High Level Equivalence Symmetric Input Identification

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

Outline

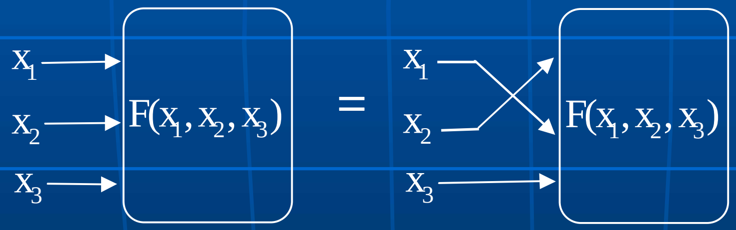
- Introduction
- Previous Work
  - BDD-Based
  - Simulation-Based
- E-Symmetry Detection Algorithm
- Experimental Results
- Conclusions

Symmetries

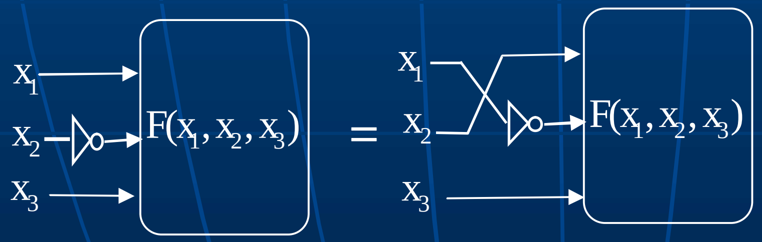
(1) Nonequivalence symmetry $\Rightarrow f_{\bar{x}_i x_j} = f_{x_i \bar{x}_j}$, denoted as $NE(x_i, x_j)$

(2) Equivalence symmetry $\Rightarrow f_{\bar{x}_i \bar{x}_j} = f_{x_i x_j}$, denoted as $E(x_i, x_j)$

(1)



(2)



Reference:C.C. Tasi "Boolean Matching Using Generalized Reed-Muller Forms", 1994

Formulation

- Input:

  Circuit (N-input, M-output)

- Output:

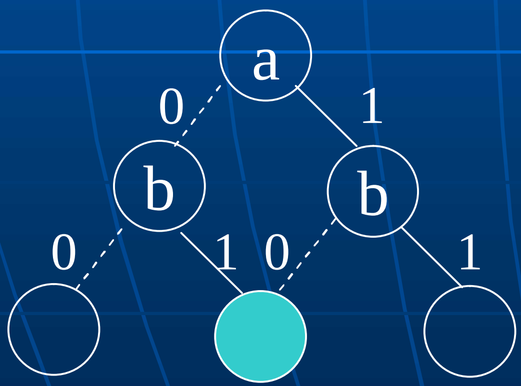
  Maximal equivalence symmetric input sets

Outline

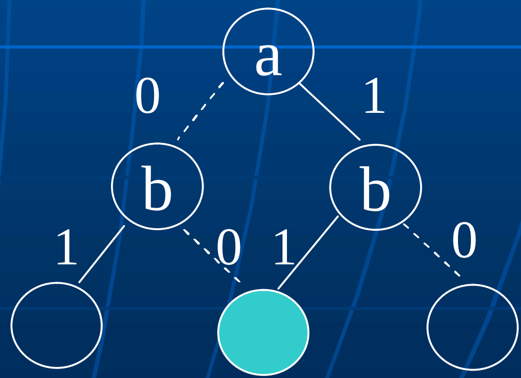
- Introduction
- Previous Work
  - BDD-Based
  - Simulation-Based
- E-Symmetry Detection Algorithm
- Experimental Results
- Conclusions

BDD-based Approach

- If the ROBDD of $f_{\bar{x}_i x_j}$ and $f_{x_i \bar{x}_j}$ are isomorphic, then x_i and x_j are NE-symmetry
- If the ROBDD of $f_{x_i x_j}$ and $f_{\bar{x}_i \bar{x}_j}$ are isomorphic, then x_i and x_j are E-symmetry



