High Level Equivalence
Symmetric Input Identification

Ming-Hong Su, Chun-Yao Wang
National Tsing Hua University, Taiwan
Outline

- Introduction
- Previous Work
  - BDD-Based
  - Simulation-Based
- E-Symmetry Detection Algorithm
- Experimental Results
- Conclusions
(1) Nonequivalence symmetry \( \Rightarrow f_{x_i x_j} = f_{x_i x_j} \), denoted as \( \text{NE}(x_i, x_j) \)

(2) Equivalence symmetry \( \Rightarrow f_{x_i x_j} = f_{x_i x_j} \), denoted as \( \text{E}(x_i, x_j) \)

Formulation

• Input:
  Circuit (N-input, M-output)

• Output:
  Maximal equivalence symmetric input sets
Outline

- Introduction
- Previous Work
  - BDD-Based
  - Simulation-Based
- E-Symmetry Detection Algorithm
- Experimental Results
- Conclusions
BDD-based Approach

- If the ROBDD of $f_{x_i x_j}$ and $f_{x_i \overline{x_j}}$ are isomorphic, then $x_i$ and $x_j$ are NE-symmetry.
- If the ROBDD of $f_{x_i x_j}$ and $f_{\overline{x_i} x_j}$ are isomorphic, then $x_i$ and $x_j$ are E-symmetry.

![NE-symmetry Diagram]

![E-symmetry Diagram]
Limitations

- For the design whose corresponding BDD cannot be built, BDD-based approaches cannot be applied
- Time of building BDD depends on the ordering of inputs
  - Optimal ordering is NP-complete
Simulation-based Approach

- Without BDD construction
  - Applicable to behavior level or RT-level

Start

Pattern Generation

Pattern Simulation

Symmetric Input Identification

Stop
Difficulties

- Generating and simulating complete patterns is time-consuming
  - Complexity is $O(2^n)$
- Comparing all patterns to obtain the symmetric relations among all inputs is intractable
  - Complexity is $O(2^n \times 2^n)$
Outline

- Introduction
- Previous Work
  - BDD Based
  - Simulation Based
- E-Symmetry Detection Algorithm
- Experimental Results
- Conclusions
Overview

- Identify two inputs as E-asymmetric is easier than to identify two inputs as E-symmetric
- Based on “negative thinking” to distinguish as many E-asymmetric inputs as possible
- The remaining inputs are possibly E-symmetric inputs
E-asymmetric Inputs (1/2)

- Legal Pattern Pair: A pair of patterns whose assignments are identical except on inputs $x_i$ and $x_j$, and $x_i = x_j$ (00 or 11) in each pattern
  - For any two inputs in an N-inputs circuit, there are $2^{N-2}$ legal pattern pairs

- Two inputs are E symmetric inputs while the outputs of each legal pattern pair are identical. Otherwise they are E - asymmetric inputs
### E-asymmetric Inputs (2/2)

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Input A and input B are E-asymmetric inputs.

Input A and input C are possibly E-symmetric inputs.
Variable Pair

- Variable Pair (VP): A pair of variables $x_i$ and $x_j$ is denoted as $VP(x_i, x_j)$ if they have not been recognized as symmetric or asymmetric.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- $VP(A, B)$
- $VP(A, C)$
Symmetric-Asymmetric Inputs (SASIs)

- SASIs: represent the maximal symmetric input sets
  - If any two inputs are not in the same group, then they are E-asymmetric inputs
  - If any two inputs are in the same group, then they are “probably” E-symmetric inputs
  - Example: for a 6-input circuit, the SASIs representation (12356)(4) indicates that inputs 1,2,3,5,6 are probably E-symmetric inputs and input 4 is E-asymmetric input to the other inputs
  - If SASIs representation could be divided into 6 groups, then all inputs are E-asymmetric inputs

Reference: C.L. Chou"Identification of maximal symmetric inputs sets using graph automorphism algorithm", 2004
VPs and SASIs

- Grouping all VPs to form the corresponding SASIs
  - Example: a 10-input circuit with 6 VPs \{(1,2),(2,3),(3,4),(5,6), (6,7),(8,9)\}

\[
\text{VPs}\{(1,2),(2,3),(3,4),(5,6),(6,7),(8,9)\}
\]

\[
(1234) \quad (567)
\]

\[
\text{SASIs} = (1234)(567)(89)(A)
\]
MEG and SEG

- MEG (Multiple Element Group): A group contains more than one element
- SEG (Single Element Group): A group contains only one element

SASIs=(1234)(567)(89)(A)
Distance (1/2)

- The distance of VP($x_i$, $x_j$) in an MEG is the difference of relative position of $x_i$ and $x_j$.

For MEG (1234567):
- Distance of VP(1,7) is 6.

For MEG (13579BD):
- Distance of VP(3,B) is 4.
Distance (2/2)

- For an MEG with $K$ elements, the maximal distance is $(K-1)$ and the number of VPs with distance $i$ is $(K - i)$

MEG (123456)

- 5 VPs with distance 1
- 4 VPs with distance 2
- 3 VPs with distance 3
- 2 VPs with distance 4
- 1 VPs with distance 5
Pattern Set

- A pattern with N bits, the set that consists of all patterns with m 1s and \((N - m)\) 0s is denoted as \(\theta^N_m\).
- Ex: \(\theta^5_1 = \{ 10000, 01000, 00100, 00010, 00001 \}\)
## Circular Pattern Set

