ARTICLE IN PRESS

INTEGRATION, the VLSI journal ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

freepapers.ir FREE papers

# Crosstalk-aware multi-bit flip-flop generation for power optimization ☆, ☆☆

Chih-Cheng Hsu [a], Mark Po-Hung Lin [a,*], Yao-Tsung Chang [b]

[a] National Chung Cheng University, Chiayi 621, Taiwan
[b] Synopsys, Inc., Hsinchu 300, Taiwan

ARTICLE INFO

*Article history:*
Received 18 December 2013
Received in revised form
28 August 2014
Accepted 28 August 2014

*Keywords:*
Power optimization
Synthesis for low power
Multi-bit flip-flop
Crosstalk
Physical design

ABSTRACT

Applying multi-bit flip-flops (MBFFs) for clock power reduction in modern nanometer ICs has been becoming a promising lower-power design technique. Many previous works tried to utilize as more MBFFs with larger number of bits as possible to gain more clock power saving. However, an MBFF with larger number of bits may lead to serious crosstalk due to the close interconnecting wires belonging to different signal nets which are connected to the same MBFF. This paper analyzes, evaluates, and compares the relationship between power consumption and crosstalk when applying MBFFs with different numbers of bits. To solve the addressed problem, a novel crosstalk-aware power optimization approach is further proposed to optimize power consumption while satisfying the crosstalk constraint. Experimental results show that the proposed approach is very effective in crosstalk avoidance when applying MBFFs for power optimization.

© 2014 Elsevier B.V. All rights reserved.

## 1. Introduction

Power/thermal minimization has been becoming one of the most important objectives in the design of modern system on chips (SOCs) which integrate huge numbers of transistors. High power/thermal dissipation of an SOC may degrade product lifetime and reliability. The power consumption of an SOC arises from dynamic, leakage, and short-circuit power, where the dynamic power is the major power source among the three [2]. The power consumption of the clock network further dominates the dynamic power [3] because of the highest switching rate of the clock signal.

To minimize the power consumption of the clock network, many techniques had been proposed, such as buffer sizing [4,5], clock gating [6,7], register clustering [8,9] and banking [10], and replacing 1-bit flip-flops with multi-bit flip-flops (MBFFs) [11–19]. Recent studies have shown the effectiveness of applying MBFFs in saving both power and area [11–13,16,18,19].

An MBFF consists of two or more 1-bit flip-flops, which share a common clock driver as shown in Fig. 1. The clock driver usually consists of two inverters that generate opposite phase clock signals for master and slave latches. According to [16], as the process technology advances to 65 nm and beyond, the minimum-sized clock driver can still drive several flip-flops. Consequently, replacing 1-bit flip-flops with MBFFs can reduce the power consumption of the clock network and the chip area due to the elimination of redundant clock drivers.

In addition to the reduction of flip-flop power consumption and chip area, some previous works [15,16] also presented other advantages of applying MBFFs. For example, common clock and enable signals for a group of flip-flops and reduced depth of a clock tree make clock skew more controllable. With fewer clock sinks and smaller capacitive load on the clock net, the delay and power consumption of the clock network can be improved. When considering design for testability (DFT) with MBFFs, the required routing resource utilization for a scan chain will be greatly reduced.

### 1.1. Previous work

The idea of applying MBFFs was first proposed in [15] to control clock skew and delay during physical synthesis. Kretchmer [14] and Chen et al. [16] introduced a design methodology to infer MBFF cells using existing logic synthesis tools. Based on the MBFF inference, it is possible to map an RTL design directly to a gate-level design with MBFF cells.

Recent studies suggested to apply MBFFs at the placement stage for better timing budgeting. All of the previous works

☆The preliminary version of this article was presented at the 2012 ACM/IEEE Asia South Pacific Design Automation Conference (ASP-DAC'12) [1].

☆☆This work was partially supported by National Science Council (now Ministry of Science and Technology) of Taiwan under Grant No's. NSC 100-2220-E-194-007-, NSC101-2220-E-194-006-, and NSC102-2220-E-194-006-.

* Corresponding author.

http://dx.doi.org/10.1016/j.vlsi.2014.08.002
0167-9260/© 2014 Elsevier B.V. All rights reserved.

Please cite this article as: C.-C. Hsu, et al., Crosstalk-aware multi-bit flip-flop generation for power optimization, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2014.08.002

ARTICLE IN PRESS



freepapers.ir FREE papers

[11–13,17–21] formulated the problem of replacing 1-bit flip-flops with MBFFs by minimizing flip-flop power consumption or the number of clock sinks while satisfying both timing and placement density constraints. Some of them additionally considered wire-length minimization [11–13,17], routability [18,19], and clock latency [20]. However, applying MBFF may also introduce substantial crosstalk noise among the signal nets. Fig. 2 illustrates examples of merging eight 1-bit flip-flops into one 8-bit flip-flops may result in serious crosstalk when an newly generated MBFF connects to the pins which are closed to each other. According to [22], serious crosstalk noise will result in unnecessary glitch and increase transition time on signal nets leading to functional failure. Fig. 3 gives two examples of signal anomalies. In Fig. 3(a), if one of the wires is switching, while the other one is stable, there will be a glitch on the stable wire. In Fig. 3(b), if two wires are simultaneously switching in opposite phases, the transition delay will be increased. Therefore, it is required to minimize crosstalk noise when applying MBFFs for power reduction.

### 1.2. Our contributions

In this paper, we address the crosstalk effect when applying MBFFs for clock power saving. We observe the ineffectiveness of applying MBFFs with larger number of bits due to serious crosstalk based on the crosstalk evaluation. We also propose a new problem formulation, which additionally consider the crosstalk constraint when applying MBFFs for clock power saving. Unlike the existing power optimization flows with MBFFs without considering crosstalk effect or merely considering routability during MBFF placement, we introduce the coupling capacitance map and present a novel algorithm to avoid crosstalk effect during Flip-Flop merging and MBFF placement. The key contributions are summarized in the following:

- We evaluate, analyze, and compare the relationship between power consumption and crosstalk when applying MBFFs with different numbers of bits, and address the ineffectiveness of applying MBFFs with larger number of bits due to the serious crosstalk.
- To solve the addressed problem, a novel crosstalk-aware power optimization flow and the corresponding algorithms are proposed to minimize power consumption while satisfying the crosstalk, timing, and placement density constraints.
- We present a crosstalk-aware bottom-up clustering algorithm for better crosstalk consideration instead of clustering a maximum number of 1-bit flip-flops by searching the maximum clique in the flip-flop intersection graph [11–13,17]. A coupling capacitance map is also introduced for better crosstalk estimation throughout the algorithms.
- Compared with our preliminary version [1], we improve the objective for flip-flop clustering, and introduce the new crosstalk-aware MBFF placement algorithm for better considering crosstalk.
- Experimental results show that the proposed approach is very effective in crosstalk avoidance when applying MBFFs for power optimization compared with the previous works.

The remainder of this paper is organized as follows. Section 2 demonstrates the crosstalk effect due to MBFFs. Section 3 presents a new problem formulation for placement-based power optimization with MBFFs to mitigate the crosstalk effect. Section 4 proposes our algorithms to solve the problem. Section 5 reports the experimental results, and finally Section 6 concludes this paper.

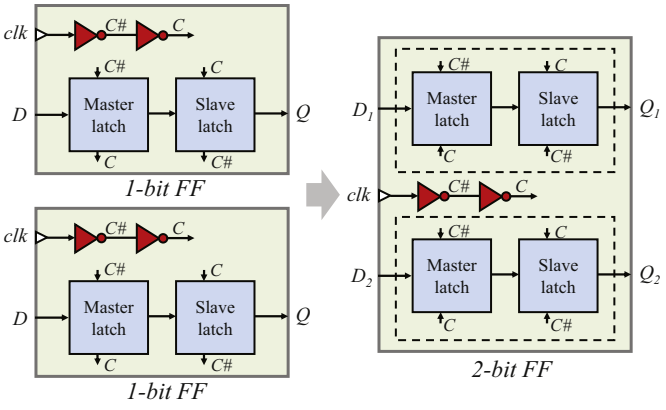


**Fig. 1.** An example of merging two 1-bit flip-flops into one 2-bit flip-flop [11,12].

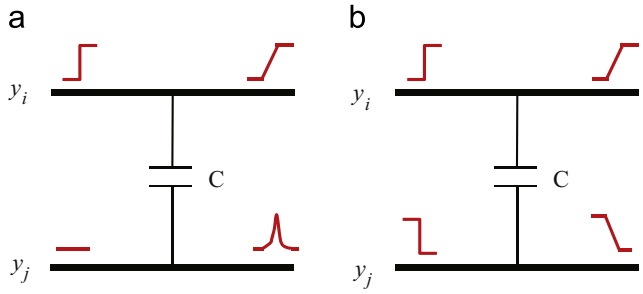


**Fig. 3.** Two examples of signal anomalies [22]. (a) Glitch. (b) Delay.

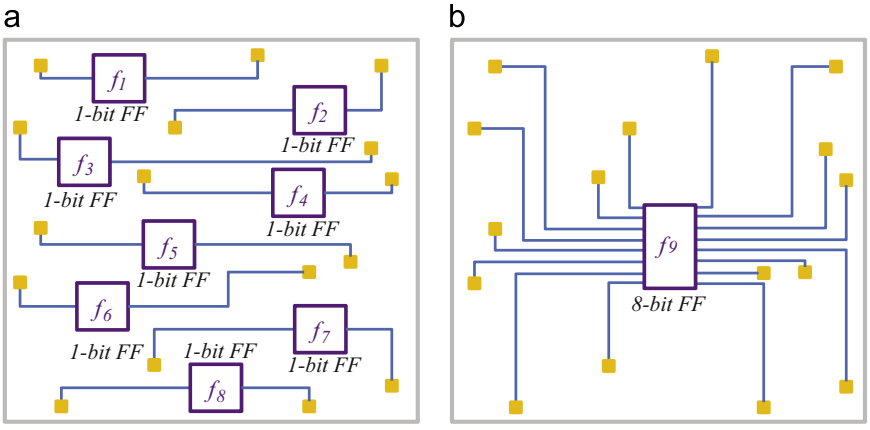


**Fig. 2.** An example of merging eight 1-bit flip-flops into one 8-bit flip-flop resulting in serious crosstalk among the connected signal nets.

