A Method to Reduce Over-testing for Path Delay Faults Using RT-level Information

Yuki Yoshikawa
COE Technical Presentation
June. 22, 2006

Outline

- Introduction
  - Importance of LSI testing
  - Path delay faults
- Reduction in over-testing
  - False paths
  - Our proposed method to identify false paths at RT-level
- Experimental results
- Conclusion

Importance of LSI testing

LSI is used in several IT equipments supporting ubiquitous networks
- CPU, Memory...

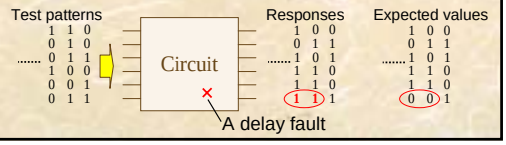
LSI testing is important
- High reliability
- Low cost test

Delay test

The speed of VLSI circuits is getting faster

Delay test is more important to guarantee the timing correctness

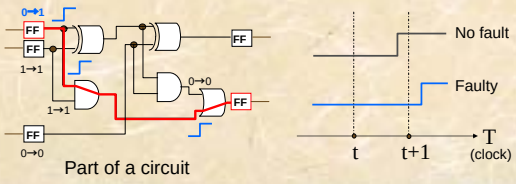
Delay test
- Detect faults that induce excessive delay in a circuit



Test for path delay fault

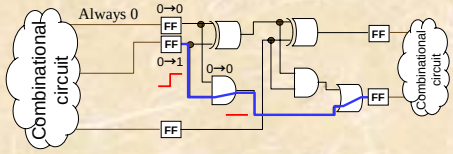
Path delay fault
- Transitions launched at the starting point of a path cannot reach the ending point within the specified time

(Path: starts at a PI or a FF and ends at a PO or a FF)



False path

False path
- Never launch a transition at the starting of the path
- Never propagate a transition along the path



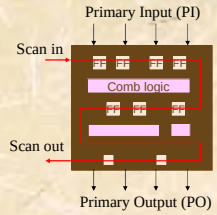
The path delay fault on a false path does not affect the circuit behavior

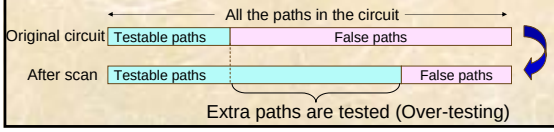
Most false paths are tested by Scan

Scan design

- Any value can be applied to FFs
- The responses can be observed

The path delay fault on a false path should not be tested !





Approach to reduction in over-testing

To reduce over-testing

1. Identify false paths before testing
2. Exclude them from the target of test

At gate level, identifying false paths within practical time is impossible

- Identifying a false path takes a lot of time
- The number of paths is very large

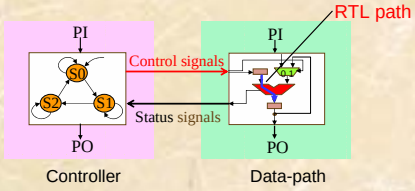
Our method

- identifies a subset of false paths within practical time using RT-level information

Target circuit

Advantages

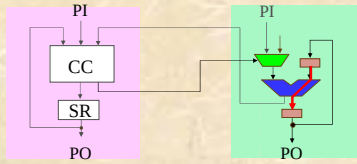
- Gate-level paths between two registers can be handled as a bundle, which is called RTL path
- Information about the timing of data transfer



