A Greedy Heuristic Algorithm for Flip-Flop Replacement Power Reduction in Digital Integrated Circuits

C.N.Kalaivani¹, Ayswarya J.J²

Assistant Professor, Dept. of ECE, Dhaanish Ahmed College of Engineering, Chennai, Tamilnadu, India¹
PG Student [Applied Electronics] Dept. of ECE, Dhaanish Ahmed College of Engineering, Chennai, Tamilnadu, India²

ABSTRACT: Power consumed by clocking has taken a major part of the whole design circuit. This paper proposed that reducing the power consumption and area by replacing some flip flops with fewer multi-bit flip-flops without affecting the performance of the original circuit. Various techniques are proposed. First to identify those flip-flops that can be merged. Next a combination table is built to enumerate all possible combinations. Finally, those flip-flops are merged in hierarchical manner. Besides the power reduction minimizing the total wire length is also considered. According to the experimental results clock power can be reduced by 20-30% and the running time can also be reduced.

KEYWORDS: Clock power reduction, merging, wire length, replacement, multi-bit flip-flop.

I. INTRODUCTION

A clock system and a logic part consumes dominant Part of the total chip power by 20–45%. In this clock system power, 90% is consumed by the flip-flops [1]. This is due to the high switching activity.

\[ P_{\text{clk}} = C_{\text{clk}} V^2 d_{\text{clk}} \]  

(1)

Where \( P_{\text{clk}} \) is clock power, \( f_{\text{clk}} \) is the clock frequency, \( V_{\text{dd}} \) is the supply voltage, and \( C_{\text{clk}} \) is the switching capacitance included in the gate capacitance of flip-flops.

During clock tree synthesis, less number of flip-flops means less number of clock sinks. Thus the resulting clock network would have smaller power consumption and uses less routing resource. 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.

Single-bit flip-flop can be reviewed before using the multi bit flip-flop. Figure 1.1 shows an example of single-bit flip-flop. A single-bit flip-flop has two latches (Master latch and slave latch). The latches need “\( \text{CLK} \)” and “\( \text{CLK}’ \)” signal to perform operations, shown in Figure 1.

![Fig. 1: Single-Bit Flip-Flop](image-url)
In order to have better delay from Clk→ Q, regenerate “Clk” from “Clk”’. There are two inverters in the clock path. Figure 2 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.

![Fig. 2: Merging flip-flops](image)

As the process technology advances into smaller geometry nodes, 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 duplication inverters and lowers the total clock dynamic power consumption.

**II. LITERATURE SURVEY**


   **Concept:**
   Replacing several one bit flip flop with one Multi Bit Flip Flop to reduce the total area and dynamic power and it can be reduced upto 50%.

   **Disadvantage:**
   Windows optimization technique is larger so that flip flop can perform slowly.


   **Concept:**
   Focuses on calculating the idle period of different flip flop and inserting the gating logic into netlist to achieve the total power by 25.3%.

   **Disadvantages:**
   The net switching power can be achieved by 25.4 % and then wirelength can also be reduced.


   **Concept:**
   Increase the flexibility that covers the clock distribution and clock generation circuit to consume total power by 40%.

   **Disadvantage:**
   Clock skew problem can be reduced by 30%.


   **Concept:**
   Replacing some flip flop with multibit flip flop without affecting the performance and total wire length can be minimized by 20-30%.
Disadvantage:
Using dual bit flip flop to save the clock power in 11.22% and the replacement of flip flop during switching rate is 10.43%.


Concept:
Focus on high frequency design to achieve high performance and to improve the complexity of the circuit.

Disadvantage:
In single supply voltage system reduce the clock power 25.45 %, and in multiple supply voltage system the clock power can be reduced by 26.15 %.

III. PROPOSED ALGORITHM

The Design flow can be roughly divided into three stages. First to use the combination table to combine all possible combinations of flip-flops. The difficulty of this problem is to repeatedly search a set of flip-flops that can be replaced by a new multi-bit flip-flop. However as the number of flip-flops in a chip increases dramatically the complexity would increase exponentially which makes the method impractical. To handle this problem more efficiently and to get better results, the following flowchart were used. The figure 3 shows the various approaches used in the algorithm.

![Flow chart of proposed method](image)

Fig. 3: Flow chart of proposed method

1) To facilitate the identification of mergeable flip-flops transform the coordinate system of cells. In this way the memory used to record the feasible placement region can also be reduced.

