at the second one; {4, 5, 6, 7, 8} is found at the third one. INTEGRA scans the *x'*-sequence and generates MBFFs at decision points. The clustering result is shown in Figure 4(e).

2.2 Problem Formulation

As mentioned in Section 1, the flexibility and solution quality of post-placement MBFF clustering are limited. To overcome the deficiency, in this paper, we perform MBFF bonding at placement.

The problem formulation is described as follows.

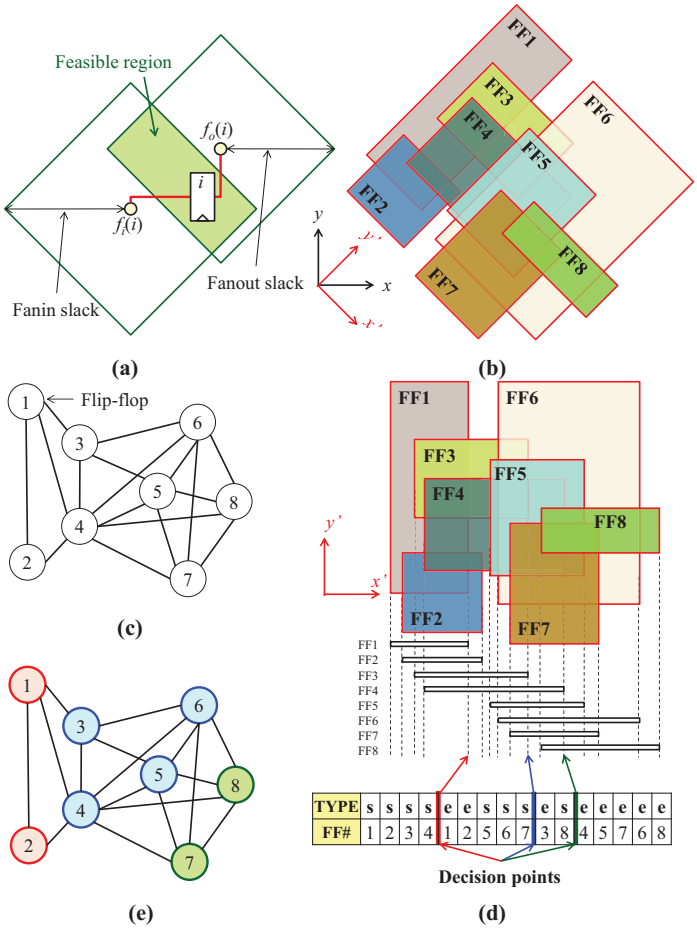


Figure 4. Post-placement MBFF clustering. (a) Feasible region. (b) Feasible region extraction. (c) Intersection graph. (d) INTEGRA. (e) MBFF clustering result: {1, 2}, {3, 4, 5, 6}, {7, 8}.

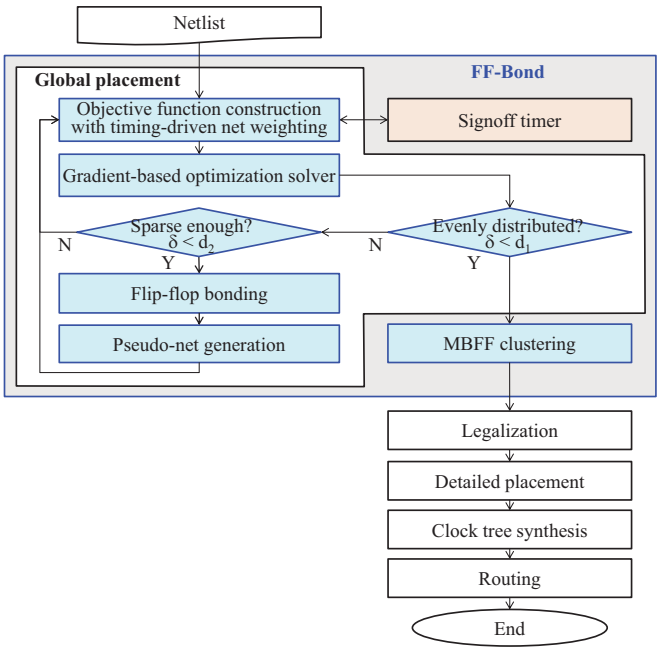


Figure 5. The overview of FF-Bond.

The MBFF Bonding at Placement Problem: Given a netlist, an MBFF library, timing constraints, and placement density constraints, find a placement and replace flip-flops with MBFFs such that the power is minimized and the timing and placement density constraints are satisfied.

3. OUR ALGORITHM—FF-BOND

In this section, we propose the MBFF bonding at placement algorithm, FF-Bond.

