A Novel Technology Mapper for Complex Universal Gates

Meng-Che Wu¹, Ai Quoc Dao¹, Mark Po-Hung Lin²

¹ Department of EE & AIM-HI, National Chung Cheng University, Chiayi, Taiwan
² AI College and Institute of Electronics National Chiao Tung University, Taiwan

w10041656@gmail.com, aiquocvt@gmail.com, mphlin@nctu.edu.tw
Outline

• Introduction
• Main Contribution
• Problem Formulation
• Proposed Algorithm
• Experimental Results
• Conclusions
Outline

• Introduction
• Main Contribution
• Problem Formulation
• Proposed Algorithm
• Experimental Results
• Conclusions
**Introduction**

**Complex universal logic gates**

- Implement various complicated Boolean functions with higher density and flexibility
- Apply to functional engineering change order (ECO), structural application-specific integrated circuits (ASICs) field programmable gate arrays (FPGAs), and other cost-effective or security-oriented VLSI design
Technology Mapping

- Standard cells
- LUT-based FPGAs

⇒ cannot achieve optimal area and delay when converting a combinational circuit into a gate-level netlist with complex universal gates

❖ *This work focuses on the challenging technology mapping problem for complex universal gates*
Outline

• Introduction
• **Main Contribution**
• Problem Formulation
• Proposed Algorithm
• Experimental Results
• Conclusions
Main Contribution

- **Boolean network simulation** finds the relationship between simulation patterns and functions, as well as the don’t care inputs (if any) on complex universal gates;
- **Permutation classification** speeds up the Boolean network simulation by classifying simulation patterns with input permutations;
- **Supergate library construction** represents all functions which can be presented by the given complex universal gates;
- **Cut enumeration** calculates all possible cuts for each node in the subject graph using dynamic programming, and filters out redundant cuts with hash function,
- **Boolean matching** determines the proper universal gates to implement the cut for each node in the subject graph of a Boolean network;
- **Universal cell covering** replaces the matched cut with the best universal gates.
Outline

• Introduction
• Main Contribution
• Problem Formulation
• Proposed Algorithm
• Experimental Results
• Conclusions
Definition 3.1. A complex universal gate is a large logic gate with \( m \) inputs, where \( m \) is usually much larger than the input numbers of basic logic gates.

\[
F_1 = A(B + DE) + C(D + BE)
\]

\[
F_2 = S_0 S_1 D + S_0 \bar{S}_1 C + \bar{S}_0 S_1 B + \bar{S}_0 \bar{S}_1 A
\]

\[
F_3 = (A \bar{S}_0 + B S_0)(S_2 + S_3) + (C \bar{S}_1 + D S_1)(S_2 + S_3)
\]
Problem Formulation (cont.)

• Input:
  – Boolean network $N$
  – A library $L$, a set of universal cells

• Objective:
  – Find the mapping network which have lowest cost calculated by equation (1)

• Solutions:
  – Supergate library constructions for complex universal gates
  – Technology mapping based on supergate library

$$TotalCost = TotalArea \times CriticalPath \quad (1)$$
Outline

• Introduction
• Main Contribution
• Problem Formulation
• Proposed Algorithm
  – Supergate library constructions for complex universal gates
  – Technology mapping based on supergate library
• Experimental Results
• Conclusions
Supergate Library Constructions for Complex Universal Gates

• Boolean network construction
• Input pattern generation
• Boolean network simulation with permutation classification
• Supergate library construction [1]

• Boolean network construction

\[ F_1 = A(B + DE) + C(D + BE) \]

An example Boolean network representing the Boolean function of \( F_1 \)
• Input pattern generation
• Boolean network simulation with permutation classification

The function of all node after Boolean network simulation based on the simulation pattern \((X_0, 0, X_2, 1, X_1)\).
Supergate Library Constructions for Complex Universal Gates

- Supergate library construction
  - A super gate is a single output combinational network of a few library gates

The number of feasible functions for three complex universal gates

<table>
<thead>
<tr>
<th>Universal cells</th>
<th># Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_1$</td>
<td>149</td>
</tr>
<tr>
<td>$F_2$</td>
<td>1006</td>
</tr>
<tr>
<td>$F_3$</td>
<td>37</td>
</tr>
</tbody>
</table>
Technology Mapping based on the Supergate Library

- Cut enumeration [2]
- Boolean matching [1]
- Cell covering: Finding the mapped network from the best matched complex universal gates according to the cost function


[2] A. Mishchenko, S. Chatterjee, and R. K. Brayton, Improvements to technology mapping for LUT-based FPGAs, TCAD 07
Cell covering