2) To avoid wasting time in finding impossible combinations of flip-flops, first build a combination table before actually merging two flip-flops. For example, if a library only provides three kinds of flip-flops which are 1-, 2-, and 3-bit first to separate the flip-flops into three groups. Therefore the combination of 1- and 3-bit flip-flops is not considered since the library does not provide the type of 4-bit flip-flop.

3) Partition a chip into several sub regions and perform replacement in each sub region to reduce the complexity. However, this method may degrade the solutions quality. To resolve the problem use a hierarchical way to enhance the result

A. Region partition to identify the mergeable flip-flop

To reduce the complexity, first divide the whole placement region into several sub-regions and then by using the combination table replace the flip-flops in each other sub-region. 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 as it is not provided by the supported library.
B. Replacement of flip-flop

After a combination has been built do the replacements of flip-flops according to the combination table. First flip-flops below the combinations corresponding to their types in the library were linked. Then for each combination n in T, serially merge the flip-flops linked below the left child and the right child of n from leaves to root. Based on its binary tree to find the combinations associated with the left child and right child of the root. Hence the flip-flops in the lists named left and right, linked below the combinations of its left child and its right child are checked. Then for each flip-flop f i in left the best flip-flop best in right which is the flip-flop that can be merged with f i with the smallest cost recorded in cbest, is picked. For each pair of flip-flops the combination cost is computed and they can be merged with the smallest cost as chosen. Finally add a new flip-flop f in the list of the combination n and remove the picked flip-flops which constitutes the f. For example, given a library containing three types of flip-flops (1-, 2-, and 4-bit), first to build a combination table T as shown in Figure5.

The above figure says that
(a) Sets of flip-flops before merging.
(b) Two 1-bit flip-flops, f1 and f2, are replaced by the 2-bit flip-flop f3.
(c) Two 1-bit flip-flops, f4 and f5, are replaced by the 2-bit flip-flop f6.
(d) Two 2-bit flip-flops, f7 and f8, are replaced by the 4-bit flip-flop f9.
(e) Two 2-bit flip-flops, f3 and f6, are replaced by the 4-bit flip-flop f10.
(f) Sets of flip-flops after merging.

In the beginning, the flip-flops with various types are, respectively, linked below n1, n2, and n3 in T according to their types. Suppose to form a flip-flop in n4 which needs two 1-bit flip-flops according to the
C. Combination table and merging flip-flop

Finally, add a new flip-flop in the list of the combination table and remove the picked flip-flops which constitutes the combination type. Pseudo type is an intermediate type which is used to enumerate all possible combinations in the combination table T, then to remove the flip-flops belonging to pseudo types. Thus after the above procedures have been applied de-replacement and replacement functions are performed if there exists any flop-flops belonging to a pseudo type is shown in figure 6.

Fig. 6: The combination table and merging

The figure says that
(a) Initialize the library L and the combination table T.
(b) Pseudo types are added into L, and the corresponding binary tree is also build.
(c) New combination n3 is obtained from combining two n1s.
(d) New combination n4 is obtained from combining n1 and n3.
(e) New combination n6 is obtained from combining n1 and n4.
(f) Last combination table is obtained after deleting unused combination in (e).

For example, if there still exists a flip-flop, fi, belonging to n3 after replacements in Fig (Fig. Last combination table is obtained after deleting the unused combination), then to de-replace fi into two flip-flops originally belongs to n1. After de-replacing, the replacements of flip-flops according to T without consideration of the combinations whose corresponding type is pseudo in L were built.
IV. COMPARISON TABLE FOR VARIOUS METHODS

This table specifies the various implementation of flip-flop to optimize the power and to achieve the net switching activity. Although the drivers are very wide devices it was found that for all technologies the share of the clock power that is due to leakage is at most 2.5%. Technology optimizations and dynamic runtime techniques for leakage reduction will become standard for clock power and will remain a major contributor to the total system power.

