DEFECT ORIENTED TEST

(IST-2000-30193 REASON)

Elena Gramatová, Institute of Informatics, Bratislava, Slovakia

Test challenge in defect oriented testing is a current topic. The tutorial doesn’t offer book solution, doesn’t solve all problems of defect oriented testing. It is targeted to some ideas and approaches how to improve test generation process to receive better quality of testing; it means how to generate a test set for ensuring better defect coverage for a manufactured design. Quality of a digital system depends also on quality of testing, but the testing cost is every year higher. And a vision for year 2014 is that transistor manufacturing will be cheaper than its testing. Defect testing should be one of the most important research topic in the testing field.

OUTLINE

- Introduction
- References
- Motivation
- Objectives of defect oriented testing
- Terminology
- Defects versus fault models
- Testing for defects
- Defect classification
- Taxonomy of fault models
- Defect manifestation
- Defect oriented test generation
- Example
- Conclusion and discussion

INTRODUCTION

Elena Gramatová: Head of Dept. of Design and Diagnostics of Digital Systems

Projects with TTU: PECO, ESPRIT, COPERNICUS, 5 FP (EEMCN, ATSEC, FUTEG, VILAB, REASON)

Research interests: TPG algorithms, ATPG systems, fault simulation, fault modeling of digital circuits and systems

Contact address:

Institute of Informatics of the Slovak Academy of Sciences

Dúbravská cesta 9, 842 37 Bratislava, Slovakia

tel./fax: + 421 2 54771008

http://www.ui.savba.sk

e-mail: elena.gramatova@savba.sk

REFERENCES

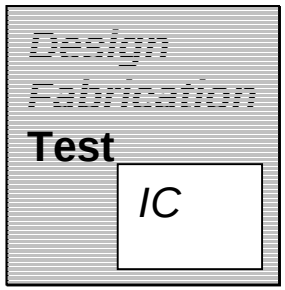
1. M. Sachdev: Defect Oriented Testing for CMOS Analog and Digital Circuits, Kluwer Academic Publishers, 1998.
2. S. Chakravarty, P.J. Thadikaran: Introduction to I_{DDQ} Testing, Kluwer Academic Publishers, 1997.
3. S.L. Hurst: VLSI Testing Digital and Mixed Analogue/Digital Techniques, The Institution of Electrical Engineers, 1998.
4. M.L. Bushell, V.D. Agrawal: Essentials of Electronic Testing for Digital, Memory and Mixed Signal VLSI Testing, Kluwer Academic Publishers, 2000
5. S. Chakravarty: Defect Based Testing, IEEE DDECS, Györ, 2001, invited talk
6. Some papers from Proceedings of ETW, DATE, ITC series.
7. T. Cibáková, M. Fischerová, E. Gramatová, W. Kuzmicz, W. Pleskacz, J. Raik, R. Ubar R: Hierarchical Test Generation for Combinational Circuits with Real Defects Coverage, Journal of Microelectronics Reliability, 42 Pergamon Press, 2002, pp. 1141-1149.

MOTIVATION - I

There are 3 major steps in realization of an IC:

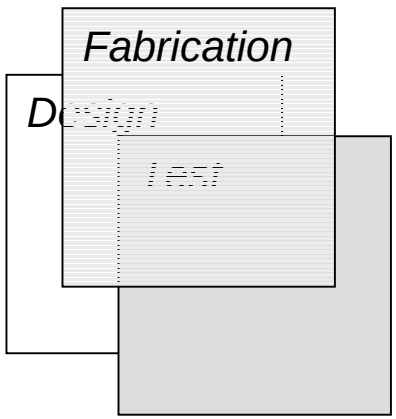
- ✓ design
- ✓ fabrication
- ✓ testing

Ideal environment - 100 % yield:

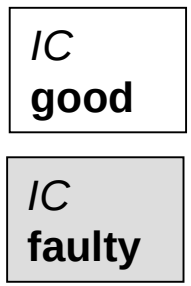


test is redundant and not required

Real world:



output



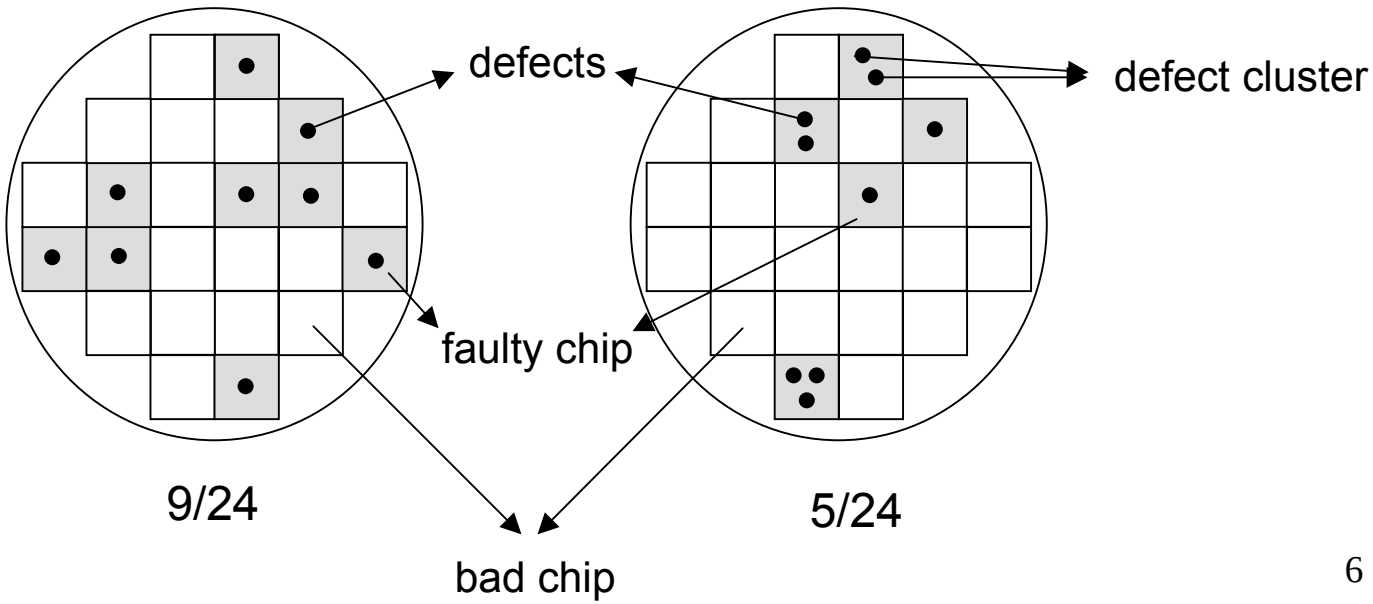
test is important at each level of design process and fabrication

