Power-Driven Flip-Flop Merging and Relocation

Shao-Huan Wang
Yu-Yi Liang
Tien-Yu Kuo
Wai-Kei Mak
@National Tsing Hua University
Outline

- Introduction
- Problem Formulation
- Algorithms
- Experimental Results
- Conclusions
Flip-Flop Merging

- Merge several 1-bit Flip-Flops into a Multi-bit Flip-Flop (MBFF)
  - Eliminate some inverters and area
  - Reduce the # clock sinks
Flip-Flop Merging

Two traditional 1-bit flip-flops

A 2-bit MBFF
Reduction of clock sinks
Related Work

- [15] Post-placement power optimization with multi-bit flip-flops, ICCAD’10

  The objective of [15] is to minimize the total FF Power

  However, our objective function is to minimize the # clock sinks and switching power of signal nets
Wirelength of Signal Nets

- Different merging solutions will affect the wirelength and switching power of signal nets differently.
Post-Placement Relocation

- After merging, we need to relocate these MBFFs
- It will affect the total switching power of signal nets
Outline

- Introduction
- Problem Formulation
- Algorithms
- Experimental Results
- Conclusions
Problem Formulation

- **Inputs**
  - A preplaced design and a MBFF Library

- **Objectives**
  - Minimize the # sinks in clock network
  - Minimize the switching power of signal nets

\[
\omega \times \frac{\text{Final \#sinks}}{\text{Original \#sinks}} + (1 - \omega) \times \frac{\sum_{\text{net}_i \in F} \alpha_i \times \text{Final WL}_i}{\sum_{\text{net}_i \in F} \alpha_i \times \text{Original WL}_i}
\]

$\alpha_i$ is the switching rate of signal nets
Constraints

- Guarantee there is no timing violation
  - Feasible region of FFs
- Control the placement density
  - Maintain the quality of legalization
  - Consider routing congestion
Feasible Region of a FF

Slack = Maximum allowed delay - $D_{AB}$
$Slack_A = Slack_B = Slack / 2$
Feasible Region of a FF (cont.)
Outline

- Introduction
- Problem Formulation
- Algorithms
- Experimental Results
- Conclusions
*Intersection Graph*

- Get the feasible regions of all FFs
- The intersection of feasible regions can be represented by an intersection graph

Feasible regions of flip-flops A~E

Intersection graph
Design Flow

1. Find all maximal cliques
2. Do Clique partition to Extract MBFFs
3. Assign MBFFs into preferred location

Converge

If No, go back to step 2.
Find all the Maximal Cliques

- Finding all the maximal cliques is NPC in general graph
  - However, it can be solved in polynomial time in the rectangle intersection graph
  - Solve by the sweep line algorithm
MBFF Extraction

- We want to extract the MBFFs by clique partitioning
  - Clique partitioning is a NP-Hard problem
- Different extraction strategies will affect
  - The number of clock sinks
  - The wirelength of signal nets
MBFF Extraction (cont.)

Cost of creating MBFF $\beta$

- $D(\beta)$: the merging possibility of FFs in $\beta$
- $B(\beta)$: the # bits of $\beta$
- Switching power of signal nets connected to $\beta$

$$cost(\beta) = \lambda \times \frac{D(\beta)}{B(\beta)} + (1-\lambda) \times \frac{\sum_{\text{net}_i \in \text{Net}(\beta)} \alpha_i \times \text{EstimatedWL}_i}{\sum_{\text{net}_i \in \text{Net}(\beta)} \alpha_i \times \text{OriginalWL}_i}$$