3.1 Overview

It can be seen that if flip-flop 2 in Figure 4 moves towards flip-flop 3, their feasible regions may overlap, and the clustering result will be improved. (Two four-bit flip-flops {1, 2, 3, 4} and {5, 6, 7, 8} can be formed.) Therefore, we propose FF-Bond to guide flip-flops towards merging friendly locations at the global placement stage without sacrificing timing.

Figure 5 shows the overview of FF-Bond. To demonstrate our flow, we develop a net-based timing-driven placer [12]. The wirelength-driven placement kernel is based on an analytical placement method, mPL5, proposed in [13]. Instead of using an approximate delay model in the placer, the net weights are adjusted according to the timing slacks computed by a signoff timing engine for more accurate timing information. By introducing a flip-flop bonding force, we guide each flip-flop to a merging friendly location. After timing-driven global placement with flip-flop bonding, flip-flops are merged. Legalization and detailed placement are then applied to remove overlaps and refine the placement. Clock network synthesis and routing are finally performed.

3.2 Timing-Driven Placement

To demonstrate our flow, we develop a timing-driven placer based on a pure wirelength-driven placement kernel mPL5 [13] and the slack-based net-weighting technique [14]. Pure wirelength-driven global placement is applied only for the first iteration, while timing-driven global placement is applied for the subsequent iterations. We shall introduce these two techniques in this subsection.

3.2.1 Wirelength-driven placement kernel

The pure wirelength-driven placement kernel is based on an analytical placement method, mPL5, proposed in [13]. A netlist is modeled by a hypergraph $H=(V, E)$, where V denotes the set of cells and hyperedges in E represent nets. (x_i, y_i) represents x-/y-coordinates of cell i. First of all, the placement region is divided into $m \times n$ non-overlapping uniform bins. The following constrained minimization problem is considered.

$$\min W(x,y)$$
$$\text{s.t. } D_{ij} = K, 1 \le i \le m, 1 \le j \le n, \quad (1)$$

where $W(x, y)$ is the wirelength function defined by half-perimeter wirelength (HPWL), D_{ij} means the average density of bin B_{ij}, and K is the target density computed by the total cell area divided by the area of the placement region. The objective function and constraints are not differentiable. The wirelength function is smoothed by log-sum-exp approximation [15].

$$\widehat{W}(x,y) = \eta \sum_{e \in E} \left(\log \sum_{v_k \in e} \exp\left(\frac{x_k}{\eta}\right) + \log \sum_{v_k \in e} \exp\left(\frac{-x_k}{\eta}\right) + \log \sum_{v_k \in e} \exp\left(\frac{y_k}{\eta}\right) + \log \sum_{v_k \in e} \exp\left(\frac{-y_k}{\eta}\right) \right). \quad (2)$$

Furthermore, the inverse Laplace transformation is applied to smooth the density function.

$$\min \widehat{W}(x,y)$$
$$\text{s.t. } \psi_{ij} = \overline{K}, 1 \le i \le m, 1 \le j \le n. \quad (3)$$

Via Lagrange multipliers, the density constraint is converted to a penalty into the objective function.

$$L(x,y,\lambda) = \widehat{W}(x,y) + \sum_{i,j} \lambda_{ij} \left(\psi_{ij} - \overline{K}\right). \quad (4)$$

A gradient-based optimization solver is then applied to solve the nonlinear program.

3.2.2 Slack-based net weighting

We adopt slack-based net weighting since this timing-driven placement approach has low computational complexity and high flexibility [12]. To reflect timing criticalities, we adjust net weights at each iteration according to the timing slacks. Instead of incorporating an approximate delay model into the placer, we rely on a signoff timing engine. We assign negative slack nets with larger net weights than positive slack nets. Thus, the placement kernel tends to shorten the negative slack nets to resolve timing violations. The net weight at an iteration is defined as follows [14].

$$net\ weight = \left(1 - \frac{slack}{T_{clk}}\right)^{\alpha}, \quad (5)$$

where T_{clk} is the clock period for a particular net, and $\alpha>1$ is the criticality exponent to emphasize critical nets. At the first iteration, slack is set to 0 for pure wirelength-driven global placement.

3.3 Flip-flop Bonding

In this subsection, we shall detail the flip-flop bonding mechanism to guide flip-flops towards merging friendly locations.

Consider the possible at-placement MBFF merging method mentioned in Section 1, where placement and post-placement MBFF clustering are directly integrated together. If doing so, the movement of flip-flops is guided by the post-placement MBFF clustering result according to the current placement. This guidance does not encourage orphan flip-flops to merge with others and may oscillate among iterations.

