Compaction of Pass/Fail-based Diagnostic Test Vectors for Combinational and Sequential Circuits

Yoshinobu Higami, Hiroshi Takahashi,
Shin-ya Kobayashi and Yuzo
Takamatsu（Ehime University）

Kewal K. Saluja
（Univ. of Wisconsin-Madison）

Department of
Computer Science
Ehime University

Outline

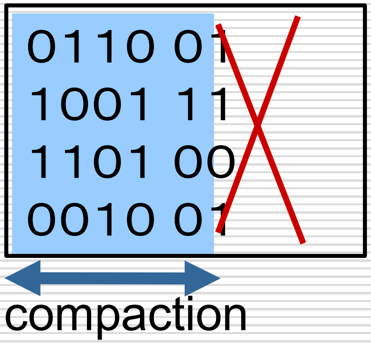
- Background
- Purpose
- Compaction for combinational circuits
  - Proposed method
  - Experimental results
- Compaction for sequential Circuits
  - Proposed method
  - Experimental results
- Conclusions

Background

- Increase of test cost
  - Expensive tester equipments
  - Long test application time
- Test compaction
  - Reduce test vectors
  - Reduce test application time
  - Reduce the memory space of test data
- Fault diagnosis
  - Locate physical defects
  - Improve design and manufacturing process

Test compaction for fault diagnosis

- **Benefits**
  - **Execution time of fault diagnosis is reduced**
  - **The memory space of test data stored on a tester is reduced**
  - **The size of fault dictionary is reduced**



0110 01
1001 11
1101 00
0010 01

Fault diagnosis of circuits failed in BIST

- **Fault detection in BIST**
  - **A large number of test vectors are applied.**
  - **Output responses are highly compacted.**
    - **Neither time information nor space information is obtained.**
- **Requirements for fault diagnosis**
  - **Information of failing vectors and passing vectors**

Extract a small number of test vectors for fault diagnosis

Purpose

- Reduce the number of test vectors from a given test set or a test sequence
- Constraints
  - Keep the number of distinguished fault pairs
  - Only use pass/fail information
    - No information of location of faulty primary outputs and faulty scan flip-flops
- Targets
  - Stuck-at faults
  - Combinational circuits and non-scanned sequential circuits

Compaction for combinational circuits

- Problem formulation
  - Given: a test set
  - Output: small number of test vectors
  - Constraint: Keep the number of distinguished fault pairs
- Ideal solution
  - If a fault distinguishing table is available, then the problem can be solved as a minimum set cover problem
- Difficulties
  - The number of fault pairs is huge.
  - Difficult to use a complete fault distinguishing table

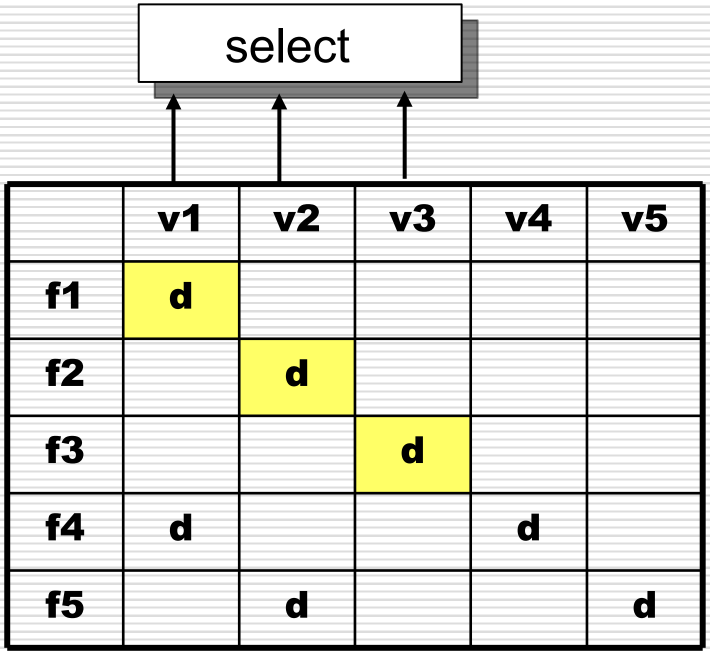
Fault distinguishing table

	v1	v2	v3	v4
pair1	D			
pair2		D	D	
pair3			D	D
pair4	D	D		

D: distinguished

Proposed algorithm

- First selection
  - Test vectors that detect ***once-detected faults***
  - Once-detected fault is a fault that is detected by only one test vector