$\alpha_i$ is the switching rate of signal nets
Example of Extraction Algorithm

- Assume we have 1/2/4-bit MBFF in library
- There are two maximal cliques
  - $c_1 = \{1,2,3,6,7\}$, $c_2 = \{4,5,6\}$
  - Random sampling 1, 2 or 4 of FFs from $c_1, c_2$
  - $\beta_1 = \{1,2,3,6\}$, $\beta_2 = \{4,6\}$
  - $\text{cost}(\beta_1) < \text{cost}(\beta_2) \Rightarrow \text{select } \beta_1$
  - Re-sampling $\beta_1' = \{7\}$ from $c_1$
  - $\text{cost}(\beta_2) < \text{cost}(\beta_1')$
  - FF6 already covered
  - Re-sampling $\beta_2' = \{4,5\}$ from $c_2$
  - Final Extraction $\{\beta_1, \beta_2', \beta_1'\}$
MBFF Relocation

- For a MBFF $\beta$, we want to minimize the switching power of its signal nets.

$$\min_{\text{net}_i \in \text{Net}(\beta)} \sum_{\text{net}_i \in \text{Net}(\beta)} \alpha_i \times WL_i$$

$\alpha_i$ is the switching rate of signal nets.
- We can formulate it as a weighted median problem.
MBFF Relocation (cont.)

The weight of P1~P5 are 2:1:1:3:1

- Fanin/fanout of a MBFF
- Bin
- Preferred region
MBFF Relocation (cont.)

- Because of bin density constraints, some MBFFs cannot be placed in preferred region.
Outline

- Introduction
- Problem Formulation
- Algorithms
- Experimental Results
- Conclusions
Experimental Setup

- Implemented in C++
- Work on Linux with 2.13GHz CPU
- We have 9 test cases
  - r1~r5 from [22] *Exact Zero-Skew*
  - t0~t3 from 2010 CAD contest of Taiwan
  - Random generate switching rates 5%~15%
## Experimental Results

- Reduction of **clock sinks** and **wirelength of clock tree**

<table>
<thead>
<tr>
<th>Test cases</th>
<th>#FF</th>
<th>FF area reduction</th>
<th>Clock tree WL(nm)</th>
<th>Run time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>original</td>
<td>final</td>
<td>reduction</td>
<td>original</td>
</tr>
<tr>
<td>r1</td>
<td>267</td>
<td>101</td>
<td>62.17%</td>
<td>1325183</td>
</tr>
<tr>
<td>r2</td>
<td>598</td>
<td>223</td>
<td>62.70%</td>
<td>2621623</td>
</tr>
<tr>
<td>r3</td>
<td>862</td>
<td>298</td>
<td>65.42%</td>
<td>3357327</td>
</tr>
<tr>
<td>r4</td>
<td>1903</td>
<td>592</td>
<td>68.89%</td>
<td>6839628</td>
</tr>
<tr>
<td>r5</td>
<td>3101</td>
<td>921</td>
<td>70.29%</td>
<td>10145960</td>
</tr>
<tr>
<td>t0</td>
<td>120</td>
<td>37</td>
<td>69.16%</td>
<td>39637</td>
</tr>
<tr>
<td>t1</td>
<td>600000</td>
<td>15040</td>
<td>74.93%</td>
<td>3981765</td>
</tr>
<tr>
<td>t2</td>
<td>5524</td>
<td>1525</td>
<td>72.39%</td>
<td>985348</td>
</tr>
<tr>
<td>t3</td>
<td>953</td>
<td>246</td>
<td>74.18%</td>
<td>201755</td>
</tr>
<tr>
<td>avg.</td>
<td></td>
<td></td>
<td>68.90%</td>
<td></td>
</tr>
</tbody>
</table>
Reduction of wirelength and estimated switching power of nets connected to FFs