★ Identify false paths for controller-datapath circuits

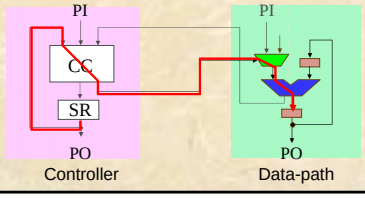
Classification of RTL path

1. Data-path register to data-path register
2. SR to data-path register
3. Data-path register to SR
4. SR to SR



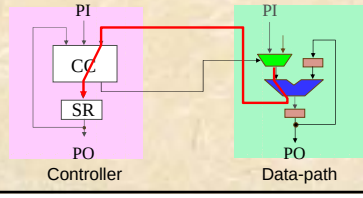
Classification of RTL path

1. Data-path register to data-path register
2. SR to data-path register
3. Data-path register to SR
4. SR to SR



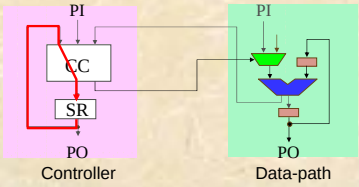
Classification of RTL path

1. Data-path register to data-path register
2. SR to data-path register
3. Data-path register to SR
4. SR to SR



Classification of RTL path

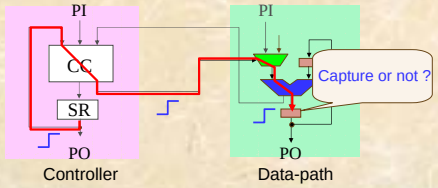
1. Data-path register to data-path register
2. SR to data-path register
3. Data-path register to SR
4. SR to SR



False path identification : SR to data-path register

1. Data-path register to data-path register
2. SR to data-path register
3. Data-path register to SR
4. SR to SR

CUPの説明



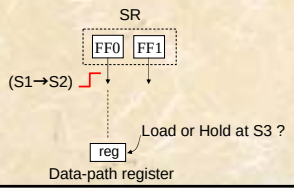
False path identification : SR to data-path register

SR loads a new value every clock
- A transition is launched every clock

Idea: Consider transitions for each FF in the SR

State assignment

	FF0	FF1
S0	0	0
S1	0	1
S2	1	1
S3	1	0



Experiments

The number of RTL path identified as a control dependent untestable path

	D_reg to D_reg		SR to D_reg		SR to SR	
Circuit	# CUPs	Ratio	# CUPs	Ratio	# CUPs	Ratio
Paulin	12 (29)	41.4 %	15 (64)	23.4 %	0 (9)	0 %
LWF	3 (19)	15.5 %	0 (24)	0 %	0 (4)	0 %
Tseng	6 (20)	30.0 %	6 (39)	15.4 %	0 (9)	0 %
JWF	117 (153)	76.5 %	249 (405)	61.5 %	0 (9)	0 %
MPEG	32 (651)	4.90 %	256 (2,138)	12.0 %	0 (100)	0 %

- For JWF, a large number of RTL paths are identified as CUPs
- The CPU time is less than 1 second.

Evaluation for gate-level paths : Data-path

The number of gate-level paths corresponding to CUPs

Circuit	# CUPs at RTL	#Gate-level paths (CUP)	#Gate-level paths (total)	Ratio
Paulin	12 (29)	65,430	117,576	55.7 %
LWF	3 (19)	536	2,766	19.3 %
Tseng	6 (20)	7,978	16,532	48.3 %
JWF	117 (153)	20,972	26,000	80.7 %

- For Paulin, Tseng and JWF, 50%－80% paths are identified as false paths within 1 second.
- If we try to identify false paths at gate-level, for Paulin, the CPU time for only 100 paths takes 498 sec

Conclusion

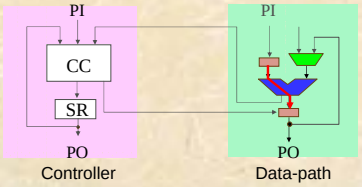
We proposed the method to identify a subset of false paths at RTL (control-dependent untestable paths) which
- can identify false paths within practical time, and (Dramatically reduce the CPU time)
- can reduce over-testing

Our future work
- Identifying not only control-dependent but also data-dependent untestable paths at RT-level

Classification of RTL path

RTL path

1. Data-path register to data-path register
2. SR to data-path register
3. Data-path register to SR
4. SR to SR



False path identification : Data-path registers

An RTL path p is a control dependent untestable path if control signals satisfy at least one of the following conditions for any state transition in the controller
(H: Hold, L: Load, *: don't care)

1. Any transition is not launched
2. Any transition is not propagated
3. Any transition is not captured