In contrast, we devise a flip-flop bonding mechanism inspired by ionic bonding in Chemistry [11]. For example, consider two maximal cliques of size 5 and 3. Based on the MBFF library given in Figure 1(b), post-placement MBFF clustering may generate one four-bit flip-flop, one dual-bit flip-flop, and two single-bit flip-flops (orphans). As illustrated in Figure 2(b), if one flip-flop in the maximal clique of size 5 is attracted to the maximal clique of size 3, we may have two four-bit flip-flops instead. Hence, by introducing a flip-flop bonding force, we direct each flip-flop towards a merging friendly location, thus forming more larger-bit flip-flops.

Given an MBFF library, the bit number of the most power efficient cell is considered as the perfect clique size. (The most power efficient flip-flop cell has the lowest normalized power per bit.) Hence, all extracted maximal cliques are classified into oversized, perfect, and undersized cliques accordingly. (e.g., Figure 6) An oversized clique can form at least one perfect-sized clique and possibly leave several single-bit flip-flops (that we try to avoid). A perfect-sized clique is desired. An undersized clique is to attract flip-flops to form a perfect-sized clique.

Flip-flop bonding tries to bond flip-flops into perfect-sized cliques. The priority of processing maximal cliques is in the following order: perfect, undersize, oversize. Perfect-sized cliques are preserved first. An investigated undersized/oversized clique selects the most adjacent flip-flops in a specified search region to form a target-sized clique. (The search region and adjacency are defined later.) Undersized cliques are handled in descending order of clique size. The target size of an undersized clique means the bit number defined in the MBFF library that is larger than and nearest to the investigated clique size. Similarly, the target size of an oversized clique is the flip-flop configuration that is larger than, nearest to, and more power efficient than the investigated clique size. For example, considering the MBFF library given in Figure 1, 2 is the target size for 1; 4 is for 2 and 3; 6 is for 5; 8 is for 6 and 7. Please note that our flip-flop bonding mechanism is general, not limited to the MBFF library given in Figure 1(b).

Figure 7 demonstrates a flip-flop bonding example. As shown in Figure 7(b), first of all, all maximal cliques are extracted based on the method presented in [9]. Figure 7(c) shows the clusters based on our flip-flop bonding strategy, where the processing order is indicated by the number beside each cluster. After the flip-flop bonding force is applied (see Section 3.4), flip-flops in each cluster are moved to each other, thus facilitating MBFF merging.

In some cases, maximal cliques may overlap. Basically, we apply the same flip-flop bonding strategy. For cliques of the same size, the clique with most independent flip-flops is processed first. An independent flip-flop means a flip-flop exists in exactly one maximal clique. Figure 7(e) shows an example with overlapping maximal cliques, while Figure 7(f) shows the resulting bonding clusters. The processing order of cliques of size 3 is indicated by the number beside each cluster. After the first two cliques of size 3 are processed, the third one has no independent flip-flops, and thus this clique is skipped.

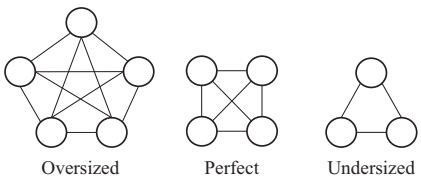


Figure 6. Clique sizes. For the MBFF library given in Figure 1, oversize means clique size > 4; perfect size means clique size = 4; undersize means clique size < 4.

