ASP-DAC 2009

Fast False Path Identification Based on Functional Unsensitizability Using RTL Information

Presenter Yuki Yoshikawa[1]

Satoshi Ohtake[2], Tomoo Inoue[1] and Hideo Fujiwara[2]

[1] Hiroshima City University, Japan

[2] Nara Institute of Science and Technology, Japan

*This work has been supported in part by STARC

Outline

Background

- False path
- Adverse effect of false paths
  - Over-testing of delay faults
  - Inaccurate estimation of a system clock period

Our proposed method

- False path identification using RTL information

Experimental result

Conclusion

Background

For high performance VLSIs,

- high quality delay fault testing, and
- accurate estimation of a circuit delay is an important issue

False paths interfere

- accurate timing delay testing
  - Over-testing
- accurate estimation of a system clock period
  - Degradation of circuit performance

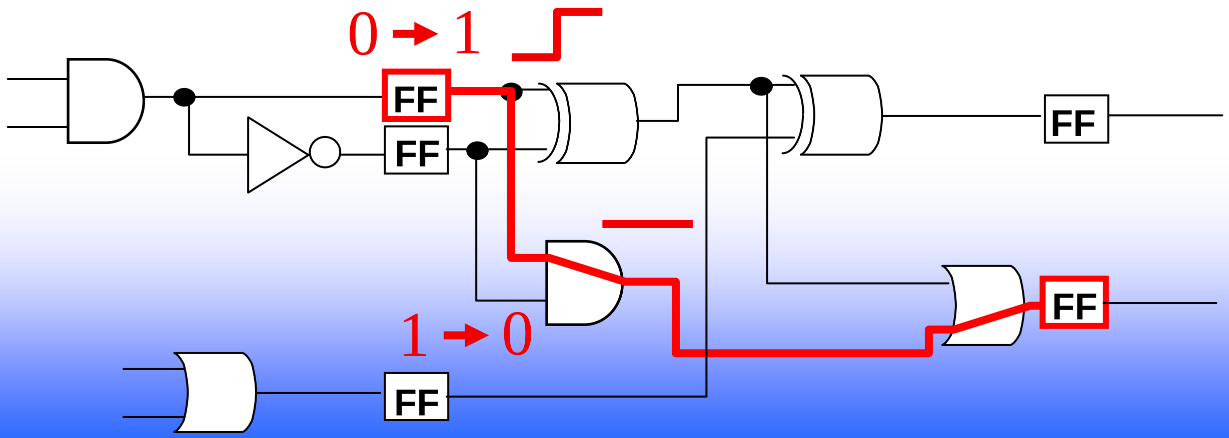
False path

No transition occurs at the start point of a path, or

a transition at the start never reaches the end of the path, or

the captured value at the end is never propagated to any PO

- Path delay faults on a false path are untestable
- Transition faults are not activated along a false path
- The propagation delay on a false path does not affect its circuit performance