E. Gramatová, II SAS, Bratislava, Slovakia

MOTIVATION - II

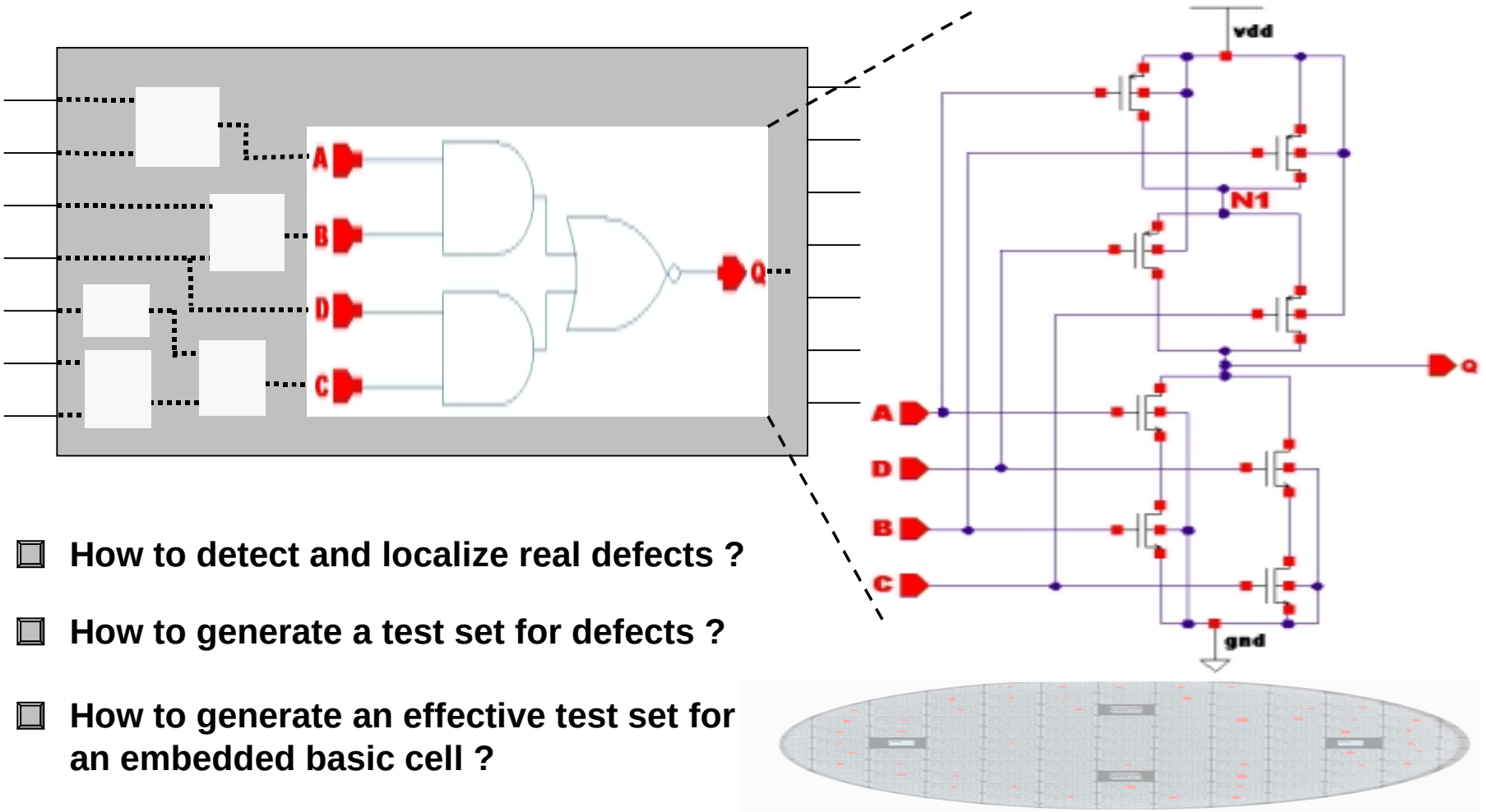
A **defect** generally refers mainly to a physical imperfection in the processed wafer. Some defects are observable through the optical or electron microscope, but others are not visible and can only be detected by electrical tests.

The **wafer yield** is very important and refers to the average number of good chips produced per wafer. The defect oriented testing can improve the yield.



E. Gramatová, II SAS, Bratislava, Slovakia

MOTIVATION - III



- How to detect and localize real defects ?
- How to generate a test set for defects ?
- How to generate an effective test set for an embedded basic cell ?

[*W. Pleskacz*]

OBJECTIVES OF DEFECT TESTING

Testing has to address:

- ✓ **Manufacturing defects**: opens, shorts, threshold/parametric failures, ...
- ✓ **Process marginalities**: line width variation, ...
- ✓ **Design marginalities**: design not validated at multiple points in process specification range.

The goal of testing is to reduce the number of **outgoing faulty parts** (refer to defects).

For VLSI chips, the test quality is specified in terms of a **defect level**, which is the fraction of faulty chips among the chips that pass the test, expressed as parts per million (ppm) - reject ratio, or **field reject rate** (*500 ppm is acceptable*).

Determination of causes of failures is used for improvements of specification, design, fabrication and test. All is aimed to **reduction of the defect level.**

[*S. Chakravarty, M.L. Bushell- W.D. Agrawal*]

TERMINOLOGY

Quality of a product is a function of the user's satisfaction. For VLSI chips, the test quality is specified in terms of **defect level.**

Defect in an electronic system is the unintended difference between the implemented hardware and its intended design.

Fault is a representation of a defect at the abstracted function level (electrical, Boolean, or functional malfunctions).

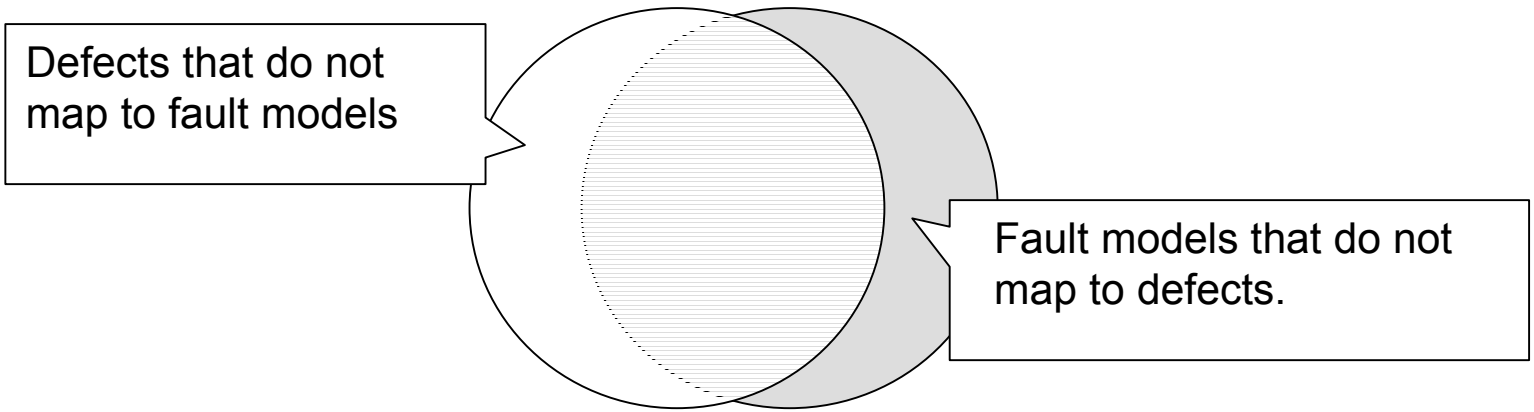
In general, **a physical defect** in a chip can produce **multiple faults**.

