Multi Flip Flop Merging based Clustering using AGC Algorithm

Susithra. E
II M.E (Applied Electronics)
Jayaram college of Engineering and Technology
Trichy, India

Mr. R. Aravindh.,M.E
Assistant Professor,ECE
Jayaram college of Engineering and Technology
Trichy, India

Abstract—Based on the elimination feature of redundant inverters in merging 1-bit flip-flops into multi-bit flip-flops, gives reduction of wired length and this result in reduction of power consumption. With the growing popularity of portable devices, power reduction has become a popular design goal for advanced design application. Multi-bit flip-flop is an effective power-saving implementation methodology by merging single-bit flip-flops in the design. Using multi-bit flip-flops can reduce clock dynamic power and the total flip-flop area effectively. In this paper, we propose agglomerative clustering algorithm to find the nearest clustering of flip-flops for merging the flip-flops. This algorithm finds the clusters of flip-flop and finally combine FFs to reduce the wire length.

Keywords— Clock power reduction, merging, multi-bit flip-flop, Wirelength.

I. INTRODUCTION

Very-large-scale integration (VLSI) is the process of creating integrated circuits by combining thousands of transistors into a single chip. VLSI began in the 1970s when complex semiconductor and communication technologies were being developed. The microprocessor is a VLSI device.

In electronics, a flip-flop or latch is a circuit that has two stable states and can be used to store state information. A flip-flop is a bistable multivibrator. The circuit can be made to change state by signals applied to one or more control inputs and will have one or two outputs. It is the basic storage element in sequential logic. Flip-flops and latches are a fundamental building block of digital electronics systems used in computers, communications, and many other types of systems.

Flip-flops and latches are used as data storage elements. Such data storage can be used for storage of state, and such a circuit is described as sequential logic. When used in a finite-state machine, the output and next state depend not only on its current input, but also on its current state (and hence, previous inputs). It can also be used for counting of pulses, and for synchronizing variably-timed input signals to some reference timing signal.

Flip-flops can be either simple (transparent or opaque) or clocked (synchronous or edge-triggered); the simple ones are commonly called latches. The word latch is mainly used for storage elements, while clocked devices are described as flip-flops. A latch is level-sensitive, whereas a flip-flop is edge-sensitive. That is, when a latch is enabled it becomes transparent, while a flip-flop’s output only changes on a single type (positive going or negative going) of clock edge.

With the growing popularity of portable devices, power reduction has become a popular design goal for advanced design application, whether mobile or not. Reducing power consumption in chips enables better, cheaper products to be designed and power-related chip failures to be minimized. As a result, how to minimize power consumption has become an important design goal that every chip designer must take care.

Several lower power design techniques have played an important role in the design flow[1]. Clock gating methodology is used for the register bank to replace the multiplexers and it can avoid the operation of reloading the same data value. The clock gating technique could reduce the dynamic power consumption efficiently[5].

The multi-Vth concept is aimed at using multi-Vth cell with satisfying performance to reduce leakage consumption, and replace lower Vth (LVT) cells by high Vth (HVT) ones, if there is room for slack. Multiple Supply Multiple Voltage (MSMV) supplies of different voltages are used for core logic, base on satisfy performance or functional requirement to adjust operating voltage for each domain, even shut off this domain.

We can see that the flip-flops on clock tree accounted for a large proportion of power consumption. Although the power distribution will vary with different ASIC design, reducing power consumption of the flip-flop on clock tree can eliminate total power consumption efficiently.

Multi-bit flip-flop is an effective power-saving implementation methodology by merging single-bit flip-flops in the design. Using multi-bit flip-flops can reduce clock dynamic power and the total flip-flop area effectively.

In this section, we will introduce multi-bit flip-flop conception. Before that, we will review single-bit flip-flop. Figure 2 shows an example of single-bit flip-flop. A single-bit flip-flop has two latches (Master latch and slave latch)[6]. The latches need “Clk” and “Clk’” signal to perform operations, such as Figure2 shows. In order to have better delay from Clk-> Q, we will regenerate “Clk” from “Clk’” . Hence we will have two inverters in the clock path.
Figure(1) shows an example of merging two 1-bit flip-flops into one 2-bit flip-flop. Each 1-bit flip-flop contains two inverters, master-latch and slave-latch. Due to the manufacturing rules, inverters in flip-flops tend to be oversized. As the process technology advances into smaller geometry nodes like 65nm and beyond, the minimum size of clock drivers can drive more than one flip-flop. Merging single-bit flip-flops into one multi-bit flip-flop can avoid duplicate inverters, and lower the total clock dynamic power consumption. The total area contributing to flip-flops can be reduced as well.

