VLSI Engineering and Design Automation Division Seminar
WUT - November 16, 2007, Warsaw, Poland

Studying CMOS Defects with DefSim

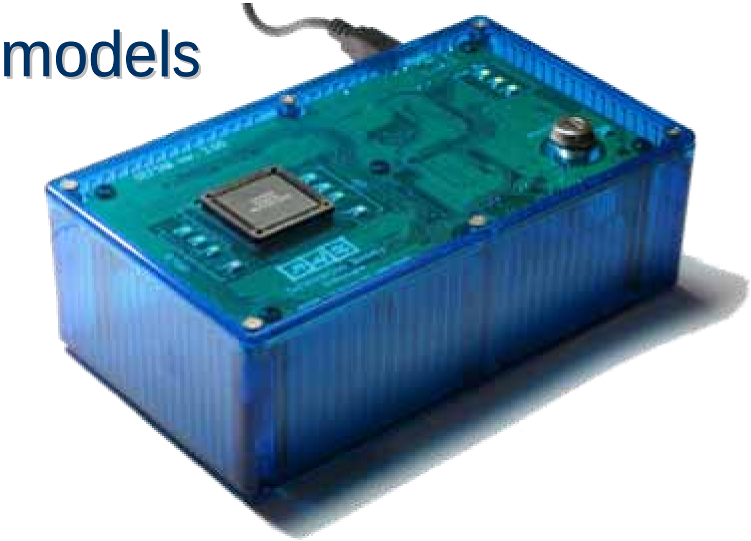
Overview and Demonstration

A. Jutman[1], W. A. Pleskacz[2]

[1]Tallinn University of Technology, Estonia
[2]Warsaw University of Technology, Poland
artur@pld.ttu.ee, pleskacz@imio.pw.edu.pl

Outline

- Introduction and Motivation
- Implementation of defects
- DefSim chip structure and design
- DefSim in a lab and classroom
- Experiments with defects and models
- DefSim software
- Conclusions and Future work



Project Funding

Financial support for DefSim hardware development was provided by:

- REASON (IST-2000-30193), an IST project funded by the European Union;
- Polish State Committee for Scientific Research (project No. 4 T11B 023 24);
- Estonian Science Foundation grants G5649, G5910 and Enterprise Estonia.

DefSim Personal and DefSim Server software was developed by Testonica Lab free of charge

Project Team

Warsaw University of Technology

Coordinator: Dr. Witold Pleskacz

T. Borejko
P. Pizon
P. Trochimiuk
T. Gugala
A. Walkanis
K. Bolczak

Tallinn University of Technology

TTÜ 1918

Coordinator: Prof. Raimund Ubar

A. Jutman
S. Devadze
E. Orasson

Slovak University of Technology

Dr. Viera Stopjakova

The team would like to thank Prof. W. Kuzmicz from Warsaw University of Technology for his support and valuable feedback

Motivation

- ITRS: testing is already one of key problems in current generation of VLSI chips and its importance will be growing.
- There is a strong demand for well-educated specialists in the area of IC testing.
- Teaching of testing topics is based on abstract models which do not reflect well physical reality.
- Simulation alone is not sufficient, “real silicon” is needed for better understanding of test problems and for training in fault simulation and test pattern generation.

Goals

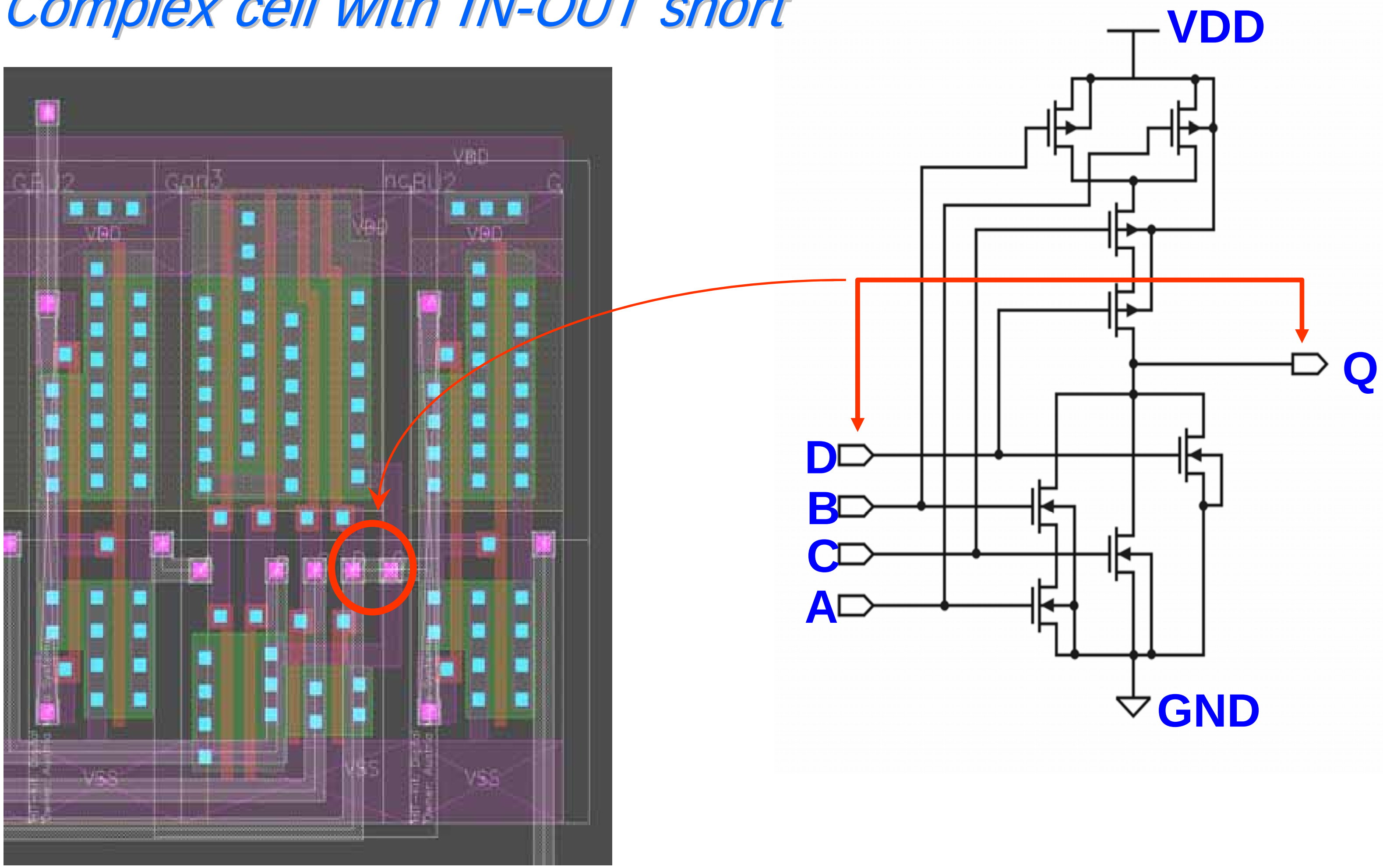
Development of a VLSI chip providing practical possibility for testing real circuits with *artificially introduced* but *real* physical defects, and in particular:

- demonstration of defects that are not easy to be modeled or for which existing fault models do not follow the real electrical effect of the respective defect precisely,
- testing of simple gates as well as gates embedded in larger digital circuits,
- comparison of theoretical and real defect coverage for various sets of test vectors,
- demonstration of I_{DDQ} testing and its efficiency,
- making the chip “student-proof” and avoiding possible damage resulting from its operation with intentionally introduced defects.

Created environment can be used in research also (e.g. in evaluation of defect-oriented test pattern generation methods and tools).

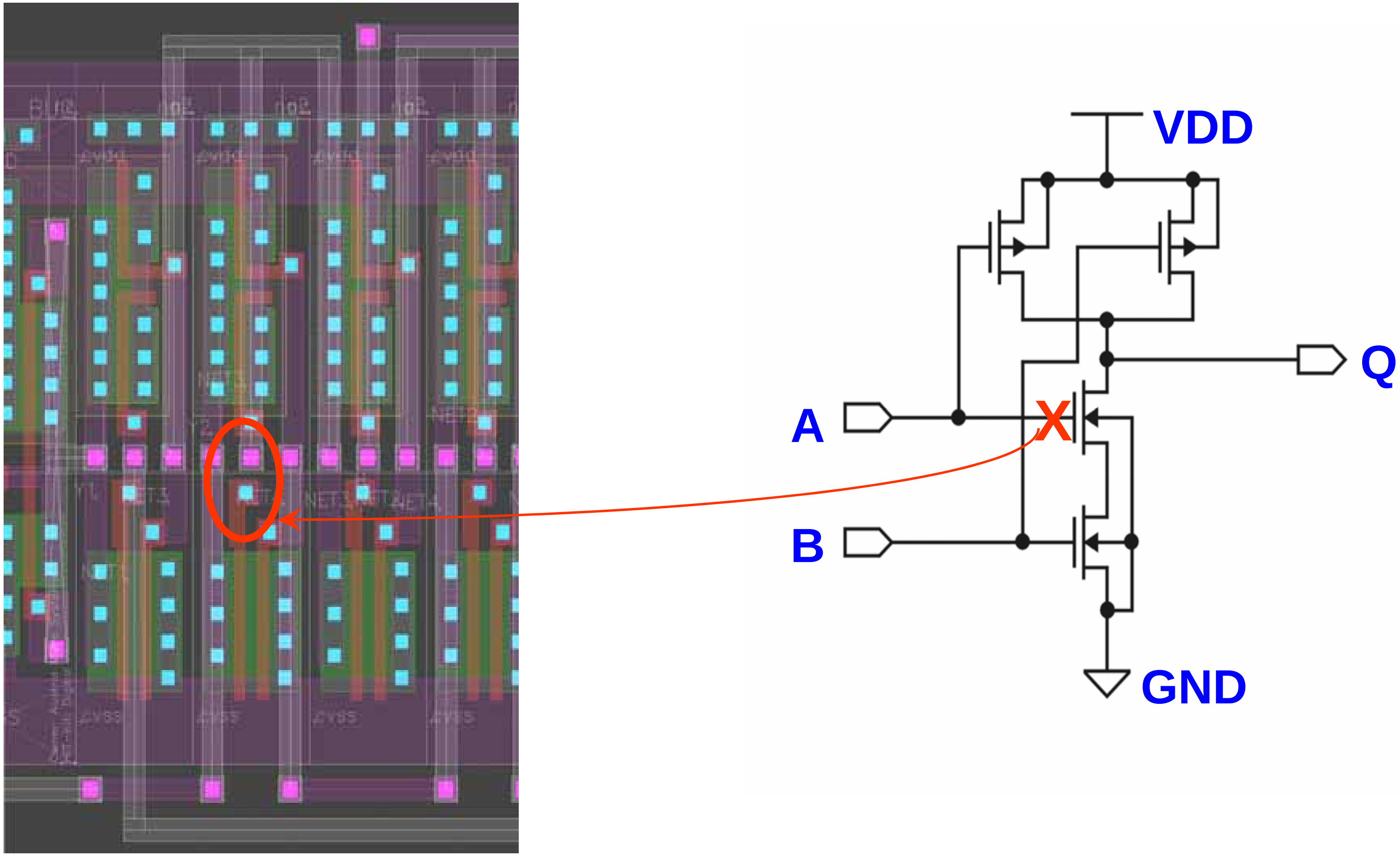
Implementation of defects

Complex cell with IN-OUT short



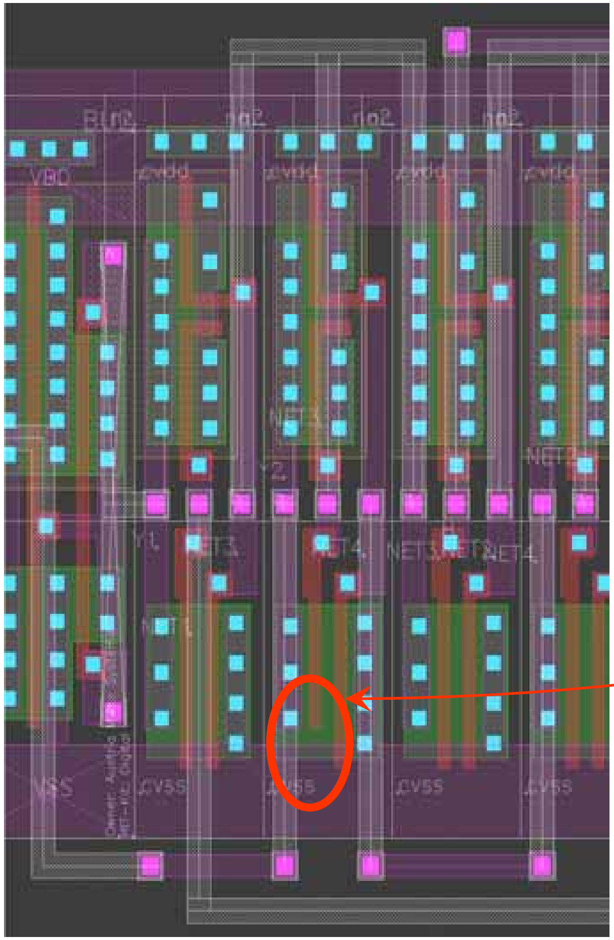
Implementation of defects

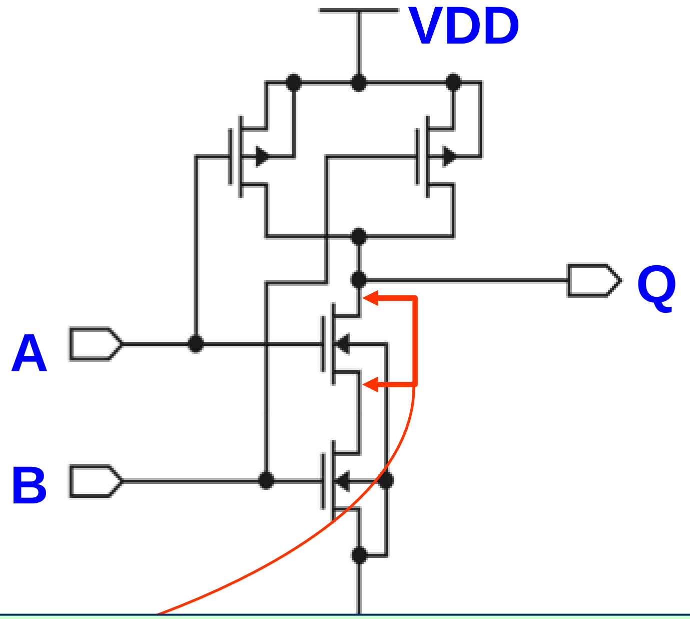
NAND2 cell with floating gate



Implementation of defects

NAND2 cell with D-S short (missing poly)

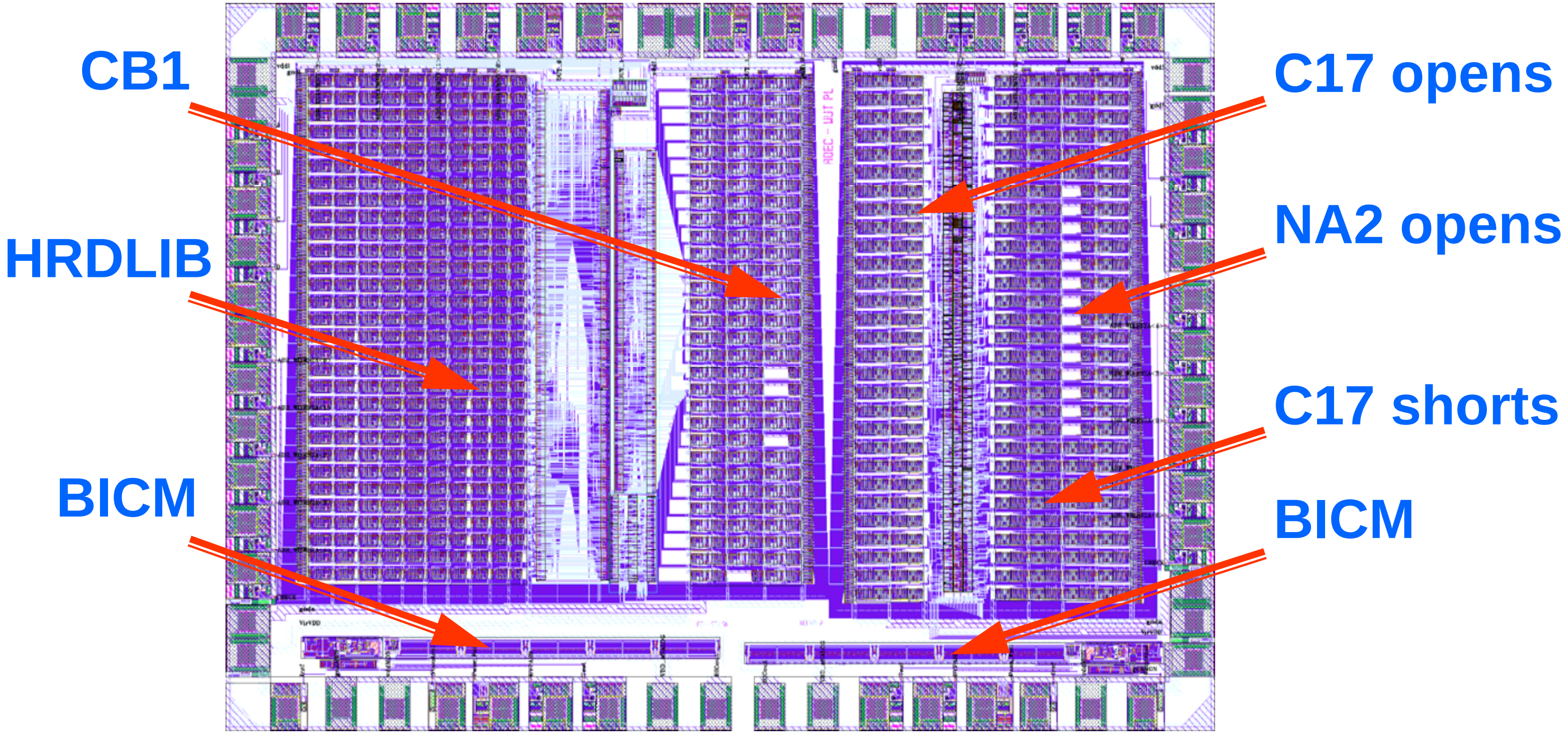




- Altogether there are **over 500** different defects on the chip
- Implemented defects are **shorts and opens** in metal and poly layers
- To be close to the silicon reality each cell is loaded and driven by standard non-inverting buffers