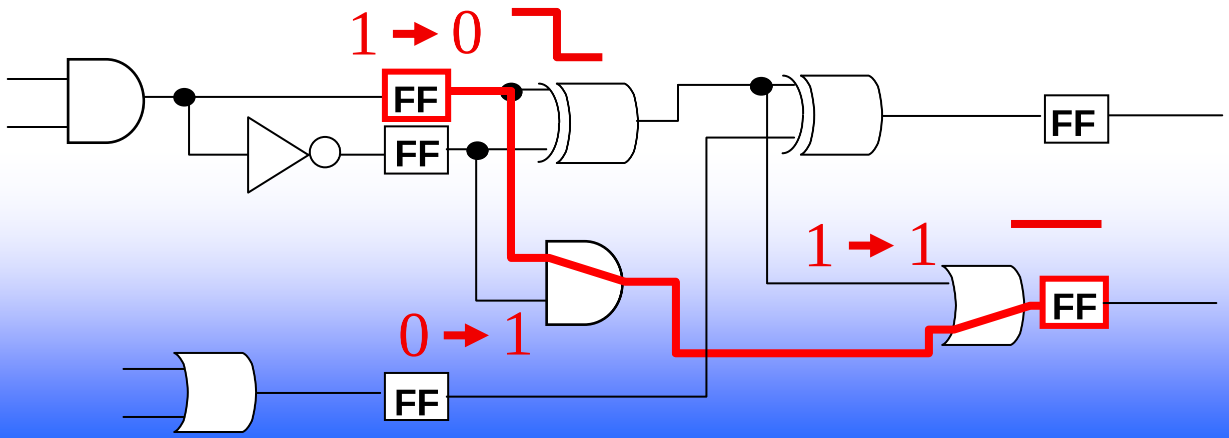
False path

No transition occurs at the start point of a path, or

a transition at the start never reaches the end of the path, or

the captured value at the end is never propagated to any PO

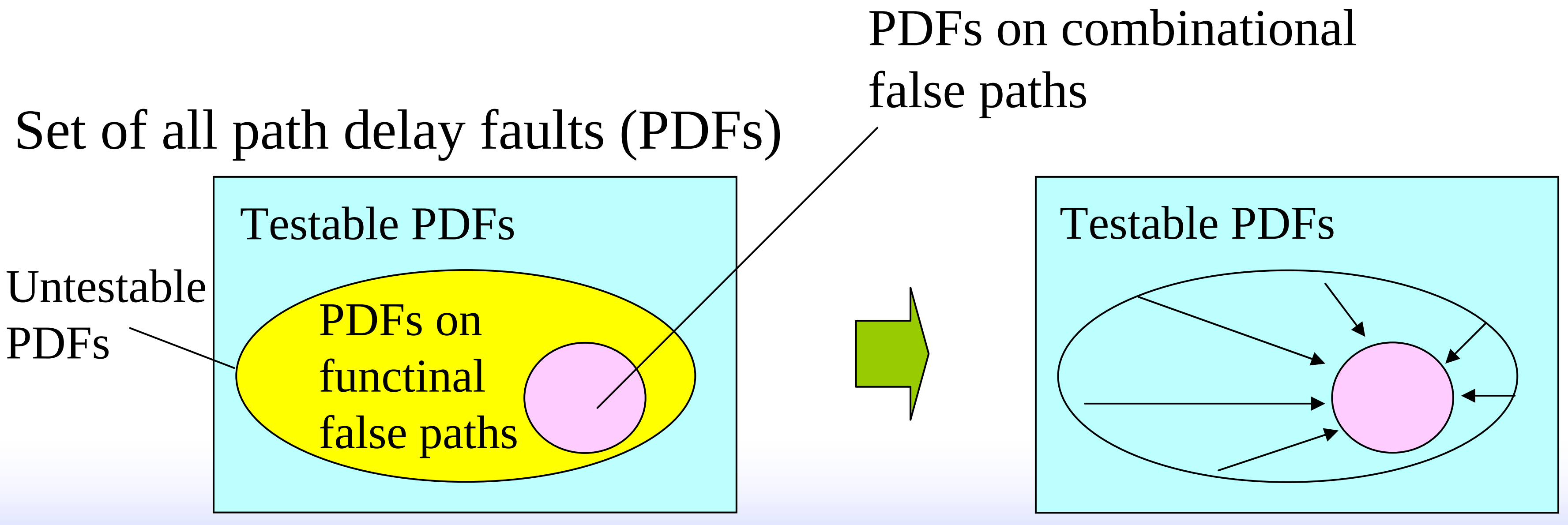
- Path delay faults on a false path are untestable
- Transition faults are not activated along a false path
- The propagation delay on a false path does not affect its circuit performance



Over-testing

Over-testing

- To test faults that are untestable in an original circuit
- It induces yield loss, futile test generation time and futile test application time


PDFs on combinational false paths
Set of all path delay faults (PDFs)
Testable PDFs
Untestable PDFs
PDFs on functional false paths
Testable PDFs

Before DFT (Original circuit)

After DFT

Approach to over-testing reduction

A strategy for over-testing reduction

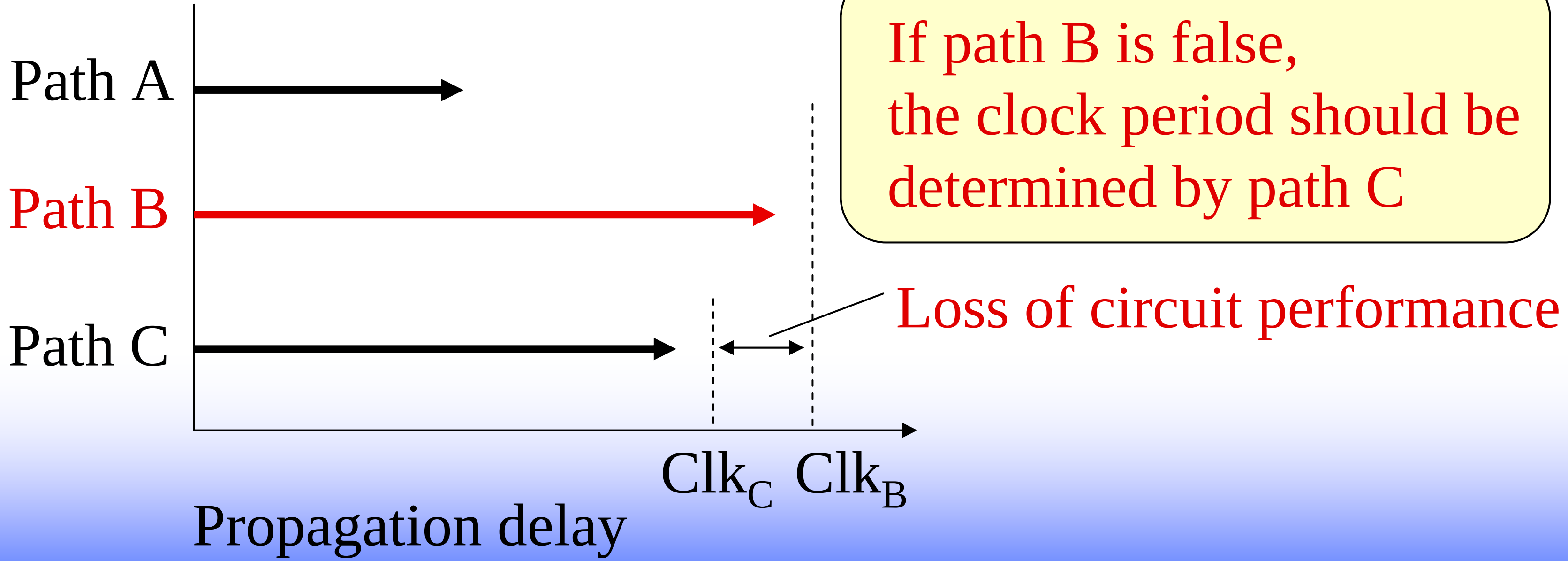
- False path identification for an original circuit
- Exclusion of PDFs on the identified false paths from the target of testing

To reduce over-testing, it is important to identify as many false paths as possible with small computational time

Inaccurate estimation of a system clock period

The system clock period of a circuit is determined according to the propagation delay on the longest path

- If the longest path in a circuit is false, the system clock period is inaccurately determined, and thus the maximum performance is missed



If path B is false, the clock period should be determined by path C

Related works

Gate-level false path identification

- For combinational circuit
  [Cheng’96], [Kajihara’97], [Reddy’01]
- For sequential circuit [Kristic’96]