Defect-oriented fault is the electrical or logical-level fault produced by physical defects (physical imperfection usually occurred during manufacture) - *e.g. opens or broken wires, bridges, improper semiconductor doping and improperly formed devices.*

Error is a wrong output signal produced by a defective system.

[*M. L. Bushell - W.D. Agrawal*]

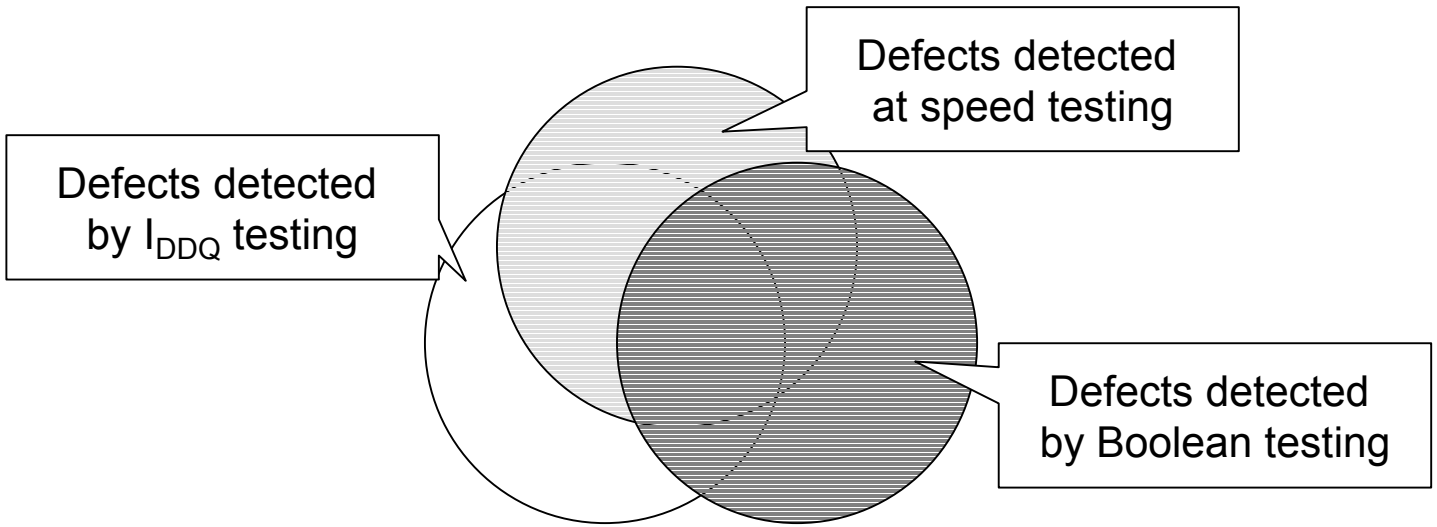
DEFECTS VERSUS FAULT MODELS



- **Determination of these two sets is very difficult !**
- **How to generate tests ?**
- **How to localize defects ?**
- **How to improve defect coverage ?**
- **Which types of faults have to be tested besides of stuck-at faults ?**
- **How to define adequate fault models for real defects ?**

TESTING OF DEFECTS - I

- Defects are manifested in different manners:
  - ✓ by changing a logical value on a node inside a circuit
  - ✓ by increasing the steady state supply current
  - ✓ by changing timing specifications



- **How to generate tests and which type of TPG is necessary to use ?**

TESTING OF DEFECTS - II

Defect Class	Destination	Test Method	100%Detect
Bridge Type-1	Transistor node, interlogic gate, logic gate to power bus, power bus-to-bus	I_{DDQ} Boolean	Yes No
Bridge Type-2	Layout identified bridges	I_{DDQ} Boolean	Yes Yes
Bridge Type-3	Sequential intranodel	I_{DDQ} Boolean	Yes Yes
Open Type-1	Transistor-on	I_{DDQ} Boolean	Yes No
Open Type-2	Transistor pair-on	I_{DDQ} Boolean	Yes Yes
Open Type-3	Transistor pair-on/off	I_{DDQ} Boolean	Yes No
Open Type-4	Sequential	I_{DDQ} Boolean	Yes Yes
Open Type-5	Transistor-off (Memory)	I_{DDQ} Boolean	No No
Open Type-6	Delay	I_{DDQ} Boolean	No No
Parametric Delay	R_{VIA}, V_T, $\Delta(W/L)$	I_{DDQ} Boolean	No No

[*Example of CMOS defect classes, M. Sachdev*]

DEFECT CLASSIFICATION

How do we target defects ? Defects can be classified into two main groups:

- **Soft defects**
  - ✓ cause speed failures or show up at high speed or some temperature,
  - ✓ need two or more test patterns for fault activation and error observation.

  These defects require carefully constructed transitions for defect activation.

- **Hard defects**
  - ✓ observable at all frequencies,
  - ✓ need only one test pattern for fault manifestation,
  - ✓ test set can be applied at slow speed.

E. Gramatová, II SAS, Bratislava, Slovakia

TAXONOMY OF FAULT MODELS

Classical Faults	**hard defects** stuck-at faults	**soft defects** STR, STF transition faults
Opens	**dynamic/transition** targets soft defects STRF	**floating gate** „extensively studied“
Bridges	**static/logic** targets hard defects wired-AND, wired-OR etc.	**dynamic/transition** targets soft defects ON path ON + OFF path

[*S. Chakravarty*]

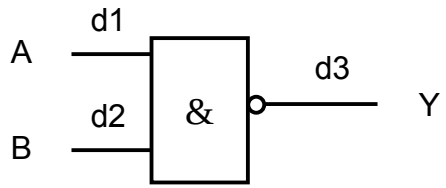
FAULT MODELING

Definition of basic fault models

- **Stuck-at faults** - one type of a logical fault. These faults affect the state of logical signals and the correct value is transformed into any other value (0,1,X,Z); it means value 0 or 1 for stuck-at fault.
- **Bridging faults** represent shorts between a group of signals (usually modeled at the gate or transistor level), non-feedback bridging faults are combinational and their coverage by **stuck-at fault tests** is normally very high. Shorts with feedback produce memory states. Bridging faults belong to the group of **defect oriented faults.**
- **Opens** (stuck-open faults) mean that a physical line in a circuit is broken. The resulting unconnected node is not tied to either V_{DD} or G_{nd}. Occurrence of such defect can result "memory effect". Opens belong also to the group of **defect oriented faults.**

DEFECT MANIFESTATION - I

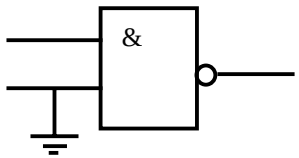
Logic testing: stuck-at faults (SAF)



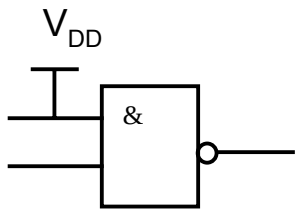
A	B	Y	Detection
1	0	1	d2, d3
0	1	1	d1, d3
1	1	0	d1,d2, d3
0	0	1	redundant

Interpretation:

- **Defect:**



short: stuck-at **0**



short: stuck-at **1**

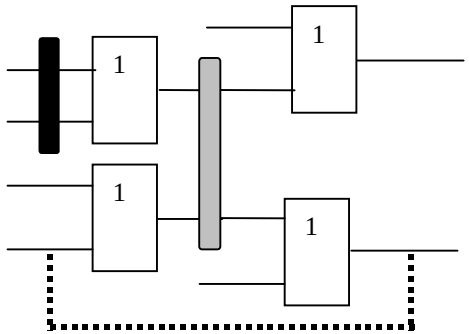
- **Fault:** signal stuck at 0, or 1.