Please cite this article as: C.-C. Hsu, et al., Crosstalk-aware multi-bit flip-flop generation for power optimization, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2014.08.002

ARTICLE IN PRESS



freepapers.ir FREE papers

## 2. Crosstalk effect due to MBFFs

This section first introduces a crosstalk model for interconnecting wires among signal nets. Based on the crosstalk model, the crosstalk evaluation flow is then presented to evaluate the crosstalk effect when applying MBFFs with different numbers of bits. Finally, the evaluation results are demonstrated to motivate our problem formulation in Section 3.

### 2.1. Crosstalk model

To calculate the crosstalk on a victim net, $n_i$, from any other aggressor net, $n_j$, we adopt the crosstalk model which was generally applied to global routing [23] and technology mapping [24]. The total crosstalk, $X_{n_i}$, on $n_i$ in worst case is the summation of all crosstalk effects from other nets, which can be obtained from Eq. (1), where $E_{ij}$ is the crosstalk coefficient, and $C_{ij}$ is the coupling capacitance between $n_i$ and $n_j$.

$$X_{n_i} = \sum_{j \neq i} E_{ij} C_{ij}. \tag{1}$$

The crosstalk coefficient, $E_{ij} \in [0, 1]$, is a real number indicating the crosstalk on $n_i$ contributed by one unit of coupling capacitance from $n_j$ and was set to be 1. The coupling capacitance $C_{ij}$ between two parallel wires, $w_i$ and $w_j$, as seen in Fig. 4, which belong to $n_i$ and $n_j$ respectively, can be further calculated by Eq. (2) [25–30], where $l$ is the parallel run length of the wires, $d$ is the distance between the wires, $\alpha$ is a constant indicating the unit capacitance, and $\beta$ was estimated to be 2 according to [31]. According to [25–30], such equation provides a rough but proper guidance to quickly estimate the coupling capacitance between two nets in different VLSI design stages, such as global routing and track assignment.

$$C_{ij} = \alpha \frac{l}{d^{\beta}}. \tag{2}$$

### 2.2. Crosstalk evaluation

Based on the crosstalk model, we present the crosstalk evaluation flows, as seen in Fig. 5, to evaluate the crosstalk effect due to MBFFs with different numbers of bits. The input of each flow includes a pre-placed design and an MBFF library, which are available from [11,12], as shown in Tables 1 and 2. In the first flow, we did not perform power optimization with MBFFs, while in the second, third, and forth flows, we implemented the algorithms proposed in [11,12], which minimize both power and wirelength in their objectives, to perform power optimization with 2-bit flip-flops, 2- and 4-bit flip-flops, and 2-, 4-, and 8-bit flip-flops, respectively.

After obtaining the optimized design with MBFFs in each flow, we further applied the coupling-free routing, which was implemented based on the pattern routing technique [32,33] to each optimized design. Finally, the power consumption of each design was calculated according to the normalized power consumption of each flip-flop cell as listed in Table 2, and the crosstalk among different nets was also analyzed based on the aforementioned crosstalk model. The analyzed results of all flows are shown in Table 3.

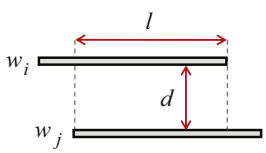


**Fig. 4.** The coupling capacitance between two parallel wires.

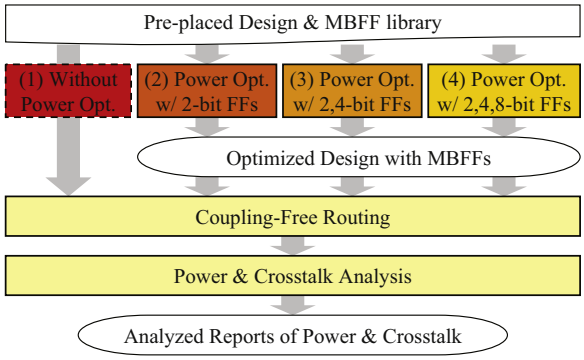


**Fig. 5.** The crosstalk evaluation flows for power optimization using MBFFs with different numbers of bits: (1) Without power optimization using MBFFs, (2) Power optimization with 2-bit flip-flops, (3) Power optimization with 2-, and 4-bit flip-flops, and (4) Power optimization with 2-, 4-, and 8-bit flip-flops.

### 2.3. Crosstalk effect

Table 3 lists the names of the benchmark circuits ("Circuit"), normalized flip-flop power consumption ("Norm. FF Power"), normalized clock wirelength ("Norm. Clock WL"), normalized signal wirelength of all FF connection nets ("Norm. Signal WL"), and normalized average crosstalk on each net ("Norm. Avg. Xtalk") of the circuits for the four flows in Fig. 5. For each circuit, the flip-flop power consumption, the clock wirelength, the wirelength of all FF connection nets, and the average crosstalk on each net are normalized with respect to the first flow. We compare the normalized flip-flop power consumption, normalized clock wirelength, normalized wirelength of all FF connection nets, and average crosstalk based on different flows by averaging those of different circuits in each flow as seen in the last row of Table 3.

According to the comparisons in Table 3, we can draw four curves, as seen in Fig. 6, to observe the relationship between the flip-flop power consumption, clock wirelength, wirelength of all FF connection nets, and average FF connection net crosstalk with respect to the number of bits of the applied MBFFs. In Fig. 6(a) and (b), the flip-flop power consumption and clock wirelength are significantly reduced when applying 2-bit or 4-bit flip-flops. However, the power and clock wirelength reduction rates keep decreasing when applying MBFFs with increasing number of bits. In Fig. 6(c), when applying 2-bit or 4-bit flip-flops, the wirelength of all FF connection nets is shorter than that when applying 1-bit flip-flops. When applying MBFF with 8-bit flip-flops, it is 2.5% larger than that applying 1-bit flip-flops. In Fig. 6(d), the crosstalk of applying 2-bit flip-flops is comparable to that of applying 1-bit flip-flops. Nevertheless, it dramatically increases when applying 4-bit and 8-bit MBFFs. Consequently, applying MBFFs with larger number of bits may achieve only a little more power saving, but introduce longer wirelength and much more serious crosstalk than applying MBFFs with smaller number of bits. It is essential to mediate the trade-off between power and crosstalk when performing power optimization with MBFFs.

## 3. Problem formulation

To mitigate the crosstalk effect due to MBFFs as described in the previous section, we define the *crosstalk-aware power optimization with MBFFs* problem as follows:

*Problem: Crosstalk-aware power optimization with MBFFs*

Given a pre-placed design with a set of flip-flops, and a MBFF library, we want to minimize total flip-flop power consumption, while satisfying placement density, timing, and crosstalk constraints,

Please cite this article as: C.-C. Hsu, et al., Crosstalk-aware multi-bit flip-flop generation for power optimization, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2014.08.002

freepapers.ir FREE papers

**Table 1**
The benchmark circuits in [11,12].

| Circuit | # of 1-bit FFs | # of 2-bit FFs | Total WL (nm) | Chip size (nm) | # of bins | Bin dimension (nm) | # of nets | Max. allowable placement density ($nm^2$) |
|---|---|---|---|---|---|---|---|---|
| c1 | 76 | 22 | 89,425 | 3000 × 3000 | 36 | 500 × 500 | 240 | 19,000 |
| c2 | 366 | 57 | 348,920 | 6000 × 6000 | 144 | 500 × 500 | 960 | 19,000 |
| c3 | 1464 | 228 | 1,395,680 | 12,000 × 12,000 | 576 | 500 × 500 | 3840 | 19,000 |
| c4 | 4378 | 751 | 4,290,655 | 21,000 × 21,000 | 1764 | 500 × 500 | 11,760 | 19,000 |
| c5 | 9150 | 1425 | 8,723,000 | 30,000 × 30,000 | 3600 | 500 × 500 | 24,000 | 19,000 |
| c6 | 146,400 | 22,800 | 139,568,000 | 120,000 × 120,000 | 57,600 | 500 × 500 | 384,000 | 19,000 |

**Table 2**
The MBFF library in [11,12,18].

| Flip-flop bit # | Normalized flip-flop power per bit | Normalized flip-flop area per bit |
|---|---|---|
| 1 | 1.00 | 1.00 |
| 2 | 0.86 | 0.96 |
| 4 | 0.78 | 0.71 |
| 8 | 0.75 | 0.59 |

**Table 3**
Comparisons of flip-flop power consumption, clock wirelength, signal wirelength, and crosstalk for different flows in Fig. 5.

| Circuit | (1) Without power opt. | | | | (2) Power opt. w/ 2-bit FFs | | | |
|---|---|---|---|---|---|---|---|---|
| | Norm. FF power | Norm. clock WL | Norm. signal WL | Norm. avg. Xtalk | Norm. FF power | Norm. clock WL | Norm. signal WL | Norm. avg. Xtalk |
| c1 | 1 | 1 | 1 | 1 | 0.911 | 0.820 | 0.837 | 1.517 |
| c2 | 1 | 1 | 1 | 1 | 0.894 | 0.788 | 0.812 | 1.102 |
| c3 | 1 | 1 | 1 | 1 | 0.894 | 0.805 | 0.819 | 0.999 |
| c4 | 1 | 1 | 1 | 1 | 0.896 | 0.813 | 0.825 | 1.023 |
| c5 | 1 | 1 | 1 | 1 | 0.894 | 0.809 | 0.822 | 0.985 |
| c6 | 1 | 1 | 1 | 1 | 0.894 | 0.804 | 0.824 | 0.975 |
| Comp. | 1 | 1 | 1 | 1 | 0.90 | 0.81 | 0.82 | 1.10 |
| | (3) Power Opt. w/ 2,4-bit FFs | | | | (4) Power Opt. w/ 2,4,8-bit FFs | | | |
| c1 | 0.852 | 0.656 | 0.917 | 4.120 | 0.835 | 0.623 | 0.997 | 6.583 |
| c2 | 0.831 | 0.646 | 0.947 | 3.229 | 0.816 | 0.582 | 1.022 | 5.302 |
| c3 | 0.829 | 0.656 | 0.948 | 3.338 | 0.816 | 0.609 | 1.030 | 5.361 |
| c4 | 0.832 | 0.670 | 0.945 | 3.420 | 0.818 | 0.622 | 1.033 | 5.342 |
| c5 | 0.829 | 0.660 | 0.949 | 3.430 | 0.816 | 0.613 | 1.034 | 5.520 |
| c6 | 0.828 | 0.657 | 0.949 | 3.094 | 0.816 | 0.616 | 1.036 | 5.588 |
| Comp. | 0.83 | 0.66 | 0.94 | 3.44 | 0.82 | 0.61 | 1.03 | 5.62 |

which are illustrated in Eqs. (3), (4) and (5), respectively.

$$D_{b_i} \le D_{max}, \ \forall b_i. \tag{3}$$

$$W_{n_i} \le W_{n_i,max}, \ \forall n_i. \tag{4}$$

$$X_{n_i} \le X_{n_i,max}, \ \forall n_i. \tag{5}$$

For placement density constraints, a chip is equally divided into a set of bins, and the placement density of each bin, $D_{b_i}$, must be less than or equal to the maximum allowable placement density, $D_{max}$, where $D_{b_i}$ is equal to the areas of all flip-flops and cells in the bin, $b_i$, divided by the area of $b_i$. In order to avoid routing congestion, when merging two or more flip-flops into one MBFF, the placement density constraint should be considered during MBFF placement because an MBFF has larger area compared with all the merged 1-bit flip-flops.