• Finding the mapped network from the best matched complex universal gates according to the cost function
• The delay optimal covering using dynamic programming is proposed
• The calculating function of delay time for the cut, c, is defined as follows

\[ Delay - TIME(c) = PCD + \max_{r \in c} Delay[r] \]

PCD : the delay of the complex universal gate, which is used to implement the cut, c
Delay[r ] : the map, that can find the cost of the best cut for the node r
Outline

- Introduction
- Main Contribution
- Problem Formulation
- Proposed Algorithm
- Experimental Results
- Conclusions
Experimental Setup

• Programming language: C++
• Platform: 2.60-GHz Intel i9-7980XE
• Benchmarks: 1) The benchmarks are provided by a research institute of the national lab (TSRI); 2) ISCAS’85 benchmarks; 3) ISCAS’89 benchmarks
• The results were verified by the Cadence Conformal tool
Experimental Results

Comparison of the proposed technology mapper against the ABC mapper on TSRI benchmarks

<table>
<thead>
<tr>
<th>Circuit name</th>
<th>Circuit information</th>
<th>ABC mapper</th>
<th>Time (s)</th>
<th>Our algorithm</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Area</td>
<td>Delay</td>
<td>Cost</td>
<td>Time</td>
</tr>
<tr>
<td>Circuit 1</td>
<td>14</td>
<td>446</td>
<td>15</td>
<td>6690</td>
<td>2</td>
</tr>
<tr>
<td>Circuit 2</td>
<td>233</td>
<td>451</td>
<td>10</td>
<td>4510</td>
<td>1</td>
</tr>
<tr>
<td>Circuit 3</td>
<td>50</td>
<td>588</td>
<td>16</td>
<td>9408</td>
<td>1</td>
</tr>
<tr>
<td>Circuit 4</td>
<td>178</td>
<td>765</td>
<td>13</td>
<td>9945</td>
<td>1</td>
</tr>
<tr>
<td>Circuit 5</td>
<td>32</td>
<td>1189</td>
<td>40</td>
<td>47560</td>
<td>3</td>
</tr>
<tr>
<td>Geomean</td>
<td></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

## Experimental Results

Comparison of the proposed technology mapper against the ABC mapper on ISCAS’85 and ISCAS’89 benchmarks

<table>
<thead>
<tr>
<th>Circuit name</th>
<th>Circuit information</th>
<th>ABC mapper</th>
<th>Time (s)</th>
<th>Our algorithm</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># PIs</td>
<td>#POs</td>
<td># gates</td>
<td>Area</td>
<td>Delay</td>
</tr>
<tr>
<td>c5315</td>
<td>178</td>
<td>123</td>
<td>2307</td>
<td>730</td>
<td>12</td>
</tr>
<tr>
<td>c6288</td>
<td>32</td>
<td>32</td>
<td>2416</td>
<td>1181</td>
<td>40</td>
</tr>
<tr>
<td>c7552</td>
<td>207</td>
<td>108</td>
<td>3513</td>
<td>849</td>
<td>12</td>
</tr>
<tr>
<td>s5378</td>
<td>35</td>
<td>49</td>
<td>2779</td>
<td>696</td>
<td>7</td>
</tr>
<tr>
<td>s9234</td>
<td>19</td>
<td>22</td>
<td>5597</td>
<td>966</td>
<td>12</td>
</tr>
<tr>
<td>s13207</td>
<td>31</td>
<td>121</td>
<td>7951</td>
<td>1643</td>
<td>11</td>
</tr>
<tr>
<td>s15850</td>
<td>14</td>
<td>87</td>
<td>9772</td>
<td>1851</td>
<td>16</td>
</tr>
<tr>
<td>s38417</td>
<td>28</td>
<td>106</td>
<td>22179</td>
<td>5061</td>
<td>12</td>
</tr>
<tr>
<td>s38584</td>
<td>12</td>
<td>278</td>
<td>19253</td>
<td>6183</td>
<td>12</td>
</tr>
<tr>
<td>Geomean</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Outline

• Introduction
• Main Contribution
• Problem Formulation
• Proposed Algorithm
• Experimental Results
• Conclusions
Conclusions

• Introduced a new technology mapper for complex universal gates
  – Boolean network simulation with permutation classification
  – Supergate library construction
  – Dynamic programming based cut enumeration
  – Boolean matching with optimal universal cell covering

• Showed experimental results on industrial benchmarks
  – Our technology mapper can achieve optimal solutions with much lower costs compared to the state-of-the-art academic technology mapper in terms of area and delay
Thanks for Your Attention!