- **Error**: if (AB) = 11, correct value is 0, faulty is 1 (SAF 0),
  if (AB) = 01, correct value is 1, faulty is 1 (SAF 1).

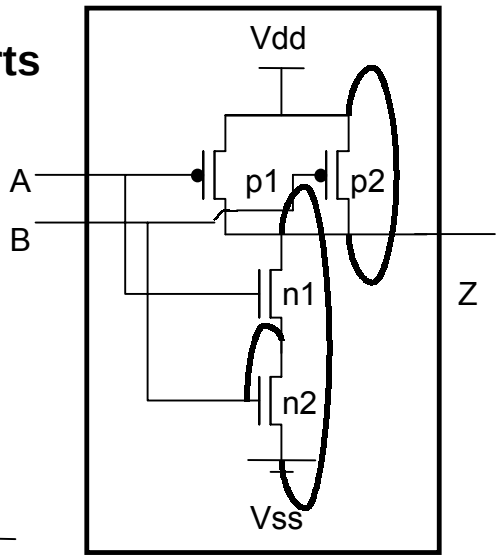
DEFECT MANIFESTATION - II

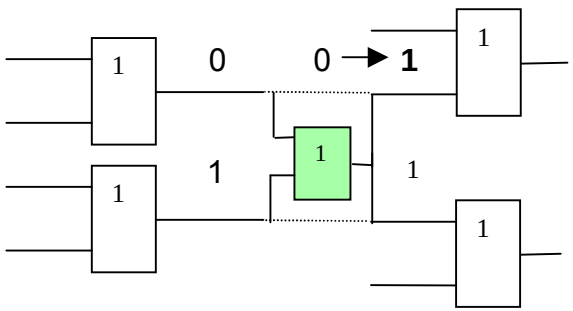
Bridges

- **Inter gate shorts**

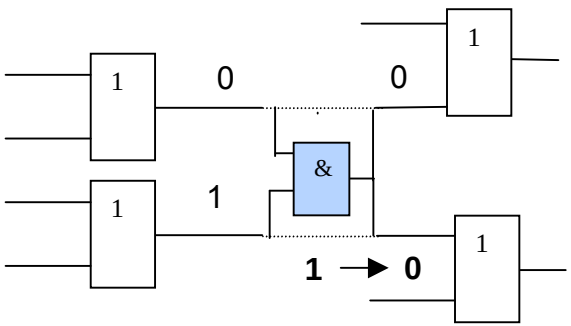


- **Intra gate shorts**





short „AND", 0-dominated



short „OR", 1-dominated

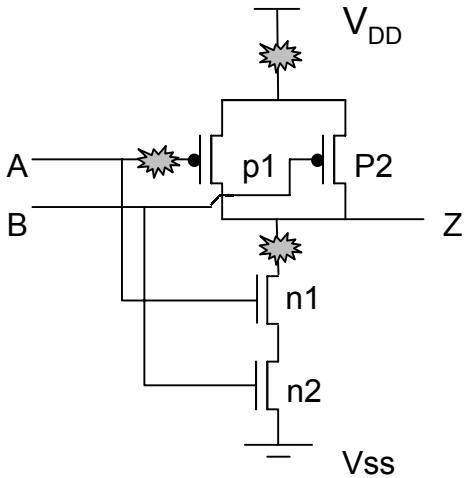
DEFECT MANIFESTATION - III

i	Fault d_i	Erroneous function f^{di}	Input patterns t_j 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	A/C	not(A*C*(B+D))				1				1					1	1		
2	A/D	not(A*D*(B+C))				1				1					1		1	
3	A/N1	not(B*(not(A)+C+D)+C*D)					1	1	1						1			
4	A/Q	not(not(A)+C*D)	1	1	1		1	1	1						1	1	1	
5	A/GND	not(C*D)													1	1	1	
6	A/VDD	not(B+C*D)					1	1	1									
7	B/C	not(B*C*(A+D))				1								1	1	1		
8	B/D	not(B*D*(A+C))				1								1	1		1	
9	B/N1	not(A+C*D)									1	1	1					
10	B/Q	not(not(B)+C*D)	1	1	1						1	1	1		1	1	1	
11	B/GND	not(C*D)													1	1	1	
12	B/VDD	not(A+C*D)									1	1	1					
13	C/N1	not((A+B+D)*(A*B+not(C)+D))		1			1	1			1	1						
14	C/Q	not(not(C)+A*B)	1	1		1	1	1		1	1	1		1				
15	C/GND	not(A*B)				1				1				1				
16	C/VDD	not(D+A*B)		1				1				1						
17	D/N1	not((A+B+C)*(A*B+C+not(D)))			1		1		1		1		1					
18	D/Q	not(not(D)+A*B)	1		1	1	1		1	1	1		1	1				
19	D/GND	not(A*B)				1				1				1				
20	D/VDD	not(C+A*B)			1				1				1					
21	N1/Q	not(A+B)				1				1				1				
22	N1/GND	SA0 for Q	1	1	1		1	1	1		1	1	1					
23	N1/ VDD	not(C*D)													1	1	1	
24	Q/GND	SA0 for Q	1	1	1		1	1	1		1	1	1					
25	Q/ VDD	SA1 for Q				1				1				1	1	1	1	1

[Example of one defect table for cell: AN1= NOR (AND (A,B), AND(C,D) -), 0,8 μ CMOS) for **inter gate shorts**; results from *Warsaw University of Technology, Poland*]

DEFECT MANIFESTATION - IV

Opens: stuck-at open fault (SOP) = difficult problem for testing

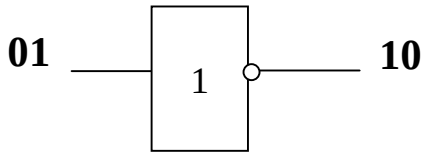


- In the presence of a SOP fault the affected transistor fails to transmit a logic value from its source terminal to its drain terminal. Therefore, the transistor can be treated as a switch which never closes and remains open in spite of all possible Boolean inputs.
- Some SOP faults in static CMOS logic gates require only one pattern test. Others need two patterns test (*one for the initialization, the second for fault excitation).*

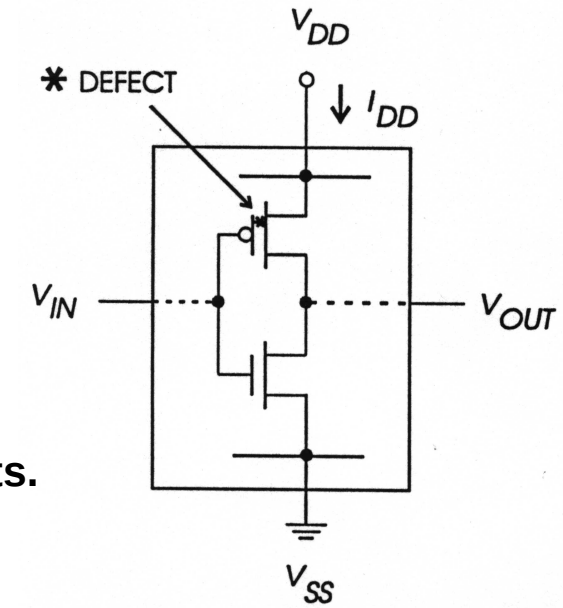
E. Gramatová, II SAS, Bratislava, Slovakia

