OPTIMIZATION OF CLOCK TREE BY USING MULTI-BIT FLIP-FLOPS WITH MERGING AND RELOCATION TECHNIQUES

Clock Power Reduction by Flip-Flop Merging

D.Mary Getsy.M.E.[A.E]*

R.Padmapriya**

Abstract—

Power consumption has become a very strong issue in current VLSI design. In electronics, the integrated circuits consume more dynamic power sources. Clock power is the major dynamic sources. In a circuit design, we can reduce its power consumption by replacing several flip-flops with less multi-bit flip-flops. However, this method may change the performance of the original circuit. Hence, the flip-flop alternate without timing and placement capacity constraints destruction becomes a quite complex problem. To avoid this difficulty very effective, we have proposed some techniques. First, we find out which of the flip-flops are capable of merging with other flip-flops. Next, we illustrate how to build a combination table to specify possible combinations of flip-flops provided by a library. Finally, we use a hierarchical way of merging the flip- flops. Besides power reduction, the objective of minimizing the total wire length is also considered. According to the experimental results, our paper significantly reduces the clock power by 20–30% and the running time is very short.

***Index Terms*—clock power, multi-bit flip-flop, combination table, merging.**

* (Guided by) Asst. Professor, Dept. of ECE, Sri Lakshmi Ammaal Engineering college, Chennai, Tamilnadu, India.

** M.E.[Applied Electronics], Sri Lakshmi Ammaal Engineering college, Chennai, Tamilnadu, India.

I. INTRODUCTION

Power has become a major source in electronic products. There are four sources of power dissipation in digital CMOS circuits: switching power, short-circuit power, leakage power and static power. The following equation describes these four components of power:

$$P_{avg} = P_{switching} + P_{short\text{-}circuit} + P_{leakage} + P_{static} \quad (1.1)$$

$P_{switching}$ is the switching power. For a properly designed CMOS circuit, this power component usually dominates, and may account for more than 90% of the total power. In this transition activity factor which is defined as the average number of power consuming transitions that is made at a node in one clock period. The circuit stored information about the previous history of input is called storage or memory elements. A primitive storage element can be constructed from a small number of gates connecting the outputs back as inputs. These circuits are binary cells capable of storing one bit of information. They have two outputs, one for the normal value and one for the complement value of bit stored in it. Primitive memory elements actually fall into two board classes: latches and flip-flop.

Static power is power consumed while there is no circuit activity. For example, the power consumed by a D flip-flop when neither the clock nor the D input have active inputs (i.e., all inputs are "static" because they are at fixed dc levels). Dynamic power is power consumed while the inputs are active. When inputs have ac activity, capacitances are charging and discharging and the power increases as a result. The dynamic power includes both the ac component as well as the static component. As technology advances, a systems-on-a-chip (SoC) design can contain more and more components that lead to a higher power density. This makes power dissipation reach the limits of what packaging, cooling or other infrastructure can support. Reducing the power consumption not only can enhance battery life but also can avoid the overheating problem, which would increase the difficulty of packaging or cooling [1], [2]. Therefore, the consideration of power consumption in complex SOCs has become a big challenge to designers. Moreover, in modern VLSI designs, power consumed by clocking has taken a major part of the whole design especially for those designs using deeply scaled CMOS technologies [3]. Thus, several methodologies [4], [5] have been proposed to reduce the power consumption of clocking. Clock gating saves power by adding more logic to a circuit to prune the clock tree. It disables the clock gates by AND gate. Switching states consumes power. Not being switched, the switching power consumption goes to zero. The advantage is when the latch is not required to switch state,

CLK-gate signal is turned off and the clock is not allowed to charge and discharge Cg, saving clock power. The drawback is there is area overhead associated with control circuitry. Apart from the latches, it adds some extra AND gates.

To optimize the clock power consumption we introduce the multi-bit flip-flops concept with several techniques. Besides, once more smaller flip-flops are replaced by larger multi-bit flip-flops; device variations in the corresponding circuit can be effectively reduced.

II. EXSISTING SYSTEM

Clock gating method is well known technique for reduce clock power consumption in circuit. Each circuit varies within and according to their applications, not all the circuits operate at the same time, giving rise to power consumption. By ANDing the clock with a gate-control signal, clock gating effectively disables the clock to an appropriate circuit when the circuit is not in used. So the unneeded power consumption can be avoided charging and discharging of unused circuits. This targets power reduction. In data-driven clock gating, employed for FFs at the gate level, which is most aggressive possible. The clock gating driving a flip-flop is gating when FFs state is not subject to change in the next clock cycle [10]. It also causes area overhead. So this technique is not effective.

