ISSN 2348-375X

Unique Journal of Engineering and Advanced Sciences

Available online: www.ujconline.net

Research Article

OPTIMIZING POWER CONSUMPTION BY USING MULTI-BIT FLIP-FLOPS

Santhoshkumar R[1*], Vijay A[2], Prasanna Venkatesan GKD[3]

[1]ME VLSI student PGP college of Engineering and Technology Nammakkal,Tamilnadu, India
[2]AP, PGP college of Engineering and Technology Nammakkal, Tamilnadu, India
[3]Vice Principal, Professor and Head/ECE, PGP College Of Engineering and Technology, Namakkal. Tamilnadu, India

Received: 08-03-2014; Revised: 06-04-2014; Accepted: 05-05-2014

*Corresponding Author: **Santhoshkumar R**
ME VLSI student PGP college of Engineering and Technology Nammakkal,Tamilnadu, India E-mail: rsanthoshkumaran@gmail.com

ABSTRACT

Clocking is the major part for the power consumption in the VLSI circuit. We can reduced its power consumption by replacing the flip-flops. A difficult problem may occurs in the replacement of flip-flops without timing and placement constraints. A combination table can store flip-flops and it can be merged in the proposed work. Combination table is build to enumerate possible combinations of flip-flops provided by a library. To merge flip-flops by using a hierarchical way. According to simulation results clock power is reduced by 19-28%.

Keywords: Clock Power Reduction, Merging, Multi-Bit Flip-Flop, Replacement, Wire Length.

INTRODUCTION

By using multi-bit flip-flops several flip-flops can replaced for reducing the power consumption. In less number of clock leads to low power consumption. Instead of smaller flipflops can be reduced by replacing larger multi-bit flip-flops. H. Kawagachi and T.sakurai (1997) proposed a techniques based on comparison between, Conventional Conditional Data Mapping Flip-Flop and Clock Pair shared D Flip-Flop (CPSFF)[6]. D.Duarte and V.Narayanan, (2002) proposed a processor to distribute the clock signal and generate the power in circuit to consume the half of the power like 30-40% in the operation[3]. Y.Cheon and A.B Kahng (2005) proposed the total net switching power, switching area and net wire length also consider the running time[2]. P.Gronowski and W.J.Bowhill (2008) proposed an important issue in modem high-frequency & low power design[4]. To reduced clock power multi-bit flip-flops are used. The scaling with multiple supply voltage is an effective way to minimize the dynamic power consumption.

Figure 1 shows the block diagrams of 1- and 2-bit flip-flops. The total power is reduced by replacing the 2 bit flip-flops with two 1-bit flip-flops since the two flip flops consume the same clock. However, the locations of some flip-flops would be changed after this replacement, and thus the wire-lengths of nets connecting pins to a flip-flop are also changed. To avoid violating the timing constraints, restriction of wire-length nets connecting pins to a flip-flop cannot be longer than specified values after this process. Besides, to guarantee that a new flip- flop can be placed within the desired region, we also need to consider the area capacity of the region. As shown in Figure 2(a), after the two 1-bit flip-flops f_1 and f_2 are replaced by the 2-bit flip-flop f_3, the wire-lengths of nets net_1, net_2, net_3, and net_4 are changed. To avoid the timing violation caused by the replacement, the Manhattan distance of new net's net_1,net_2, net_3, and net_4 cannot be longer than the specified values.

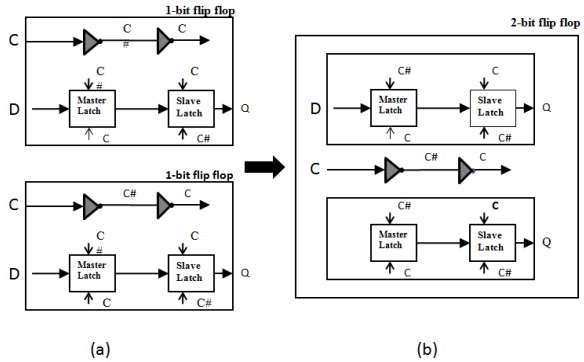

Figure 1(a) Two 1-bit flip-flops (before merging). (b) 2-bit flip-flop (after merging).