For timing constraints, the timing model with the consideration of coupling capacitance was presented in [29]. According to [11–13,17,19], the delay of a net can be modelled with respect to the corresponding wirelength. The wire length of a net, $W_{n_i}$, must be less than or equal to the maximum allowable wire length of the net, $W_{n_i,max}$, which can be estimated based on the timing model [29]. When replacing several 1-bit flip-flops with an MBFF, the wirelength between the flip-flops and its connected pins may become much longer, as seen in Fig. 7. In Fig. 7(a), there are four 1-bit flip-flops, $f_1$, $f_2$, $f_3$ and $f_4$, and the corresponding connected pins, $p_1$, $p_2$, $p_3$, $p_4$, $p_5$, $p_6$, $p_7$ and $p_8$. After replacing $f_1$ and $f_2$ with the 2-bit flip-flop, $f_5$, and replacing $f_3$ and $f_4$ with the 2-bit flip-flop, $f_6$, as shown in Fig. 7(b), the wirelength from $f_5$ to $p_4$ and the wirelength from $f_6$ to $p_5$ become much longer. Longer wirelength will introduce much larger delay leading to timing violation.

In addition to the placement density and timing constraints, we shall further consider the crosstalk constraint. The crosstalk of each net, $X_{n_i}$, must be less than or equal to the maximum allowable crosstalk of the net, $X_{n_i,max}$, where $X_{n_i}$ has been derived in Eq. (1). We empirically set the value of $X_{n_i,max}$ to 65%, which is the largest crosstalk value in the original design. Fig. 7 shows an example of unsuitably merging four 1-bit flip-flops into two 2-bit flip-flops, $f_5$ and $f_6$, and placing $f_5$ and $f_6$ at poor location, which may result in serious crosstalk among the interconnecting wires.

## 4. The proposed algorithms

The flowchart of our algorithms is shown in Fig. 8. Given an initial placement of a design, the coupling capacitance map is first generated to indicate the degree of net coupling throughout the chip area (see Section 4.1). The flip-flop intersection graph is then constructed according to the timing constraint of each net which connects to a flip-flop (see Section 4.2). Based on the coupling capacitance map and the flip-flop intersection graph, the crosstalk-aware flip-flop clustering algorithm (see Section 4.3) is developed to iteratively cluster flip-flops while satisfying timing and crosstalk constraints until there is no MBFF which can be further applied to the design. When a new MBFF is generated, the coupling capacitance map and the flip-flop intersection graph should be updated accordingly. Finally, the MBFF placement algorithm (see Section 4.4) optimizes the locations of all MBFFs.

In order to effectively merge local 1-bit flip-flops and reduce the problem size, we apply the progressive window-based optimization technique [11,12], as shown in Fig. 9. For each iteration, the flip-flop clustering algorithm is performed within a predefined window in the chip area. The window is initialized with 2 × 2 bins in a chip corner, and then slid horizontally and vertically throughout the chip area. After clustering local flip-flops within the window size of 2 × 2 bins, the window size is then progressively enlarged to further cluster more flip-flops. Such technique can help to minimize flip-flop displacement after replacing 1-bit flip-flops with MBFFs.

Please cite this article as: C.-C. Hsu, et al., Crosstalk-aware multi-bit flip-flop generation for power optimization, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2014.08.002

freepapers.ir FREE papers

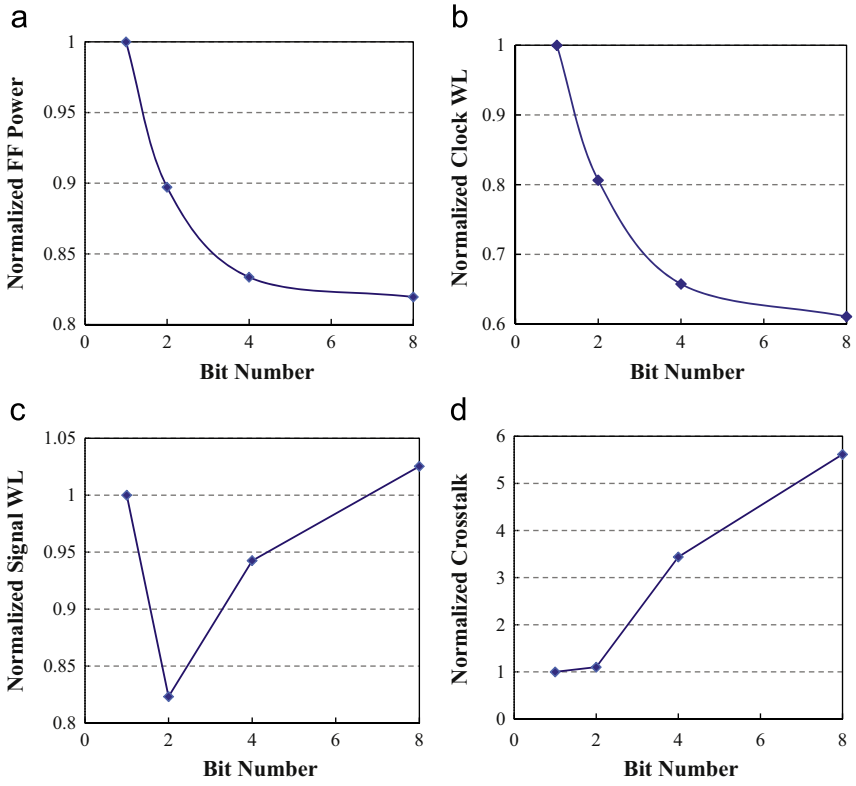


**Fig. 6.** The relationship between (a) flip-flop power consumption, (b) total clock wirelength, (c) total signal wirelength, and (d) crosstalk, with respect to the maximum number of bits of applied MBFFs.

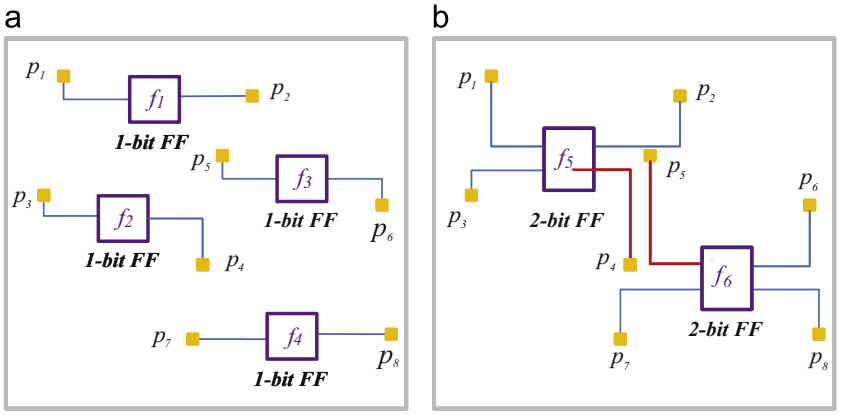


**Fig. 7.** (a) Four 1-bit flip-flops and their interconnections. (b) Merging the four 1-bit flip-flops in (a) into two 2-bit flip-flops may result in longer wirelength and serious crosstalk among the interconnections.

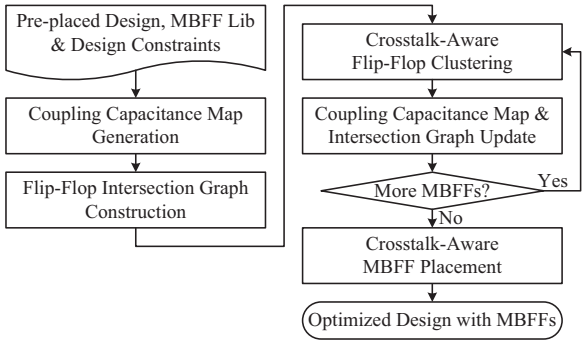


**Fig. 8.** The flowchart for crosstalk-aware power optimization with MBFFs.

### 4.1. Coupling capacitance map generation

Similar to the coupling capacitance map in [34,35] and the noise map in [36], which were proposed for crosstalk-aware placement, our coupling capacitance map acts as a guide for both crosstalk-aware flip-flop clustering and MBFF placement. It should be noted that a coupling capacitance map is different from a congestion map because optimization with a congestion map may not satisfactorily reduce coupling capacitance [34–36].