In clock distribution networks, several small-swing clocking schemes have been proposed as in [4]. Half swing method needs four clock signal. Due to skew problems among the four clock signal it needs an extra chip. A reduced clock-swing flip-flop requires extra high power-supply voltage to reduce the leakage current [4].

Another technique called multi-bit flip-flop (MBFF) has recently proposed for clock power reduction. By grouping of FFs, we can reduce the clock power consumption. MBFF is physically merging FFs into a single cell.

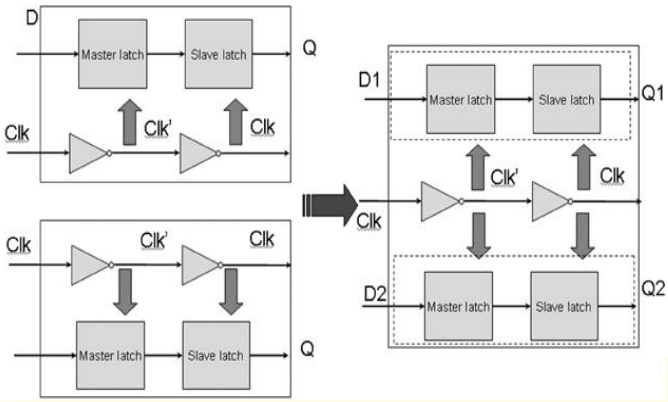

Fig. 1. Merging of two 1 bit flip-flops into one 2 bit flip-flop (a) Two 1-bit flip-flops (before merging). (b) 2-bit flip-flop(after merging)

The FF grouping problem was NP-hard also occurred [6] first proposed the problem of using multi-bit flip-flops to reduce power consumption in the post-placement stage. They use the graph-based approach to deal with this problem. In a graph, each node represents a flip-flop. If two flip-flops can be replaced by a new flip-flop without violating timing and capacity constraints, they build an edge between the corresponding nodes. After the graph is built, the problem of replacement of flip-flops can be solved by finding an *m*-clique in the graph. The flip-flops corresponding to the nodes in an *m*-clique can be replaced by an *m*-bit flip-flop. They use the branch-and-bound and backtracking algorithm [8] to find all *m*-cliques in a graph. Because one node (flip-flop) may belong to several *m*-cliques (*m*-bit flip-flop), they use greedy heuristic algorithm to find the maximum independent set of cliques, which every node only belongs to one clique, while finding *m*-cliques groups. However, if some nodes correspond to k-bit flip-flops that k1, the bit width summation to flip-flops corresponding to nodes in an m-clique, j , may not equal m. if the library type of a j-bit flip-flops is not supported by the library, it may be wasting in finding impossible combinations of flip-flops.

This paper make following contributions:

- To ease the identification of mergeable flip-flops, we find which of the FFs has same clock synchronization. In this way we find out the same clocked FFs.
- Build a combination table before merging two flip-flops, by this we can avoid wasting time in finding impossible combination of flip-flops.

- A chip is partitioned into several sub regions and performs replacement in each sub region to reduce the complexity. This may degrade the solution quality. To resolve it, we use a hierarchical way to enhance the result.

In next section describes the problem formulation. Section IV describes the proposed technique. Section V shows comparison results. Finally, section VI concludes this paper.

III. PROBLEM FORMULATION

Before giving our problem formulation, given a cell library L and a placement which contains a lot of flip-flops, our target is to merge as many flip-flops as possible in order to reduce the total power consumption. If we want to replace some flip-flops by a new flip-flops constraints will occur. Besides, since the replacement would change the routing length of the nets that connect to a flip-flop. It inevitably changes timing of some paths. Finally, to ensure that a legalized placement can be obtained after the replacement, there should exist enough space in each bin. To consider these issues, we define two constraints as follows.

- Timing Constraint for Net Connecting to a Flip-flop.
- Placement Capacity Constraint will occur.

To avoid these problem number of several techniques are proposed in Multi-bit flip-flops.

IV. PROPOSED SYSTEM

A. Multi-Bit Flip-Flops

Reducing clock power and decreasing the total flip-flops by using number of techniques in merging multi-bit flip-flops.

B. Flow Chart

Our design flow [see Fig. 2.] can be ro**u**ghly divided into three stages. In the beginning, we have to identify the flip-flops which of the clock pulse are synchronized in the same region. 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 synchronized 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 mergeable.