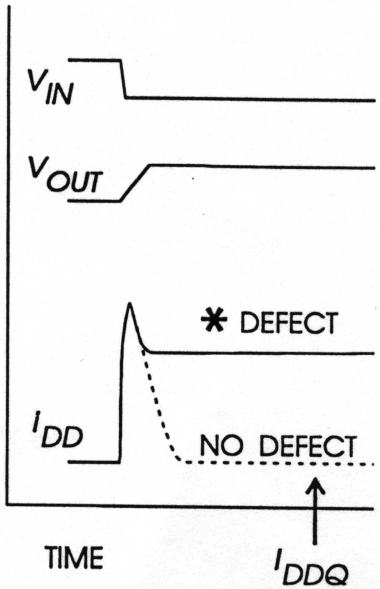
DEFECT MANIFESTATION - V

I_{DDQ} testing



Faults belong to
the group of **defect oriented faults.**





- **Advantages**
  - ✓ Simple generation of the **test set** (it is not necessary to propagate values to primary outputs).
  - ✓ Detection of defects not detectable by logical testing.
- **Disadvantages**
  - ✓ Longer test application time.
  - ✓ Usage of a sensor (a monitor) for measurement.
  - ✓ Threshold value estimation for correct evaluation - good and faulty chips.

DEFECT MANIFESTATION - VI

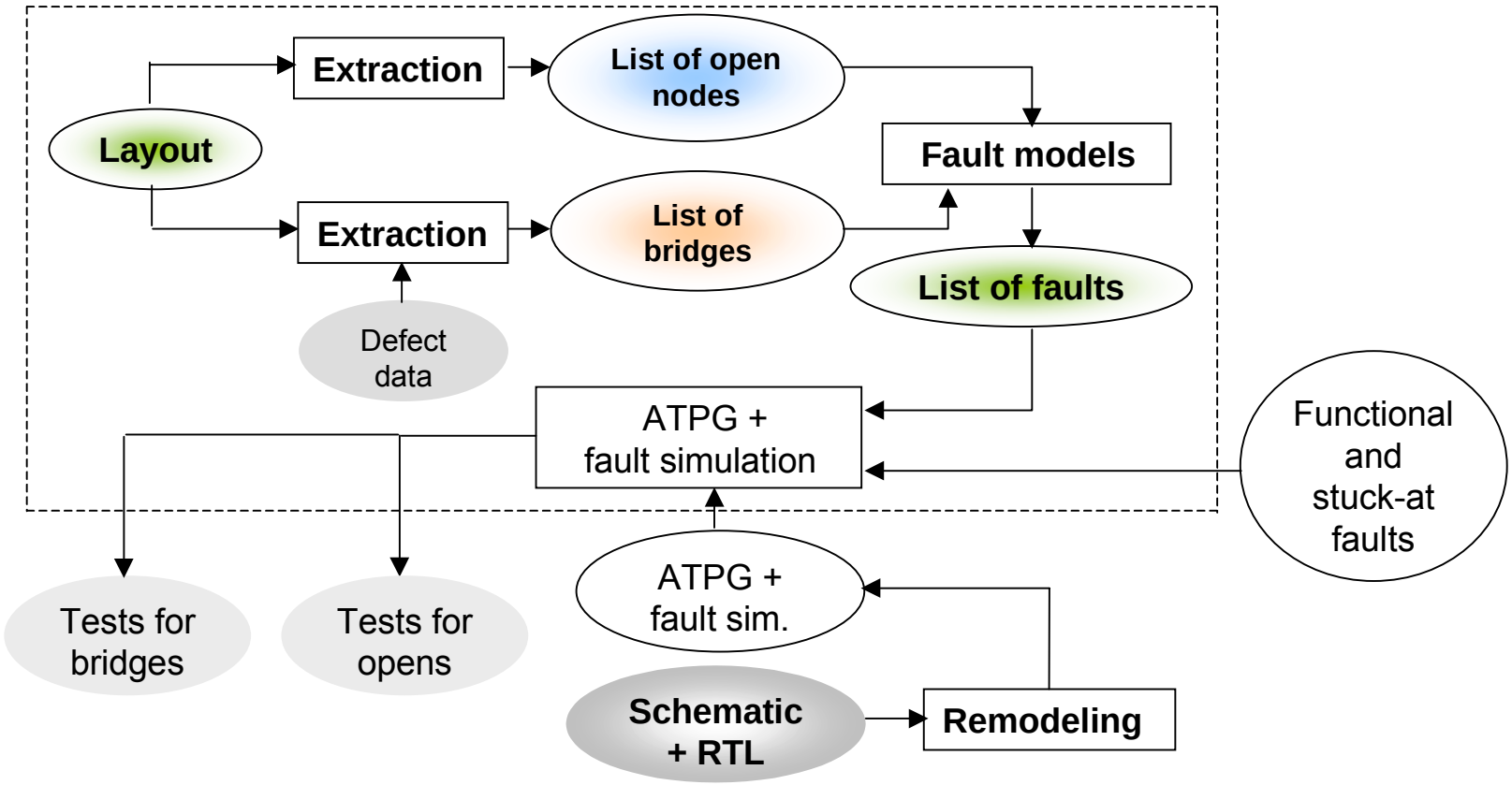
		Logical	a/0	a/1	b/0	b/1	y/0	y/1	mp1off	mp2off	mn1off	mn2off	mp1on	mp2on	mn1on	mn2on	mp1gds	mp2gds	mn1gds	mn2gds	mn1dss	mn2dss
0	00/1		0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
1	01/1		0	1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0
2	10/1		0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1
3	11/0		1	0	1	0	0	1	0	0	1	1	0	0	0	0	1	1	1	1	0	0

		Current	a/0	a/1	b/0	b/1	y/0	y/1	mp1off	mp2off	mn1off	mn2off	mp1on	mp2on	mn1on	mn2on	mp1gds	mp2gds	mn1gds	mn2gds	mn1dss	mn2dss
0	00/1		0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
1	01/1		0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0
2	10/1		0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
3	11/0		0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0

[Example of one defect table for the cell NAND for **intra gate shorts**; results of the CP 940391 UBISTA project.]