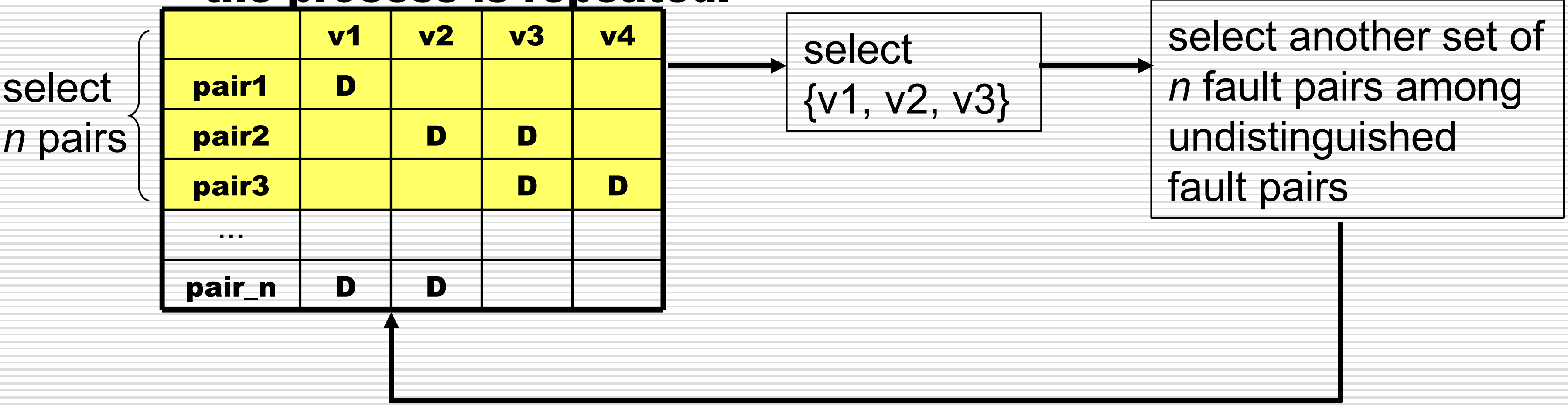
select

	v1	v2	v3	v4	v5
f1	d				
f2		d			
f3			d		
f4	d			d	
f5		d			d

d: detected

Proposed algorithm

- □ **Use of a partial fault distinguishing table**
  - **A partial distinguishing table includes information about only a subset of fault pairs**
- □ **Repeat**
  - **Make a distinguishing table for *n* fault pairs**
  - **Select test vectors that distinguish the selected n fault pairs**
  - **If the original diagnostic resolution is not achieved, then the process is repeated.**