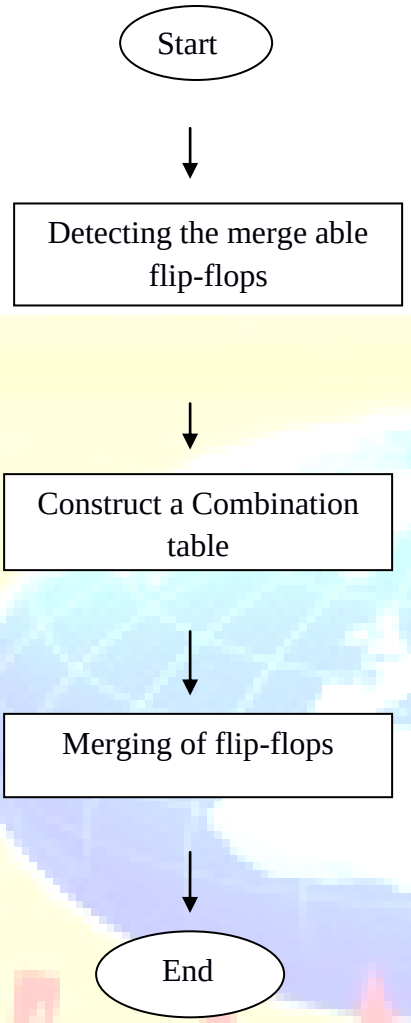

Fig. 2. Flow chart

Thus, we have to combine several bins into a larger bin and repeat this step until no flip-flop can be merged anymore.

C. Identifying the Mergeable Flip-Flops

The replacement of some flip-flops with multi-bit flip- flops would change the routing length of the nets that connect to a flip-flop; it inevitably changes timing of some paths. To avoid that timing is affected after the replacement, several techniques are used. First, identify the flip-flops by clock synchronization in circuits. The clock distribution network (or clock tree, when this network forms a tree) distributes the clock signal(s) from a common point to all the elements that need it. Since this function is vital to the operation of a synchronous system, much attention

has been given to the characteristics of these clock signals and the electrical networks used in their distribution. Clock signals are often regarded as simple control signals; however, these signals have some very special characteristics and attributes.

Clock signals are typically loaded with the greatest fan-out and operate at the highest speeds of any signal within the synchronous system. Since the data signals are provided with a temporal reference by the clock signals, the clock waveforms must be particularly clean and sharp. Furthermore, these clock signals are particularly affected by technology scaling, in that long global interconnect lines become significantly more resistive as line dimensions are decreased. This increased line resistance is one of the primary reasons for the increasing significance of clock distribution on synchronous performance. Finally, the control of any differences and uncertainty in the arrival times of the clock signals can severely limit the maximum performance of the entire system.

In a synchronous digital system clock signal is used to synchronize the movement of data within the system. Clock signals are required to be distributed at physically remote locations of an integrated circuit. Clock signals transitions drive all the synchronous elements of a digital circuit like Flip Flops and Memories. These elements are referred to *as* Sinks. Clock Distribution Network (CDN) is circuits that distribute a clock signal from a central global clock source at the centre of the Integrated circuit to all the sinks which use it.

In the process of Clock distribution, the clock signal traverses through a lot of interconnect networks and buffers which are a part of the clock distribution network. These elements introduce delay in the clock signal path. Ideally, a clock signal should arrive at all the sinks at the same time. But due to the variations in parameters like wire interconnect length, temperature variations, capacitive coupling and process variations; the arrival time of the clock transition at different sink locations varies. This spatial variation in the arrival time of the clock transition on an integrated circuit is commonly referred to as Clock Skew. For two points i and j, if the arrival times of the clock signals are a_i and a_j respectively then the clock skew between two points is given by $d(i,j) = a_i - a_j$ The Fig. 3. illustrates the reason for clock skew.

Clock signals typically have the highest fan out and operate at the highest speeds in a synchronous digital system. Since the clock signals are used to synchronize the operations of the entire digital circuit the clock transitions should be sharp and should have minimum possible skew to avoid any data integrity errors. As the frequency of operation of the synchronous circuit

increases the circuit becomes more and more susceptible to clock skew i.e. the timing becomes more and more critical. By differential circuit XOR gate, find the difference of clock pulse of each flip-flop. By compare with the reference clock value of the difference is calculated.