GL approaches would take much time to handle many paths

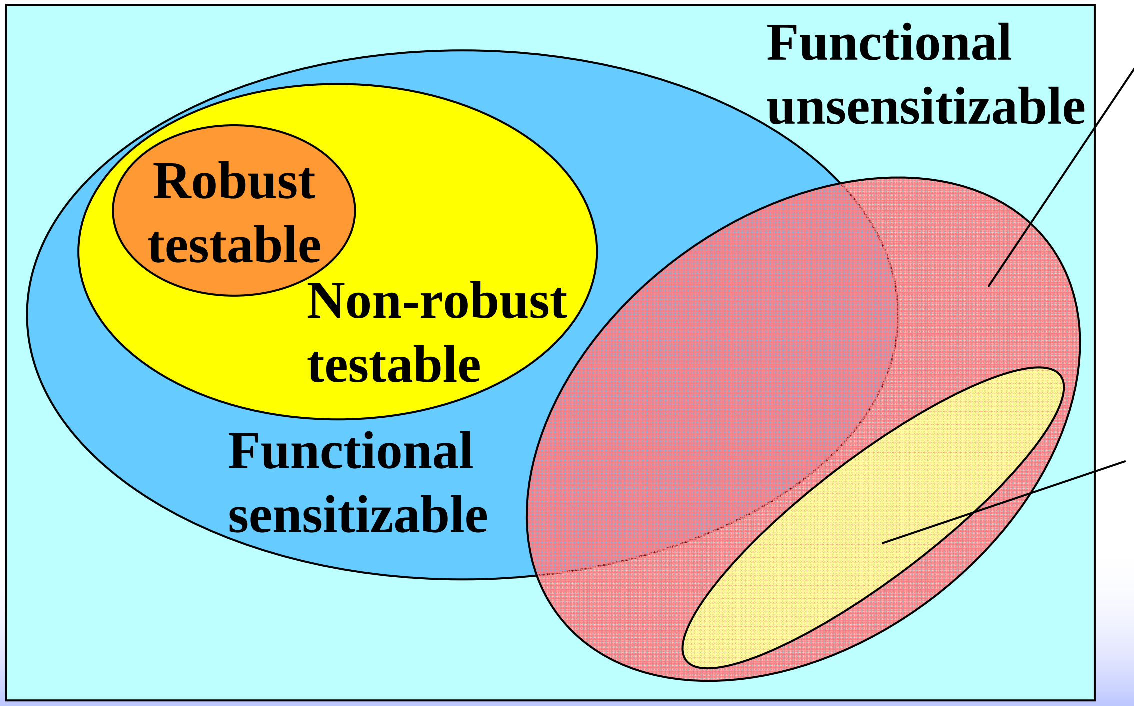
RT-level false path identification

- False path identification using RTL information and its application to over-testing reduction for delay faults [Yoshikawa’07]

The method identifies non-robust untestable paths at RTL
The time required for the identification is much faster than GL

Research objective

- Our objective is to identify false paths at RTL based on functional unsensitizability of PDFs



A subset of non-robust untestable PDFs [Yoshikawa’ 07]

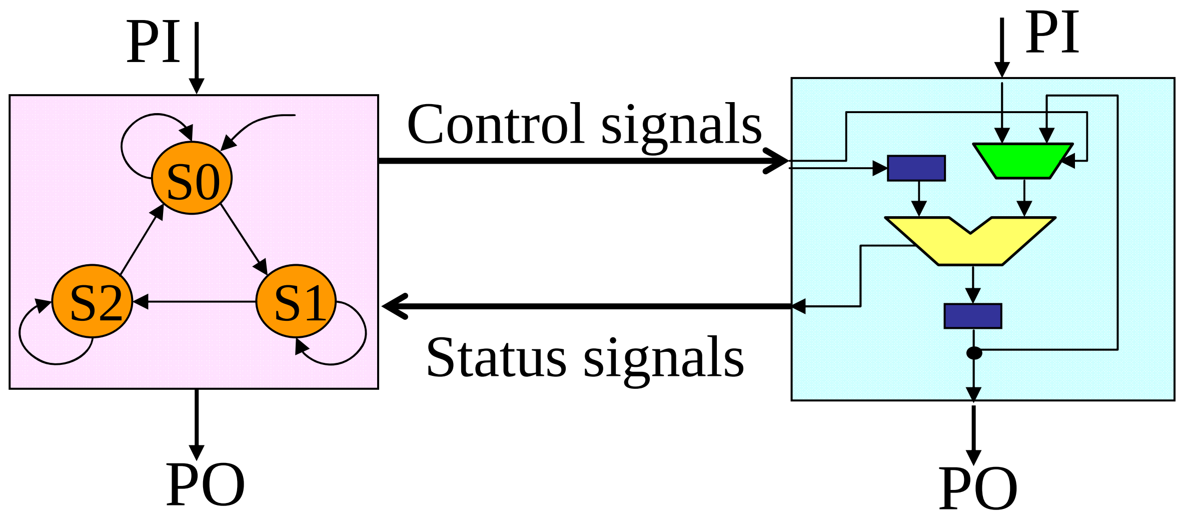
- Not all the paths are false because functional sensitizable (FU) PDFs are included

A subset of FU PDFs

- All the paths whose PDFs are FU with rising and falling transitions are false

Path delay fault classification [K. T. Cheng]