Before generating the coupling capacitance map, the chip area is first equally divided into a set of rectangular bins, $B = \{b_1, b_2, \ldots, b_n\}$, or global routing cells, as shown in Fig. 10. We apply coupling-free global routing [32,33] to fast generate the routing topology of each net. Fig. 10 compares the routing topologies of four nets, $n_1$,

freepapers.ir FREE papers

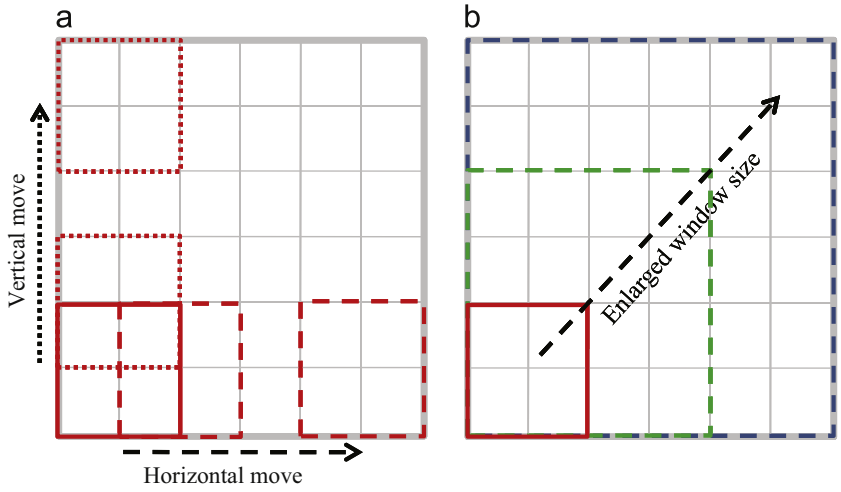


**Fig. 9.** Progressive window-based optimization technique. (a) Window sliding with the size of 2 × 2 bins. (b) Enlarged windows with the size of 4 × 4 and 6 × 6 bins.

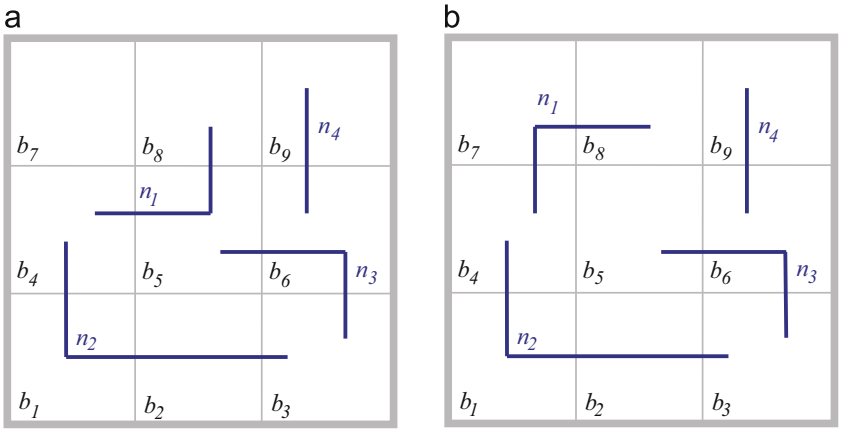


**Fig. 10.** Routing topologies of the nets, $n_1$, $n_2$, $n_3$, and $n_4$, after performing (a) routability-driven and (b) coupling-free global routing on nine rectangular bins.

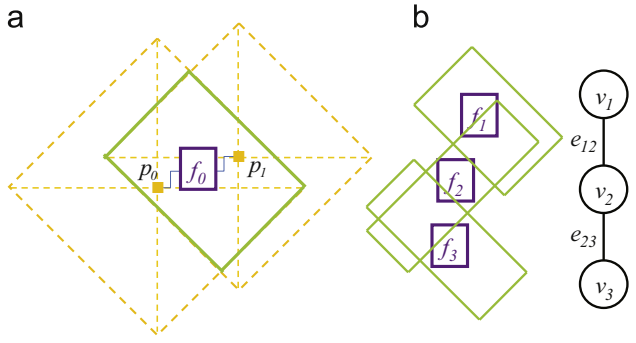


**Fig. 11.** (a) The feasible region of the flip-flop, $f_0$. (b) The feasible regions of the flip-flops, $f_1$, $f_2$, and $f_3$, and the corresponding flip-flop intersection graph.

$n_2$, $n_3$, and $n_4$, based on the routability-driven global routing [37] and the coupling-free global routing [32,33]. The routability-driven global routing may generate the routing topology in either Fig. 10(a) or (b) because the number of nets across each bin boundary is less than or equal to one in both cases, and the total wirelengths of both cases are also the same. However, the coupling-free global routing will only produce the routing topology in Fig. 10(b) because it always minimizes the coupling capacitance among different nets.

According to the resulting routing topology in Fig. 10(b), the coupling capacitance map can be generated by calculating the coupling capacitance of each bin, which is the sum of the coupling capacitances on all wire segments inside the bin. For example, in Fig. 10(b), the coupling capacitance of $b_2$, $C_2^b$, is equal to $C_{n_2}^{b_2}$, which is the capacitance on the horizontal wire segment of $n_2$ inside $b_2$. $C_{n_2}^{b_2}$ is obtained by summing up the coupling capacitances between the parallel wire segments of $n_2$ inside $b_2$ and $n_3$ inside $b_5$, and those of $n_2$ inside $b_2$ and $n_1$ inside $b_8$. The coupling capacitance between two parallel wires can be calculated by Eq. (2).

### 4.2. Flip-Flop intersection graph

To satisfy the timing constraint, which is defined in Section 3, a flip-flop must be placed in the feasible region, as seen in Fig. 11(a), which is the tilted rectangular region intersected by the maximum allowable Manhattan rings [38,9] from all its connected pins, $p_0$ and $p_1$. When merging several flip-flops with an MBFF, the MBFF should be placed at a common feasible region of all merged flip-flops such that the timing constraints of all its connected nets are satisfied. We construct a flip-flop intersection graph to represent the relationship among the feasible regions of all flip-flops.

Fig. 11(b) shows the feasible regions of the flip-flops, $f_1$, $f_2$, and $f_3$, and the corresponding flip-flop intersection graph, $G(V,E)$, where each vertex, $v_i \in V$, corresponds to the feasible region of a flip-flop, $f_i$. There is an edge, $e_{ij} \in E$, between $v_i$ and $v_j$, if there exists an intersection between the feasible regions of $f_i$ and $f_j$, and the total numbers of bits of $f_i$ and $f_j$ is less than or equal to the largest MBFF number of bits in the library. For example, $f_1$ and $f_3$ cannot be grouped and merged by an MBFF since the feasible regions of $f_1$ and $f_3$ has no intersection. On the contrary, $f_1$ and $f_2$

Please cite this article as: C.-C. Hsu, et al., Crosstalk-aware multi-bit flip-flop generation for power optimization, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2014.08.002

ARTICLE IN PRESS



freepapers.ir FREE papers

can be grouped and merged by an MBFF because the merged MBFF can be placed in the intersection of the feasible regions of $f_1$ and $f_2$ such that the timing constraint of the merged MBFF is met.

### 4.3. Crosstalk-aware flip-flop clustering

According to Section 2, an MBFF with larger number of bits may introduce much more serious crosstalk than that with smaller number of bits. Therefore, we propose a crosstalk-aware bottom-up clustering algorithm, as illustrated in Algorithm 1, for better crosstalk consideration, instead of clustering a maximum number of 1-bit flip-flops by searching the maximum clique in the flip-flop intersection graph [11–13,17].

**Algorithm 1.** Crosstalk-aware Flip-Flop Clustering.

**Require:** A flip-flop intersection graph, $G(V,E)$.
**Ensure:** A set of non-conflicting flip-flop clusters.
1: $E' \leftarrow \phi$; // $E'$ is a set of clustered edges.
2: $V' \leftarrow \phi$; // $V'$ is a set of clustered vertices.
3: $H \leftarrow \phi$; // $H$ is a heap for all edges.
4: **for all** $e_{ij} \in E$ **do**
5: ComputeKey($e_{ij}$);
6: $H.push(e_{ij})$;
7: **end for**
8: **while** $H \neq \phi$ **do**
9: $e_{ij} \leftarrow H.extractMin()$;
10: **if** $(v_i \cap V' = \phi) \cap (v_j \cap V' = \phi)$ **then**
11: EstimatClusteringeCrosstalk($e_{ij}$);
12: **if** NoCrosstalkViolation($e_{ij}$) **then**
13: AddFlipFlopCluster($e_{ij}$);
14: $E' = E' \cup e_{ij}$;
15: $V' = V' \cup v_i$;
16: $V' = V' \cup v_j$;
17: UpdateRouting($e_{ij}$);
18: UpdateCouplingCapacitanceMap($e_{ij}$);
19: **end if**
20: **end if**
21: **end while**

The input of our algorithm is a flip-flop intersection graph, while the output is a set of non-conflicting flip-flop clusters. The algorithm starts with computing the key values of all edges in $E$ and storing the edges in a heap, $H$ (see Lines 4–7). The key value of an edge, $e_{ij}$, can be calculated by Eq. (6), where $W_{e_{ij}}$ is the HPWL ratio of all pins connected to $f_i$ and $f_j$ before and after newly MBFF generation, $A_{e_{ij}}$ is the area ratio of the intersected feasible regions of $f_i$ and $f_j$ before and after MBFF generation, $X_{e_{ij}}$ is the estimated crosstalk ratio of all nets connected to $f_i$ and $f_j$ before and after MBFF generation, and $\gamma$ and $\delta$ are constant coefficients, which were set to be 0.6 and 0.2, respectively. An edge, $e_{ij}$, in the flip-flop intersection graph with a minimum key value indicates that merging the corresponding flip-flops into an MBFF will result in shorter interconnecting wirelength, larger feasible region for MBFF placement, and less impact on crosstalk after merging the corresponding flip-flops.

$$Key(e_{ij}) = \gamma W_{e_{ij}} - \delta A_{e_{ij}} + (1-\gamma-\delta) X_{e_{ij}}. \quad (6)$$

After all edges are stored in the heap, $H$, the edge with the minimum key value will be iteratively extracted from $H$ (see Lines 8–21). During each iteration, we consider the edge, $e_{ij}$, which connects $v_i$ and $v_j$ corresponding to the feasible regions of $f_i$ and $f_j$. If $f_i$ and $f_j$ have not been clustered, we estimate the crosstalk of each net connecting to $f_i$ and $f_j$, after they are merged into an MBFF. If there is no crosstalk constraint violation, a flip-flop cluster containing $f_i$ and $f_j$ is then added to the set of non-conflicting flip-flop cluster.