<table>
<thead>
<tr>
<th>Implementation</th>
<th>Description</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Post Placement Power Optimization</td>
<td>Progressive Windows Based Optimization</td>
<td>Reduce The Power &amp; Interconnecting Wire length</td>
</tr>
<tr>
<td>Power Aware Placement</td>
<td>Register clustering &amp; net weighting</td>
<td>Reduced area &amp; wire length</td>
</tr>
<tr>
<td>Impact Of Technology Scaling</td>
<td>Scaling interconnect &amp; impact of leakage</td>
<td>Reduced leakage power</td>
</tr>
<tr>
<td>Flip-Flop Merging And Relocation</td>
<td>Net Switching technique</td>
<td>Switching rate less &amp; save clock power</td>
</tr>
<tr>
<td>Clock Power Using Multibit Flip-flop</td>
<td>Three phase algorithm</td>
<td>Reduced power single &amp; multiple supply voltage system</td>
</tr>
<tr>
<td>Clock Power Flip-Flop For Future Soc Applications</td>
<td>CDMFF + CPSFF Proposed in LCPTFF</td>
<td>Reduced power &amp; area</td>
</tr>
<tr>
<td>A Low-Swing Clock Double-Edge Triggered Flip-Flop</td>
<td>LCDFF performed charging &amp; discharging Technique</td>
<td>Saving power in flip-flop operation &amp; clock network</td>
</tr>
</tbody>
</table>

CONVENTIONAL CONDITIONAL DATA MAPPING D FLIP-FLOP

In conditional data mapping flip-flop (CDMFF) uses only seven clocked transistors, resulting in about 50% reductions in the number of clocked transistors. This shows the effectiveness of reducing clocked transistor numbers to achieve low power. The figure 7 shows the circuit diagram for CDMFF.

![Fig. 7: Circuit diagram of CDMFF](image-url)
In a Conventional D flip-flop part of the clock energy is consumed by the internal clock buffer to control the transmission gates.

**CLOCKED PAIR SHARED FLIP-FLOP DESIGN**

To ensure efficient and robust implementation of low power sequential element propose a Clocked Pair Shared flip-flop to use less clocked transistor than CDMFF and to overcome the floating problem in CDMFF. The figure 8 shows the block diagram of CPSFF

![CPSFF Circuit Diagram](image)

**Fig. 8**: Circuit diagram of CPSFF

By reducing the no of transistor count the overall switching delay, power, and area consumption can be reduced. **LOW POWER CLOCKED PASS TRANSISTOR FLIP- FLOP** Low Power Clocked Pass Transistor flip-flop design shows much less power & Area constraints than the Existing two Flip-Flop designs. LCPTFF will be having very less clock delay when compared to all other circuits.

![LCPTFF Circuit Diagram](image)

**Fig. 9**: Circuit diagram of LCPTFF

<table>
<thead>
<tr>
<th>Type</th>
<th>Power Consumption</th>
<th>Area Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conventional CDMFF Design</td>
<td>0.45mW</td>
<td>270µm²</td>
</tr>
<tr>
<td>Clock Pair Share Flip-Flop</td>
<td>15.232µW</td>
<td>225 µm²</td>
</tr>
<tr>
<td>(CPSFF)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proposed Design (LCPTFF)</td>
<td>9.581 µW</td>
<td>84 µm²</td>
</tr>
</tbody>
</table>

**Table 1**: Comparison table for Power and Area
IV. SIMULATION RESULTS

![Diagram](image)

Fig: 10 a) Influence of the region size on power
FOR S BIT FLIP-FLOP

Fig: 11 b) Influence of the weighting factor on Wire-length reduction

OUTPUT WAVEFORM

The values PR_Ratio and WR_Ratio can be computed by the following equations:

\[ PR_{Ratio} = \frac{Power_{original} - Power_{merged}}{Power_{original}} \times 100\% \]
International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering  
(An ISO 3297: 2007 Certified Organization)  
Vol. 3, Issue 8, August 2014

\[ WRatio(\%) = \frac{wire_{length_merged}}{wire_{length_original}} \times 100\% \]

Table 2: Comparison of simulated results

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Existing Flip-Flop</th>
<th>Merged Flip-Flop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power</td>
<td>32</td>
<td>35</td>
</tr>
<tr>
<td>Wire length</td>
<td>35</td>
<td>42</td>
</tr>
</tbody>
</table>

Fig:12 Simulation result of combinational Table

Fig:13 Simulation results of merged flipflop

V. CONCLUSION

The number of flip-flops in a chip increases dramatically the complexity would increase exponentially, which makes the method impractical. To handle this problem more efficiently and get better results, the following approaches are used. 1) To facilitate the identification of mergeable flip-flops transform the coordinate system of cells. In this way the memory used to record the feasible placement region can also be reduced. 2) To avoid wasting time in finding impossible combinations of flip-flops first build a combination table before actually merging two flip-flops. 3) Partition a chip into several sub-regions and perform replacement in each sub-region to reduce the complexity. However this method may degrade the solution quality. To resolve the problem use a hierarchical way to enhance the result and processing time can be reduced.
REFERENCES