Clustering of Flip-Flops for Useful-Skew Clock Tree Synthesis

Chuan Yean Tan
Rickard Ewetz
Cheng-Kok Koh • Purdue University
• University of Central Florida
• Purdue University
Outline

• MBFF Introduction
• Previous MBFF/Clustering Solutions
• Bounded Arrival Time Useful Skew Tree
• Clustering of Flip-Flops Mechanics
• Proposed Methodology
• Evaluation & Experimental Results
• Summary
Outline

• MBFF Introduction
• Previous MBFF/Clustering Solutions
• Bounded Arrival Time Useful Skew Tree
• Clustering of Flip-Flops Mechanics
• Proposed Methodology
• Evaluation & Experimental Results
• Summary
Multi-Bit Flip Flop (MBFF)

<table>
<thead>
<tr>
<th>No. of Bits</th>
<th>1-Bit FF</th>
<th>2-Bit FF</th>
<th>4-Bit FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Per Bit</td>
<td>1.00</td>
<td>0.86</td>
<td>0.78</td>
</tr>
<tr>
<td>Area Per Bit</td>
<td>1.00</td>
<td>0.96</td>
<td>0.71</td>
</tr>
</tbody>
</table>
Benefits of utilizing MBFF

• Reduce **power consumption**
• Reduce **routing complexity** and **clock tree complexity**, less elements to connect leads to less routing complexity

<table>
<thead>
<tr>
<th>No. of Bits</th>
<th>1-Bit FF</th>
<th>2-Bit FF</th>
<th>4-Bit FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Per Bit</td>
<td>1.00</td>
<td>0.86</td>
<td>0.78</td>
</tr>
<tr>
<td>Area Per Bit</td>
<td>1.00</td>
<td>0.96</td>
<td>0.71</td>
</tr>
</tbody>
</table>

**MBFF Power And Area Savings[3]**

Outline

• MBFF Introduction
• Previous MBFF/Clustering Solutions
• Bounded Arrival Time Useful Skew Tree
• Clustering of Flip-Flops Mechanics
• Proposed Methodology
• Evaluation & Experimental Results
• Summary
Previous Works Summary

Key Methodologies of Previous Works:

• Timing-Driven Intersecting of Feasible Regions[14]
• Maximum Independent Set based on heuristic approach [2]
• x- and y- interval graphs to determine maximum clique [9]
• K-means Algorithm [15,10]

Timing-Driven Clustering of Flip-Flops

Feasible Region

Fanout’s slack

fanin$_i$

fanout$_i$

D

Q

CLK
Flip-Flop is allowed to moved with the region
Analysis of Previous Works

• Previous work cluster flip-flop without consideration of clock tree synthesis (CTS)
• Timing slacks are obtained pre-CTS, timing does not correlate as accurately after CTS
• Most modern designs adopt useful-skew tree to reduce area and power consumption
• However, once the UST has been built, updating or displacing FF’s placement is expensive!
Outline

• MBFF Introduction
• Previous MBFF/Clustering Solutions
• **Bounded Arrival Time Useful Skew Tree**
• Clustering of Flip-Flops Mechanics
• Proposed Methodology
• Evaluation & Experimental Results
• Summary
Setup Constraints: \( t_i + t_i^{\text{CQ}} + t_{ij}^{\text{max}} + t_j^S \leq t_j + T \)

Hold Constraints: \( t_i + t_i^{\text{CQ}} + t_{ij}^{\text{min}} \geq t_j + t_j^H \)
Setup Constraints: \[ t_i + t_i^{CQ} + t_{ij}^{\text{max}} + t_j^S \leq t_j + T \]

Hold Constraints: \[ t_i + t_i^{CQ} + t_{ij}^{\text{min}} \geq t_j + t_j^H \]