To estimate the net crosstalk induced by an MBFF, we place the newly generated MBFF at a bin with the least coupling capacitance inside the feasible region, and re-route the previously removed nets by performing the coupling-free global routing. Based on the routing results, the crosstalk on each net can be re-calculated by Eqs. (1) and (2).

After all edges in the heap are extracted, $G(V,E)$ will be updated if there is any clustered edge in $E'$. When updating $G(V,E)$, we create a super vertex for each clustered edge, $e'_{ij}$ in $E'$, and merge its connected vertices, $v_i$ and $v_j$, into the super vertex if the total numbers of bits of $f_i$ and $f_j$ is less than the maximum number of bits of the available MBFFs in the library. Once the super vertex is created, if there is a vertex, $v_k$, in $G(V,E)$ which is connected to both $v_i$ and $v_j$, a new edge is created to connect the super vertex and $v_k$. After the connection of the super vertex is built up, the merged vertices and all their connected edges are removed. Consequently, a new graph is obtained for the next clustering iteration.

Fig. 12 illustrates the details of Algorithm 1 with an example. Assuming that there are four 1-bit flip-flops in a design, and the

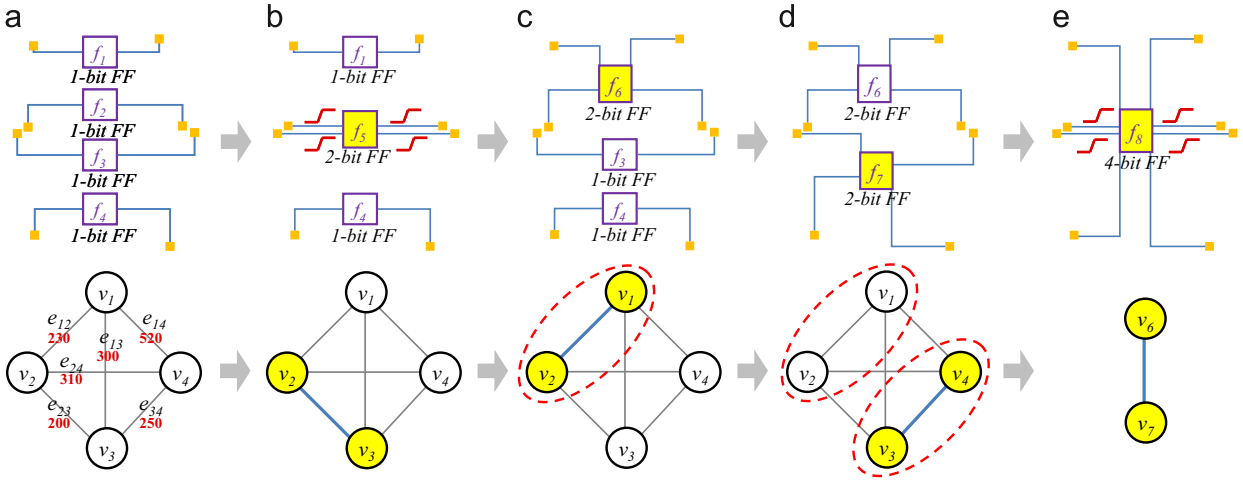


**Fig. 12.** An example for the crosstalk-aware flip-flop clustering algorithm. (a) A design containing four 1-bit flip-flops, and the corresponding flip-flop intersection graph. (b) Merging $f_2$ and $f_3$ into a 2-bit flip-flop, $f_5$, resulting in crosstalk constraint violation. (c) Merging $f_1$ and $f_2$ into a 2-bit flip-flop, $f_6$, without crosstalk constraint violation. (d) Merging $f_3$ and $f_4$ into a 2-bit flip-flop, $f_7$, without crosstalk constraint violation. (e) Merging $f_6$ and $f_7$ into a 4-bit flip-flop, $f_8$, resulting in crosstalk constraint violation.

Please cite this article as: C.-C. Hsu, et al., Crosstalk-aware multi-bit flip-flop generation for power optimization, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2014.08.002

ARTICLE IN PRESS



freepapers.ir FREE papers

feasible region of each flip-flop intersects one another, the corresponding flip-flop intersection graph of the design can be constructed with a key value annotated on each edge, as shown in Fig. 12(a). To simply the example, we do not show other signal nets in the design and the corresponding coupling capacitance map.

In Fig. 12(b), the edge with the smallest key value, $e_{23}$, is first selected, and the corresponding flip-flops, $f_2$ and $f_3$, are considered to be merged into a 2-bit flip-flop, $f_5$. The crosstalk of each net after merging $f_2$ and $f_3$ is then estimated by performing the coupling-free global routing. Since the crosstalk constraint in this case is violated due to long parallel wires connecting to $f_5$, $e_{23}$ is discarded as a merging candidate.

Next, the edge with the second smallest key value, $e_{12}$, is selected. When merging $f_1$ and $f_2$ with a 2-bit flip-flop, $f_6$, the crosstalk of each net satisfies the crosstalk constraint after performing the coupling-free global routing as shown in Fig. 12(c). Therefore, $e_{12}$ can be considered as a merging candidate, and the coupling capacitance map is also updated. We then consider the next edge, $e_{34}$. Similarly, merging $f_3$ and $f_4$ with a 2-bit flip-flop, $f_7$, does not cause any crosstalk constraint violation as shown in Fig. 12(d). Hence, $e_{34}$ is also considered as a merging candidate, and the coupling capacitance map is updated accordingly. Since $f_1$, $f_2$, $f_3$, and $f_4$ have been clustered, $e_{13}$, $e_{24}$, and $e_{14}$ will not be considered.

Once all the edges in the heap are traversed, the graph is then updated according to the clustered edges in $E'$. Fig. 12(e) shows the updated graph, where the super vertices, $v_6$ and $v_7$, are created for the clustered edges, $e_{12}$ and $e_{34}$, respectively, and there is an edge between $v_6$ and $v_7$ because the original graph is a complete graph. Based on the updated graph, Algorithm 1 is further performed.

Since merging $f_2$ and $f_3$ may cause crosstalk constraint violation as shown in Fig. 12(b), we cannot avoid the crosstalk when merging $f_6$ and $f_7$ into a 4-bit flip-flop. Consequently, the four 1-bit flip-flops in Fig. 12(a) can only be replaced with two 2-bit flip-flops as shown in Fig. 12(d) such that both timing and crosstalk constraints are satisfied.

The overall time complexity of the crosstalk-aware flip-flop clustering algorithm in Algorithm 1 is $O(n^2m)$, where $n$ is the number of flip-flops, and $m$ is the largest bit number of MBFFs. When generating a non-conflicting set of all the explored flip-flip clusters, it takes $O(n^2)$ to compute the key value, $Key(e_{ij})$, since there are $n$ flip-flops and $C_2^n$ edges in the worst case (Lines 4–7). It takes $O(n^2lgn)$ to sort all $e_{ij}$ based on their key values. For each $e_{ij}$ and the corresponding flip-flips, $f_i$ and $f_j$ , it takes $O(m)$ to estimate the crosstalk induced by newly generated $m$-bit flip-flop. Finally, it further takes $O(m)$ to generate a non-conflicting of flip-flip cluster and update the coupling capacitance map if there is no crosstalk constraint violation (Lines 8–21).

### 4.4. Crosstalk-aware MBFF placement

When searching the best bin to accommodate an MBFF inside its feasible region, all the previous works [11–13,17] only considered the placement density constraint, as illustrated in Equation (3), while minimizing interconnecting wirelength. Fig. 13(a) shows the preferred region of an MBFF for wirelength minimization, which is within the median coordinates of the fan-in and fan-out gates of the MBFF. If the preferred region is not inside the feasible region of the MBFF, the MBFF will be placed in a bin which is close to the preferred region.

**Algorithm 2.** Crosstalk-aware MBFF Placement.

**Require**: A set of non-conflicting flip-flop clusters, $C$.
**Ensure**: All MBFFs with corresponding positions.

```
1: while C ≠ φ do
2:   P ← φ; // P is a set of grids.
3:   c_i ← C.extract();
4:   for all p_j in the preferred region of c_i do
5:       ComputeWeight (c_i, p_j);
6:       P.push(p_j);
7:   end for
8:   while P ≠ φ do
9:       p_j ← P.extractMin();
10:      EstimatePlacementCrosstalk (c_i, p_j);
11:      if (NoConstraintViolation(c_i) ∩ LegalGrid(p_j)) then
12:          PlaceMBFF (c_i, p_j);
13:          UpdateRouting(c_i);
14:          UpdateCouplingCapacitanceMap(c_i);
15:          break;
16:      end if
17:  end while
18:  if PositionNotFound(c_i) in preferred region then
19:      for all p_k in the feasible region of c_i do
20:          EstimatePlacementCrosstalk (c_i, p_k);
21:          if (NoConstraintViolation(c_i) ∩ LegalGrid (p_k))
 then
22:              PlaceMBFF (c_i, p_k);
23:              UpdateRouting(c_i);
24:              UpdateCouplingCapacitanceMap(c_i);
25:              break;
26:          end if
27:      end for
28:  end if
29:  if PositionNotFound(c_i) then
30:      Decluster(c_i);
31:  end if
32: end while
```

In addition to the placement density constraint, we additionally consider the crosstalk constraint when searching for the best placement bin and grid for an MBFF. In Fig. 13, if there already exists a fixed MBFF, $f_2$, a newly generated MBFF, $f_1$, is selected to place within the corresponding preferred region. It may lead to crosstalk constraint violation between the interconnecting wires of $f_1$ and $f_2$ when we purely consider placement density constraint as shown in Fig. 13(b) whereas we apply coupling-free global routing and use coupling capacitance map as a guidance to additionally consider the crosstalk constraint in order to avoid the crosstalk violation between the interconnecting wires of $f_1$ and $f_2$, as shown in Fig. 13(c), both placement density constraint and crosstalk constraint are satisfied when searching for the placement bin and grid for $f_2$.