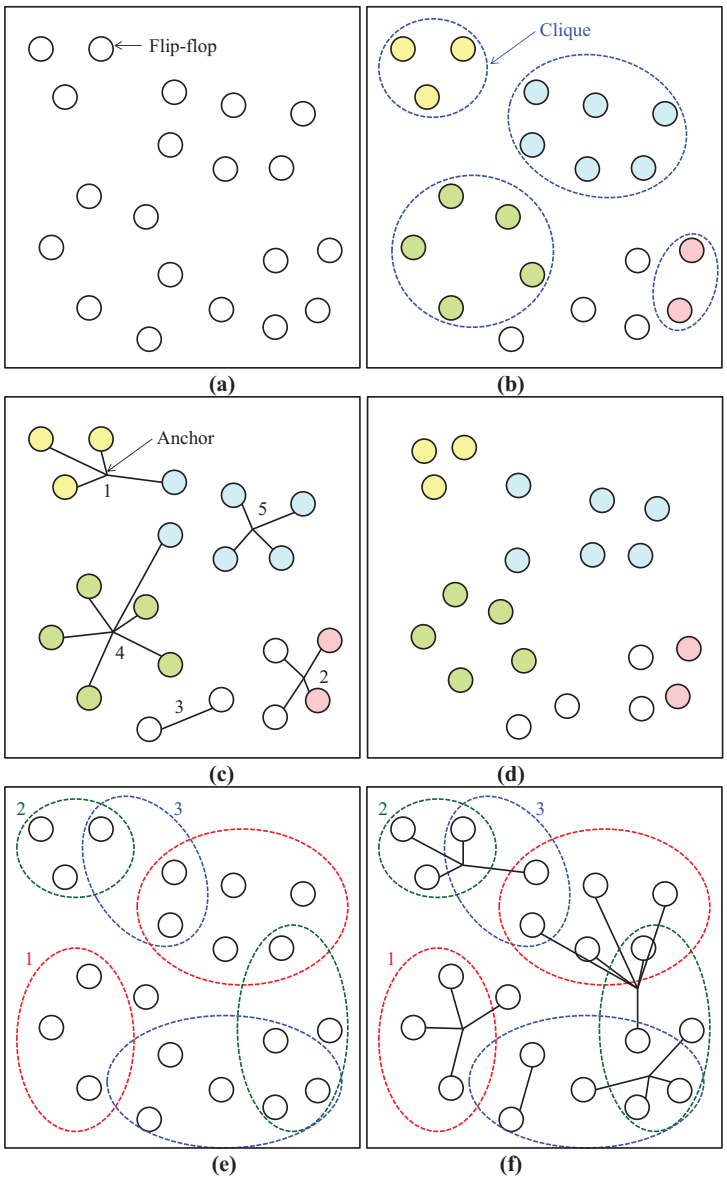


Figure 7. Flip-flop bonding. (a) Flip-flops before flip-flop bonding. (b) Maximal cliques. (c) Flip-flop bonding clusters. (d) Flip-flops after flip-flop bonding. (e) Maximal cliques overlap. (f) Flip-flops after flip-flop bonding.

For the flip-flop bonding strategy, we define a search region and adjacency. The search region prevents flip-flops from attracting distant flip-flops. The adjacency reflects the physical distance and timing information. Let (x_C, y_C) denote the average x-/y-coordinates of clique C and (x_i, y_i) denote the x-/y-coordinates of flip-flop i. Assume that the fanin and fanout slack of flip-flop i is $s_{fi}(i)$ and $s_{fo}(i)$, respectively. The adjacency between clique C and flip-flop i is defined as follows.

$$A(C,i) = |x_C - x_i| + |y_C - y_i| - \varepsilon_i \left(s_{fi}(i) + s_{fo}(i)\right),$$
$$\varepsilon_i = \begin{cases} -\infty, & \text{if } s_{fi}(i) < 0 \text{ or } s_{fo}(i) < 0 \\ \tau, & \text{otherwise} \end{cases}. \quad (6)$$

τ is the delay-wirelength conversion parameter used in Section 2.1.

3.4 Pseudo-Net Generation

After flip-flop bonding, we introduce a flip-flop bonding anchor for each flip-flop cluster. Each flip-flop within a cluster is linked

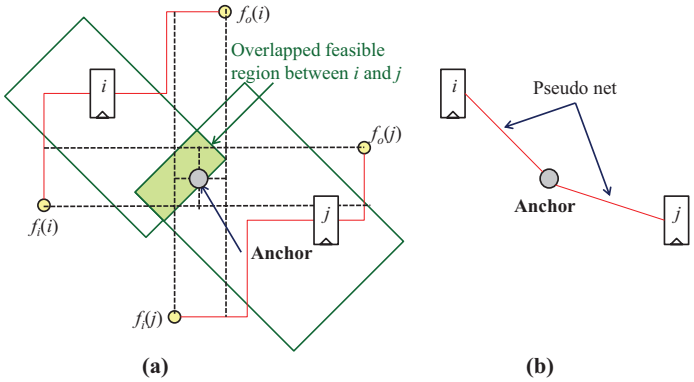


Figure 8. Pseudo net. (a) Flip-flop bonding anchor. (b) Pseudo net.

to the anchor using a two-pin pseudo net that is assigned a high net weight. Within FF-Bond, these two-pin nets are viewed as flip-flop bonding attractions. To emphasize the attractions, their weights should be greater than the default net weight for a positive slack net, say 5X in our experiments.

The anchor is set to the desired location of the potentially formed MBFF. The post-placement MBFF clustering methods place MBFFs within feasible regions due to timing constraints. However, at global placement, cells can flexibly be moved and timing is still maintained. Therefore, the anchor is set to the (center of) median of all fanin/fanout x-/y-coordinates (for signal wirelength consideration). Figure 8 shows a flip-flop cluster with two flip-flops. Figure 8(a) illustrates the anchor introduced, while Figure 8(b) shows the generated pseudo nets.

3.5 The Condition to Apply Flip-flop Bonding

At the early iterations of global placement, cells strongly overlap. While the gradient-based optimization solver computes Lagrange multipliers, cells are gradually moved towards optimal locations. During this process, the overlap among cells is iteratively reduced. Finally, when the amount of overlap is small enough, the optimizer stops. Because the placement at early iterations is quite different from the final result, flip-flop bonding is applied when cells are sparse enough.