Figure 2(b), divide the whole placement region into several bins and each bin has an area capacity denoting the remaining area that additional cells can be placed within it.

Suppose the area of f_3 is 7 and f_3 is assigned to be placed

in the same bin as f_1, f_3 cannot be placed in that bin since the remaining area of the bin is smaller than the area of f_3. In addition to the considerations mentioned in the above, we also need to check whether the cell library provides the type of the new flip-flop. For example, the availability of a 3-bit flip-flop is checked in the cell library when we desire to replace 1- and 2-bit flip-flops by a 3-bit flip-flop.

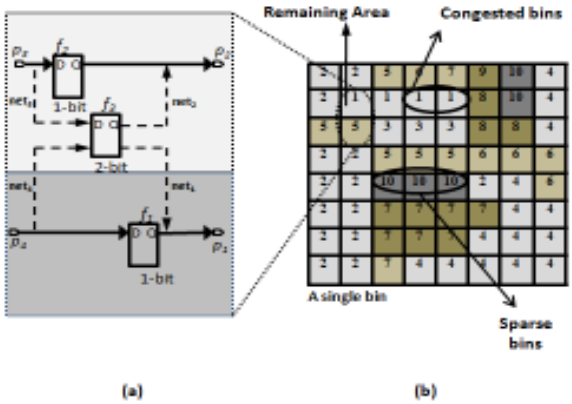

Figure 2(a) Combination of flip-flops possibly increases the wire length.(b) Combination of flip-flops also changes the density.

Our Algorithm

Our design flow can be roughly divided into three stages. Please see Figure 3 for our flow. In the beginning, we have to identify a legal placement region for each flip-flop f_i. First, the feasible placement region of a flip-flop associated with different pins is found based on the timing constraints defined on the pins. Then, the legal placement region of the flip-flop f_i can be obtained by the overlapped area of these regions. However, because these regions are in the diamond shape, it is not easy to identify the overlapped area. Therefore, the overlapped area can be identified more easily if we can transform the coordinate system of cells to get rectangular regions.

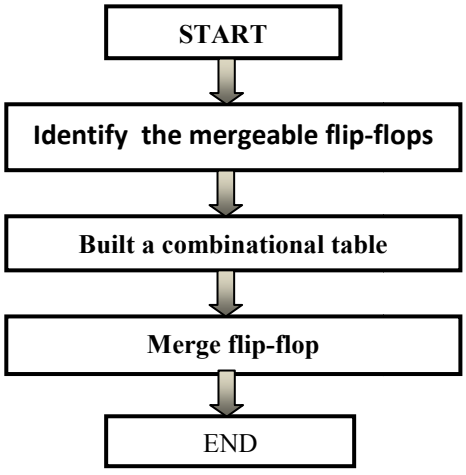

Figure 3: Flow chart of our algorithm

In the second stage, we would like to build a combination table, which defines all possible combinations of flip-flops in order to get a new multi-bit flip-flop provided by the library. The flip-flops can be merged with the help of the table. After the legal placement regions of flip-flops are found and the combination table is built, we can use them to merge flip-flops. To speed up our program, we will divide a chip into several bins and merge flip-flops in a local bin. However, the flip-flops in different bins may be merge-able. Thus, we have to combine several bins into a larger bin and repeat this step until no flip-flop can be merged anymore.

Build a Combination Table

To create a combination table, consider 1-bit and 4-bit in the library. Let n_1 be 1-bit and n_2 be 4-bit. Remaining 2-bit and 3-bit are considered as pseudo bits. Now we build a new combination n_3 by combining teo 1-bit flip-flops as shown in the figure 2.