select
n pairs

	v1	v2	v3	v4
pair1	D			
pair2		D	D	
pair3			D	D
…				
pair_n	D	D		

select
{v1, v2, v3}

select another set of
n fault pairs among
undistinguished
fault pairs

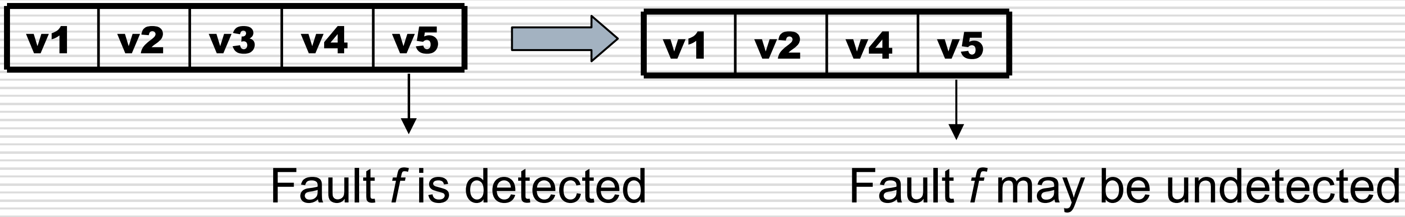
Experimental results

circuit	coverage	pair	vectors	CPU(s)
c432	97.52	1.30E+05	71	0.1
c880	97.52	4.45E+05	70	0.3
c1355	98.57	1.25E+06	89	1.3
c2670	84.34	3.08E+06	79	3.9
c3540	94.69	5.97E+06	207	16.6
c5315	98.83	1.57E+07	199	25.1
c6288	99.56	2.97E+07	38	3560.0
c7552	92.03	2.77E+07	208	21.0
cs15850	85.05	4.97E+07	279	389.4
cs35932	89.81	6.16E+08	96	2023.0
cs38417	86.58	3.64E+08	453	7021.0
cs38584	90.60	5.41E+08	537	10030.0

1024 random vectors were used as a given test set.

Test compaction for sequential circuits

- More difficult than that for combinational circuits
  - When test vectors are removed
    - State transition is changed
    - Faults that are originally detected become undetected.
    - Fault simulation is needed to check if all the faults are still detected.



v1 v2 v3 v4 v5

v1 v2 v4 v5

Fault *f* is detected

Fault *f* may be undetected

Reverse order restoration

[Guo et al., 1998]

- First remove all the test vectors except for initialization vectors
- Restore test vectors in order to detect a subset of faults
- Perform fault simulation to see if all the faults are detected

Ex：T= v1 – v2（only initialization vectors）

T = v1 – v2 – v9 （v9 detects f3）

If T detects f3, then add v6.

Otherwise, add v8.

T= v1 – v2 – v8 - v9

If T detects f3, then add v6

T= v1 – v2 - v8 - v9 – v6

vector	faults
v1	
v2	
v3	f1
v4	
v5	
v6	f2
v7	
v8	
v9	f3

Proposed algorithm（DCOMP-S)

- Apply reverse order restoration for diagnostic sequences
  - Remove all the test vectors except for initialization vectors
  - Restore test vectors
  - Perform fault simulation to see if all the distinguished fault pairs are distinguished.

vector	pair
v1	
v2	
v3	pair1
v4	
v5	
v6	pair2
v7	
v8	
v9	pair3

Ex： T= v1 – v2（Only initialization vectors）

T = v1 – v2 – v9 （Add v9 for distinguishing pair3）

Unless T distinguishes pair3, then add v8

T1= v1 – v2 – v8 - v9

Check if T distinguishes pair3

T1= v1 – v2 - v8 - v9 – v6

Experimental results

circuit	length	compacted	%	pair	CPU(s)
s344	127	82	35.4	50,721	0.31
s382	2,074	846	59.2	48,516	16.37
s386	286	164	42.7	49,141	0.73
s400	2,214	845	61.8	61,075	24.91
s526	2,258	631	72.1	64,980	19.42
s641	209	123	41.1	80,601	0.74
s713	173	113	34.7	113,050	0.79
s820	1,115	745	33.2	330,078	33.89
s1196	435	295	32.2	766,941	18.19
s1238	475	316	33.5	822,403	22.88
s1423	150	134	10.7	261,003	4.92
s1488	1,170	787	32.7	1,041,846	75.68
s1494	1,245	772	38.0	1,054,878	106.2