E. Gramatová, II SAS, Bratislava, Slovakia

DEFECT ORIENTED TEST GENERATION - I

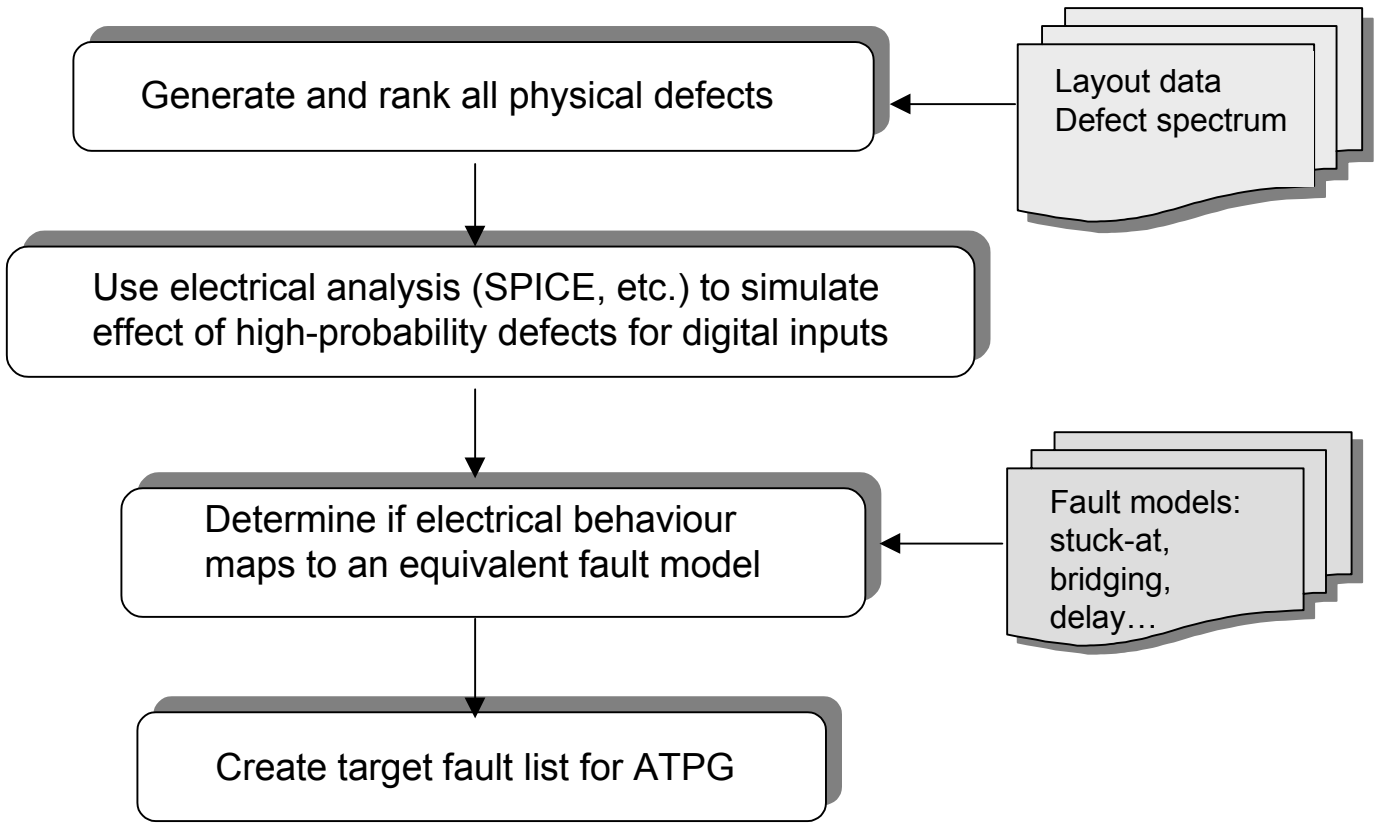


How to receive the best list of faults ?

[*S. Chakravarty*]

DEFECT ORIENTED TEST GENERATION - II

Inductive Analysis



[*M. Sachdev*]

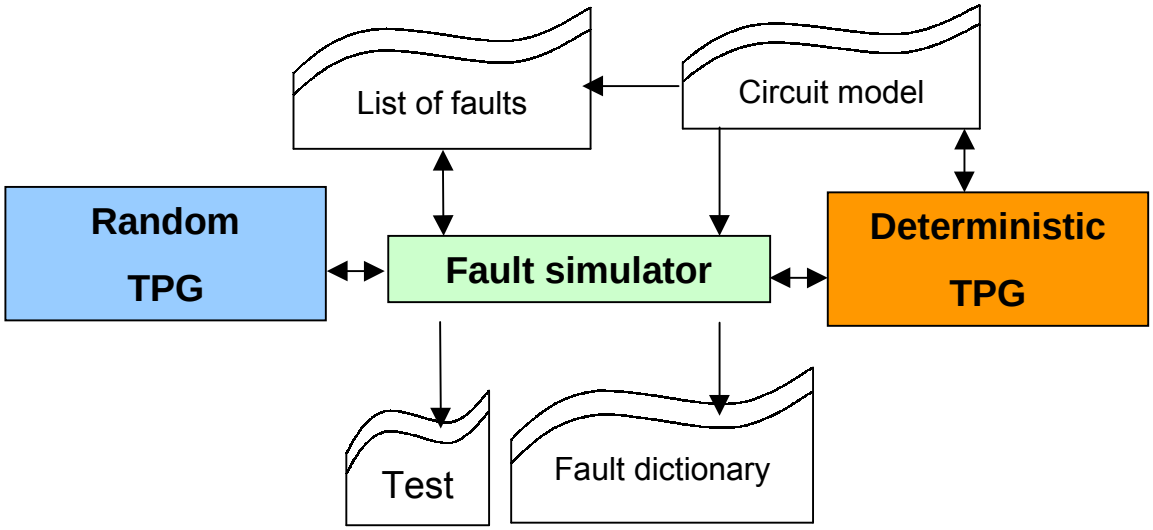
DEFECT ORIENTED TEST GENERATION - III

i	Fault d_i	Erroneous function f^{di}	P_i	Input patterns t_j															
				0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	A/C	not(A*C*(B+D))	3,1100E-07				1				1					1	1		
2	A/D	not(A*D*(B+C))	1,1940E-07				1				1					1		1	
3	A/N1	not(B*(not(A)+C+D)+C*D)	4,9059E-08					1	1	1						1			
4	A/Q	not(not(A)+C*D)	6,9159E-08	1	1	1		1	1	1						1	1	1	
5	A/GND	not(C*D)	2,6895E-08													1	1	1	
6	A/VDD	not(B+C*D)	1,9847E-08					1	1	1									
7	B/C	not(B*C*(A+D))	1,0477E-07				1								1	1	1		
8	B/D	not(B*D*(A+C))	5,7931E-08				1								1	1		1	
9	B/N1	not(A+C*D)	5,2930E-08									1	1	1					
10	B/Q	not(not(B)+C*D)	3,3912E-08	1	1	1						1	1	1		1	1	1	
11	B/GND	not(C*D)	4,6466E-08													1	1	1	
12	B/VDD	not(A+C*D)	1,8972E-08									1	1	1					
13	C/N1	not((A+B+D)*(A*B+not(C)+D))	3,9147E-08		1			1	1			1	1						
14	C/Q	not(not(C)+A*B)	9,1480E-08	1	1		1	1	1		1	1	1		1				
15	C/GND	not(A*B)	1,9862E-08				1				1				1				
16	C/VDD	not(D+A*B)	1,4727E-08		1				1				1						
17	D/N1	not((A+B+C)*(A*B+C+not(D)))	2,7604E-08			1		1		1		1		1					
18	D/Q	not(not(D)+A*B)	2,0036E-07	1		1	1	1		1	1	1		1	1				
19	D/GND	not(A*B)	2,1443E-08				1				1				1				
20	D/VDD	not(C+A*B)	9,9504E-09			1				1				1					
21	N1/Q	not(A+B)	1,3697E-07				1				1				1				
22	N1/GND	SA0 for Q	8,4883E-09	1	1	1		1	1	1		1	1	1					
23	N1/ VDD	not(C*D)	2,1532E-07													1	1	1	
24	Q/GND	SA0 for Q	1,0145E-07	1	1	1		1	1	1		1	1	1					
25	Q/ VDD	SA1 for Q	3,5661E-08				1				1				1	1	1	1	1