NE-symmetry



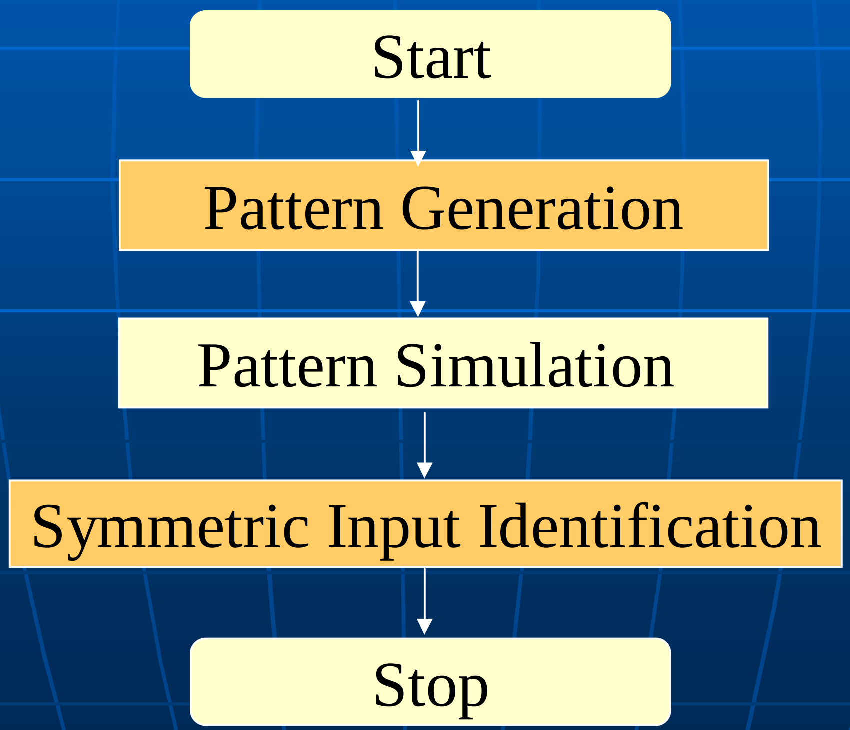
E-symmetry

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



Difficulties

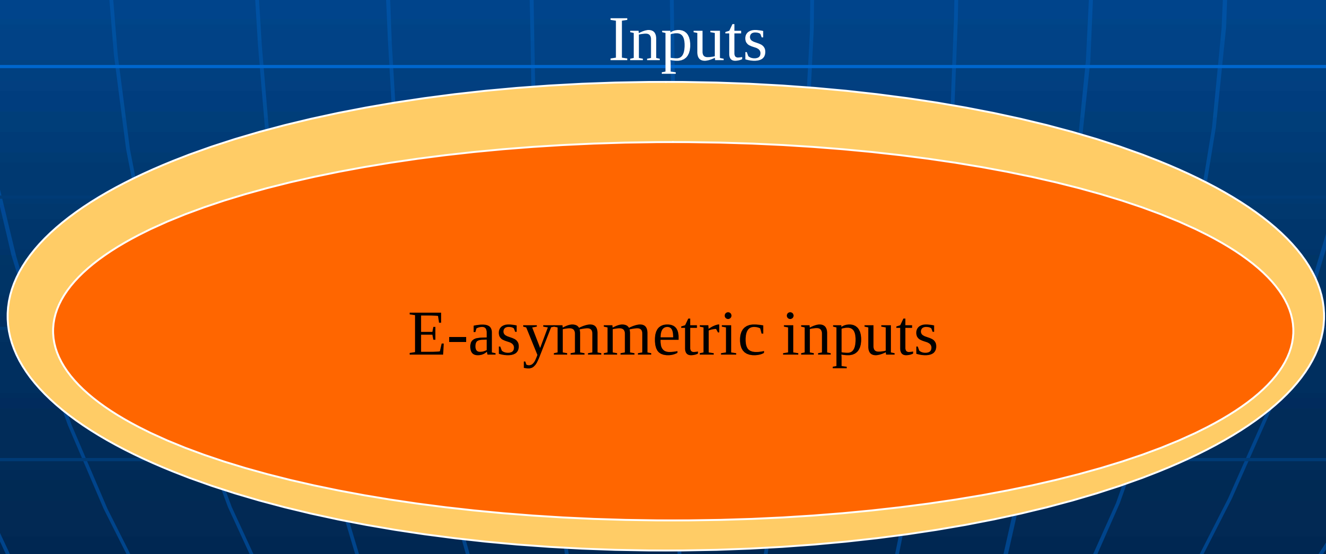
- Generating and simulating complete patterns is time-consuming
  - Complexity is $O(2^n)$
- Comparing all patterns to obtain the symmetric relations among all inputs is intractable
  - Complexity is $O(2^n \times 2^n)$

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)

Input				Output	
A	B	C	D	f1	f2
0	0	0	0	0	1
1	1	0	0	1	0
1	0	1	0	0	1
1	0	0	1	1	1
0	1	1	0	0	0
0	1	0	1	1	1
0	0	1	1	1	0
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