We use an overlap index to control the global placement flow. The overlap index δ is defined by the total overlap cell area divided by the total cell area. During the global placement optimization, cells are gradually spread out, and thus δ decreases iteration by iteration. When $\delta < d_2$, flip-flop bonding is applied. d_2 is a user-specified parameter. The greater d_2, the earlier iteration flip-flop bonding is applied (the potentially larger flexibility to merge flip-flops). Later, in our experiments, d_2 is set to 0.5.

The timing-driven incremental placement and flip-flop bonding are repeated until cells are evenly distributed, $\delta < d_1$. d_1 is a user-specified parameter. Usually, d_1 is very small such that the overlap of cells is small and the density constraint is satisfied. Later, in our experiments, d_1 is set to 0.1.

4. EXPERIMENTAL RESULTS

FF-Bond was implemented in the C++ programming language on a Linux workstation with an Intel Xeon 2.4GHz CPU and 16GB memory. Experiments are conducted on the circuits from IWLS 2005 benchmark [16]. The MBFF library is designed by [17]. (see Figure 1) These circuits are synthesized and legalized by state-of-the-art commercial tools [18][19] based on UMC 55nm technology. The signoff timing engine is [20]. To test the effectiveness, FF-Bond is compared with two representative flows: Post-placement MBFF clustering (PMC) performs timing-driven global placement followed by post-placement MBFF clustering. Interleaving placement and post-placement MBFF clustering (IMC) interleaves timing-driven global placement and post-placement MBFF clustering. The post-placement MBFF clustering method used in our experiments is based on [9] because of its superior power efficiency. The parameters in FF-Bond used in our experiments are set as follows: $d_1 = 0.1$, $d_2 = 0.5$, $\alpha = 1.2$, the search region is bounded by 20% of chip dimension, and the net weight of a pseudo net is 5X the default value for a positive slack net.

Table 1 compares the three flows on flip-flop power and generated MBFFs. FF-Bond obtains the best power efficiency among the three flows. 'FF power ratio' means the total power of generated flip-flops divided by the power of using only single-bit flip-flops. Without timing consideration, the lower bound of FF power ratio is 0.78. (All are four-bit flip-flops.) The FF power ratio of FF-Bond is very close to the lower bound. Moreover, the main constituents of formed flip-flops of FF-Bond are four-bit flip-flops (compared with single-bit flip-flops for PMC, and dual-bit flip-flops for IMC). These results show that flip-flop bonding indeed effectively guides flip-flops to merging friendly locations. Although FF-Bond results in longer wirelength, the increased wirelength induces less than 1% chip power increase in our experiments (because data signals do not always toggle in every cycle). Hence, the tradeoff between signal power and clock power is good. Figure 9 shows the global placement and MBFF merging results of s38417 of the three flows.

Table 2 compares the three flows on clock power (including clock network and MBFFs). The clock network is synthesized based on [21]. FF-Bond obtains the lowest clock power among the three flows. The fewer clock sinks, the simpler clock network. Hence, FF-Bond saves 19% flip-flop power and saves even more in terms of clock power. For s38584, IMC consumes slightly lower clock power than FF-Bond because of smaller clock buffers used, but FF-Bond still achieves fewer clock sinks. FF-Bond can totally save 27% clock power on average. Compared with post-placement MBFF clustering, FF-Bond can further reduce 14% clock power.

Table 3 compares PMC and FF-Bond on timing slacks. The timing slack of each endpoint means the worst slack over all paths ending at this endpoint. 'Worst slack' represents the worst timing slack over all endpoints, while 'Average slack' means the average. Because we consider timing during FF-Bond, the slacks of the two flows are quite similar.

Table 4 shows the impact of d_2 on the MBFF bonding results of FF-Bond. For a smaller d_2, flip-flop bonding starts at later iterations but does not guide flip-flops well because the low flexibility of moving flip-flops. In contrast, for a larger d_2, flip-flop bonding starts from earlier iterations but results in longer wirelength because distant flip-flops are attracted.

Table 5 and Figure 10 show the impact of the search region on flip-flop power, slack, and wirelength. For s38417, when the search region is set to 0.2X chip dimension, the balance between flip-flop power, slack, and wirelength is good.

Tables 1–5 show that FF-Bond is promising to merge flip-flops.

5. CONCLUSION

Applying MBFFs can effectively reduce clock power. Unlike state-of-the-art work performed MBFF clustering at the post-

placement stage, in this paper, we did MBFF bonding at placement. Inspired by ionic bonding in Chemistry, we directed flip-flops to merging friendly locations. Experimental results showed that FF-Bond can save 27% clock power on average. Compared with post-placement MBFF clustering, FF-Bond can further reduce 14% clock power. Future work includes MBFF bonding with routability consideration.