Algorithm 2 shows our crosstalk-aware MBFF placement algorithm. The input is a set of non-conflicting flip-flop clusters, $C$, while the output is newly generated MBFFs and their corresponding positions. For each flip-flop cluster, $c_i \in C$, we first collect the set of grids, $P$, in the preferred region for $c_i$ which results in the shortest wirelength, as seen in Fig. 13(a), and then analyze the local pin and wire densities around each grid (see Lines 4–7). The grids in $P$ are the candidates that the newly generated MBFF will be placed. The weight of a candidate grid, $p_j$, of a flip-flop cluster, $Weight(c_i, p_j)$, can be calculated by Eq. (7), where $N_k^v$ ($N_k^h$) is the number of pins and/or flip-flops in the $k$th vertical (horizontal) grid, and $(x_1, y_1)$ and $(x_2, y_2)$ are left-bottom and right-top coordinates when the newly generated MBFF is placed at $p_j$. A grid, $p_j$, with smaller weight indicates lower pin and wire densities around $p_j$ resulting in better routability around $p_j$.

$$Weight(c_i, p_j) = \sum_{k=x_1}^{x_2} N_k^v + \sum_{k=y_1}^{y_2} N_k^h. \quad (7)$$

Please cite this article as: C.-C. Hsu, et al., Crosstalk-aware multi-bit flip-flop generation for power optimization, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2014.08.002

freepapers.ir FREE papers

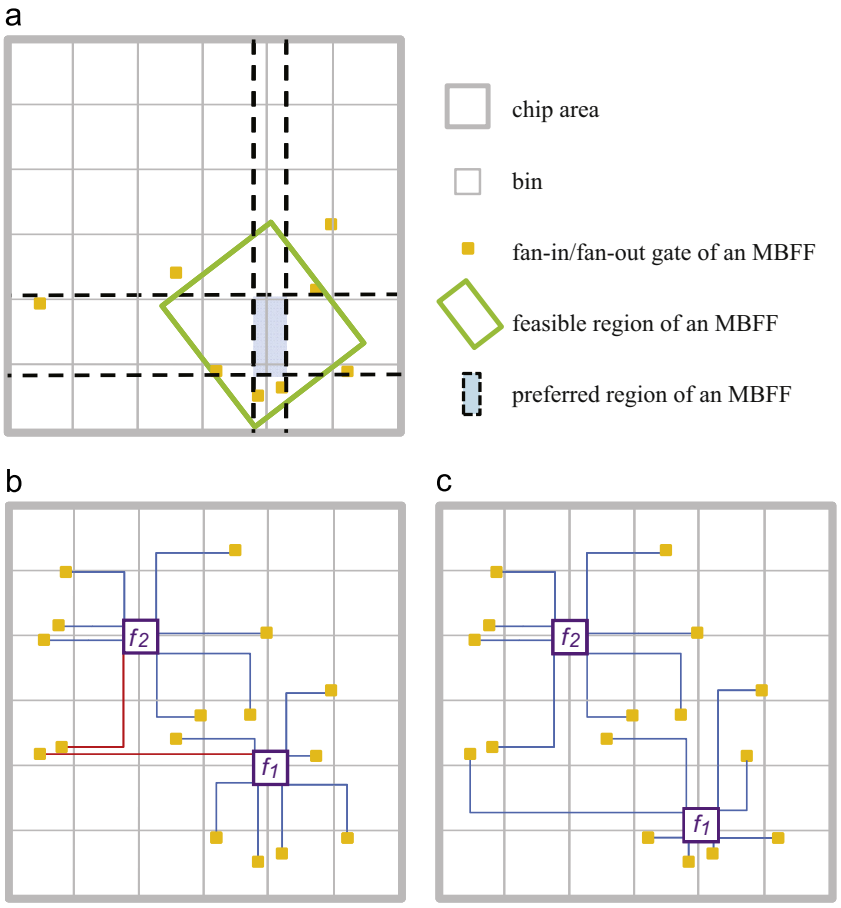


**Fig. 13.** (a) The preferred region of the MBFF, $f_1$, for wirelength minimization. (b) The placement of $f_1$ with crosstalk constraint violation. (c) The placement of $f_1$ without crosstalk constraint violation.

**Table 4**
Comparisons of power ratios, wirelength ratios, average crosstalk ratios, and CPU times for three different approaches, [1], our approach without crosstalk-aware MBFF placement, and our approach with crosstalk-aware MBFF placement.

| Circuit | Our preliminary version [1] | | | | Our approach w/o crosstalk-aware MBFF placement | | | | Our approach w/ crosstalk-aware MBFF placement | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Power ratio | WL ratio | Xtalk ratio | Time (s) | Power ratio | WL ratio | Xtalk ratio | Time (s) | Power ratio | WL ratio | Xtalk ratio | Time (s) |
| c1 | 0.869 | 0.920 | 3.459 | 0.09 | 0.869 | 0.923 | 3.649 | 0.13 | 0.877 | 0.929 | 2.815 | 0.16 |
| c2 | 0.855 | 0.916 | 2.181 | 0.63 | 0.857 | 0.931 | 2.617 | 0.79 | 0.859 | 0.915 | 1.886 | 0.81 |
| c3 | 0.855 | 0.920 | 2.015 | 2.74 | 0.855 | 0.933 | 2.443 | 2.85 | 0.858 | 0.921 | 1.986 | 2.92 |
| c4 | 0.859 | 0.919 | 2.072 | 8.87 | 0.859 | 0.930 | 2.303 | 9.00 | 0.860 | 0.924 | 2.094 | 9.028 |
| c5 | 0.856 | 0.923 | 2.016 | 18.80 | 0.855 | 0.936 | 2.453 | 18.73 | 0.857 | 0.925 | 2.014 | 19.05 |
| c6 | 0.856 | 0.925 | 2.007 | 320.42 | 0.855 | 0.937 | 2.431 | 329.64 | 0.857 | 0.926 | 2.052 | 336.23 |
| Comp. | 1 | 1 | 1.06 | 0.867 | 1 | 1.01 | 1.24 | 0.951 | 1 | 1 | 1 | 1 |

Once the set grids, $P$, in the preferred region for a flip-flop cluster, $c_i$, are collected, the grid with the minimal weight is then iteratively extracted from $P$ (see Lines 8–17) until a legal grid without constraint violation is found. A legal grid means that placing the corresponding MBFF at the grid will not overlap other cells. For each extracted grid, we estimate the induced crosstalk when the corresponding MBFF is placed at the grid based on the coupling capacitance map, as seen in Fig. 10. An MBFF is generated for $c_i$ and placed at the grid if both placement density and crosstalk constraints are satisfied. In addition, the coupling capacitance map should be updated for the newly placed MBFF accordingly. If no valid placement grid can be found in preferred region, we then find grids in the feasible region for newly MBFF (see Lines 18–28). Although placing an MBFF out of its preferred region may increase wirelength, it can avoid declustering MBFFs to save clock power. On the contrary, if no valid placement grid can be found in both the preferred and feasible regions, $c_i$ is then declustered. (see Lines 29–31). After all the MBFFs are placed, we may continue with crosstalk-aware flip-flop clustering for those discarded flip-flop clusters until no more MBFF can be applied. Based on our crosstalk-aware MBFF placement algorithm, both wirelength and crosstalk can be minimized.

The overall time complexity of the MBFF placement algorithm in Algorithm 2, is $O(n)$. We have at most $n/m$ flip-flip clusters (i.e. $n/m$ $m$-bit flip-flips). It takes constant time to compute the weight of all grids because the numbers of grids in the preferred region

Please cite this article as: C.-C. Hsu, et al., Crosstalk-aware multi-bit flip-flop generation for power optimization, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2014.08.002

ARTICLE IN PRESS



freepapers.ir FREE papers

are small (Lines 4–7). When searching a placement grid for a flip-flip cluster, it takes $O(m)$ to estimate the crosstalk induced by the newly generated $m$-bit flip-flop, to create a new MBFF for the flip-flip cluster, and also to update the netlist (Lines 8–17). If no valid placement grid can be found in the preferred region, it takes constant time to find a valid placement grid in the common feasible region, and then it also takes $O(m)$ to estimate the crosstalk induced by the newly generated $m$-bit flip-flop, to create a new MBFF for the flip-flip cluster, and also to update the netlist (Lines 18–28). If no valid placement grid can be found in the common feasible region for the newly generated MBFF, it takes $O(m)$ to decluster the flip-flop cluster (Lines 29–31).

## 5. Experimental results

We implemented our algorithms in the C++ programming language, and performed on a 2.26 GHz Intel Xeon machine under the Linux operating system. We conducted three experiments to show the effectiveness of our approach for crosstalk avoidance. The first experiment compares our improved algorithms in this work with our preliminary version [1]. The second experiment compares our approach with the approaches in [11,13,17], which do not consider the crosstalk constraint, and the last one compares ours with [19], which considers similar objectives and constraints toward the routing problem.

### 5.1. Comparisons with our preliminary version [1]

We empirically tested our approach on six industrial circuits with the MBFF library, as seen in Tables 1 and 2. We compared power ratios, wirelength ratios, average crosstalk ratios, and CPU times for three different approaches, including our preliminary vision [1], and our approach without and with performing crosstalk-aware MBFF placement, as shown in Table 4. Table 4 shows the comparisons of the power ratios ("Power Ratio"), wirelength ratios ("WL Ratio"), average crosstalk ratios ("Xtalk Ratio"), and CPU times ("Time (s)") for the three approaches.

According to the results in Table 4, the power consumption and interconnecting wirelength based on our approach with performing crosstalk-aware MBFF placement are comparable to those based on the other two approaches. For average crosstalk ratios, our approach with performing crosstalk-aware MBFF placement is 6% and 19% better than our preliminary vision [1] and our approach without performing crosstalk-aware MBFF placement, respectively. Consequently, our preliminary vision in [1] has been further improved in this work.