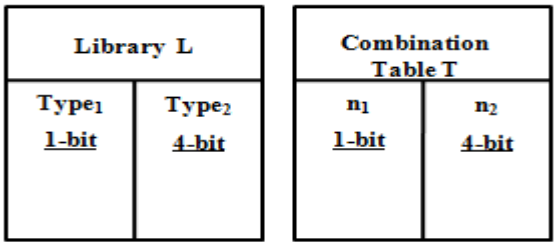

Figure 4: Initialize the library L and the combination table T.

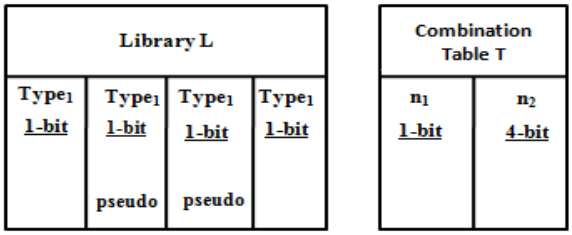

Figure 5: Pseudo types are added into L, and the corresponding binary tree is also build for each combination in T

Similarly we can get a new combination n4 by combining n3 and n1 as shown in figure 3. All the combinations are shown in figure 4. In a binary tree two single bits are used to form a 2-bit binary tree.

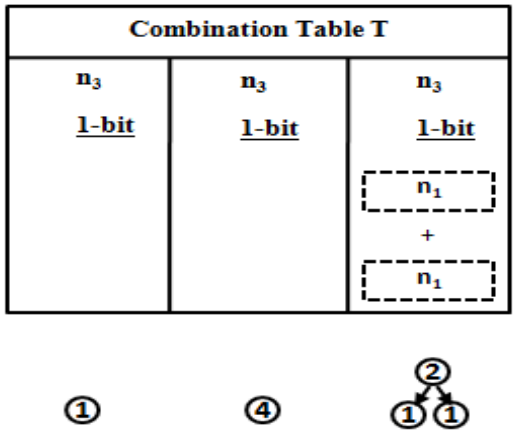

Figure 6: New combination n_3 is obtained from combining two n_1s

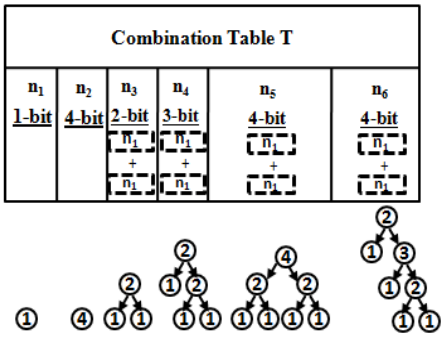

Figure 7: Combination Table

Similarly 1-bit and 2-bit are used to form a 3-bit binary tree. A 4-bit binary tree can be formed in two different ways; either by combining two n_3's or by combining n_1 and n_4 the height of the binary tree is larger, so we go for the combination of two n_3's. By avoiding unused combination like n_1 and n_4, we can speed the execution.

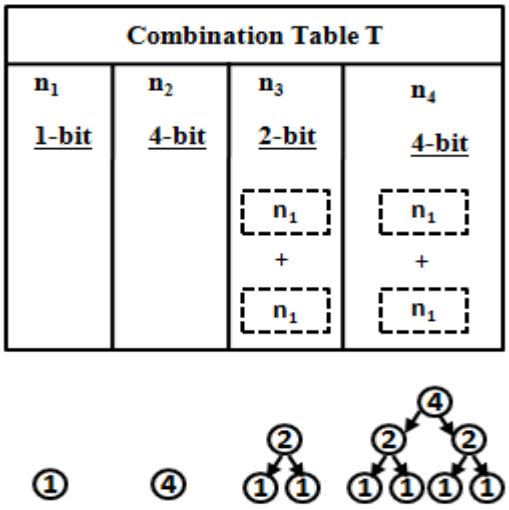

Figure 8: Last combination table is obtained after deleting the unused combination

Merge Flip-Flops

Here, with help of combination table to combine flip-flops. To reduce the complexity, the whole placement region into several sub-regions, Then with help of combination table to replace flip-flops in each sub-region. ,after that several sub-regions are combined into a larger sub-region ana vice versa, In final stage all flip-flops with pseudo types are deleted in the last stage because they are not provided by the supported library.