6. REFERENCES

[1] L.-T. Wang, Y.-W. Chang, and K.-T. Cheng (editors). *Electronic Design Automation: Synthesis, Verification, and Test*, Elsevier/Morgan Kaufmann, 2009.

[2] Y.-S. Cheon, P.-H. Ho, A. B. Kahng, S. Reda and Q. Wang. Power-aware placement. In *Proc. Design Automation Conf. (DAC)*, pp. 795–800, 2005.

[3] D. Papa, N. Viswanathan, C. Sze, Z. Li, G.-J. Nam, C. Alpert, I.L. Markov. Physical synthesis with clock-network optimization for large systems on chips. *IEEE Micro*, vol. 31, no. 4, Apr. 2011, pp. 51–62.

[4] D.-J. Lee and I. L. Markov. Obstacle-aware clock-tree shaping during placement. *IEEE Trans. Computer-Aided Design (TCAD)*, vol. 31, no. 2, Feb 2012, pp. 205–216.

[5] 2010 CAD contest of Taiwan, http://cad contest.ee.ntu.edu.tw/cad10/Problems/B1_Faraday_091223_ MultiBitFF.pdf.

[6] J.-T. Yan and Z.-W. Chen. Construction of constrained multibit flipflops for clock power reduction. In *Proc. Int. Conf. Green Circuits Syst. (ICGCS)*, pp. 675–678, 2010.

[7] 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. Int. Conf. on Computer-Aided Design (ICCAD)*, pp. 218–223, 2010.

[8] S.-H. Wang, Y.-Y. Liang, T.-Y. Kuo, and W.-K. Mak. Power-driven flip-flop merging and relocation. In *Proc. Int. Symp. on Physical Design (ISPD)*, pp. 107–114, 2011.

[9] Iris H.-R. Jiang, C.-L. Chang, Y.-M. Yang. INTEGRA: Fast multibit flip-flop clustering for clock power saving. *IEEE Trans. Computer-Aided Design (TCAD)*, vol. 31, no. 2, Feb 2012, pp. 192–204. Also see *Proc. Int. Symp. on Physical Design (ISPD)*, pp. 115–121, 2011.

[10] S. S.-Y. Liu, C.-J. Lee and H.-M. Chen. Agglomerative based flip-flop merging for power optimization. In *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1391–1396, 2012.

[11] Ionic and covalent bonds. http://chemwiki.ucdavis.edu/.

[12] D. Z. Pan, B. Halpin, and H. Ren. Timing driven placement. Handbook of Algorithms for VLSI Physical Automation, CRC Press, 2007.

[13] T. Chen, J. Cong, and K. Sze. Multilevel generalized force-directed method for circuit placement. In *Proc. Int. Symp. on Physical Design (ISPD)*, pp. 185–192, 2005.

[14] A. Marquardt V. Betz and J. Rose. Timing driven placement for FPGAs. In *Proc. Int. Symp. on Field Programmable Gate Arrays (FPGA)*, pp. 203–213, 2000.

[15] W. C. Naylor, R. Donelly, and L. Sha. Non-linear optimization system and method for wire length and delay optimization for an automatic electric circuit placer. *U.S. Patent* 6301693, Oct. 9, 2001.

[16] IWLS 2005 benchmarks. http://iwls.org/iwls2005/benchmarks.html.

[17] Faraday Technology Corp.

[18] Design Compiler. Synopsys, Inc.

[19] SoC Encounter. Cadence Design Systems, Inc.

[20] Prime Time. Synopsys, Inc.

[21] Y.-C. Chang, C.-K. Wang and H.-M. Chen. On constructing lower power and robust clock tree via slew budgeting. In *Proc. Int. Symp. on Physical Design (ISPD)*, pp. 129–136, 2012.

Table 3. Slack Comparison.

		PMC		FF-Bond	
Circuit Name	**Clock period (ns)**	**Worst slack (ns)**	**Average slack (ns)**	**Worst slack (ns)**	**Average Slack (ns)**
s13207	1.5	0.042	0.580	0.041	0.579
s15850	1.8	0.158	0.336	0.154	0.334
s38417	2.0	0.164	1.049	0.163	1.047
s38584	2.0	0.217	0.871	0.209	0.869
b17	3.0	0.122	1.272	0.128	1.269
b19	2.7	0.112	1.278	0.109	1.273

Table 5. FF-Bond: Search Region vs. Flip-flop Power, Wirelength, and Slack. (s38417)

Search region (Unit: chip_width+chip_height)	FF power ratio	HPWL	Worst slack (ns)
0.05	0.817	4.89E+07	0.163
0.08	0.819	4.98E+07	0.163
0.10	0.814	5.07E+07	0.164
0.15	0.812	5.23E+07	0.163
0.18	0.811	5.34E+07	0.164
0.20	0.808	5.41E+07	0.163
0.25	0.807	5.41E+07	0.163
0.28	0.810	5.41E+07	0.163