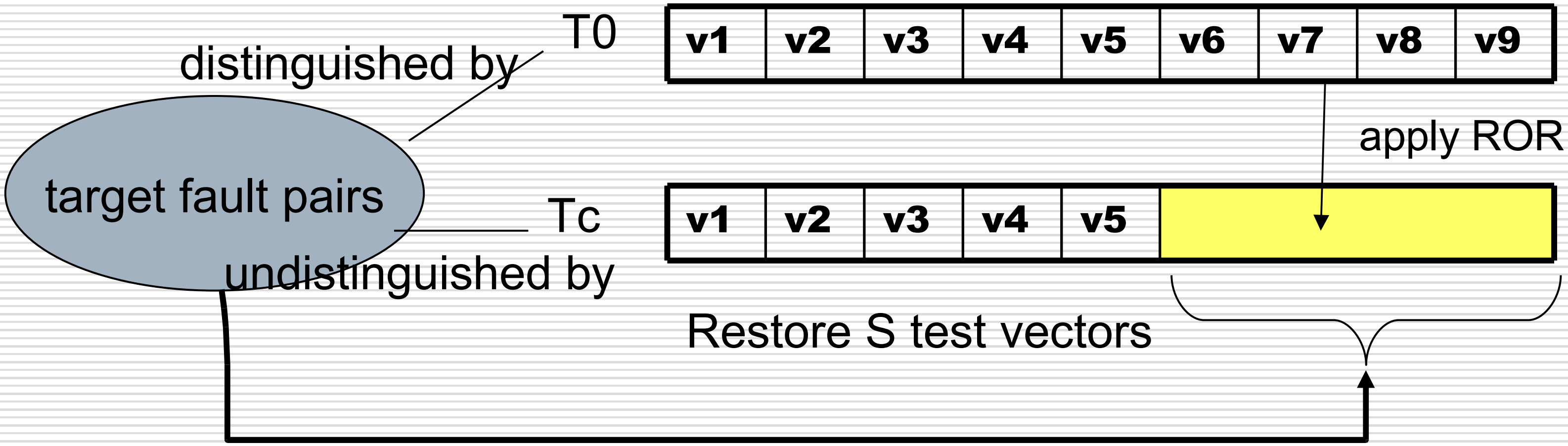
Test sequences generated by HITEC were used as original test sequences.

DCOMP-S

- Results
  - About 10~72 % test vectors were removed.
- Shortcomings
  - It needs information about when each fault pair is distinguished
  - It is difficult to store such information for a large number of fault pairs.

Proposed algorithm（DCOMP-LS）

- **Most of the steps in DCOMP-LS is same as those in DCOMP-S**
- **Differences**
  - **First restore test vectors v1 to vs as an initial compacted test sequence (Tc)**
  - **Collect fault pairs distinguished by T0 but not by Tc**



Results by DCOMP-LS

	compacted test length				CPU(s)	
circuit	DCOMP-S	%	DCOMP-LS	%	DCOMP-S	DCOMP-LS
s344	82	35.4	98	22.8	0.3	0.1
s382	846	59.2	1580	23.8	16.4	18.5
s386	164	42.7	233	18.5	0.7	0.6
s400	845	61.8	1865	15.8	24.9	42.4
s526	631	72.1	1468	35.0	19.4	19.6
s641	123	41.1	177	15.3	0.7	0.4
s713	113	34.7	154	11.0	0.8	0.4
s820	745	33.2	940	15.7	33.9	19.1
s1196	295	32.2	366	15.9	18.2	12.5
s1423	134	10.7	145	3.3	4.9	2.6
s1488	787	32.7	992	15.2	75.7	35.2
s1494	772	38.0	1074	13.7	106.2	47.1

Results by DCOMP-LS

Test sequences generated by HITEC, S=90%

circuit	length	compacted	%	pairs	CPU(s)
s5378	912	839	8.0	5.064E+06	20
s35932	496	496	0.0	6.090E+08	3409

HITEC+ 50 random vectors, 1024 random vectors, S=90%

circuit	length	compacted	%	pair	CPU(s)
s35932	546	535	2.0	6.098E+08	2502
s35932	1024	928	9.4	3.753E+08	3853

Conclusion

- Proposed diagnostic test compaction methods
  - Methods for combinational circuits and sequential circuits
  - Heuristics for reducing target fault pairs to be considered at a time
- Future work
  - Improve the method for large sequential circuits

Experimental results

- Comparison of results with different *n*.

	n=100,000		*n*=1,000	
circuit	vectors	CPU(s)	vectors	CPU(s)
c6288	37	1659.0	38	3560.0
c7552	198	33.8	208	21.0
cs15850	261	141.8	279	389.4
cs35932	91	962.2	96	2023.0
cs38417	436	1848.0	453	7021.0
cs38584	509	2069.0	537	10030.0

1024 random vectors were used as a given test set.