<table>
<thead>
<tr>
<th>Test cases</th>
<th>$\sum_{\text{nets} \in F} W_{Li}$ (original)</th>
<th>$\sum_{\text{nets} \in F} W_{Li}$ (final)</th>
<th>Reduction</th>
<th>$\sum_{\text{nets} \in F} \alpha_i \times W_{Li}$ (original)</th>
<th>$\sum_{\text{nets} \in F} \alpha_i \times W_{Li}$ (final)</th>
<th>Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>r1</td>
<td>1743703</td>
<td>1756925</td>
<td>-0.75%</td>
<td>179802</td>
<td>176661</td>
<td>1.74%</td>
</tr>
<tr>
<td>r2</td>
<td>3930879</td>
<td>3732569</td>
<td>5.04%</td>
<td>400928</td>
<td>370500</td>
<td>7.58%</td>
</tr>
<tr>
<td>r3</td>
<td>5672241</td>
<td>5191913</td>
<td>8.46%</td>
<td>574642</td>
<td>511426</td>
<td>11.00%</td>
</tr>
<tr>
<td>r4</td>
<td>12616681</td>
<td>12066921</td>
<td>4.35%</td>
<td>1266302</td>
<td>1187960</td>
<td>6.18%</td>
</tr>
<tr>
<td>r5</td>
<td>20528314</td>
<td>19012768</td>
<td>7.38%</td>
<td>2061324</td>
<td>1856472</td>
<td>9.93%</td>
</tr>
<tr>
<td>t0</td>
<td>83285</td>
<td>74365</td>
<td>10.71%</td>
<td>8755</td>
<td>7482</td>
<td>14.53%</td>
</tr>
<tr>
<td>t1</td>
<td>53624875</td>
<td>33077705</td>
<td>38.31%</td>
<td>5356145</td>
<td>3157927</td>
<td>41.04%</td>
</tr>
<tr>
<td>t2</td>
<td>3562985</td>
<td>2099595</td>
<td>41.07%</td>
<td>357151</td>
<td>204907</td>
<td>42.62%</td>
</tr>
<tr>
<td>t3</td>
<td>576710</td>
<td>448090</td>
<td>22.30%</td>
<td>58576</td>
<td>43931</td>
<td>25.00%</td>
</tr>
<tr>
<td>avg.</td>
<td>222804.5</td>
<td>170474.5</td>
<td>23.5%</td>
<td>215852.3</td>
<td>160682.8</td>
<td>26.02%</td>
</tr>
</tbody>
</table>
Comparison with [15]

- Our algorithm can be modified to target the objectives of [15]

<table>
<thead>
<tr>
<th>Test cases</th>
<th>#FFs</th>
<th>FF Power Red.</th>
<th>HPWL Red.</th>
<th>Run time(s)</th>
<th>FF Power Red.</th>
<th>HPWL Red.</th>
<th>Run time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>c1</td>
<td>98</td>
<td>14.8%</td>
<td>8.7%</td>
<td>0.01</td>
<td>15.64%</td>
<td>8.2%</td>
<td>0.01</td>
</tr>
<tr>
<td>c2</td>
<td>423</td>
<td>16.9%</td>
<td>5.3%</td>
<td>0.04</td>
<td>17.52%</td>
<td>11.1%</td>
<td>0.05</td>
</tr>
<tr>
<td>c3</td>
<td>1692</td>
<td>17.1%</td>
<td>5.2%</td>
<td>0.10</td>
<td>17.41%</td>
<td>11.5%</td>
<td>0.22</td>
</tr>
<tr>
<td>c4</td>
<td>5129</td>
<td>16.8%</td>
<td>5.5%</td>
<td>0.28</td>
<td>17.07%</td>
<td>11.5%</td>
<td>0.72</td>
</tr>
<tr>
<td>c5</td>
<td>10575</td>
<td>17.1%</td>
<td>5.1%</td>
<td>0.60</td>
<td>17.29%</td>
<td>13.4%</td>
<td>1.89</td>
</tr>
<tr>
<td>c6</td>
<td>169200</td>
<td>17.2%</td>
<td>5.1%</td>
<td>78.92</td>
<td>17.52%</td>
<td>11.8%</td>
<td>36.12</td>
</tr>
<tr>
<td>avg.</td>
<td></td>
<td>16.65%</td>
<td>5.82%</td>
<td></td>
<td>17.03%</td>
<td>11.25%</td>
<td></td>
</tr>
</tbody>
</table>
Conclusions

- We present a power-driven flip-flop merging and relocation approach to reduce the switching power consumption of the entire circuit.
Q&A

- Thanks for your attention