### 5.2. Comparisons with Chang et al.'s [11], Jiang et al.'s [13], and Wang et al.'s [17] Approaches

In the second experiment, we compared the numbers of flip-flops, power ratios, wirelength ratios, average crosstalk ratios, and CPU times for four different approaches, including Chang et al.'s approach [11,12], INTEGRA [13], Wang et al.'s approach [17] and ours, as shown in Tables 5 and 6. Table 5 lists the names of the benchmark circuits ("Circuit"), the numbers of ("# of") 1-bit, 2-bit, and 4-bit flip-flops resulting from the four approaches. Compared with [11] and [17], the results of our approach contain 38–50% less 1-bit flip-flops, 3.53–4.13 × more 2-bit flip-flops, and 47–49% less 4-bit flip-flops on average. Compared with [13], the results of our approach contain 200 × more 1-bit flip-flop, 27 × more 2-bit flip-flops, and 57% less 4-bit flip-flops on average. In summary, our approach may adopt much less 4-bit flip-flops, much more 2-bit flip-flops because 4-bit flip-flops may introduce much more serious crosstalk than 1-bit and 2-bit flip-flops.

Table 6 compares the ratios between the corresponding data before and after applying the proposed algorithms, including the power ratios ("Power Ratio"), the wirelength ratios ("WL Ratio") based on the HPWL model, and the average crosstalk ratios ("Xtalk Ratio"), and the CPU times ("Time (s)") for the four approaches.

**Table 6**
Comparisons of power ratios, wirelength ratios, average crosstalk ratios, and CPU times for four different approaches, [11] (on Intel Core i7 2.66 GHz), [13] (on Intel Xeon 3.8 GHz), [17] (on Intel Xeon 2.40 GHz), and ours (on Intel Xeon 2.26 GHz).

| Circuit | Chang et al., ICCAD'10 [11] | | | | Jiang et al., TCAD'12 [13] | | | |
|---|---|---|---|---|---|---|---|---|
| | Power ratio | WL ratio | Xtalk ratio | Time (s) | Power ratio | WL ratio | Xtalk ratio | Time (s) |
| c1 | 0.852 | 0.917 | 4.12 | 0.01 | 0.828 | 0.964 | 6.146 | 0.01 |
| c2 | 0.831 | 0.947 | 3.229 | 0.04 | 0.809 | 1.020 | 3.730 | 0.01 |
| c3 | 0.829 | 0.948 | 3.338 | 0.10 | 0.808 | 1.036 | 4.169 | 0.01 |
| c4 | 0.832 | 0.945 | 3.420 | 0.28 | 0.810 | 1.041 | 4.349 | 0.02 |
| c5 | 0.829 | 0.949 | 3.430 | 0.60 | 0.807 | 1.048 | 4.416 | 0.05 |
| c6 | 0.828 | 0.949 | 3.094 | 78.92 | 0.807 | 1.053 | 4.552 | 1.11 |
| Comp. | 0.97 | 1.02 | 1.62 | 0.07 | 0.94 | 1.11 | 2.13 | 0.01 |
| | Wang et al., TCAD'12 [17] | | | | Ours | | | |
| c1 | 0.844 | 0.899 | 4.935 | 0.02 | 0.877 | 0.929 | 2.815 | 0.16 |
| c2 | 0.827 | 0.844 | 2.409 | 0.06 | 0.859 | 0.915 | 1.886 | 0.81 |
| c3 | 0.826 | 0.853 | 2.555 | 0.29 | 0.858 | 0.921 | 1.986 | 2.92 |
| c4 | 0.829 | 0.859 | 2.365 | 0.92 | 0.860 | 0.924 | 2.094 | 9.028 |
| c5 | 0.825 | 0.855 | 2.337 | 1.67 | 0.857 | 0.925 | 2.014 | 19.05 |
| c6 | 0.825 | 0.856 | 2.360 | 24.92 | 0.857 | 0.926 | 2.052 | 336.23 |
| Comp. | 0.96 | 0.93 | 1.29 | 0.09 | 1 | 1 | 1 | 1 |

**Table 5**
Comparisons of the numbers of 1-bit, 2-bit, and 4-bit flip-flops for four different approaches, [11,13,17], and ours.

| Circuit | Chang et al., ICCAD'10 [11] | | | Jiang et al., TCAD'12 [13] | | | Wang et al., TCAD'12 [17] | | | Ours | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | # of 1-bit FFs | # of 2-bit FFs | # of 4-bit FFs | # of 1-bit FFs | # of 2-bit FFs | # of 4-bit FFs | # of 1-bit FFs | # of 2-bit FFs | # of 4-bit FFs | # of 1-bit FFs | # of 2-bit FFs | # of 4-bit FFs |
| c1 | 8 | 10 | 23 | 0 | 4 | 28 | 6 | 7 | 25 | 6 | 31 | 13 |
| c2 | 24 | 36 | 96 | 0 | 6 | 117 | 18 | 35 | 98 | 12 | 134 | 50 |
| c3 | 84 | 146 | 386 | 0 | 14 | 473 | 68 | 124 | 401 | 40 | 536 | 202 |
| c4 | 242 | 469 | 1175 | 2 | 21 | 1459 | 204 | 424 | 1207 | 118 | 1649 | 616 |
| c5 | 480 | 920 | 2420 | 2 | 33 | 2983 | 388 | 804 | 2501 | 220 | 3350 | 1270 |
| c6 | 7320 | 14,780 | 38,780 | 18 | 113 | 47,939 | 6324 | 12,798 | 40,020 | 3280 | 53,600 | 20,380 |
| Comp. | 1.98 | 0.28 | 1.89 | 0.01 | 0.04 | 2.32 | 1.60 | 0.24 | 1.96 | 1 | 1 | 1 |

Please cite this article as: C.-C. Hsu, et al., Crosstalk-aware multi-bit flip-flop generation for power optimization, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2014.08.002

ARTICLE IN PRESS



freepapers.ir FREE papers

**Table 7**
The benchmark circuits in [19].

| Circuit | FF Number (1-bit FF) | Normalized FF power | Original FF area |
|---|---|---|---|
| Ex01 | 120 | 120 | 12,000 |
| Ex02 | 120 | 120 | 12,000 |
| Ex03 | 800 | 800 | 80,000 |
| Ex04 | 900 | 900 | 90,000 |
| Ex05 | 1000 | 1000 | 100,000 |

**Table 8**
Comparisons of power ratios, wirelength ratios, average crosstalk ratios, and CPU times for Chen & Yan's approach [19] and ours.

| Circuit | Chen and Yan's approach [19] | | | | Ours | | | |
|---|---|---|---|---|---|---|---|---|
| | Power ratio | WL ratio | Xtalk ratio | Time (s) | Power ratio | WL ratio | Xtalk ratio | Time (s) |
| Ex01 | 0.778 | 1.239 | 21.191 | 0.05 | 0.832 | 0.835 | 2.934 | 0.16 |
| Ex02 | 0.777 | 1.229 | 14.464 | 0.03 | 0.827 | 0.890 | 2.186 | 0.13 |
| Ex03 | 0.755 | 0.803 | 1.171 | 1.99 | 0.760 | 0.616 | 0.488 | 17.09 |
| Ex04 | 0.753 | 0.792 | 1.373 | 2.44 | 0.758 | 0.585 | 0.508 | 22.06 |
| Ex05 | 0.755 | 0.851 | 1.190 | 3.47 | 0.770 | 0.707 | 0.686 | 48.57 |
| Comp. | 0.97 | 1.35 | 4.14 | 0.17 | 1 | 1 | 1 | 1 |

According to the results in Table 6, the power consumption based on our approach is comparable to those based on all the previous work [11,13,17]. The interconnecting wirelength based on our approach is 2%–10% shorter than that based on [11,13] and comparable to those based on [17]. It should be noted that our approach leads to significant crosstalk reduction compared with the previous works. The average net crosstalk based on our approach is 38%, 53%, and 23% less than those based on [11,13,17], respectively. The CPU time based on our approach is 10.64–71.43 × longer than that based on [11,13,17]. The reason is that we additionally performed global routing and constructed/updated the coupling capacitance map for crosstalk minimization and avoidance, which had been known to be one of the most time-consuming parts in physical synthesis. Consequently, our approach is very effective in crosstalk avoidance for power optimization with MBFFs.

### 5.3. Comparisons with [Chen and Yan 2012]

We further compare the effectiveness of crosstalk avoidance of our approach and Chen and Yan's approach [19], which considers the routability constraint based on their benchmark circuits, as shown in Table 7. Table 8 compares the ratios between the corresponding data before and after applying the proposed algorithms, including the power ratios ("Power Ratio"), wirelength ratios ("WL Ratio"), average crosstalk ratios ("Xtalk Ratio"), and CPU times ("Time (s)") for Chen and Yan's approach [19] and ours. The results show that the power ratios based on our approach are comparable to those based on Chen and Yan's approach [19]. It should be noted that our approach leads to 34% wirelength and significant average crosstalk reduction in all benchmark circuits. The reason is because Chen and Yan's approach [19] merely considers routability constraint during MBFF placement, and the routability constraint cannot reflect the severeness of the crosstalk effect. The CPU time based on our approach is 5.88 × longer than that based on [19]. The reason is that Chen and Yan [19] estimated routability by simply computing the number of nets goes across the bin edge, while we performed global routing and constructed/updated the coupling capacitance map for crosstalk avoidance and wirelength minimization. Consequently, our approach, which considers the crosstalk constraint during both flip-flop clustering and MBFF placement, is very effective in crosstalk avoidance.

## 6. Conclusions

In this paper, we have addressed the crosstalk effect when applying MBFFs for clock power saving. We have observed the ineffectiveness of applying MBFFs with larger number of bits due to serious crosstalk based on the crosstalk evaluation. We have also proposed a new problem formulation, which additionally considers the crosstalk constraint when applying MBFFs for clock power saving. To solve the addressed problem, we introduced the coupling capacitance map and presented novel algorithms to effectively handle the crosstalk constraint during both flip-flop merging and MBFF placement. Experimental results have shown that our approach is very effective in crosstalk avoidance for power optimization with MBFFs.