<table>
<thead>
<tr>
<th>$\alpha_{1,1}^5 = \theta_1^5$</th>
<th>$\alpha_{2,1}^5$</th>
<th>$\alpha_{3,1}^5$</th>
<th>$\alpha_{4,1}^5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000 {1}</td>
<td>11000 {1,2}</td>
<td>11100 {1,2,3}</td>
<td>11110 {1,2,3,4}</td>
</tr>
<tr>
<td>01000 {2}</td>
<td>01100 {2,3}</td>
<td>01110 {2,3,4}</td>
<td>01111 {2,3,4,5}</td>
</tr>
<tr>
<td>00100 {3}</td>
<td>00110 {3,4}</td>
<td>00111 {3,4,5}</td>
<td>10111 {3,4,5,1}</td>
</tr>
<tr>
<td>00010 {4}</td>
<td>00011 {4,5}</td>
<td>10011 {4,5,1}</td>
<td>11011 {4,5,1,2}</td>
</tr>
<tr>
<td>00001 {5}</td>
<td>10001 {5,1}</td>
<td>11001 {5,1,2}</td>
<td>11101 {5,1,2,3}</td>
</tr>
<tr>
<td></td>
<td>$\alpha_{2,2}^5$</td>
<td>$\alpha_{3,2}^5$</td>
<td></td>
</tr>
<tr>
<td>10100 {1,3}</td>
<td>11010 {1,2,4}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01010 {2,4}</td>
<td>01101 {2,3,5}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>00101 {3,5}</td>
<td>10110 {3,4,1}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10010 {4,1}</td>
<td>01011 {4,5,2}</td>
<td></td>
<td></td>
</tr>
<tr>
<td>01001 {5,2}</td>
<td>10101 {5,1,3}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Theorem

- For an MEG with K elements, circular pattern sets \( \{ \alpha_{m,1}^K, \alpha_{m+2,i}^K \} \) could be used to recognize VPs with distance \( i \) and \( (K - i) \).
Example (1/2)

- For a 10-input circuit, initial SASIs=(123456789A)

- Step 1: generate \( \{ \theta_0^{10}, \theta_2^{10} \} \) and compare
  - assume VPs\{\( (1,2),(2,3),(3,4),(4,5),(5,6),(6,7) \)\} could not be recognized as asymmetric VPs

- Updated SASIs = (1234567)(8)(9)(A)
Example (2/2)

- Step 2: SASIs=(1234567)(8)(9)(A)
  - For the MEG (1234567)
    - Generate $\alpha_{1,1}^7 \Rightarrow \{(1), (2), (3), (4), (5), (6), (7)\}$ in $\theta_1^7$
    - Generate $\alpha_{3,1}^7, \alpha_{3,2}^7, \alpha_{3,3}^7$ in $\theta_3^7$
  - Others are randomly assigned
  - Comparing $(\alpha_{1,1}^7, \alpha_{3,1}^7)$ covers VPs with distance 1 and 6
  - Comparing $(\alpha_{1,1}^7, \alpha_{3,2}^7)$ covers VPs with distance 2 and 5
  - Comparing $(\alpha_{1,1}^7, \alpha_{3,3}^7)$ covers VPs with distance 3 and 4
  - Updated SASIs = (1)(2)(3)(4)(5)(6)(7)(8)(9)(A)
Flowchart

Start

i = 0

All asymmetric or i > bound

No

Yes

Pattern Generation

i++

Pattern Simulation

Update SASIs

Stop
Outline

- Introduction
- Previous Work
  - BDD-Based
  - Simulation-Based
- E-Symmetry Detection Algorithm
- Experimental Results
- Conclusions
Experiment Setup

- ISCAS’85 benchmarks in Verilog HDL
- SUN SPARC II workstation
- Compared with [10]

## Experimental Results

<table>
<thead>
<tr>
<th>circuit</th>
<th># in</th>
<th># out</th>
<th>Time (s)</th>
<th>Symmetry pair</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Reading</td>
<td>[10]</td>
</tr>
<tr>
<td>c880</td>
<td>60</td>
<td>26</td>
<td>11.57</td>
<td>0.03</td>
</tr>
<tr>
<td>c1355</td>
<td>41</td>
<td>32</td>
<td>1.30</td>
<td>0.05</td>
</tr>
<tr>
<td>c1908</td>
<td>33</td>
<td>25</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>c432</td>
<td>36</td>
<td>7</td>
<td>--</td>
<td>--</td>
</tr>
<tr>
<td>c499</td>
<td>41</td>
<td>32</td>
<td>1.17</td>
<td>0.05</td>
</tr>
<tr>
<td>c3540</td>
<td>50</td>
<td>22</td>
<td>18.96</td>
<td>0.08</td>
</tr>
<tr>
<td>c5315</td>
<td>178</td>
<td>123</td>
<td>&gt;1hr</td>
<td>0.02</td>
</tr>
<tr>
<td>c2670</td>
<td>233</td>
<td>140</td>
<td>&gt;1hr</td>
<td>0.08</td>
</tr>
<tr>
<td>c7552</td>
<td>207</td>
<td>108</td>
<td>&gt;1hr</td>
<td>0.17</td>
</tr>
<tr>
<td>c6288</td>
<td>32</td>
<td>32</td>
<td>--</td>
<td>--</td>
</tr>
</tbody>
</table>
Outline

- Introduction
- Previous Work
  - BDD Based
  - Simulation Based
- E-Symmetry Detection Algorithm
- Experimental Results
- Conclusions
Conclusions

- Simulation with randomly generated patterns is inefficient due to many redundant patterns are generated for recognized asymmetric VPs
- Propose a systematic pattern generation algorithm to identify E-symmetric inputs