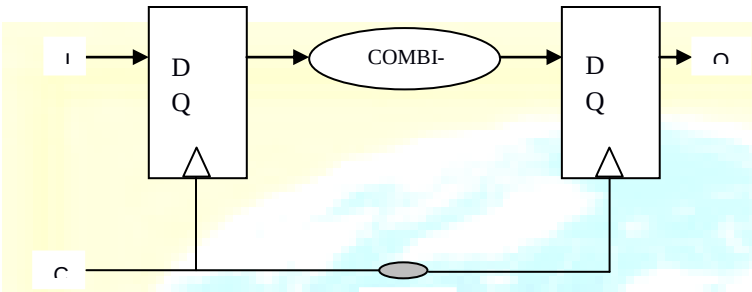

Fig. 3. Clock skew illustration

D. Build a Combination Table

If we want to replace several flip-flops by a new flip-flop f_i' (note that the bit width of f_i' should equal to the summation of bit widths of these flip-flops), we have to make sure that the new flip-flop f_i' is provided by the library $\boldsymbol{L}$ when the feasible regions of these flip-flops overlap. Now a combination table is to be built, which records all possible combinations of flip-flops to get feasible flip-flops before replacements. Thus, we can gradually replace flip-flops according to the order of the combinations of flip-flops in this table. We build a combination table, which records all possible combinations of flip-flop to get feasible flip-flops before replacements. Thus, we can gradually replace flip-flops according to the order of the combinations of flip-flops in this table. Since only one combination of flip-flops needs to be considered in each time, the search time can be reduced greatly.

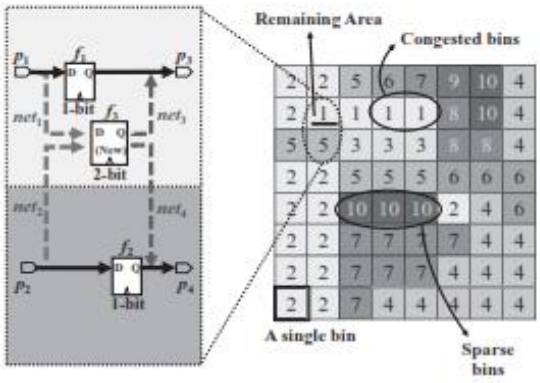

Fig. 4. Combinational of flip-flops in CDN

Thus the fig 4 shows the clock distribution network with number of sub regions. Each region operates at different clock pulse. The arrival time of each clock pulse compare with the main clock pulse of CDN. Each flip-flop acted at different clock zone. The clock difference is calculated by the differential circuit. The differential circuit is XOR. By using reference clock, each flip-flop clock difference is compared and counted. In this section, we illustrate how to build a combination table. Figure 5 shows difference of clock pulse value, from those same bins capable of merging the flip-flop. Initialize the library L and combinational table.

1 4

Fig 5 Example of library L

For example consider the library L that provides 2 types of flip-flops, whose bit widths are 1 and 4 in Fig. 4. We first initialize two combinations *n*1 and *n*2 to represent these two types of flip-flops in the table ***T*** [see Fig. 6(a)]. Next, the function is performed to check whether the flip-flop types between 1 and 4 exist or not. Now, for each combination in ***T***, we would build a binary tree with 0-level, and the root of the binary tree denotes the combination. Next, we try to build new legal combinations according to the present combinations. By combing two1-bit flip-flops in the first combination, a new combination *n*3 can be obtained [see Fig. 6(b)]. Similarly, we can get a new combination *n*4 (*n*5) by combining *n*1 and *n*3 (two *n*3's) [see Fig. 6(c)]. Finally, *n*6 is obtained by combining n1 and n4.

Library L		Combinational table T	
Type 1 1 bit	Type 2 4 bit	n1 1 bit	n2 4 bit

(a)

Combination table T		
n1 1 bit	n2 4 bit	n3 2 bit n1 + n1

(b)

Combinational Table T					
n1	n2	n3	n4	n5	n6
1 bit	4 bit	2 bit n1 + n1	3 bit n1 + n3	4 bit n3 + n3	4bit n1 + n4

(c)

Combination table T			
n1	n2	n3	n4
1 bit	2 bit	3 bit n1 + n1	4 bit n3 + n3

(d)

Fig. 6. Example of building combination table (a) initialize the library L and combination table

T. (b) new combination n3 is obtained from combining two n1s. (c) New combination n_6 is obtained from combining n_1 and n_4. (d)) Last combination table is obtained after deleting the unused combination in (e).

