



Loosely coupled multi-bit flip-flop allocation for power reduction



Hyungseok Moon, Taewhan Kim*

Department of Electrical and Computer Engineering, Seoul National University, Korea

ARTICLE INFO

Keywords:

Flip-flop
Multi-bit
Power
Clock polarity assignment
Power/ground noise

ABSTRACT

Merging 1-bit flip-flops into multi-bit flip-flops in the post-placement stage is one of the most effective techniques for minimizing clock power. In this work, we introduce a new style of multi-bit flip-flop, called loosely coupled multi-bit flip-flop (LC-MBFF). The merit of LC-MBFF is that the logically constituent 1-bit flip-flops in LC-MBFF can be physically apart (i.e., no relocation), providing no need to set aside white space. Utilizing LC-MBFFs, we propose a multi-bit flip-flop allocation algorithm which fully explores the diverse allocation of LC-MBFF structures to maximally reduce clock power consumption. Experimental results with ISCAS89 and IWLS2005 benchmark circuits show that our proposed allocation algorithm using the newly designed multi-bit flip-flops is able to reduce on average the clock power by 8.51% while the best known multi-bit flip-flop allocation algorithm [7] reduces by 5.37%. Additionally, we extend our algorithm to support the multi-bit flip-flop allocation for circuits with clock polarity assignment.

1. Introduction

Minimizing power consumption is an utmost important task in modern VLSI system design. Since a dominating portion of power in high-speed synchronous systems is contributed by the clock network including storages (e.g., flip-flops, latches), a considerable research effort has been made in conjunction with clock networks to reduce power consumption.

Recently, allocating multi-bit flip-flops (MBFFs) as opposed to 1-bit flip-flops has been recognized as one of effective design optimization techniques to reduce clock power. The key idea in designing an MBFF is that only two inverters are allocated in the MBFF to drive all master and slave latches in the MBFF, rather than allocating two distinct inverters to each pair of master and slave latches. Fig. 1(b) shows the structure of 2-bit MBFF which plays the same role of the two 1-bit flip-flops in Fig. 1(a), but driving inverters are shared. The rationale of adopting this MBFF structure is supported by the following fact: As the CMOS process technology scales deeply, the driving capability of logic devices such as inverter-based buffers has increased significantly. For instance, a minimum-sized inverter of 90 nm process technology can drive almost twice as many gates as that of 0.18 μm process technology can. As many system-on-chip (SoC) designs for mobile devices are manufactured using submicron technology nowadays, more driving capability of logic cells is available [8], which means not only 2-bit MBFFs but also 4 or more-bit MBFFs can replace 1-bit flip-flops.

After Kretchmer [1] firstly proposed to map MBFFs to a bundle of flip-flops in RTL synthesis stage, several approaches have proposed the

allocation of MBFFs in either the in-placement or post-placement stage. It has been noted that addressing MBFF allocation in the pre-placement stage (i.e., RTL and logic synthesis) makes it very hard to reliably estimate its effect on the changes of timing, power, and placement. For this reason, most works have focused on the MBFF allocation problem in the in-placement or post-placement stage. Yan et al. [2] formulated the MBFF allocation problem with the consideration of maximizing the wire length reduction by MBFFs while avoiding congestion. Precisely, they defined a timing *feasible region* for each flip-flop by analyzing the connection locations of the input and output ports of the flip-flop. Thus, if the feasible regions of two 1-bit flip-flops have a common area, an MBFF can be allocated to the area, replacing the flip-flops. However, if the common area is too small to accommodate an MBFF, the MBFF allocation is given up. They constructed a *merging graph*, in which nodes represent flip-flops and edges represent overlapping relation between the feasible regions corresponding to the two terminal nodes of the edge. They select a pair of nodes which are connected in the merging graph with the least connectivity with the rest of nodes. They merge the selected nodes and allocate an MBFF for those. Then, they update the merging graph accordingly and repeat the merging process until there is no edge on the graph. The idea of the feasible region of flip-flop has been adopted by a number of subsequent approaches of MBFF allocation. Chang et al. [3] proposed a window-based approach which controlled the searching region and size of mergeable flip-flops by sliding and expanding a window. Jiang, Chang, and Yang [4] applied a technique of coordinate transformation which transforms the original diamond-shapes of the intersections of feasible

* Corresponding author.