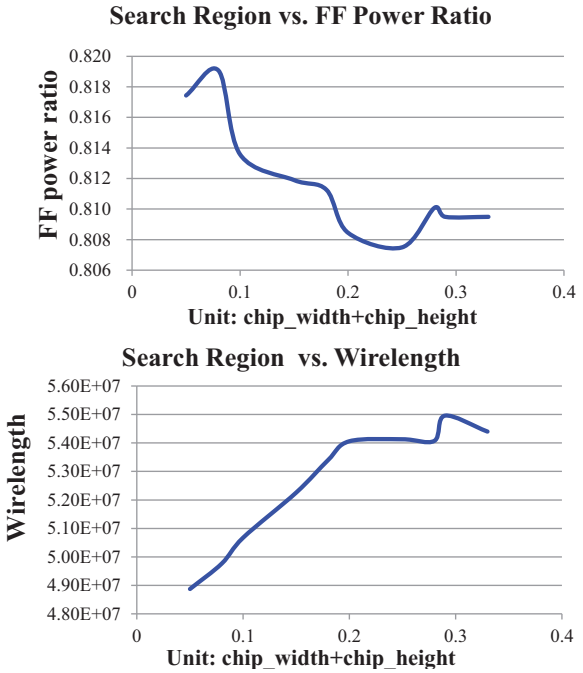


Figure 10. FF-Bond: Search region analysis. (s38417)

Table 1. Flip-flop Power Comparison.

		PMC			IMC			FF-Bond		
Circuit Name	#Flip-flops	#MBFFs 4-/2-/1-bit	FF power ratio	HPWL	#MBFFs 4-/2-/1-bit	FF power ratio	HPWL	#MBFFs 4-/2-/1-bit	FF power ratio	HPWL
s13207	212	8/57/66	0.892	4.569E+06	23/51/18	0.837	4.975E+06	35/31/10	0.814	5.344E+06
s15850	128	10/29/30	0.868	2.117E+06	18/23/10	0.826	2.869E+06	23/15/6	0.809	2.903E+06
s38417	881	35/252/237	0.885	4.599E+07	105/179/103	0.838	4.789E+07	159/105/35	0.808	5.406E+07
s38584	1069	46/291/303	0.886	5.992E+07	96/282/121	0.847	6.213E+07	203/116/25	0.803	6.926E+07
b17	1068	53/264/328	0.887	1.346E+08	137/201/118	0.834	1.363E+08	196/124/36	0.806	1.470E+08
b19	4384	378/886/1100	0.868	7.187E+08	593/742/528	0.834	7.267E+08	851/425/130	0.802	8.023E+08
Avg. ratio	-	0.21/0.91/1.00	0.881	0.85	1.20/2.05/1.00	0.836	0.92	5.33/3.33/1.00	0.807	1.00

Table 2. Clock Power Comparison. (Flip-flops and Clock Networks)

	Without MBFF clustering				PMC				IMC				FF-Bond			
Circuit Name	Total Cap. (pF)	Sinks	Buffer	Wire	Total Cap. (pF)	Sinks	Buffer	Wire	Total Cap. (pF)	Sinks	Buffer	Wire	Total Cap. (pF)	Sinks	Buffer	Wire
s13207	1.333	48.5%	36.4%	15.1%	1.223	46.8%	39.7%	13.5%	1.094	49.6%	38.8%	11.7%	1.056	49.8%	40.2%	10.0%
s15850	0.901	43.3%	47.1%	9.6%	0.837	40.9%	50.6%	8.5%	0.806	39.6%	52.6%	7.8%	0.799	39.5%	53.1%	7.4%
s38417	5.051	53.2%	31.6%	15.2%	4.113	57.9%	26.8%	15.3%	3.884	58.2%	27.4%	14.5%	3.711	58.5%	28.6%	12.9%
s38584	6.100	53.5%	28.9%	17.6%	5.352	54.3%	29.9%	15.8%	4.576	60.6%	24.1%	15.3%	4.870	53.9%	32.8%	13.3%
b17	6.273	51.9%	28.1%	20.0%	5.574	51.8%	28.7%	19.5%	5.241	51.9%	30.5%	17.6%	4.513	58.2%	26.3%	15.5%
b19	25.611	52.2%	26.8%	21.0%	22.081	52.4%	28.1%	19.5%	19.410	57.4%	23.9%	18.7%	18.277	58.7%	24.5%	16.8%
Avg. Ratio	1.00	-	-	-	0.87	-	-	-	0.77	-	-	-	0.73	-	-	-

Table 4. FF-Bond: Flip-Flop Power and Wirelength Comparison under Different d_2.