Region partition:

To rise our speed the whole chip is divided in to several sub regions. The computation complexity of merging flip-flops can be reduced ideally by suitable partition. From the above mentioned Figure, we divide the region into several sub regions, and each sub region contains 6 bins ,[bin is the smallest unit of a sub region].

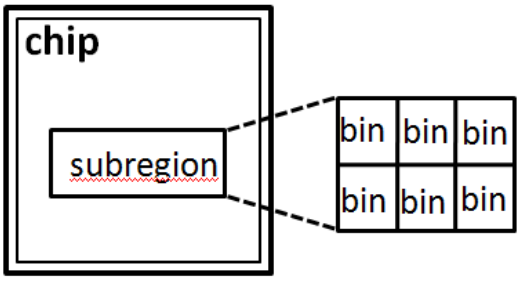

Figure 9: Example of region partition with six bins in one sub-region.

Replacement of flip-flops in each sub region:

The equation to measure the quality, if 2 flip-flops are replaced by new flip-flop is

Cost = routing length – α x √available area (5)

Where, routing length indicates total routing length between the new flip-flop and connected pins. α indicates weighting factor, available area indicates feasible region for placing the new flip-flop. In smaller cost we combine with other flip-flops and more power reduction. Once the flip-flops are not merged in to higher bit type [as the 4-bit combination n_4], Hence we set the α in to 0. After the builted combination we replace the flip-flops. Then link the flip-flops with help of combination related to their types. Then for each combination nin T and serially merge the flip- flops linked below the left child and right child of n from leaves to root. From the algorithm 3 we can get new flip-flop corresponding to n, Based on its binary tree we can find the combinations of left and right child of root. Thus the flip-flops in the from of lists namely l left and l right are linked below the combinations of left and right child. Then each flip-flop fbest in lright. That is the flip-flop is merged with f I with the smallest cost in c best .

Finally, we add and remove the flip-flop f in the list of combination n, flip-flops with various types are linked below n1,n2and n3 in T. If we want to form a flip-flop in n4, it needs two 1-bit flip-flops. Each pair of flip-flops in n1 are selected and check wether it can be combined or not. There are so many ways to the pair with smallest cost value is used to break the tie. In f1 and f2 are closed, its contains the combination gains the smallest cost . Then we add new node f3 in the list below n4 and delete f1and f2 from their original list. Like wise f4 and f5 combined to form a new flip-flop f6. Later all flip-flops in the combinations of I-level trees[n4 and n5] was generated then start to from II-level trees[n6 and n7]. After from the list of n2 and n4 some flip-flops are exist then we will merge them to get flip-flops in n6 and n7.If there is no any overlap between the couple of flip-flops in n2 and n4 then the 4-bit flip-flop are not form in n6,so the 2-bit flip-flops f3 and f6 are mergeable, we can combined to form 4 bit flip-flop f10 in n7,finally the procedure get finished when the no coule of flip-flops that can be combined future.

New flip-flop is obtained to replace those flip-flops from the available overlap region of two flip-flop. Once we place the new flip-flop in the available region, replacement will be performed smoothly and new flip-flop is placed in the grid that makes wire length between the flip-flop and connected

pins smallest. Suppose the capacity constraint of the bin, BK,the grid will be violated after the new flip-flop is placed on that grid. The bins near BK is used to find a new available grid for new flip-flop. Suppose no bins are overlapped with the available region of new flip-flop the capacity constraint is satisfy after the placement of new flip finally the satisfy after the placement of new flip-flop. Finally the program gets stopped the replacement of two flip-flops.