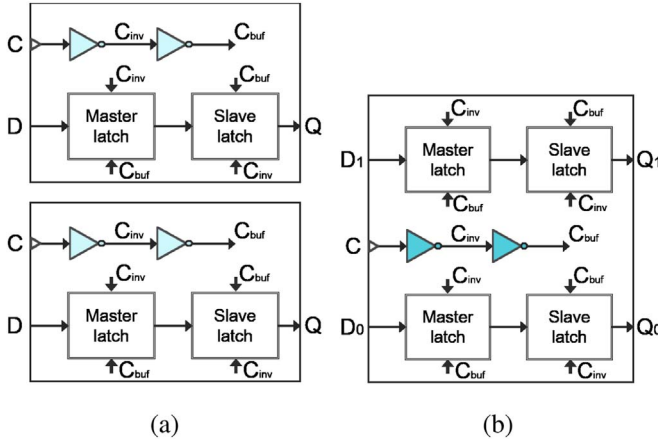


Fig. 1. Comparison of the internal structures of the 1-bit master-slave based flip-flop and the 2-bit master-slave flip-flop.

regions into rectangular forms to speed up the flip-flop clustering. Wang et al. [5] focused on minimizing the clock network size and thus the clock power consumption by selectively merging flip-flops based on their activity patterns and preferred locations. Hsu, Chen, and Lin [6] also took into account the potential simplification of resulting clock tree during the MBFF allocation. In addition, after the MBFF allocation, they performed a timing driven incremental placement to meet the timing and placement constraints and legalize the placement. On the other hand, Chen and Yan [7] constrained the signal routability on partitioned bins in the course of MBFF placement. Shyu et al. [8] suggested to use a combination table to enumerate possible combinations of flip-flops provided by a library. By listing and limiting the combinations of flip-flops, they attempted to reduce the computation time. Lu and Lin [22] considered delay and slack of paths among the flip-flops. By translating timing slack to equivalent wire length, they tried to allocate MBFFs in a way to minimize the impact of resulting timing change on the system's timing vulnerability.

The three constraints vested in all previous multi-bit flip-flop allocation works in [2–8,22] are (1) *the input and output timing constraint on every flip-flop*, (2) *the area constraint on every partitioned bin in the placement plane*, and (3) *the routing capacity constraint on every bin edge*. Consequently, the tighter the timing, available space, and routing are, the less MBFF allocation is, inhibiting the full exploitation of MBFFs to reduce power consumption. In this work, we propose a *new style of multi-bit flip-flop*, called *loosely coupled multi-bit flip-flop* (LC-MBFF) for the first time to mitigate *constraint-1* and eliminate *constraint-2* in the MBFF allocation so that the benefit of MBFF allocation can be fully acquired (A preliminary version of this work was presented in [21]). The strength of LC-MBFF is that the logically constituent 1-bit flip-flops in LC-MBFF can be physically apart, i.e., the 1-bit flip-flops can stay with no relocation, providing no need to prepare empty space for *constraint-2* and less strict timing checking for *constraint-1*. Utilizing LC-MBFFs, we propose a new multi-bit allocation algorithm to maximally reduce the clock power. In addition, the structure of LC-MBFF has the advantage that can handle merging flip-flops with different (negative, positive) clock polarities. Thus, our MBFF allocation algorithm can fully apply to the circuits with clock polarity assignment while the application of conventional MBFF allocation algorithms is limited only to the flip-flops with identical clock polarity. The contributions of this work are summarized as:

1. We introduce a new structure of multi-bit flip-flops, called *loosely coupled multi-bit flip-flop* (LC-MBFF) for the first time. The use of LC-MBFFs as opposed to the conventional MBFF facilitates less strict timing checking and removal of area constraint, which should be met whenever a new MBFF is allocated in prior works. Thus, the

multi-bit flip-flop allocation can be vitalized, opening up more opportunity to reduce power consumption. We also provide a set of analyses to support the practical feasibility of the use of LC-MBFFs.