Artur Jutman, Warsaw, Poland, November 16, 2007

DefSim IC layout



CB1

HRDLIB

BICM

C17 opens

NA2 opens

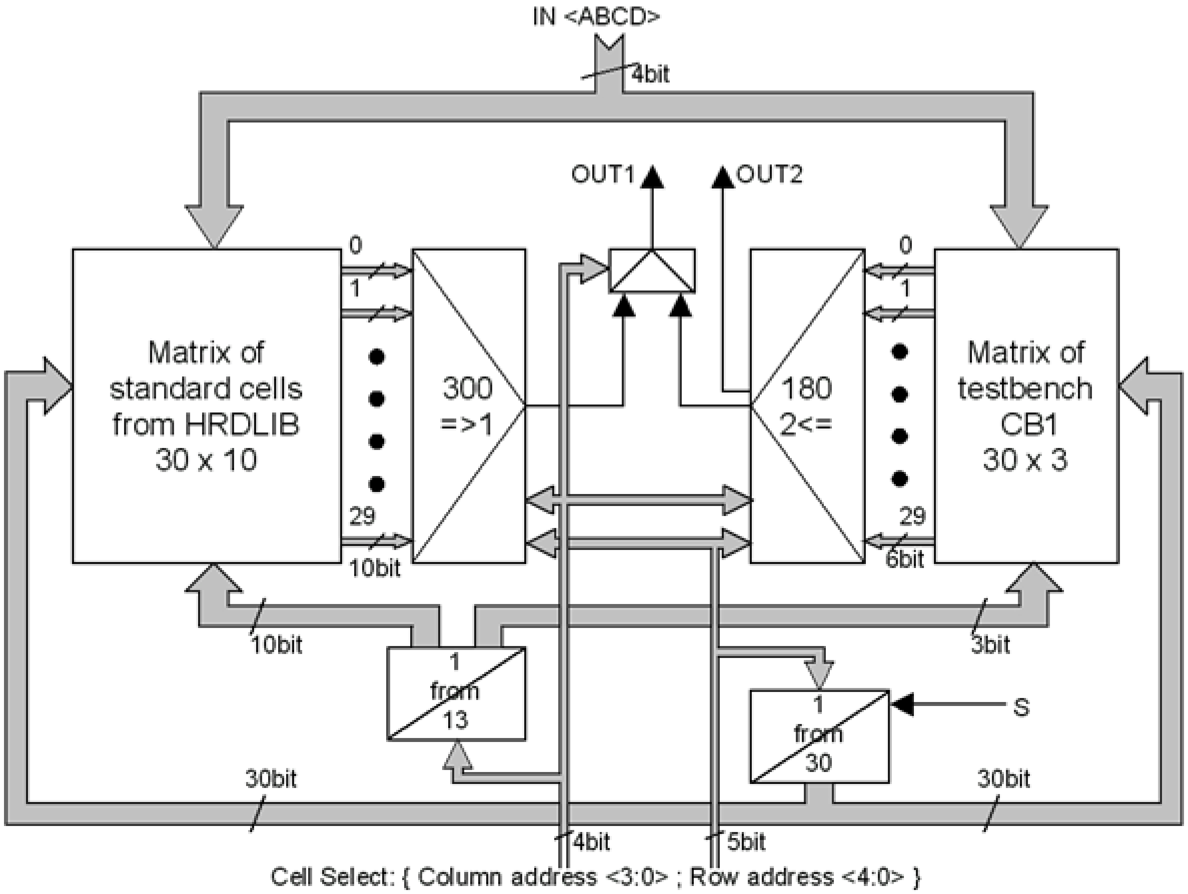
C17 shorts

BICM

16 different complex cells and small circuits implemented

For each cell there is a fault-free circuit and many copies of the same cell with intentionally injected defects.

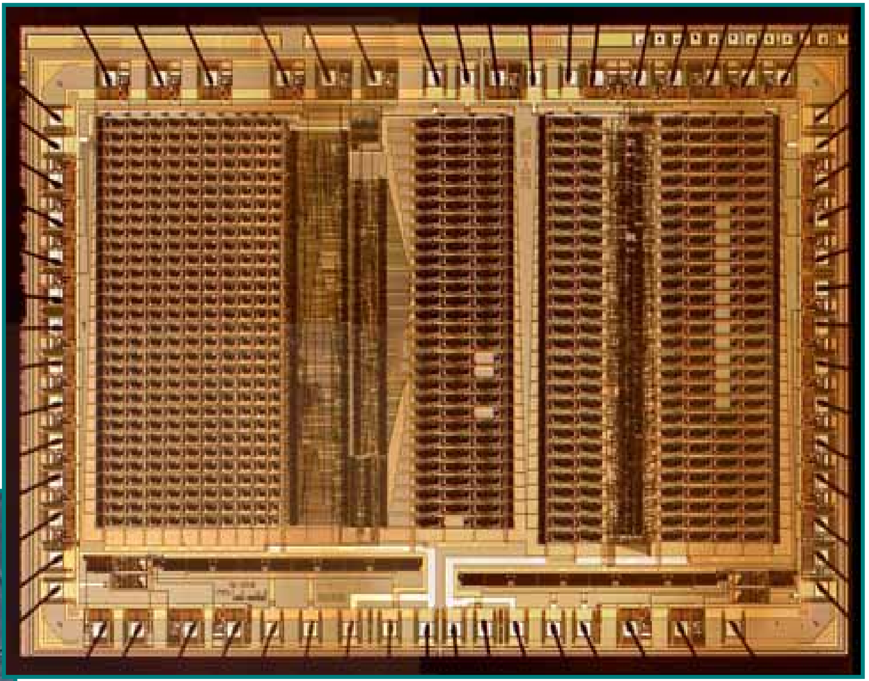
Schematic diagram of *block1*


IN <ABCD>
4bit
OUT1
OUT2
0
1
29
10bit
Matrix of standard cells from HRDLIB 30 x 10
300 =>1
180 2<=
0
1
29
6bit
Matrix of testbench CB1 30 x 3
10bit
3bit
1 from 13
1 from 30
S
30bit
30bit
4bit
5bit
Cell Select: { Column address <3:0> ; Row address <4:0> }

Only one circuit at a time can be active (selected)