## References


[1] C.-C. Hsu, Y.-T. Chang, M.P.-H. Lin, Crosstalk-aware power optimization with multi-bit flip-flops, in: Proceedings of IEEE/ACM Asia South Pacific Design Automation Conference, 2012, pp. 431–436.
[2] A. Krishnamoorthy, Minimize IC power without sacrificing performance, EE Times.
[3] D. Liu, C. Svensson, Power consumption estimation in CMOS VLSI chips, IEEE J. Solid-State Circuits 29 (6) (1994) 663–670.
[4] J.G. Xi, W.-M. Dai, Buffer insertion and sizing under process variations for low power clock distribution, in: Proceedings of ACM/IEEE Design Automation Conference, 1995, pp. 491–496.
[5] K. Wang, M. Marek-Sadowska, Buffer sizing for clock power minimization subject to general skew constraints, in: Proceedings of ACM/IEEE Design Automation Conference, 2004, pp. 159–164.
[6] M. Donno, A. Ivaldi, L. Benini, E. Macii, Clock-tree power optimization based on RTL clock-gating, in: Proceedings of ACM/IEEE Design Automation Conference, 2003, pp. 622–627.
[7] F. Emnett, M. Biegel, Power reduction through RTL clock gating, in: Proceedings of Synopsys Users Group, 2000.
[8] Y. Cheon, P.-H. Ho, A.B. Kahng, S. Reda, Q. Wang, Power-aware placement, in: Proceedings of ACM/IEEE Design Automation Conference, ACM, New York, NY, USA, 2005, pp. 795–800.
[9] Y. Lu, C.N. Sze, X. Hong, Q. Zhou, Y. Cai, L. Huang, J. Hu, Navigating registers in placement for clock network minimization, in: Proceedings of ACM/IEEE Design Automation Conference, ACM, New York, NY, USA, 2005, pp. 176–181.
[10] W. Hou, D. Liu, P.-H. Ho, Automatic register banking for low-power clock trees, in: Proceedings of IEEE/ACM International Symposium on Quality of Electronic Design, IEEE Computer Society, Washington, DC, USA, 2009, pp. 647–652.
[11] Y.-T. Chang, C.-C. Hsu, M.P.-H. Lin, Y.-W. Tsai, S.-F. Chen, Post-placement power optimization with multi-bit flip-flops, in: Proceedings of IEEE/ACM International Conference on Computer-Aided Design, 2010, pp. 218–223.
[12] M.P.-H. Lin, C.-C. Hsu, Y.-T. Chang, Post-placement power optimization with multi-bit flip-flops, IEEE Trans. Comput-Aided Des. 30 (12) (2011) 1870–1882.
[13] I.H.-R. Jiang, C.-L. Chang, Y.-M. Yang, INTEGRA: fast multi-bit flip-flop clustering for clock power saving, IEEE Trans. Comput.-Aided Des. 31 (2) (2012) 192–204.
[14] Y. Kretchmer, Using multibit register inference to save area and power, EE Times Asia.
[15] R. Pokala, R. Feretich, R. McGuffin, Physical synthesis for performance optimization, in: Proceedings of IEEE International ASIC Conference and Exhibit, 1992, pp. 34–37.
[16] L. Chen, A. Hung, H.-M. Chen, E. Tsai, S.-H. Chen, M.-H. Ku, C.-C. Chen, Using multi-bit flip-flop for clock power saving by Design Compiler, Proceedings of Synopsys Users Group.
[17] S.-H. Wang, Y.-Y. Liang, T.-Y. Kuo, W.-K. Mak, Power-driven flip-flop merging and relocation, IEEE Trans. Comput.-Aided Des. 31 (2) (2012) 180–191.
[18] J.-T. Yan, Z.-W. Chen, Construction of constrained multi-bit flip-flops for clock power reduction, in: IEEE International Conference on Green Circuits and Systems, 2010, pp. 675–678.
[19] Z.-W. Chen, J.-T. Yan, Routability-constrained multi-bit flip-flop construction for clock power reduction, Integr., VLSI J.
[20] C.-C. Hsu, Y.-C. Chen, M.P.-H. Lin, In-placement clock-tree aware multi-bit flip-flop generation for power optimization, in: Proceedings of IEEE/ACM International Conference on Computer-Aided Design, 2013.
[21] C.-C. Tsai, Y. Shi, G. Luo, I.H.-R. Jiang, Ff-bond: multi-bit flip-flop bonding at placement, in: Proceedings of ACM International Symposium on Physical Design, 2013, pp. 147–153.


Please cite this article as: C.-C. Hsu, et al., Crosstalk-aware multi-bit flip-flop generation for power optimization, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2014.08.002

ARTICLE IN PRESS



[22] X. Bai, S. Dey, High-level crosstalk defect simulation methodology for system-on-chip interconnects, IEEE Trans. Comput.-Aided Des. 23 (9) (2004) 1355–1361.
[23] H. Zhou, D. Wong, Global routing with crosstalk constraints, IEEE Trans. Comput.-Aided Des. 18 (11) (1999) 1683–1688.
[24] F.-Y. Fan, H.-M. Chen, I.-M. Liu, Technology mapping with crosstalk noise avoidance, in: Proceedings of IEEE/ACM Asia South Pacific Design Automation Conference, 2010, pp. 319–324.
[25] A. Kumar, K. Chakrabarty, C.R. Mohan, An ECO technique for removing crosstalk violations in clock networks, in: Proceedings of IEEE VLSI Design, 2007, pp. 283–288.
[26] Q. Zhao, J. Hu, Track assignment considering crosstalk-induced performance degradation, in: Proceedings of IEEE International Conference on Computer Design, 2012, pp. 506–507.
[27] Y.-L. Li, Y.-N. Chang, W.-N. Cheng, A gridless routing system with nonslicing floorplanning-based crosstalk reduction on gridless track assignment, ACM Trans. Des. Autom. Electron. Syst. 16 (2) (2011) 19.
[28] Y.-N. Chang, Y.-L. Li, W.-T. Lin, W.-N. Cheng, Non-slicing floorplanning-based crosstalk reduction on gridless track assignment for a gridless routing system with fast pseudo-tile extraction, in: Proceedings of ACM International Symposium on Physical Design, 2008, pp. 134–141.
[29] X. Gao, L. Macchiarlo, Track routing optimizing timing and yield, in: Proceedings of IEEE/ACM Asia South Pacific Design Automation Conference, 2011, pp. 627–632.
[30] Z. Qi, Q. Zhou, Y. Jia, Y. Cai, Z. Li, H. Yao, A novel fine-grain track routing approach for routability and crosstalk optimization, in: Proceedings of IEEE/ACM International Symposium on Quality of Electronic Design, 2011, pp. 621–626.
[31] T. Sakurai, K. Tamaru, Simple formulas for two- and three-dimensional capacitance, IEEE Trans. Electron Devices 30 (2) (1983) 183–185.
[32] R. Kastner, E. Bozorgzadeh, M. Sarrafzadeh, An exact algorithm for coupling-free routing, in: Proceedings of ACM International Symposium on Physical Design, 2001, pp. 10–15.
[33] R. Kastner, E. Bozorgzadeh, M. Sarrafzadeh, Pattern routing: Use and theory for increasing predictability and avoiding coupling, IEEE Trans. Comput.-Aided Des. 21 (7) (2002) 777–790.
[34] J. Lou, W. Chen, Cross talk driven placement, in: Proceedings of IEEE/ACM Asia South Pacific Design Automation Conference, 2003, pp. 73–740.
[35] J. Lou, W. Chen, Crosstalk-aware placement, IEEE Des. Test. Comput. 21 (1) (2004) 24–32.
[36] H. Ren, D. Pan, P. Villarubia, True crosstalk aware incremental placement with noise map, in: Proceedings of IEEE/ACM International Conference on Computer-Aided Design, 2004, pp. 402–409.
[37] Z. Cao, T. Jing, J. Xiong, Y. Hu, L. He, X. Hong, DpRouter: A fast and accurate dynamic-pattern-based global routing algorithm, in: Proceedings of IEEE/ACM Asia South Pacific Design Automation Conference, 2007, pp. 256–261.
[38] T.-H. Chao, Y.-C. Hsu, J.-M. Ho, A. Kahng, Zero skew clock routing with minimum wirelength, IEEE Trans. Circuits Syst. II: Analog Digit. Signal Process. 39 (11) (1992) 799–814.

**Chih-Cheng** Hsu received the B.S. degree in Electronics Engineering from the Fu Jen Catholic University, Taipei, Taiwan, in 2009. Since 2009, he has been working toward the Ph.D. degree with the EDA Laboratory, Department of Electrical Engineering, National Chung Cheng University, Chiayi, Taiwan. His current research interests include low power design optimization and physical design automation.

**Mark Po-Hung Lin** received the B.S. and M.S. degrees in Electronics Engineering from National Chiao Tung University (NCTU), Hsinchu, Taiwan, respectively, and the Ph.D. degree in the Graduate Institute of Electronics Engineering, National Taiwan University (NTU), Taipei, Taiwan. He has been with the Department of Electrical Engineering, National Chung Cheng University, Chiayi, Taiwan, since 2009, where he is currently an Associate Professor. He was with SpringSoft, Inc. (now Synopsys, Inc.) during 2000–2007 as an Engineer, a Senior Engineer, an Associate Manager, and a Technical Manager. He was a Visiting Scholar with the University of Illinois at Urbana-Champaign, Champaign, IL, USA, during 2007–2009, and a Humboldt Research Fellow with the Technical University of Munich (TUM), Germany, during 2013–2014.

Since 2007, he has published over 40 technical papers and patents, including six IEEE TCAD papers, six U.S. patents, four DAC papers, and five ICCAD papers. His current research interests include design automation for analog/mixed-signal integrated circuits and low-power circuit and system design optimization.

Dr. Lin was the recipient of IEEE Tainan Section Macronix Award, Humboldt Research Fellowship for Experienced Researchers, Outstanding Young Scholar Award of Taiwan IC Design Society, and Outstanding Young Faculty Award of National Chung Cheng University.

**Yao-Tsung Chang** received the B.S. degree in Electronics Engineering from the Fu Jen Catholic University (FJU), Taipei, Taiwan, in 2006, and the M.S. degree in Electronics Engineering from the National Chung Cheng University (CCU), Chiayi, Taiwan, in 2011. His research interest lies in electronic design automation, with an emphasis on low power design. He worked for SpringSoft from 2011 to 2012, and joined Synopsys as an R&D engineer in 2012.

freepapers.ir FREE papers دانلود کننده مقالات علمی

Please cite this article as: C.-C. Hsu, et al., Crosstalk-aware multi-bit flip-flop generation for power optimization, INTEGRATION, the VLSI journal (2014), http://dx.doi.org/10.1016/j.vlsi.2014.08.002