Target RTL circuit



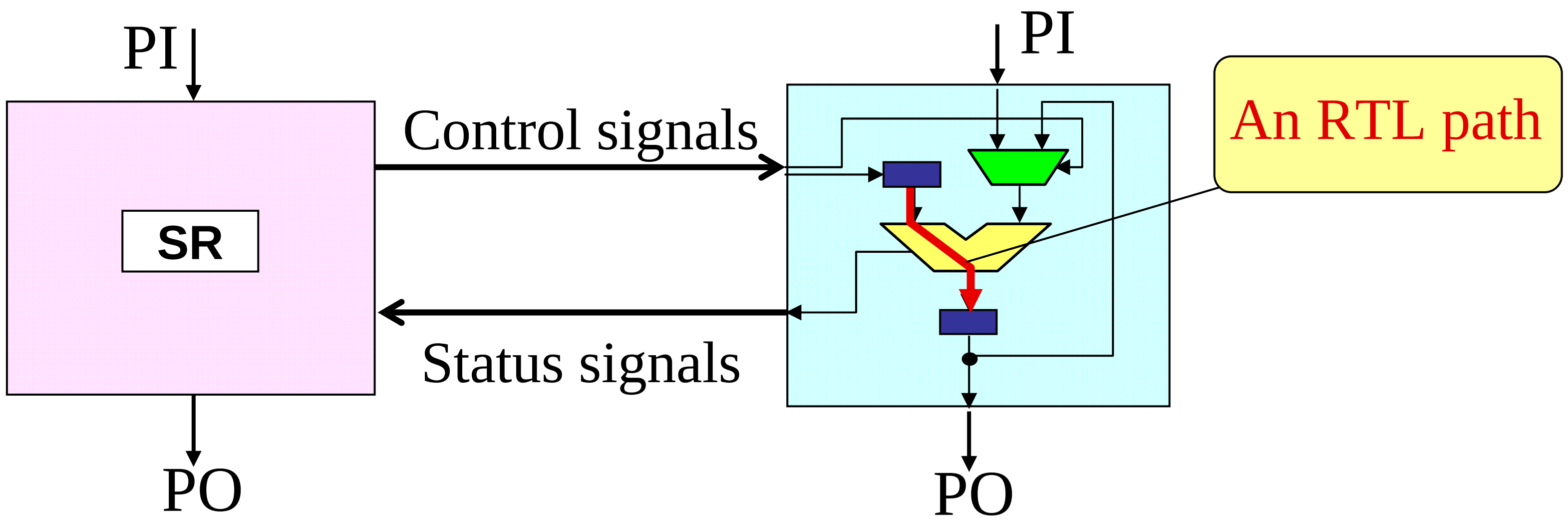
Controller

- Represented by an FSM
- State transitions are completely specified

Data path

- Represented by interconnection of registers, multiplexers and combinational operation modules

RTL path



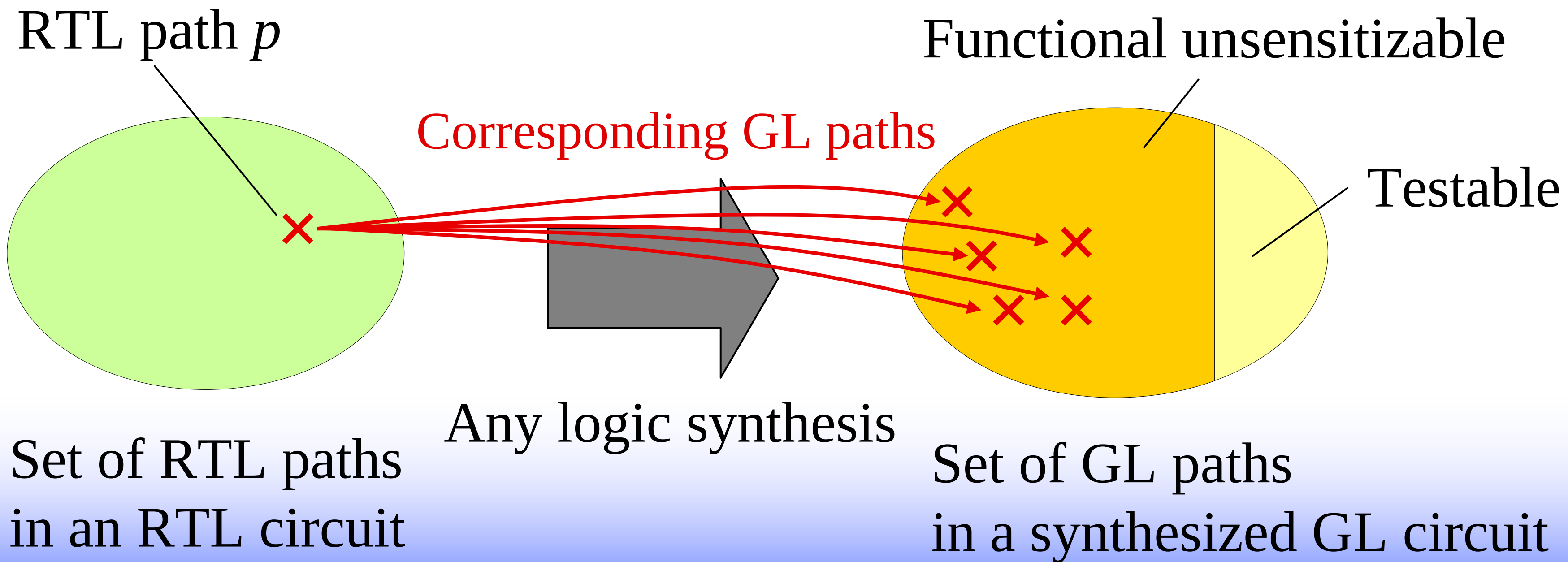
RTL path

- starts from a register or a PI and ends at a register or a PO
- only passes through combinational modules
- is a bundle of gate-level paths