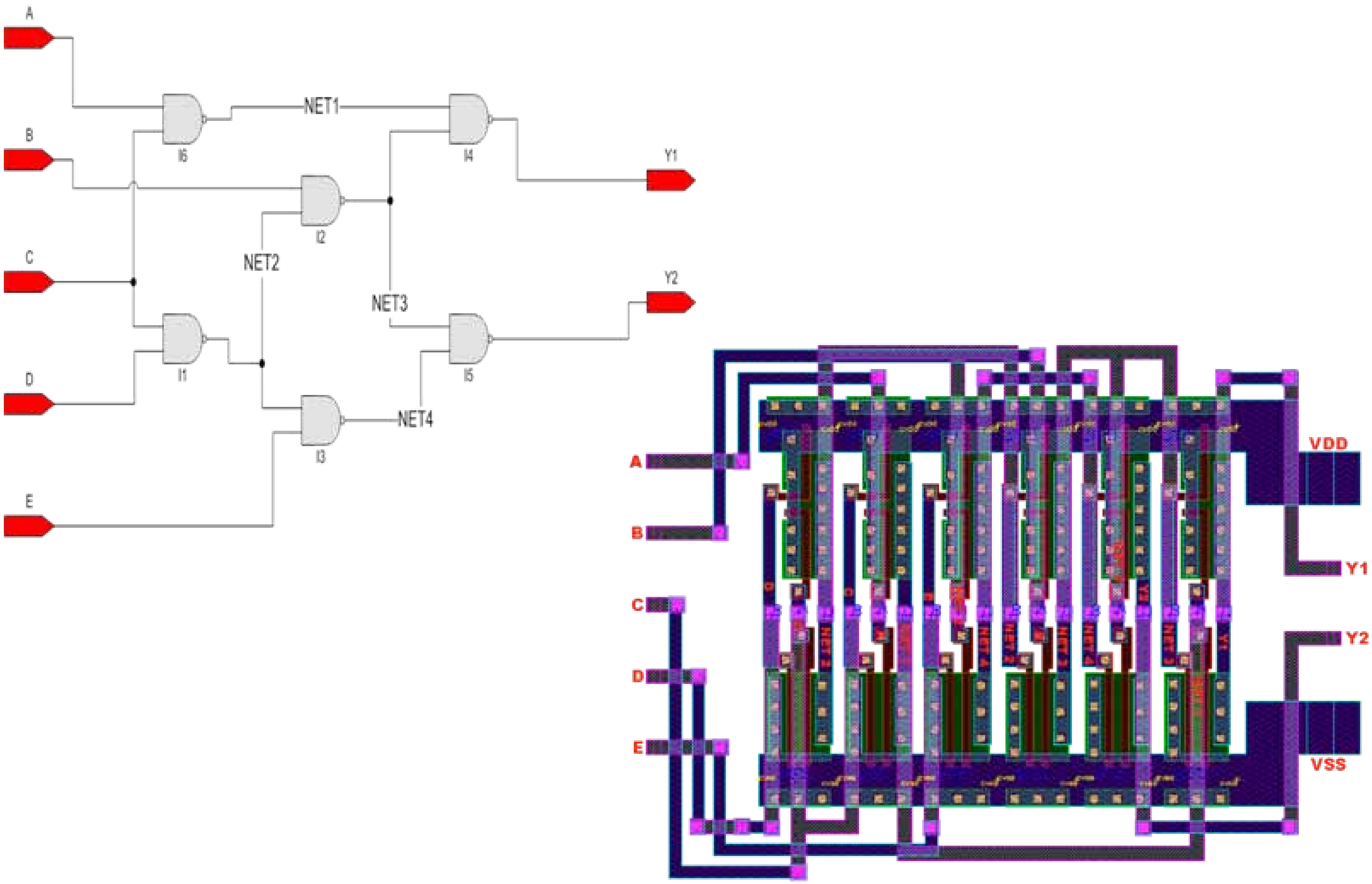
DefSim IC details

- Standard industrial technology (AMS CMOS 0.8µm CYE - 2M/2P)
- Area 19.90 mm^2
- Approx. 48000 transistors
- 62 pins, JLCC68 package

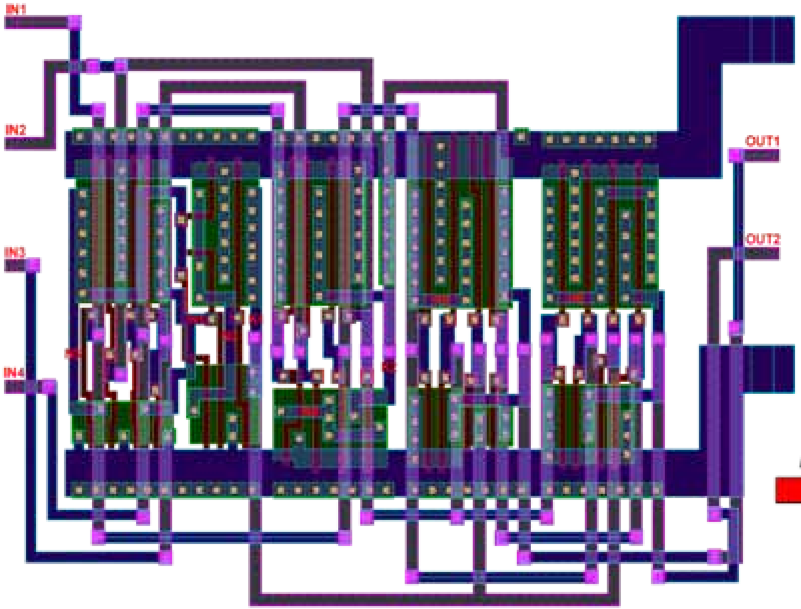


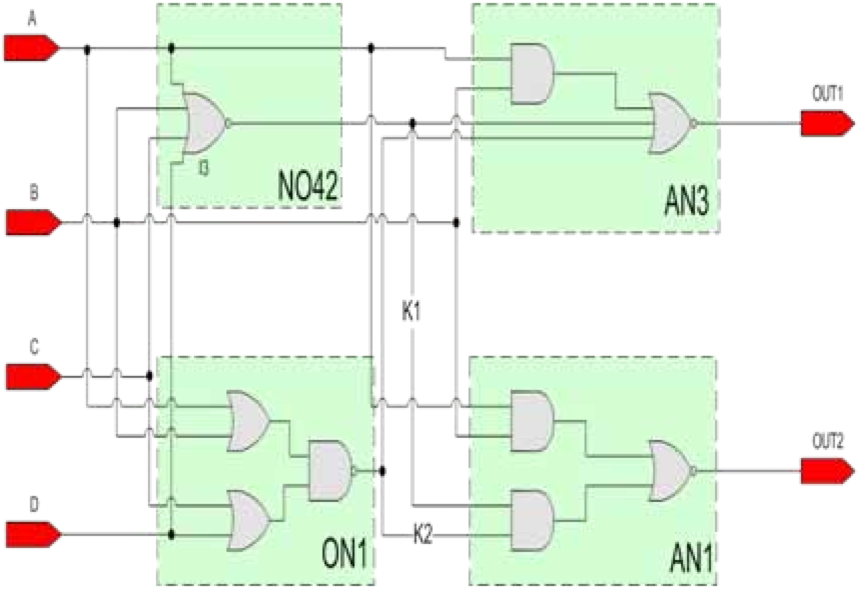
A built-in current monitor for *I_{DDQ} testing* is implemented in each block.