2. The power saving by the use of MBFFs is contributed by two factors. One is the MBFF itself and the other is the clock tree. As the number of sinks (flip-flops) that should be driven by clock decreases, the driving clock tree will be simplified accordingly. This work carefully treats the clock tree when allocating MBFFs in the post-placement stage, instead of completely resynthesizing the clock tree during the placement as done in [5].
3. We propose an allocation algorithm to maximally utilize LC-MBFFs. The algorithm iteratively selects a pair of flip-flops at the same clock tree level, which can be merged with the most power saving under the given routability constraint, and replace them with an LC-MBFF. Unlike the conventional MBFF merging algorithms which perform MBFF allocation, clock tree generation, and routing of flip-flop connections in a sequence of separate steps, ours simultaneously performs them.
4. We show that the application of LC-MBFF allocation can be extended to the circuits that have processed clock polarity assignment to mitigate the power/ground noise (i.e., peak current). With a slight modification of the connection to LC-MBFF, it is possible to support LC-MBFF allocation for circuits with clock polarity assignment.
5. A set of experiments using benchmark circuits is performed to check the effectiveness of the proposed algorithms utilizing LC-MBFFs. In summary, ours outperforms the best known post-placement MBFF allocation algorithm [7], saving 3.14% more clock power on average.

The rest of the paper is organized as follows. The limitations of the conventional MBFF allocation are described in Section 2, followed by proposing a new style of MBFF that is able to overcome the limitations in Section 3. Then, in Section 4 we propose an MBFF allocation algorithm that uses the newly designed MBFFs. An extension of our algorithm to support the circuits with clock polarity assignment is described in Section 5. A set of experimental results to show the effectiveness of our proposed allocation algorithms is presented in Section 6. Finally, a conclusion of the work is given in Section 7.

2. Illustration of limited applicability of MBFF allocation

In general, designers want to keep the logical and physical designs as much as they are during the post-placement optimization since the designs have already been optimized according to a certain set of design rules and constraints. This means that only a minimal design perturbation is permitted in post-placement optimization. In this context, the task of MBFF allocation is not an exception. The MBFF allocation problem in post-placement entails the satisfaction of three stringent constraints: timing, area, and routing.

Using an example, we illustrate how the constraints adversely affect the applicability of the conventional MBFF allocation and how the limitation can be overcome by the exploitation of LC-MBFFs. Fig. 2(a) shows an input circuit design for MBFF allocation in post-placement. The design has 15 1-bit flip-flops f_1 through f_{15} placed on 16 partitioned bins B_1 through B_{16} where the number in the top-left corner on each bin indicates the area slack normalized to the area of 1-bit flip-flop, and the number on each bin edge indicates the routing slack in terms of the number of nets to cross the edge. Fig. 2(b) shows the MBFF allocation result produced by the application of the flip-flop merging algorithm in [7]. The MBFF allocation replaces only three pairs of 1-bit flip-flops with MBFFs. This inefficient allocation of MBFFs is caused by the lack of empty space (as marked with red numbers) and tight timing constraints. We performed an experiment to see how many cells (in terms of unit area) undergo replacement by the MBFF allocation. The area portion of the logic cells relocated is listed in

Download English Version:

<https://daneshyari.com/en/article/4970669>

Download Persian Version:

<https://daneshyari.com/article/4970669>

[Daneshyari.com](https://daneshyari.com)