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

Input				Output	
A	B	C	D	f1	f2
0	0	0	0	0	0
1	1	0	0	0	0
1	0	1	0	0	0
1	0	0	1	0	0
0	1	1	0	0	0
0	1	0	1	0	0
0	0	1	1	0	0
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

VP(A,B)

VP(A,C)

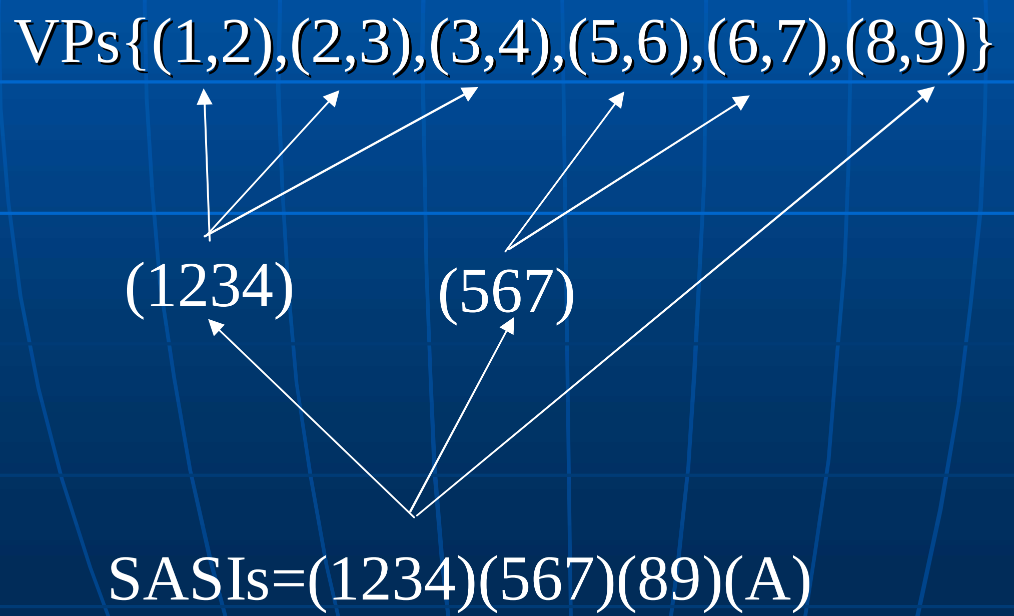
Symmetric-Asymmetric Inputs (SASIs)

- 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 automprphism algorithm",2004