By using multi-bit flip-flop to implement ASIC design, users can enjoy the following benefits:
1) Lower power consumption by the clock in sequential banked components
2) Smaller area and delay, due to shared transistors and optimized transistor-level layout
3) Reduced clock skew in sequential gates

Figure(2) shows an example of dual-bit flip-flop cell. It has two data input pins, two data output pins, one clock pin and reset pin. Use dual-bit flip-flop can get the benefits of lower power consumption then single-bit, and almost no other additional costs to pay. Figure(3) shows the true table of dual bit flip-flop cell. We could find that when CK is positive edge, the value of Q1 will pass to D1, and the value of Q2 will pass to D2. Or Q1 and Q2 will keep original value.

We implement the multi-bit Flip Flop concept in merging of two or more flip-flops according to application to reduce the block size and power consumption. Here the merging of flip-flops are done by forming clusters by agglomerative clustering algorithm. This clustering algorithm forms the separation of clusters according to the minimum distance between Flip-Flops. From the neighboring information, we merge the flip-flops.

II. PROBLEM FORMULATION
Before giving our problem formulation, we need the following notations.
1) Let fi denote a flip-flop and bi denote its bit width.
2) Let A(fi) denote the area of fi.
3) Let P(fi) denote all the pins connected to fi.
4) Let M(pi, fi) denote the Manhattan distance between a pin pi and fi, where pi is an I/O pin that connects to fi.
5) Let S(pi) denote the constraint of maximum wirelength for a net that connects to a pin pi of a flip-flop.
6) Given a placement region, we divide it into several bins and each bin is denoted by Bk.
7) Let RA(Bk) denote the remaining area of the bin Bk that can be used to place additional cells.
8) Let L denote a cell library which includes different flip-flop types (i.e., the bit width or area in each type is different).

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 f1, ..., fj−1 by a new flip-flop fj, the bit width of fj must be equal to the summation of bit widths in the original ones (i.e., bi = bj, i = 1 to j−1). Besides, since the placement 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.
1) Timing Constraint for a Net Connecting to a Flip-Flop fj from a Pin pi: To avoid that timing is affected after the replacement, the Manhattan distance between pi and fj cannot be longer than the given constraint S(pi) defined on the pin pi (i.e., M(pi, fj) ≤ S(pi)). Based on each timing constraint defined on a pin, we can find a feasible placement region for a flip-flop fj. See Fig. 4 for example. Assume pins p1 and p2
connect to a 1-bit flip-flop \( f_1 \). Because the length is measured by Manhattan distance, the feasible placement region of \( f_1 \) constrained by the pin \( p_i \) [i.e., \( M(p_i, f_1) \leq S(p_i) \)] would form a diamond region, which is denoted by \( R(p_i) \), \( i = 1 \) or 2. See the region enclosed by dotted lines in the figure. Thus, the legal placement region of \( f_1 \) would be the overlapping region enclosed by solid lines, which is denoted by \( R(f_1) \). \( R(f_1) \) is the overlap region of \( R(p_1) \) and \( R(p_2) \).

2) Capacity Constraint for Each Bin \( B_k \): The total area of flip-flops intended to be placed into the bin \( B_k \) cannot be larger than the remaining area of the bin \( B_k \) (i.e., \( A(f_i) \leq RA(B_k) \)).

III. DESIGN FLOW

Our design flow can be roughly divided into three stages. 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 are 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[2].

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

In this section, we would detail each stage of our method. In the first subsection, we show a simple formula to transform the original coordination system into a new one so that a legal placement region for each flip-flop can be identified more easily[2]. The second subsection presents the flow of building the combination table.

In the beginning, they 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.

IV. OUR ALGORITHM

In this paper, we propose agglomerative clustering algorithm to find the nearest clustering of flip-flops for merging the flip-flops[2]. The algorithm forms clusters in a bottom-up manner, as follows:
1). Initially, put each article in its own cluster.
2). Among all current clusters, pick the two clusters with the smallest distance.
3). Replace these two clusters with a new cluster, formed by merging the two original ones.
4). Repeat the above two steps until there is only one remaining cluster in the pool.
5). This algorithm finds the clusters of flip-flop and finally combine FFs to reduce the wire length.