Constraints Formulation:
\[
\begin{align*}
    u_{ij} &= T - t_i^{CQ} - t_{ij}^{\text{max}} - t_j^S \\
    l_{ij} &= t_j^H - t_i^{CQ} - t_{ij}^{\text{min}}
\end{align*}
\]

Skew Constraints:
\[
\begin{align*}
    l_{ij} &\leq t_i - t_j \leq u_{ij} \\
    l_{ij} &\leq \text{skew}_{ij} \leq u_{ij}
\end{align*}
\]
Skew Constraint Graph (SCG)

Weighted Edges:
\[ w_{12} = u_{12} \]
\[ w_{21} = -l_{12} \]

From the skew constraints:
\[ t_1 - t_2 \leq u_{12} \]
\[ t_2 - t_1 \leq -l_{12} \]

Generalize weighted constraint:
\[ t_i - t_j \leq w_{ij} \]
Useful-Skew Tree based on Arrival Time Constraints [5]

Skew Constraint Graph (SCG)

Arrival Time Ranges

\[ l_{12} \leq t_1 - t_2 \leq u_{12} \]
\[ l_{13} \leq t_1 - t_3 \leq u_{13} \]
\[ l_{24} \leq t_2 - t_4 \leq u_{24} \]
\[ l_{34} \leq t_3 - t_4 \leq u_{34} \]

LP Formulation
No negative cycle from SCG

Useful-Skew Tree based on Arrival Time Constraints [5]

 Arrival Time Ranges

ff₁⁺ub
ff₁⁻lb
ff₂⁺ub
ff₂⁻lb
ff₃⁺ub
ff₃⁻lb
ff₄⁺ub
ff₄⁻lb

Clock Tree Construction

Outline

• MBFF Introduction
• Previous MBFF/Clustering Solutions
• Bounded Arrival Time Useful Skew Tree
• Clustering of Flip-Flops Mechanics
• Proposed Methodology
• Evaluation & Experimental Results
• Summary
Clustering of FFs Mechanics

$\text{FF}_i$ Merge FF Candidate 1

$\text{fanin}_i$

$\text{D}$

$\text{Q}$

$\text{CLK}$

$\text{fanout}_i$

$\text{fanin}_j$

$\text{D}$

$\text{Q}$

$\text{CLK}$

$\text{fanout}_j$

$\text{FF}_j$ Merge FF Candidate 2
Displacing FFs

Worst Case: FF’s displacement introduces increase in wirelength
Displacing FFs

Moving each FF will require update to SCG to avoid “out-dated” skew constraints!!
Displacing FFs

Furthermore, displacing FF\textsubscript{i} does not guarantee that all skew constraints will be met after update\textsubscript{1}
Timing Delays

Let $\Delta k = \text{timing delay for increase in wirelength}$
Timing Delays

\[ \Sigma \Delta k_i = \text{total timing delay increased from displacing FFs} \]
Comparing FF’s Arrival Time Ranges

Example of arrival time ranges derived from skew constraints
Merging Flip-Flop Candidates
Overlapping Arrival Time Ranges Between FFs
Updating New Arrival Time Range

Fast computation to update Arrival Time Ranges after merging FFs
Outline

• MBFF Introduction
• Bounded Arrival Time Useful Skew Tree
• Previous MBFF/Clustering Solutions
• Clustering of Flip-Flops Mechanics
• **Proposed Methodology**
• Evaluation & Experimental Results
• Summary
Proposed Methodology

1. Design Netlist
2. Specify Arrival Time Range
3. Select a FF Candidate Randomly
   - if not all FF evaluated
   - Cluster Flip-Flops Algorithm
4. Clock Tree Synthesis
Cluster Flip-Flops Algorithm

- New FF Candidate
  - Find Flip-Flop Nearest Neighbor
    - Arrival Time Ranges Overlaps?
      - YES: Compute Cluster Location
      - NO: Scaled Arrival Time Ranges Overlaps?
        - YES: Cluster Flip-Flops
Cluster Flip-Flops Algorithm

New FF Candidate

Find Flip-Flop Nearest Neighbor

Arrival Time Ranges Overlaps?