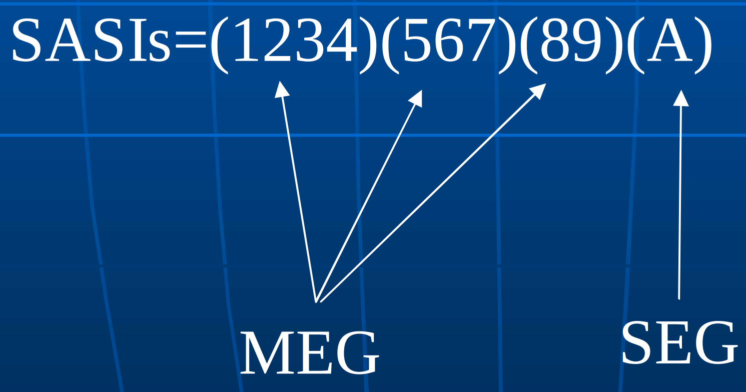
VPs and SASIs

- 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)}



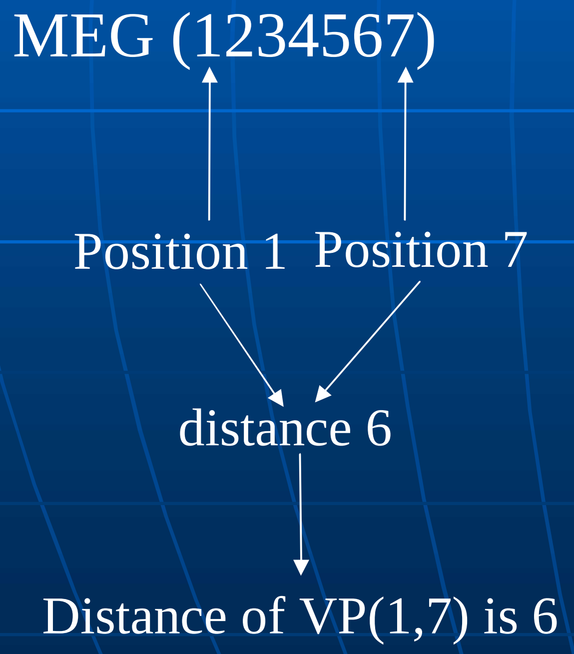
MEG and SEG

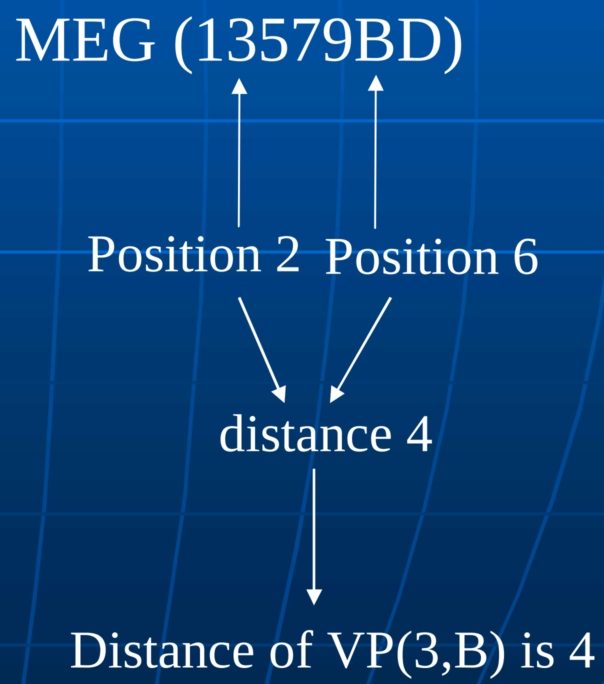
- MEG (Multiple Element Group): A group contains more than one element
- SEG (Single Element Group): A group contains only one element



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

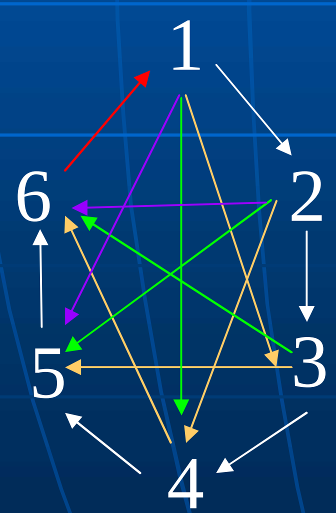




Distance (2/2)

- 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 θ_m^N
- Ex: $\theta_1^5 = \{$ 10000,
  01000,
  00100,
  00010,
  00001 $\}$

Circular Pattern Set

$\alpha^5_{1,1} = \theta^5_1$		$\alpha^5_{2,1}$		$\alpha^5_{3,1}$		$\alpha^5_{4,1}$	
10000	{1}	11000	{1,2}	11100	{1,2,3}	11110	{1,2,3,4}
01000	{2}	01100	{2,3}	01110	{2,3,4}	01111	{2,3,4,5}
00100	{3}	00110	{3,4}	00111	{3,4,5}	10111	{3,4,5,1}
00010	{4}	00011	{4,5}	10011	{4,5,1}	11011	{4,5,1,2}
00001	{5}	10001	{5,1}	11001	{5,1,2}	11101	{5,1,2,3}
		$\alpha^5_{2,2}$		$\alpha^5_{3,2}$			
		10100	{1,3}	11010	{1,2,4}		
		01010	{2,4}	01101	{2,3,5}		
		00101	{3,5}	10110	{3,4,1}		
		10010	{4,1}	01011	{4,5,2}		
		01001	{5,2}	10101	{5,1,3}		