RESULTS AND DISCUSSIONS

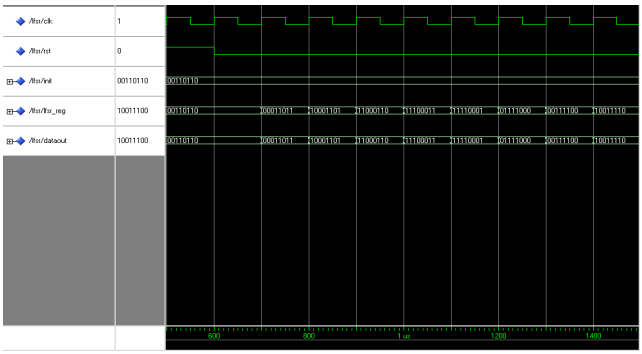

Figure 10: Simulation result

Figure 11: Power summary of existing flip-flop

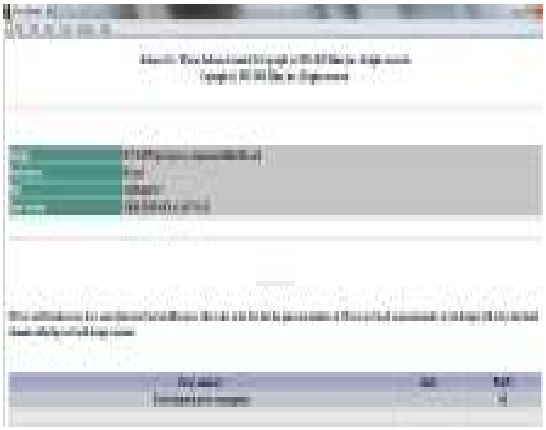

Figure 12: Power summary of merged flip-flop

Table 1: Comparison Result

Parameters	Existing Flip-flop	Merged Flip-Flop
Power(mw)	85	65

CONCLUSION

This paper has proposed an algorithm for flip-flop replacement for power reduction in digital integrated circuit design. The procedure of flip-flop replacements is depending on the combination table, which records the relationships among the flip-flop types. The concept of pseudo type is introduced to help to enumerate all possible combinations in the combination table. By the guidelines of replacements from the combination table, the impossible combinations of flip-flops will not be considered that decreases execution time. Besides power reduction, the objective of minimizing the total wirelength also be considered to the cost function. This algorithm can achieve a balance between power reduction and wirelength reduction and the running time is very short.

REFERENCES

1. Chang YT, Hsu CC, Lin PH, Tsai YW, and Chen SF, Postplacement power optimization with multibit flipflops, in Proc. IEEE/ACM Comput. Aided Design Int. Conf., San Jose, CA, 2010; 218–223.
2. Cheon Y, Ho PH, Kahng AB, Reda S, and Wang Q, Power aware placement, in Proc. Design Autom. Conf., 2005; 795–800.
3. Duarte D, Narayanan V, and Irwin MJ, Impact of technology scaling in the clock power, in Proc. IEEE VLSI Comput. Soc. Annu. Symp., Pittsburgh, PA,2002; 52–57.
4. Gronowski WJ, Bowhill, Preston RP, Gowan MK, and Allmon RL, High- performance microprocessor design," IEEE J. Solid-State Circuits, 1998; 33(5):676– 686.
5. Hou W, Liu D, and Ho PH, Automatic register banking for lowpower clock trees, in Proc. Quality Electron. Design, San Jose, CA, 2009; 647–652.
6. Kawagachi H and Sakurai T, A reduced clock swing flip-flop (RCSFF) for 63% clock power reduction, in VLSI Circuits Dig. Tech. Papers Symp., 1997;97–98.
7. Shyu YT, Lin JM, Huang CP, Lin CW, Lin YZ, and Chang SJ, Effective and efficient approach for power education by using multi bit flip flops, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2013; 21, (4): 218–223.
8. Ding W. and Marchionini G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

Source of support: Nil, Conflict of interest: None Declared