YES

Compute Cluster Location

NO

Scaled Arrival Time Ranges Overlaps?

YES

Cluster Flip-Flops

NO
Nearest Neighbor FF Selection Heuristic

Cluster Pair 1

Cluster Pair 2

Cluster Pair 3
Cluster Flip-Flops Algorithm

1. New FF Candidate
2. Find Flip-Flop Nearest Neighbor
3. Arrival Time Ranges Overlaps?
   - YES: Compute Cluster Location
   - NO: Scaled Arrival Time Ranges Overlaps?
     - YES: Cluster Flip-Flops
     - NO: Proceed to next step

Compute Clustering Location

Cluster Location Algorithm:
Weighted FF candidate based on FF’s fanout
Higher fanout → Higher capacitive load
Compute Clustering Location

Merge FF Candidate 1, $FF_i$
No. of Fanout = 5

Cluster Location

Merge FF Candidate 2, $FF_j$
No. of Fanout = 2

Displacing FF with **higher capacitive load** will incur **higher delay change**
Cluster Flip-Flops Algorithm

1. New FF Candidate
2. Find Flip-Flop Nearest Neighbor
3. Arrival Time Ranges Overlaps?
   - NO
   - Scaled Arrival Time Ranges Overlaps?
     - NO
     - YES
     - Compute Cluster Location
     - YES
     - Scaled Arrival Time Ranges Overlaps?
       - NO
       - YES
       - Cluster Flip-Flops

Arrival Time Range Overlaps

Overlapping Region
Updating Scaled Arrival Time Ranges

Scaled Arrival Time Range update after FF has been moved

Scaled from FF’s displacement
Cluster Flip-Flops Algorithm

New FF Candidate

Find Flip-Flop Nearest Neighbor

NO

Arrival Time Ranges Overlaps?

YES

Compute Cluster Location

NO

Scaled Arrival Time Ranges Overlaps?

YES

Cluster Flip-Flops
Outline

• MBFF Introduction
• Previous MBFF/Clustering Solutions
• Bounded Arrival Time Useful Skew Tree
• Clustering of Flip-Flops Mechanics
• Proposed Methodology
• Evaluation & Experimental Results
• Summary
Evaluation Framework

Monte Carlo Framework:
• Process Variation (10%)
• Voltage Variation (15%)
• Temperature Variation (30%)

Tree Structure:
• Zero-Skew Tree (ZST)
• Useful-Skew Tree (UST) [5]

[16] R. Ewetz et al. MCMM clock tree optimization based on slack redistribution using a reduced slack graph. ASP-DAC ’16
### Clustering FF Results

<table>
<thead>
<tr>
<th>Design</th>
<th>No. of FF</th>
<th>Total Capacitance (pF)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>USB Fast</td>
<td>1752</td>
<td>5.62</td>
<td>5.08</td>
<td>3.53</td>
</tr>
<tr>
<td>DMA</td>
<td>2121</td>
<td>6.94</td>
<td>5.56</td>
<td>4.31</td>
</tr>
<tr>
<td>openMSP430</td>
<td>686</td>
<td>2.66</td>
<td>1.98</td>
<td>1.44</td>
</tr>
<tr>
<td>Ethernet</td>
<td>10544</td>
<td>31.67</td>
<td>26.23</td>
<td>22.54</td>
</tr>
<tr>
<td>AES256</td>
<td>13216</td>
<td>40.47</td>
<td>33.75</td>
<td>25.09</td>
</tr>
<tr>
<td>PCI Bridge</td>
<td>3582</td>
<td>10.55</td>
<td>9.15</td>
<td>7.48</td>
</tr>
<tr>
<td>VGA enhanced</td>
<td>17071</td>
<td>49.89</td>
<td>41.08</td>
<td>34.83</td>
</tr>
</tbody>
</table>