Thus two kinds of flip-flops whose clock rates are equal are added into library L. In above combinations n5 and n6 are duplicated because they both represent the same condition, it replaced by combining four 1-bit flip-flops by a 4 bit flip-flop. To speed up our program n6 is deleted from T rather than n5 because its process is longer. After this change n4 is unused combination. So after deleting n6, n4 also need to be deleted. The last combination table T shown in Fig. 6(d).

E. Merge Flip-Flops

After the combination table is built the combination of flip-flops are used for merging and replacing. To reduce the complexity the whole placement region is divided into several sub regions. Then, several sub regions are combined into a larger sub region and the flip-flops are replaced again so that those flip-flops in the neighboring sub regions can be replaced further. Finally, those flip- flops with pseudo types are deleted in the last stage because they are not provided by the supported library.

1) Region Partition: To speed up our problem, we divide the whole chip into several sub regions. By suitable partition, the computation complexity of merging flip-flops can be reduced significantly. We divide the region into several sub regions, and each sub region contains six bins, where a bin is the smallest unit of a sub region.

2) Replacement of Flip-flops in Each Sub Region: Before illustrating the procedure to merge flip-flops, first an equation is given to measure the quality if two flip-flops are going to be replaced by a new flip-flop as follows:

Besides, if the region has larger available space to place a new flip-flop, it implies that it has higher opportunities to combine with other flip-flops in the future and more power reduction. Thus, we will give it a smaller cost. First the flip-flops are linked below the combinations corresponding to their types in the library. Then, for each combination ***n*** in ***T,*** we serially merge the flip-flops.

V. COMPARISON RESULTS

The values of the power ratio can be computed by the following equations:

Measurement of power

$$PR_{RATIO(\%)} = \frac{\text{poweroriginal} - \text{power merged}}{power\ original}.100\%$$

Thus using above equations we can measure the power and wire length of original case (Before merging of flip-flop) and compare with the resulted distribution of flip-flop (After merging of flip-flop). The number of clocked inverter has been reduced by merging of flip-flop. So the area also has reduced, then obviously number inverter has reduced, it intimates consumption of clock power also reduced, which is denoted by PR-Ratio.

VI. 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. The experimental results show that our algorithm can achieve a balance between power reduction and area reduction. Besides power reduction, the objective of minimizing the total area is also considered.

REFERENCES

[1] P. Gronowski, W. J. Bowhill, R. P. Preston, M. K. Gowan, and R. L. Allmon, "High-performance microprocessor design," *IEEE J. Solid-State Circuits*, vol. 33, no. 5, pp. 676–686, May 1998.

[2] W. Hou, D. Liu, and P.-H. Ho, "Automatic register banking for low-power clock trees," in *Proc. Quality Electron. Design*, San Jose, CA, Mar. 2009, pp. 647–652.

[3] D. Duarte, V. Narayanan, and M. J. Irwin, "Impact of technology scaling in the clock

power," in *Proc. IEEE VLSI Comput. Soc. Annu. Symp.*, Pittsburgh, PA, Apr. 2002, pp. 52–57.

[4] H. Kawagachi and T. Sakurai, "A reduced clock-swing flip-flop (RCSFF) for 63% clock power reduction," in *VLSI Circuits Dig. Tech. Papers Symp.*, Jun. 1997, pp. 97–98.

[5] Y. Cheon, P.-H. Ho, A. B. Kahng, S. Reda, and Q. Wang, "Power-aware placement," in *Proc. Design Autom. Conf.*, Jun. 2005, pp. 795–800.

[6]Y.-T. Chang, C.-C. Hsu, P.-H. Lin, Y.-W. Tsai, and S.-F. Chen, "Post-placement power optimization with multi-bit flip-flops," in *Proc. IEEE/ACM Comput.-Aided Design Int. Conf.*, San Jose, CA, Nov. 2010, pp. 218–223.

[7]*Faraday Technology Corporation* [Online]. Available: http://www. faraday-tech.com/index.html

[8]C. Bron and J. Kerbosch, "Algorithm 457: Finding all cliques of an undirected graph," *ACM Commun.*, vol. 16, no. 9, pp. 575–577, 1973.

[9]*CAD Contest of Taiwan* [Online]. Available: http://cad_contest.cs. nctu.edu.tw/cad11

[10] M.Dono, E.Macii. and L. Mazzoni, "power-aware clock tree planning," in *proc.int. symp. Phys*. Design, 2004, pp, 138-147.