We propose agglomerative-based clustering for number of flip-flop reduction and signal wirelength minimization is proposed. Comparing to previous works on flip-flop reduction, our method can obtain an optimal tradeoff curve between flip-flop number reduction and increase in signal wirelength. Our proposed methodology performs in both reducing number of flip-flops and minimizing increase in signal wirelength[1]. In our methodology obtains a tradeoff of 15.8% reduction in flip-flop's signal wirelength with 16.9% additional flip-flops.
The algorithm used advantages:
1). The agglomerative clustering algorithm will result in a binary cluster tree with single article clusters as its leaf nodes and a root node containing all the articles.
2). In the clustering algorithm, we use a distance measure based on log likelihood.

E. Form cluster FF:
In this module we form flip flops in clusters according to the distance information. This clustering represents what are the flip flops can be made to merge and form as a multi bit flip flops. This clustering was found by agglomerative clustering algorithm to form clustered flip flop.

Agglomerative hierarchical clustering is a bottom-up clustering method where clusters have sub-clusters, which in turn have sub-clusters, etc. The classic example of this is species taxonomy. Gene expression data might also exhibit this hierarchical quality (e.g. neurotransmitter gene families). Agglomerative hierarchical clustering starts with every single object (gene or sample) in a single cluster. Then, in each successive iteration, it agglomerates (merges) the closest pair of clusters by satisfying some similarity criteria, until all of the data is in one cluster.

The hierarchy within the final cluster has the following properties:
1) Clusters generated in early stages are nested in those generated in later stages.
2) Clusters with different sizes in the tree can be valuable for discovery.

F. Create table:
In this module, we construct a table called as combinational table which contains cluster information of each flip flop and distances between them. From this, the decision was taken to merge FFs. Then from this information, we merge FF at a mergable range of clusters.

G. Merge flip flop:
In this stage, we finally merge flip flops according to the acceptable range from combinational table. This formation made the individual FFs into multi-bit flip-flops which contains common clock for FFs and form multi bit FF.

V. SIMULATION RESULTS
Simulation results shows the ISim GUI opens and loads the design. The simulator time remains at 0 ns until you specify a run time.

For comparison purposes, you can browse to the completed folder for a completed version of the simulate_isim.bat batch file.
Table-1: COMPARISION OF SPEED, AREA AND POWER

<table>
<thead>
<tr>
<th>METHODS</th>
<th>AREA</th>
<th>SPEED</th>
<th>POWER</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXISTING SYSTEM</td>
<td>FFs(264)</td>
<td>195.32(MHZ)</td>
<td>43 (mwatt)</td>
</tr>
<tr>
<td>PROPOSED SYSTEM</td>
<td>FFs(148)</td>
<td>233.05(MHZ)</td>
<td>36 (mwatt)</td>
</tr>
</tbody>
</table>

The above tabulation shows that if the number of area increases the corresponding speed is also increase.

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 by AGC algorithm.

In this paper, we proposed merging of flip flops to reduce power reduction of single bit FF to form multi bit FF. This merging was found by acceptable distance range each neighboring flip flops. This was done by forming clusters of flip flops by using agglomerative clustering algorithm.

ACKNOWLEDGEMENT

First and foremost I thank GOD, the almighty who stands behind and strengthens me to complete the project successfully. I would like to express my sincere respect and gratitude towards my HOD Mrs. M. Geetha, M.Tech., (Ph.D). Her wide knowledge, serious research attitude and enthusiasm in work deeply impressed me and taught what a true scientific research should be. I am very thankful for the support she extended to me and the freedom to express my views. Words are inadequate to express the gratitude to my beloved husband, children, Parents, Brother and friends for their excellent and never ending co-operation.

REFERENCES


BIOGRAPHIES

E. Susithra received her B.E. degree in Electronics and Communication Engineering from Info Institute of Engineering, Coimbatore, Anna University, Chennai, India, in 2012 and M.E degree in Communication Systems from Jayaram College of Engineering & Technology, Trichy, Anna University, Chennai, India, in 2014. Her research area of interest is Very Large Scale Integrated Circuits.

R. Aravindh received his B.E. degree in Electronics and Communication Engineering from Jayaram College of Engineering & Technology, Trichy, Anna University, Chennai, India, in 2008, and M.E degree in Communication Systems from Jayaram College of Engineering & Technology, Anna University, Trichy, India, in 2011. Currently, he is working as Assistant professor in Jayaram College of Engineering & Technology, Trichy, India. His research interest includes Digital Communication, High Speed networks, Digital Image Processing.