		d_2=0.3			d_2=0.5			d_2=0.7		
Circuit Name	#Flip-flops	#MBFFs 4-/2-/1-bit	FF power ratio	HPWL	#MBFFs 4-/2-/1-bit	FF power ratio	HPWL	#MBFFs 4-/2-/1-bit	FF power ratio	HPWL
s13207	212	27/41/22	0.834	5.243E+06	35/31/10	0.814	5.344E+06	38/25/10	0.809	5.518E+06
s15850	128	20/20/8	0.819	2.620E+06	23/15/6	0.809	2.903E+06	22/16/8	0.814	2.895E+06
s38417	881	171/85/27	0.802	5.258E+07	159/105/35	0.808	5.406E+07	164/89/47	0.808	5.569E+07
s38584	1069	194/135/23	0.805	6.861E+07	203/116/25	0.803	6.926E+07	192/134/33	0.807	6.944E+07
b17	1068	186/135/23	0.809	1.460E+08	196/124/36	0.806	1.470E+08	202/116/28	0.803	1.485E+08
b19	4384	847/427/142	0.803	7.927E+08	851/425/130	0.802	8.023E+08	851/431/118	0.802	8.037E+08
Avg. ratio	-	4.99/3.44/1.00	0.812	0.97	5.33/3.33/1.00	0.807	1.00	5.04/3.04/1.00	0.807	1.01

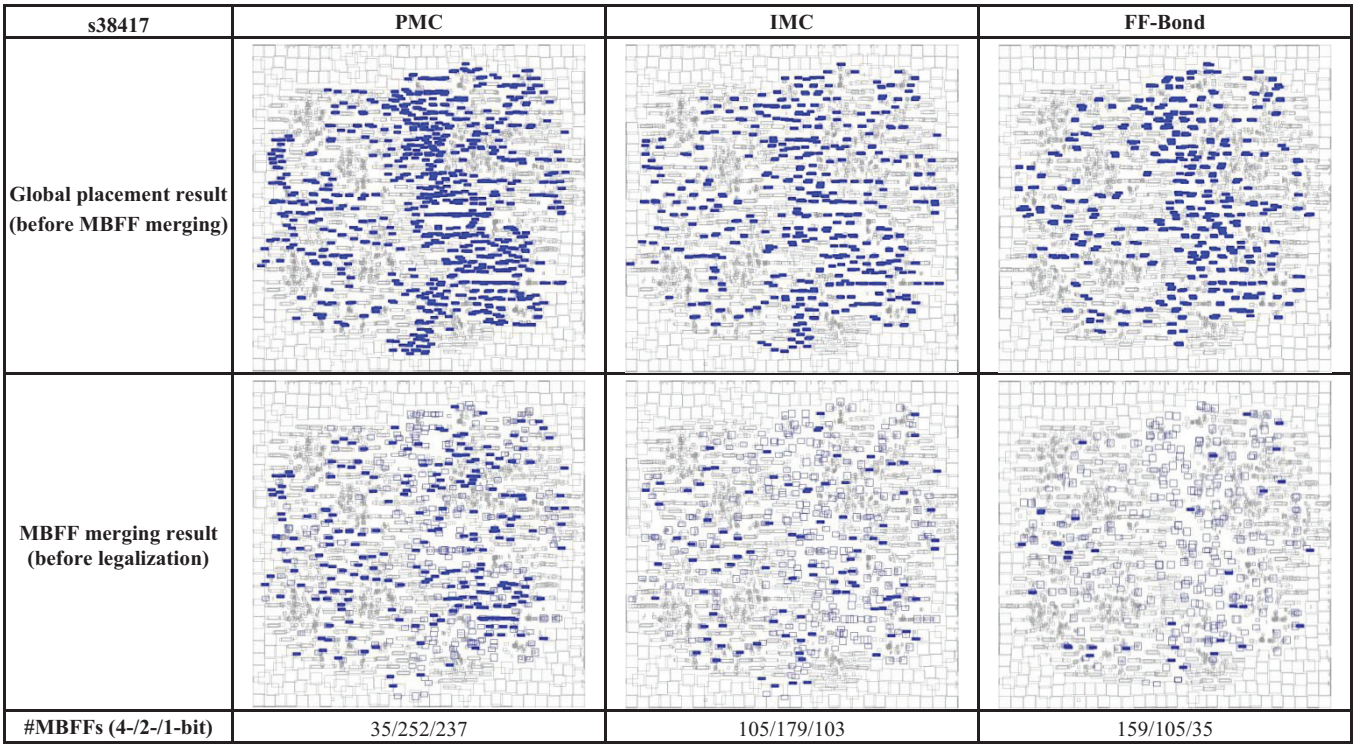


Figure 9. Global placement and MBFF merging results of s38417. Grey boxes indicate combinational cells and IOs. Solid boxes indicate single-bit flip-flops, while dark squares indicates MBFFs.