RTL functional unsensitizable (RTL-FU) path

RTL-FU path

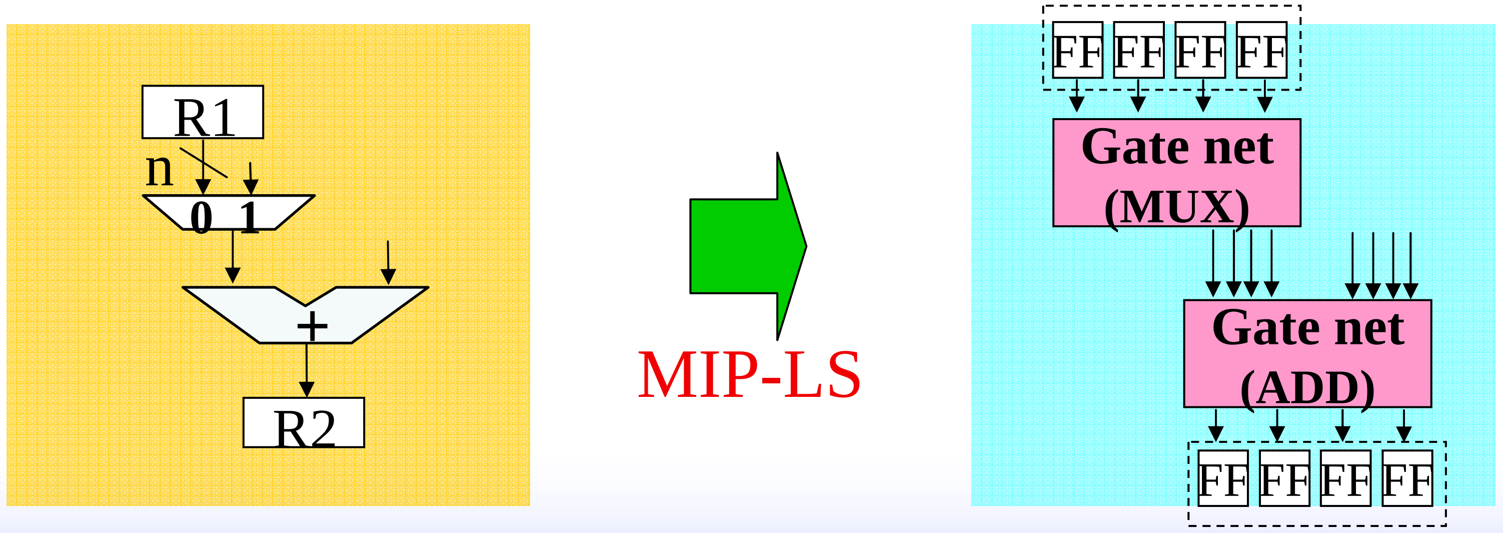
- An RTL path *p* is RTL-FU if all gate-level paths corresponding to *p* are functional unsensitizable for any logic synthesis



Logic synthesis

To clarify the correspondence between an RTL path and its GL paths, we consider Module Interface Preserving Logic Synthesis (MIP-LS)

- It transforms each RTL module and RTL signal line into its own gate-level netlist and single-bit signal lines



A mapping from an RTL path to its GL paths is also proposed [Iwata' 08]

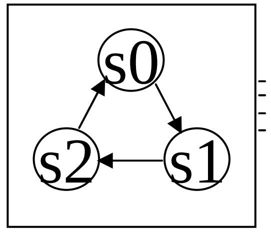
Strategy for RTL-FU path identification

- Check control signals of registers and multiplexers

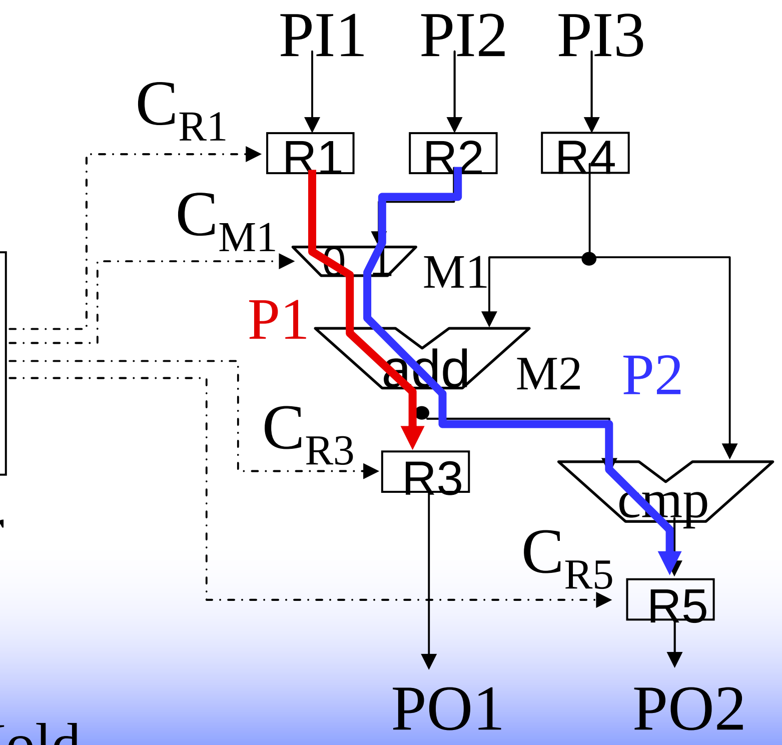
Example: P1 is not RTL-FU and P2 is RTL-FU