**Benchmarks from IWLS 2005 [8]**

8 – 14% decrease in total capacitance

### Clustering FF + MBFF Transformation Results

<table>
<thead>
<tr>
<th>Design</th>
<th>No. of FF</th>
<th>Total Capacitance (pF)</th>
<th>UST[^{[1]}]</th>
<th>Cluster + MBFF + UST</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB Fast</td>
<td>1752</td>
<td>3.53</td>
<td>2.48</td>
<td></td>
</tr>
<tr>
<td>DMA</td>
<td>2121</td>
<td>4.31</td>
<td>2.85</td>
<td></td>
</tr>
<tr>
<td>openMSP430</td>
<td>686</td>
<td>1.44</td>
<td>1.10</td>
<td></td>
</tr>
<tr>
<td>Ethernet</td>
<td>10544</td>
<td>22.54</td>
<td>14.25</td>
<td></td>
</tr>
<tr>
<td>AES256</td>
<td>13216</td>
<td>25.09</td>
<td>17.8</td>
<td></td>
</tr>
<tr>
<td>PCI Bridge</td>
<td>3582</td>
<td>7.48</td>
<td>5.00</td>
<td></td>
</tr>
<tr>
<td>VGA enhanced</td>
<td>17071</td>
<td>34.83</td>
<td>22.92</td>
<td></td>
</tr>
</tbody>
</table>

\[^{[1]}\]: Design notation

\[^{[3]}\]: UST notation

20% – 34% decrease in total capacitance
Data & Clock Path Cap Reduction

Assuming *worst case* datapath wirelength increase

<table>
<thead>
<tr>
<th>Design[1]</th>
<th>No. of FF</th>
<th>Data Path Wire Cap Reduction (%)</th>
<th>Clock Path Wire Cap Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>USB Fast</td>
<td>1752</td>
<td>-4.17</td>
<td>13.25</td>
</tr>
<tr>
<td>DMA</td>
<td>2121</td>
<td>-9.86</td>
<td>19.47</td>
</tr>
<tr>
<td>openMSP430</td>
<td>686</td>
<td>-7.29</td>
<td>19.81</td>
</tr>
<tr>
<td>Ethernet</td>
<td>10544</td>
<td>-6.02</td>
<td>22.25</td>
</tr>
<tr>
<td>AES256</td>
<td>13216</td>
<td>-1.8</td>
<td>13.03</td>
</tr>
<tr>
<td>PCI Bridge</td>
<td>3582</td>
<td>-4.81</td>
<td>18.05</td>
</tr>
<tr>
<td>VGA enhanced</td>
<td>17071</td>
<td>-8.98</td>
<td>19.56</td>
</tr>
</tbody>
</table>

Clock Path WireCap Reduction >> Data Path Wire Cap Reduction
Outline

• MBFF Introduction
• Bounded Arrival Time Useful Skew Tree
• Previous MBFF/Clustering Solutions
• Clustering of Flip-Flops Mechanics
• Proposed Methodology
• Evaluation & Experimental Results
• Summary
Summary

- Displacing FFs requires update to skew constraints to ensure timing is met
- Clustering of Flip-Flops based on Bounded Arrival Time Range Clock Tree Synthesis provides fast update to the skew constraints while ensuring timing correctness
- Clustering of FFs reduces routing complexity
- MBFF transformation further reduces power consumption
Backup Slides
Bounded Arrival Time Ranges

Resulting Merge FF Arrival Time Range
Skew Constraint Graph (SCG)

\[ l_{12} \leq t_1 - t_2 \leq u_{12} \]

\[ l_{13} \leq t_1 - t_3 \leq u_{13} \]

\[ l_{24} \leq t_2 - t_4 \leq u_{24} \]

\[ l_{34} \leq t_3 - t_4 \leq u_{34} \]