[Example of the defect table for the cell: AN1= NOR (AND (A,B), AND(C,D)) for **inter gate shorts with probabilistic evaluation** ; results from *WUT, Poland*]

DEFECT ORIENTED TEST GENERATION - IV

General diagram of test pattern generation system



HUGE NUMBER OF PATTERNS

HUGE NUMBER OF DEFECTS

E. Gramatová, II SAS, Bratislava, Slovakia

DEFECT ORIENTED TEST GENERATION - V

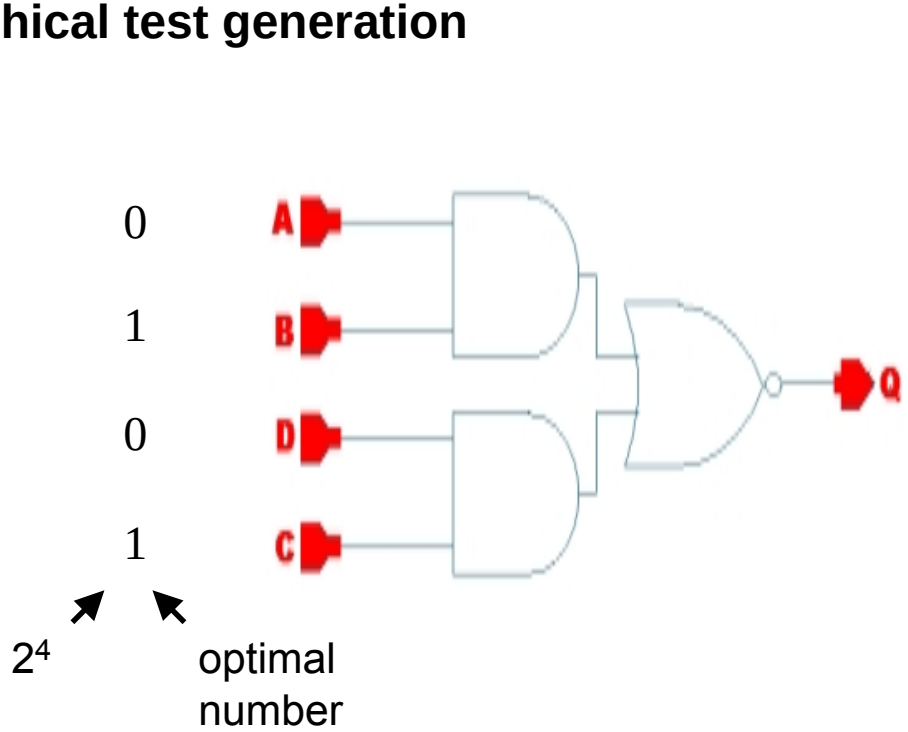
Defect table analysis

i	Fault d_i	Erroneous function f^{di}	P_i	Input patterns t_j															
				0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	A/C	not(A*C*(B+D))	3,1100E-07				1				1					1	1		
2	A/D	not(A*D*(B+C))	1,1940E-07				1				1					1		1	
3	A/N1	not(B*(not(A)+C+D)+C*D)	4,9059E-08					1	1	1						1			
4	A/Q	not(not(A)+C*D)	6,9159E-08	1	1	1		1	1	1						1	1	1	
5	A/GND	not(C*D)	2,6895E-08													1	1	1	
6	A/VDD	not(B+C*D)	1,9847E-08					1	1	1									
7	B/C	not(B*C*(A+D))	1,0477E-07				1								1	1	1		
8	B/D	not(B*D*(A+C))	5,7931E-08				1								1	1		1	
9	B/N1	not(A+C*D)	5,2930E-08									1	1	1					
10	B/Q	not(not(B)+C*D)	3,3912E-08	1	1	1						1	1	1		1	1	1	
11	B/GND	not(C*D)	4,6466E-08													1	1	1	
12	B/VDD	not(A+C*D)	1,8972E-08									1	1	1					
13	C/N1	not((A+B+D)*(A*B+not(C)+D))	3,9147E-08		1			1	1			1	1						
14	C/Q	not(not(C)+A*B)	9,1480E-08	1	1		1	1	1		1	1	1		1				
15	C/GND	not(A*B)	1,9862E-08				1				1				1				
16	C/VDD	not(D+A*B)	1,4727E-08		1				1				1						
17	D/N1	not((A+B+C)*(A*B+C+not(D)))	2,7604E-08			1		1		1		1		1					
18	D/Q	not(not(D)+A*B)	2,0036E-07	1		1	1	1		1	1	1		1	1				
19	D/GND	not(A*B)	2,1443E-08				1				1				1				
20	D/VDD	not(C+A*B)	9,9504E-09			1				1				1					
21	N1/Q	not(A+B)	1,3697E-07				1				1				1				
22	N1/GND	SA0 for Q	8,4883E-09	1	1	1		1	1	1		1	1	1					
23	N1/ VDD	not(C*D)	2,1532E-07													1	1	1	
24	Q/GND	SA0 for Q	1,0145E-07	1	1	1		1	1	1		1	1	1					
25	Q/ VDD	SA1 for Q	3,5661E-08				1				1				1	1	1	1	1