DefSim Cell Examples: c17



DefSim Cell Examples: CB_1



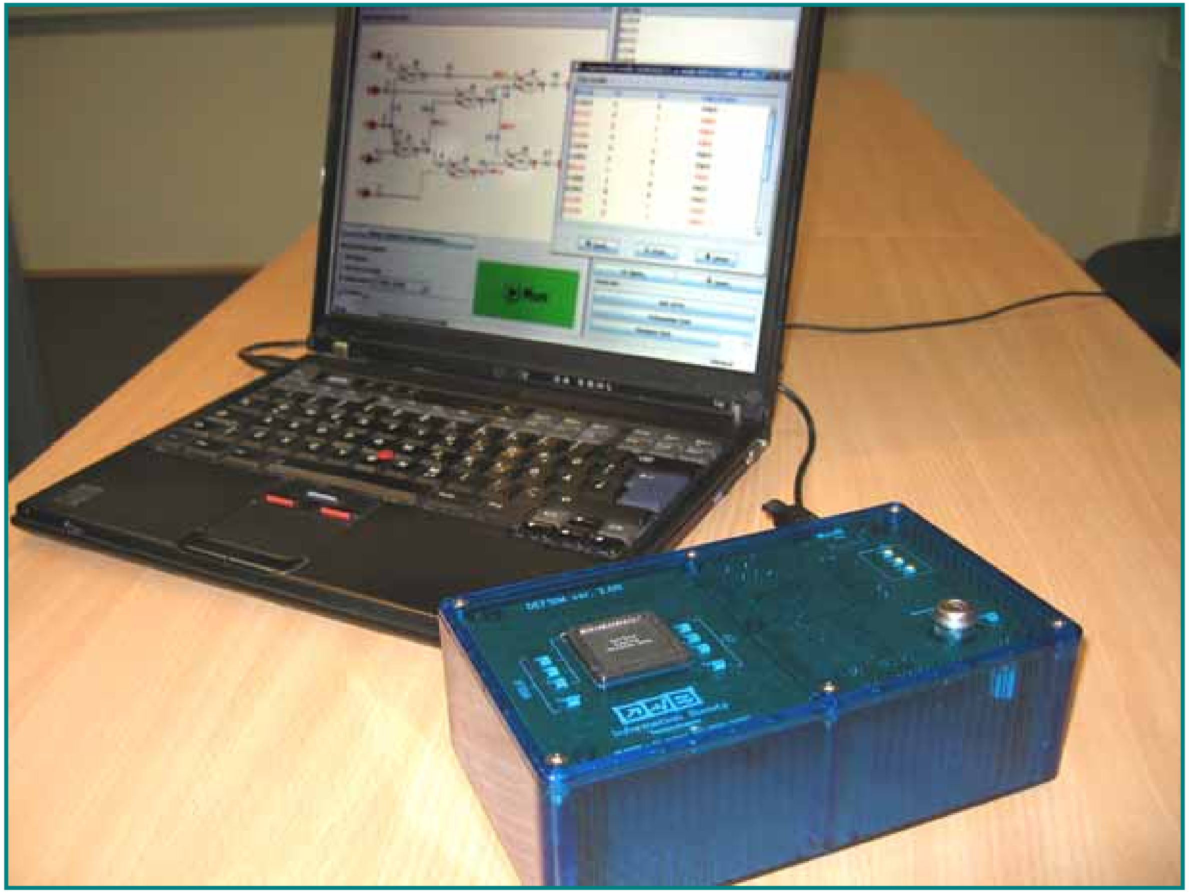


DefSim in the classroom

With DefSim you can

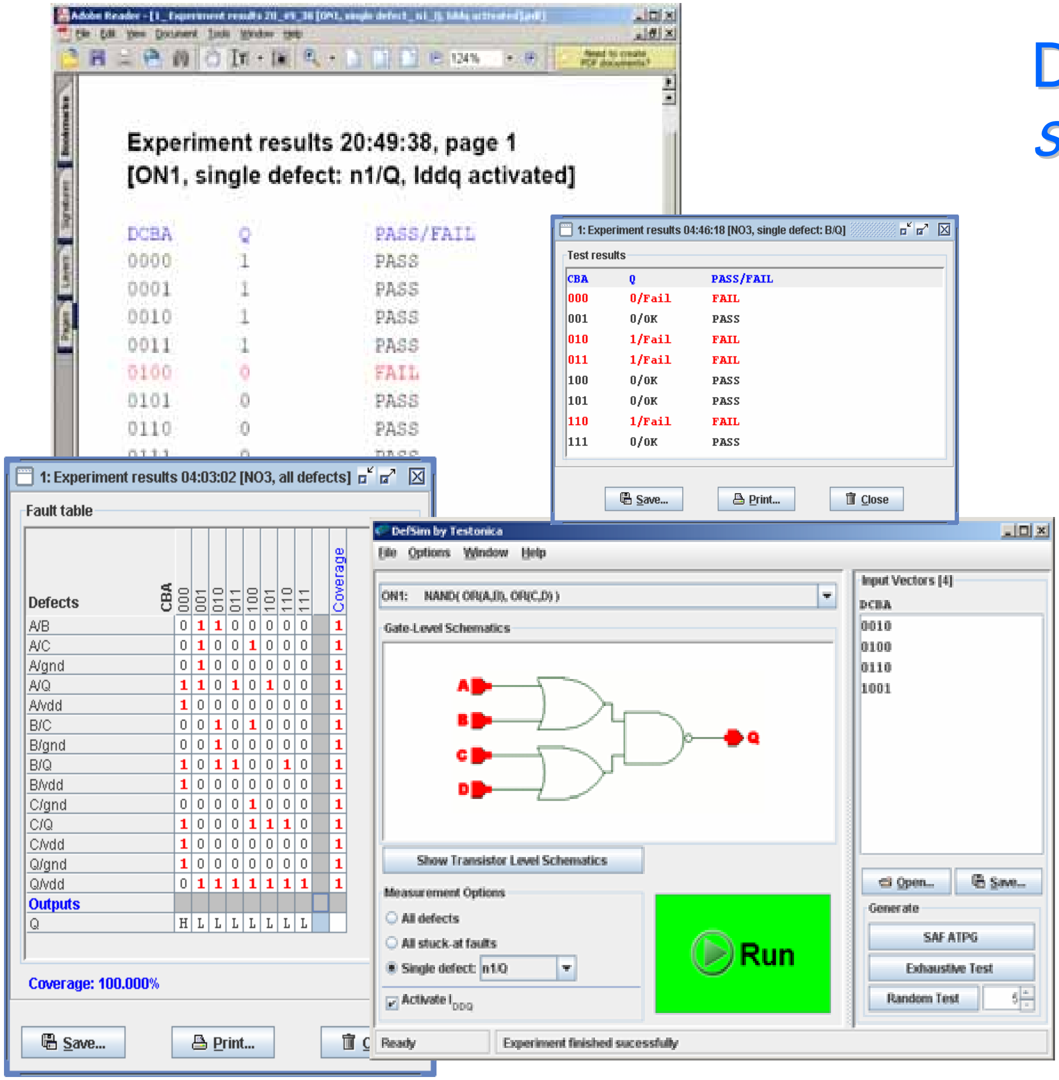
- Observe the truth table of correct circuit
- Observe the truth table of defective circuit
- Obtain defect/fault tables for all specific defects
- Define test patterns automatically or manually
- Activate IDDQ and voltage measurements
- Study behavior of bridging and open faults
- Study and compare different fault models