State transition table

PS	NS		Output
	R=0	R=1	C_{R1}, C_{R3}, C_{R5}, C_{M1}
S0	S1	S0	1101
S1	S2	S0	0100
S2	S0	S0	0010



Controller



PS: Present state, NS: Next state,

R : Reset, Load enable ‘1’ is Load, ‘0’ is Hold

Strategy for RTL-FU path identification

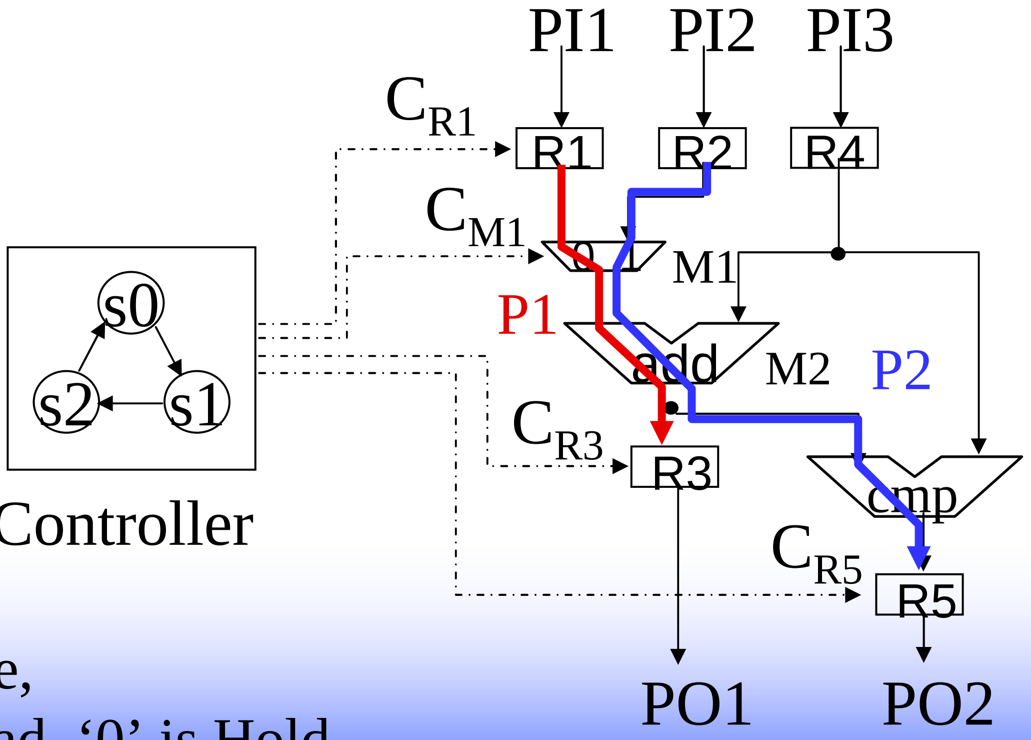
- Check control signals of registers and multiplexers

Example: P1 is not RTL-FU and P2 is RTL-FU

State transition table

PS	NS		Output
	R=0	R=1	C_{R1}, C_{R3}, C_{R5}, C_{M1}
S0	S1	S0	1101
S1	S2	S0	0100
S2	S0	S0	0010

PS: Present state, NS: Next state,
R : Reset, Load enable ‘1’ is Load, ‘0’ is Hold



Strategy for RTL-FU path identification

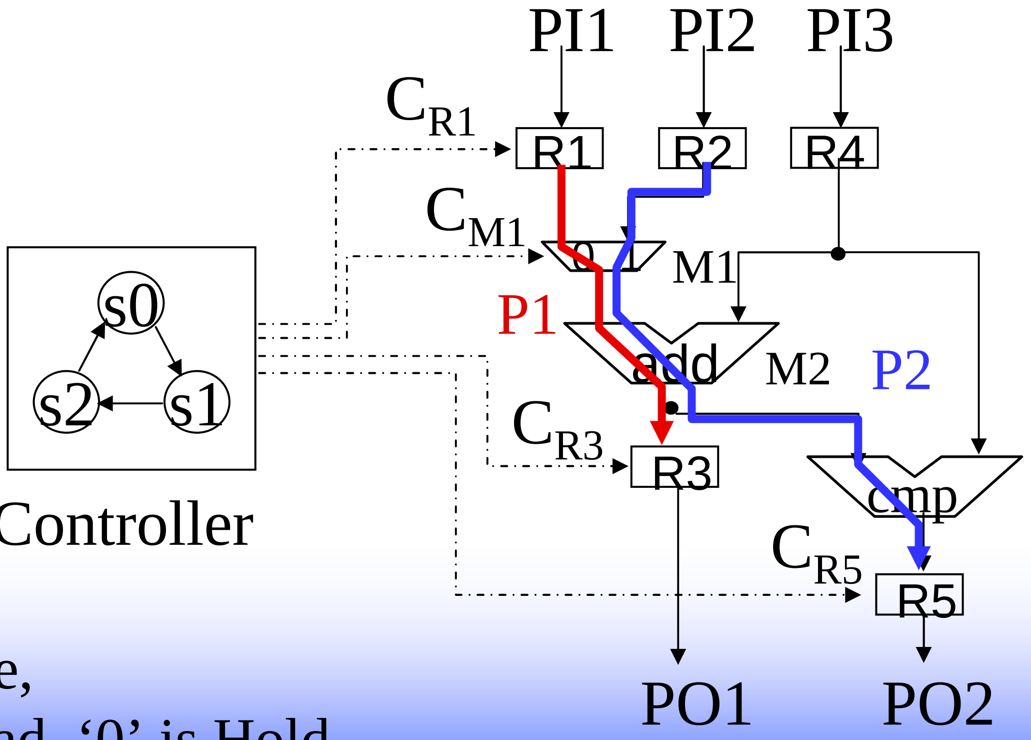
- Check control signals of registers and multiplexers

Example: P1 is not RTL-FU and P2 is RTL-FU

State transition table

PS	NS		Output
	R=0	R=1	C_{R1}, C_{R3}, C_{R5}, C_{M1}
S0	S1	S0	1101
S1	S2	S0	0100
S2	S0	S0	0010

PS: Present state, NS: Next state,
R : Reset, Load enable '1' is Load, '0' is Hold



Strategy for RTL-FU path identification

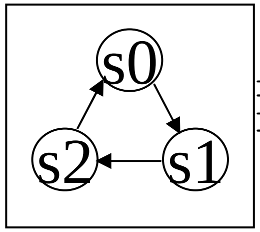
- Check control signals of registers and multiplexers