5 test patterns can be selected

DEFECT ORIENTED TEST GENERATION - VI

Hierarchical test generation



1100
0011
0101
0010
1000

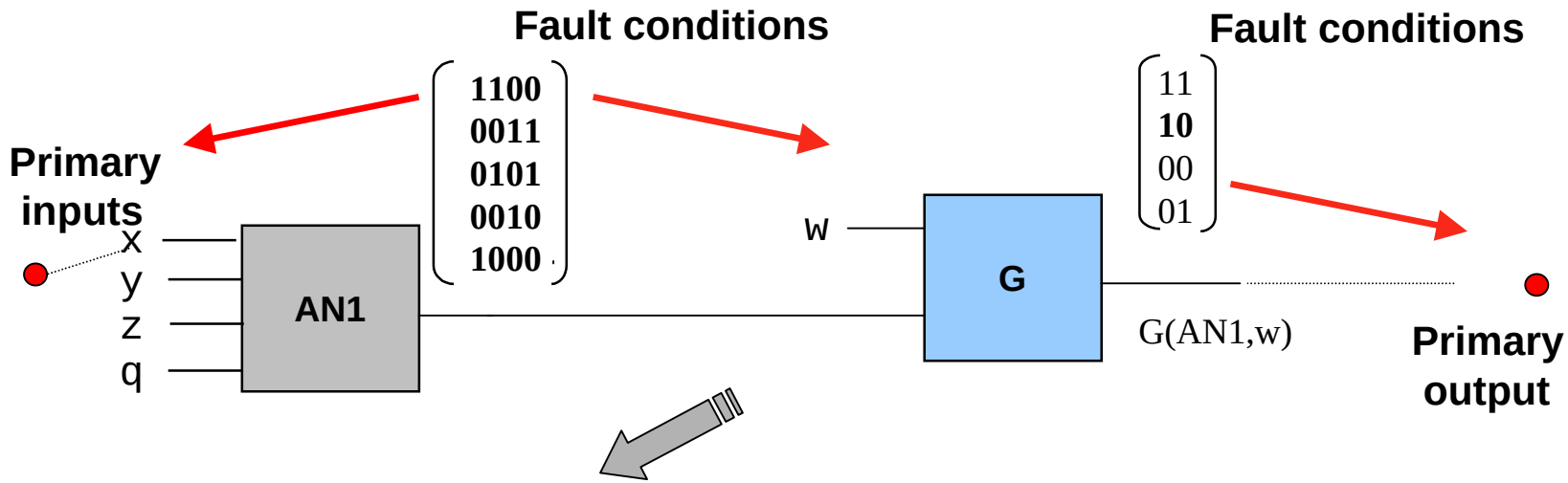
Fault conditions

Functional fault model

- **Finding an optimal test set for a cell - e.g. for AN1.**
- **Test generation with an implicit fault model - a functional fault model.**

DEFECT ORIENTED TEST GENERATION - VII

Hierarchical test generation



- How to propagate the values to primary outputs based on path sensitization ?
- How to justify desired values inside a circuit from the primary inputs ?
- How to cover other fault conditions on the another cells in paths ?

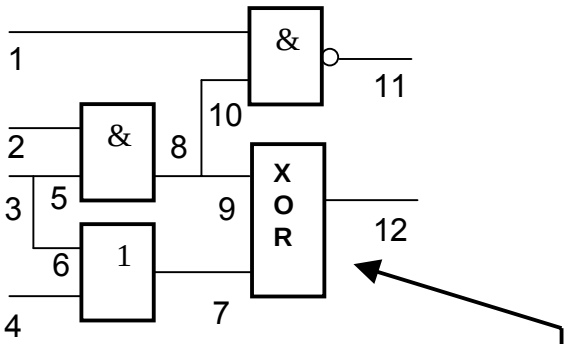
DEFECT ORIENTED TEST GENERATION - VIII

Defect localization uses only the reduced defect table, and the relative probabilistic values can be used for calculation of more realistic defect coverage.

i	Fault d_i	Erroneous function f^{di}	P_i	2	3	5	8	12
1	A/C	not(A*C*(B+D))	3,1100E-07		1			1
2	A/D	not(A*D*(B+C))	1,1940E-07		1			1
3	A/N1	not(B*(not(A)+C+D)+C*D)	4,9059E-08			1		1
4	A/Q	not(not(A)+C*D)	6,9159E-08	1		1		1
5	A/GND	not(C*D)	2,6895E-08					1
6	A/VDD	not(B+C*D)	1,9847E-08			1		
7	B/C	not(B*C*(A+D))	1,0477E-07		1			1
8	B/D	not(B*D*(A+C))	5,7931E-08		1			1
9	B/N1	not(A+C*D)	5,2930E-08				1	
10	B/Q	not(not(B)+C*D)	3,3912E-08	1			1	1
11	B/GND	not(C*D)	4,6466E-08					1
12	B/VDD	not(A+C*D)	1,8972E-08				1	
13	C/N1	not((A+B+D)*(A*B+not(C)+D))	3,9147E-08			1	1	
14	C/Q	not(not(C)+A*B)	9,1480E-08		1	1	1	
15	C/GND	not(A*B)	1,9862E-08		1			
16	C/VDD	not(D+A*B)	1,4727E-08			1		
17	D/N1	not((A+B+C)*(A*B+C+not(D)))	2,7604E-08	1			1	
18	D/Q	not(not(D)+A*B)	2,0036E-07	1	1		1	
19	D/GND	not(A*B)	2,1443E-08		1			
20	D/VDD	not(C+A*B)	9,9504E-09	1				
21	N1/Q	not(A+B)	1,3697E-07		1			
22	N1/GND	SA0 for Q	8,4883E-09	1		1	1	
23	N1/ VDD	not(C*D)	2,1532E-07					1
24	Q/GND	SA0 for Q	1,0145E-07	1		1	1	
25	Q/ VDD	SA1 for Q	3,5661E-08		1			1

EXAMPLE - I

Defect detection



- ***Non detectable fault conditions***
(number of gate, fault condition)

XOR 12 C01/1 V01/1

*- *)Aborted*

for I_{DDQ} testing

for logical testing

```
* TIME: 0:00:00
* Coverage: 92.31 %
* C:\Users\frrn\example\example.isc
 IDDQ/Voltage Test Set
* Library: C:\Users\frrn\defgen\FCLib\sf.ds
** REMARK - ORDER OF VALUES OF TEST SET:
* NUMBER OF TEST, V/C TEST, INPUTS
*  OUTPUTS
* V/C TEST:  01 Current Test, 10 Voltage Test11
Current/Voltage Test
* #
* INPUTS: 1 2 3 6
* OUTPUTS: 11 12
*
* TEST SET
0 11    1100
 HL
1 11    X010
 1H
2 11    XX01
 1H
3 11    011X
 HL
4 11    111X
 L0
```

H = 1
L = 0

X - don‘t care value

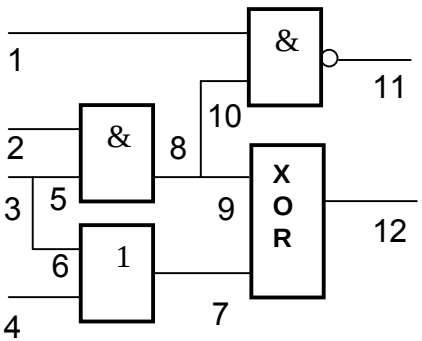
EXAMPLE - II

Defect localization for logical testing

- Non detectable fault conditions (number of gate, fault condition).

XOR 12 C01/1 V01/1

*- *)Aborted*



--------- Undetectable defects ------

XOR 12: mp1ds

--------- Untestable defects ------
OR2 mp1off mp2off mp3off mn1off mn2off mp1on mp2on mp3on mn1on mn2on mn3on
AND2 mp1off mp2off mp3off mp1on mp2on mp3on mn1on mn2on mn3on
NAND2 mp1off mp2off mp1on mp2on mn1on mn2on
XOR2 mp1do mp1ds mp1so mp2do mp2ds mp2so mp4do mp4so mp6do mp6so

Whole number of defects	: 110
Number of non-covered defects	: 37
Number of untestable defects	: 36
Number of aborted defects	: 0
Percentual covering of defects	: 66. 363636
Percentual covering of "one test pattern" defects	: 98. 648648

I_{DDQ} tests should be used for untestable defects

(18 untestable faults 98,92 % fault coverage)

E. Gramatová, II SAS, Bratislava, Slovakia

CONCLUSION

Defect oriented testing should play a central role in the future test strategy.

Hierarchical defect oriented test generation is a promising solution in testing complex digital systems.

New technologies bring new defects = new problems with fault modeling.

It is a gap between academic research and industry.

Research in this field needs commercial support (some results from chip fabrication, new defects, ...).

Thank you for your attention.