Lab environment



“Plug and Play” – dedicated hardware and software

DefSim software



Experiment results 20:49:38, page 1
[ON1, single defect: n1/Q, Iddq activated]

DCBA	Q	PASS/FAIL
0000	1	PASS
0001	1	PASS
0010	1	PASS
0011	1	PASS
0100	0	FAIL
0101	0	PASS
0110	0	PASS
0111	0	PASS

1: Experiment results 04:46:18 [NO3, single defect: B/Q]

Test results

CBA	Q	PASS/FAIL
000	0/Fail	FAIL
001	0/OK	PASS
010	1/Fail	FAIL
011	1/Fail	FAIL
100	0/OK	PASS
101	0/OK	PASS
110	1/Fail	FAIL
111	0/OK	PASS

1: Experiment results 04:03:02 [NO3, all defects]

Fault table

Defects	CBA 000	001	010	011	100	101	110	111		Coverage
A/B	0	1	1	0	0	0	0	0		1
A/C	0	1	0	0	1	0	0	0		1
A/gnd	0	1	0	0	0	0	0	0		1
A/Q	1	1	0	1	0	1	0	0		1
A/vdd	1	0	0	0	0	0	0	0		1
B/C	0	0	1	0	1	0	0	0		1
B/gnd	0	0	1	0	0	0	0	0		1
B/Q	1	0	1	1	0	0	1	0		1
B/vdd	1	0	0	0	0	0	0	0		1
C/gnd	0	0	0	0	1	0	0	0		1
C/Q	1	0	0	0	1	1	1	0		1
C/vdd	1	0	0	0	0	0	0	0		1
Q/gnd	1	0	0	0	0	0	0	0		1
Q/vdd	0	1	1	1	1	1	1	1		1
Outputs										
Q	H	L	L	L	L	L	L	L		

Coverage: 100.000%

DefSim by Testonica

ON1: NAND(OR(A,B), OR(C,D))

Gate-Level Schematics

Measurement Options

○ All defects
○ All stuck-at faults
● Single defect: n1/Q
☑ Activate I_{DDQ}

Input Vectors [4]

```
DCBA
0010
0100
0110
1001
```

Random Test 5

Ready

Experiment finished sucessfully

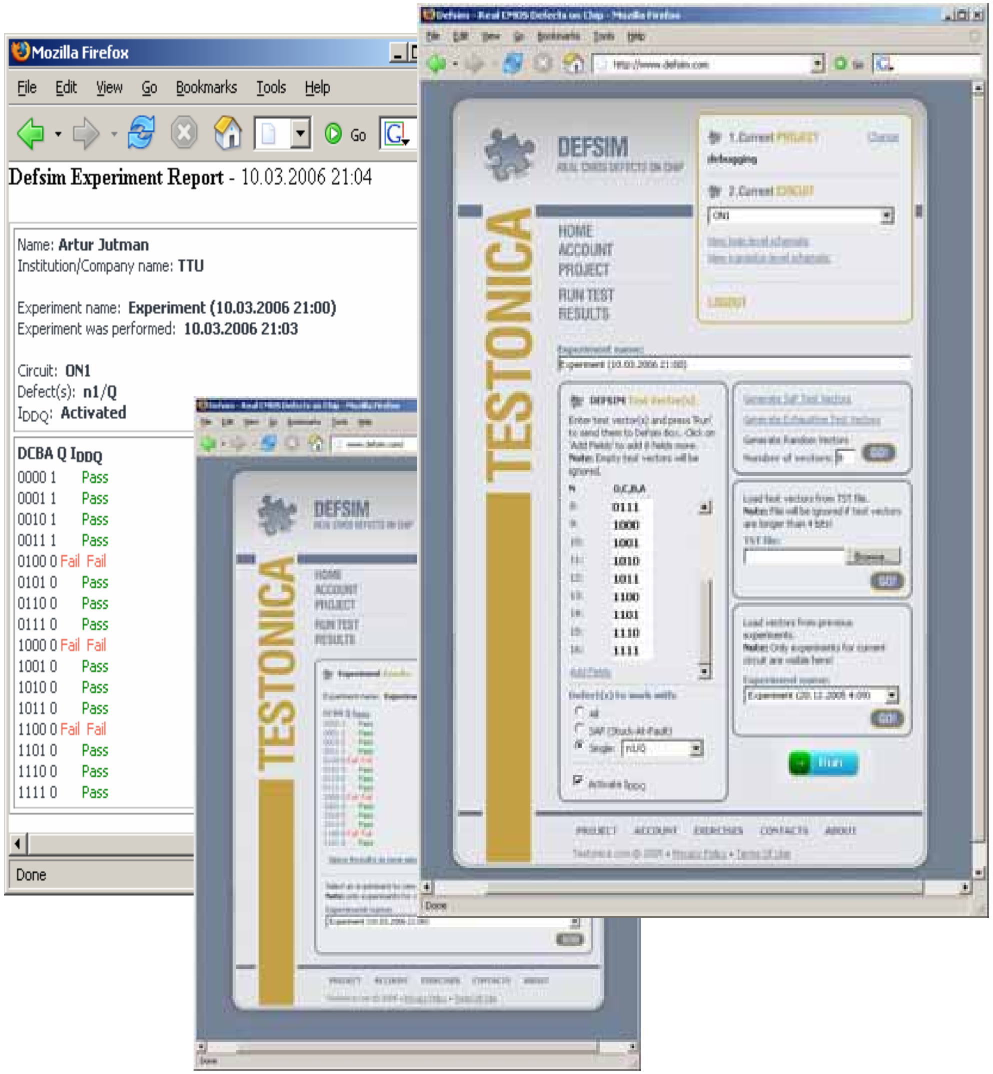
DefSim Personal – *simple but powerful.*

Personal solution allows to experiment with different aspects of IC testing shortly after connecting DefSim hardware with computer. Ideal for standalone use or for small groups of users.