Example: P1 is not RTL-FU and P2 is RTL-FU

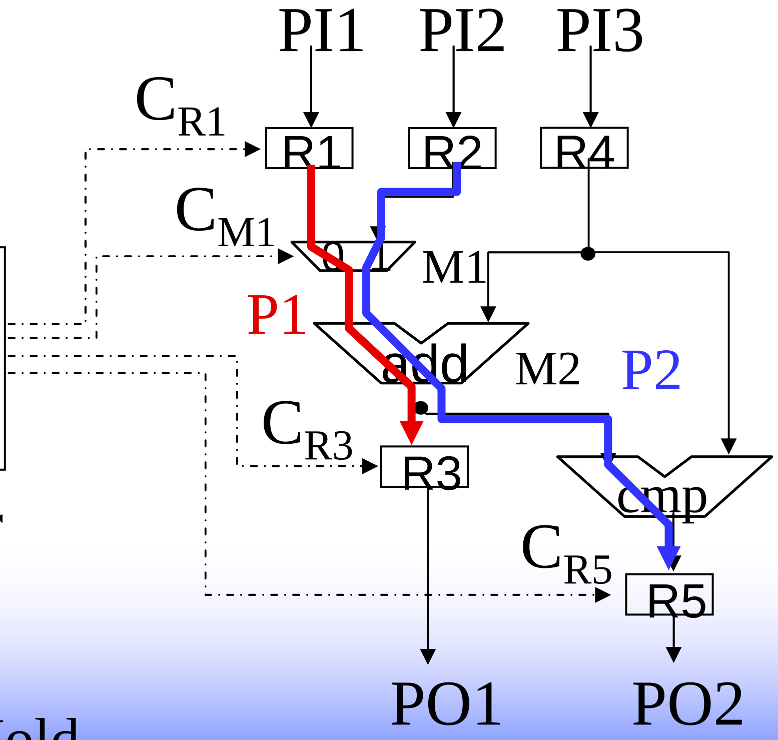
State transition table

PS	NS		Output
	R=0	R=1	C_{R1}, C_{R3}, C_{R5}, C_{M1}
S0	S1	S0	1101
S1	S2	S0	0100
S2	S0	S0	0010

PS: Present state, NS: Next state,
R : Reset, Load enable ‘1’ is Load, ‘0’ is Hold

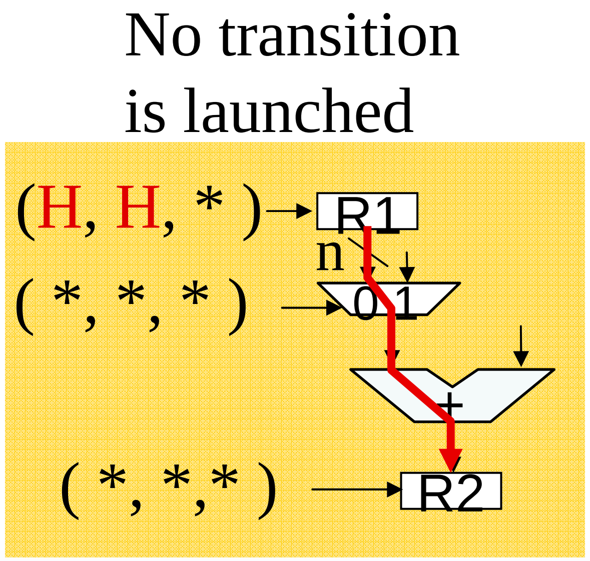


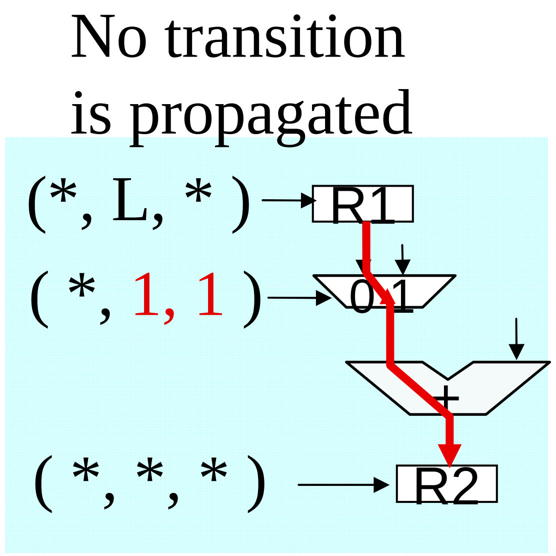
Controller

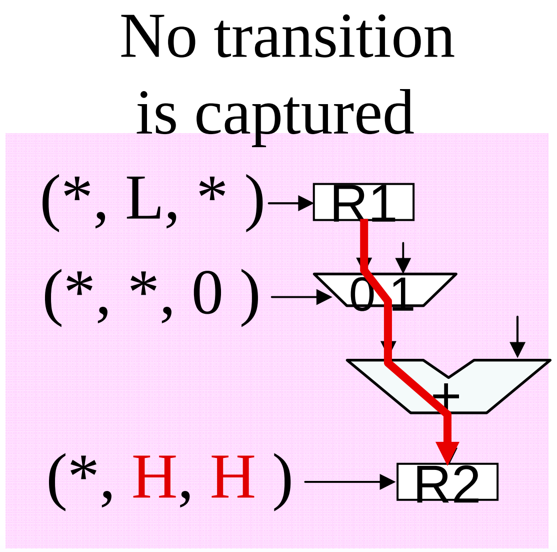


Control condition

- An RTL path *p* is RTL-FU if at least one of the following three conditions is satisfied for any state transition