Theorem

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

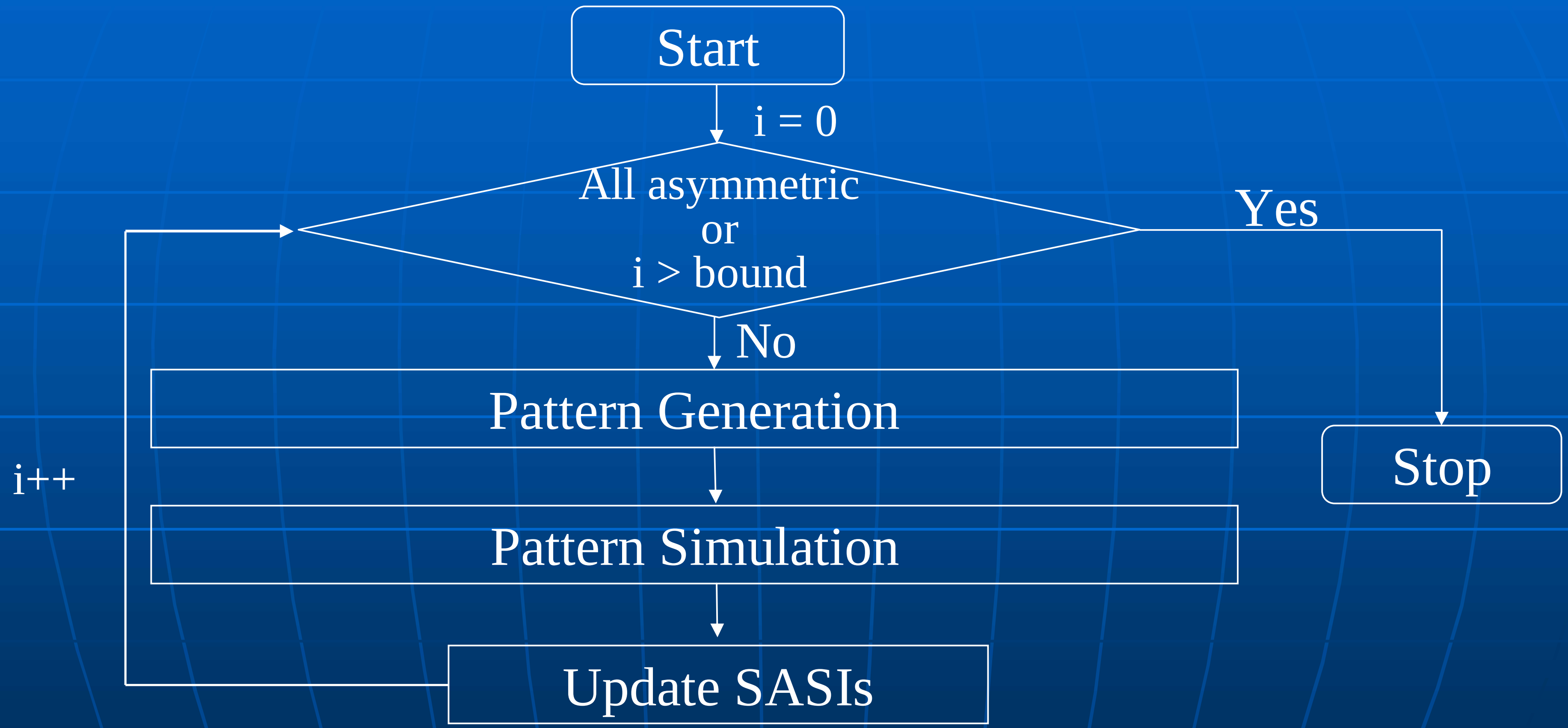
Example (1/2)

- 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 $\alpha^{7}_{1,1} \Rightarrow \{(1),(2),(3),(4),(5),(6),(7)\}$ in θ^{7}_{1}
    - Generate $\alpha^{7}_{3,1}$, $\alpha^{7}_{3,2}$, $\alpha^{7}_{3,3}$ in θ^{7}_{3}
  - Others are randomly assigned
  - Comparing $(\alpha^{7}_{1,1}, \alpha^{7}_{3,1})$ covers VPs with distance 1 and 6
  - Comparing $(\alpha^{7}_{1,1}, \alpha^{7}_{3,2})$ covers VPs with distance 2 and 5
  - Comparing $(\alpha^{7}_{1,1}, \alpha^{7}_{3,3})$ 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
Yes
No
Pattern Generation
Stop
i++
Pattern Simulation
Update SASIs

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]

[10] K.H. Wang, "Symmetry Detection for Imcompletely Specified Functions with K-Disjointness Paradigm," ASPDAC'05

Experimental Results

circuit	# in	# out	Time (s)			Symmetry pair	
			Reading	[10]	ours	[10]	ours
c880	60	26	11.57	0.03	1.75	0	0
c1355	41	32	1.30	0.05	0.68	0	0
c1908	33	25	--	--	0.28	--	0
c432	36	7	--	--	0.19	--	0
c499	41	32	1.17	0.05	0.66	0	0
c3540	50	22	18.96	0.08	2.42	0	0
c5315	178	123	>1hr	0.02	49.38	0	0
c2670	233	140	>1hr	0.08	593.04	28	227
c7552	207	108	>1hr	0.17	633.61	6	160
c6288	32	32	--	--	0.25	--	0

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