（H: Hold, L: Load, *: L or H or 0 or 1）

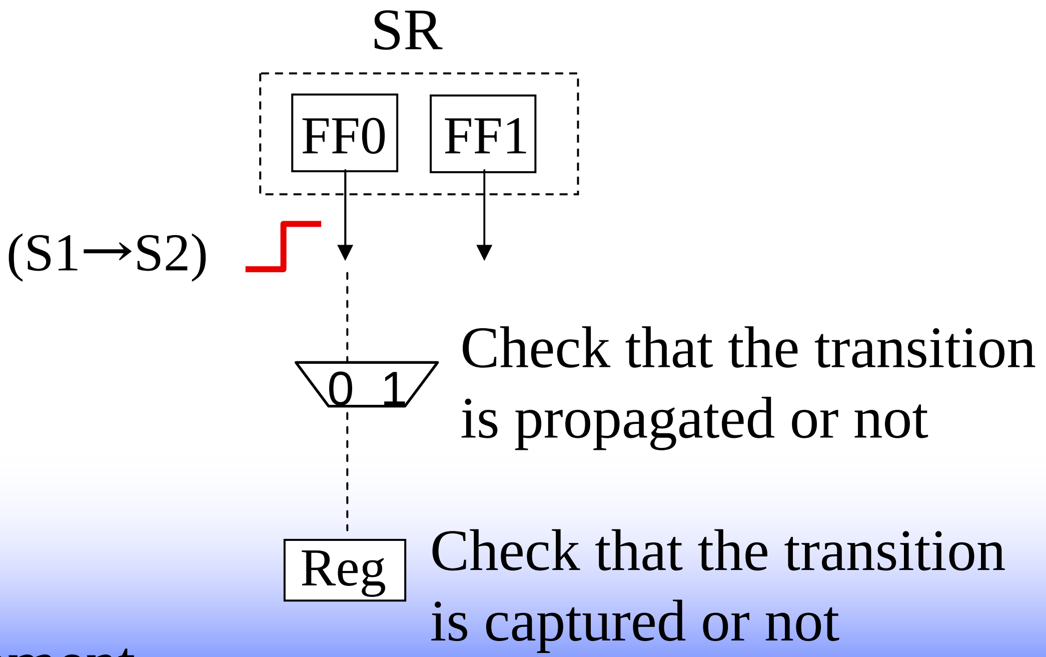
Identification for RTL path starting from state register

For each FF(SR-ff) in an SR,
by considering a state assignment and state transitions,
we can know the time when a transition occurs

State transition table

PS	NS
S0 (00)	S1 (01)
S1 (01)	S2 (10)
S2 (10)	S3 (11)
S3 (11)	S0 (00)

(FF0, FF1): State assignment


SR
FF0
FF1
(S1→S2)
0 1
Reg
Check that the transition is propagated or not
Check that the transition is captured or not

Experimental results

- Evaluate the number of RTL paths that are identified as RTL-FU by our method
- Evaluate the number of gate-level paths that are corresponding to the identified RTL-FU paths

Circuit characteristics of benchmarks

Circuit	Bit width	# Regs	#States	Area NOT gate: 2	# RTL paths	
					DR to DR	SR-ff to DR
Tseng	8	7	5	2,975	20	42
Paulin	8	8	6	3,391	29	67
JWF	8	15	8	4,758	153	408
MPEG	8	241	163	77,554	651	2,152
RISC	32	39	10	81,086	10,181	38,122

Number of RTL paths identified as RTL-FU

	DR			SR-ff : Rise			SR-ff : Fall		
Circuit (RTL)	#RTL path	#RTL FU	#RTL NRU	#RTL path	#RTL FU	#RTL NRU	#RTL path	#RTL FU	#RTL NRU
Tseng	20	2	6	42	5	13	42	6	11
Paulin	29	0	13	67	17	25	67	19	30
JWF	153	69	119	408	172	285	408	226	319
MPEG	651	0	32	2,152	0	64	2,152	0	64
RISC	10,181	707	1,233	38,122	28,217	28,411	38,122	15,176	18,968

・DR: RTL paths starting from registers in a data path

・SR: RTL paths starting from FFs of the state register in a controller

・RTL-NRU (Non-robust untestable): identified by our previous method

- For JWF and RISC, many RTL paths were identified as RTL-FU
- For MPEG, there is no RTL paths identified as RTL-FU
- The time required for identifying RTL-FU paths is a few seconds

Number of GL paths corrsp. to identified RTL-FU paths

	DR			SR-ff			Total		
Circuit (Gate)	#GL path	#GL FU	#GL NRU	#GL path	#GL FU	#GL NRU	#GL path	#GL FU	#GL NRU
Tseng	13,056	534	5,910	944	139	465	14,000	673	6,375
Paulin	96,912	0	41,278	95,310	24,135	39,434	192,232	24,135	80,712
JWF	60,150	12,710	18,182	101,622	53,064	79,404	161,772	65,774	97,586
MPEG	833,696	0	2,048	2,602,624	0	70,624	3,436,320	0	72,672
RISC	57.6 *B*	2.1 *B*	3.8 *B*	223.7 *B*	140.8 *B*	141.4 *B*	281.3 *B*	142.9 B	145.2 *B*

- We extracted #GL paths of RISC under the constraint of PrimeTime.
- Unit *B* means Billion

- For Paulin and JWF, our method identified 24,135 (13%) and 65,774 (41%) GL FU paths within 1 second
- For RISC, 142.9 billion GL FU paths were identified by our proposed method within 10 seconds
- For Paulin, it takes 50 hours to identify 10,000 as NRU by TetraMax

Conclusion

We have proposed a method for identifying functional unsensitizable paths using RTL information

- The identified paths are false paths
- The time required for identification is much faster than GL approaches
- The information of the identified false paths can be used in order to reduce over-testing, and
  area and performance optimization during logic synthesis

Condition of RTL-FU path

RTL path *p* is RTL-FU if at least one of the following four properties is satisfied for any consecutive two cycles t and t+1

- No transition is launched at the output of the start register Rs in cycle t irrespective of the delay of the load-enable signal applied to Rs and/or input data delivered to Rs
- Even if a transition is launched at Rs, it never reachesthe end register Re along p in cycle t+1 irrespective of the delay of the off-inputs on p
- The reached value is never captured into Re in cycle t+1 irrespective of the delay of the load-enable signal applied to Re
- The captured value of Re at cycle t+1 never affects any PO at the latter cycles irrespective of the delay of the off-inputs of RTL modules on all the propagation paths from Re to any PO