DefSim software

DefSim Server – *advantage by sharing*.

Built on a top of modern Web-technologies DefSim Server is capable to share single or several hardware devices between multiple users. Best suitable for organizing study process in classrooms, distance e-learning or remote Web-access services.



Defsim Experiment Report - 10.03.2006 21:04

Name: **Artur Jutman**
Institution/Company name: **TTU**

Experiment name: **Experiment (10.03.2006 21:00)**
Experiment was performed: **10.03.2006 21:03**

Circuit: **ON1**
Defect(s): **n1/Q**
I_{DDQ}: **Activated**

```
DCBA Q IDDQ
0000 1      Pass
0001 1      Pass
0010 1      Pass
0011 1      Pass
0100 0 Fail Fail
0101 0      Pass
0110 0      Pass
0111 0      Pass
1000 0 Fail Fail
1001 0      Pass
1010 0      Pass
1011 0      Pass
1100 0 Fail Fail
1101 0      Pass
1110 0      Pass
1111 0      Pass
```

Experiment name: Experiment (10.03.2006 21:00)

Enter test vector(s) and press 'Run' to send them to Defsim Box. Click on 'Add Fields' to add 8 fields more.
Note: Empty test vectors will be ignored.

```
N    D,C,B,A
8:   0111
9:   1000
10:  1001
11:  1010
12:  1011
13:  1100
14:  1101
15:  1110
16:  1111
```

Add Fields

Defect(s) to work with:
○ All
○ SAF (Stuck-At-Fault)
● Single: n1/Q

☑ Activate I_{DDQ}

Load test vectors from TST file.
Note: File will be ignored if test vectors are longer than 4 bits!
TST file:

Load vectors from previous experiments.
Note: Only experiments for current circuit are visible here!
Experiment name: Experiment (20.12.2005 4:00)

Artur Jutman, Warsaw, Poland, November 16, 2007

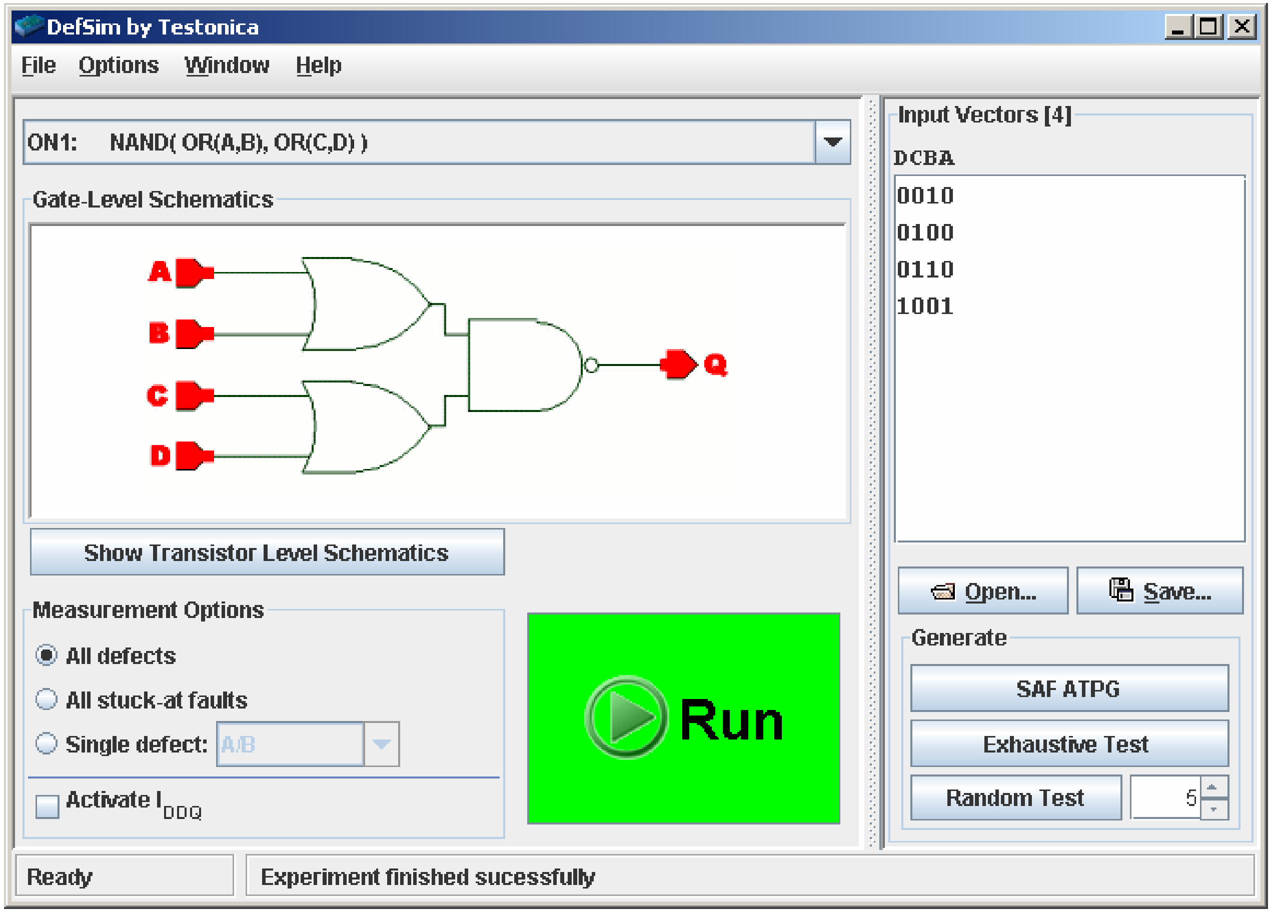
Outline

- Introduction and Motivation
- Implementation of defects
- DefSim chip structure and design
- DefSim in a lab and classroom
- DefSim software
- Experiments with defects and models
- Conclusions and Future work

Basic experiments in the classroom

Evaluating stuck-at fault model

DCBA
0010
0100
0110
1001



DefSim by Testonica

ON1: NAND(OR(A,B), OR(C,D))

Input Vectors [4]

```
DCBA
0010
0100
0110
1001
```

Measurement Options

● All defects
○ All stuck-at faults
○ Single defect: A/B

□ Activate I_{DDQ}

Random Test: 5

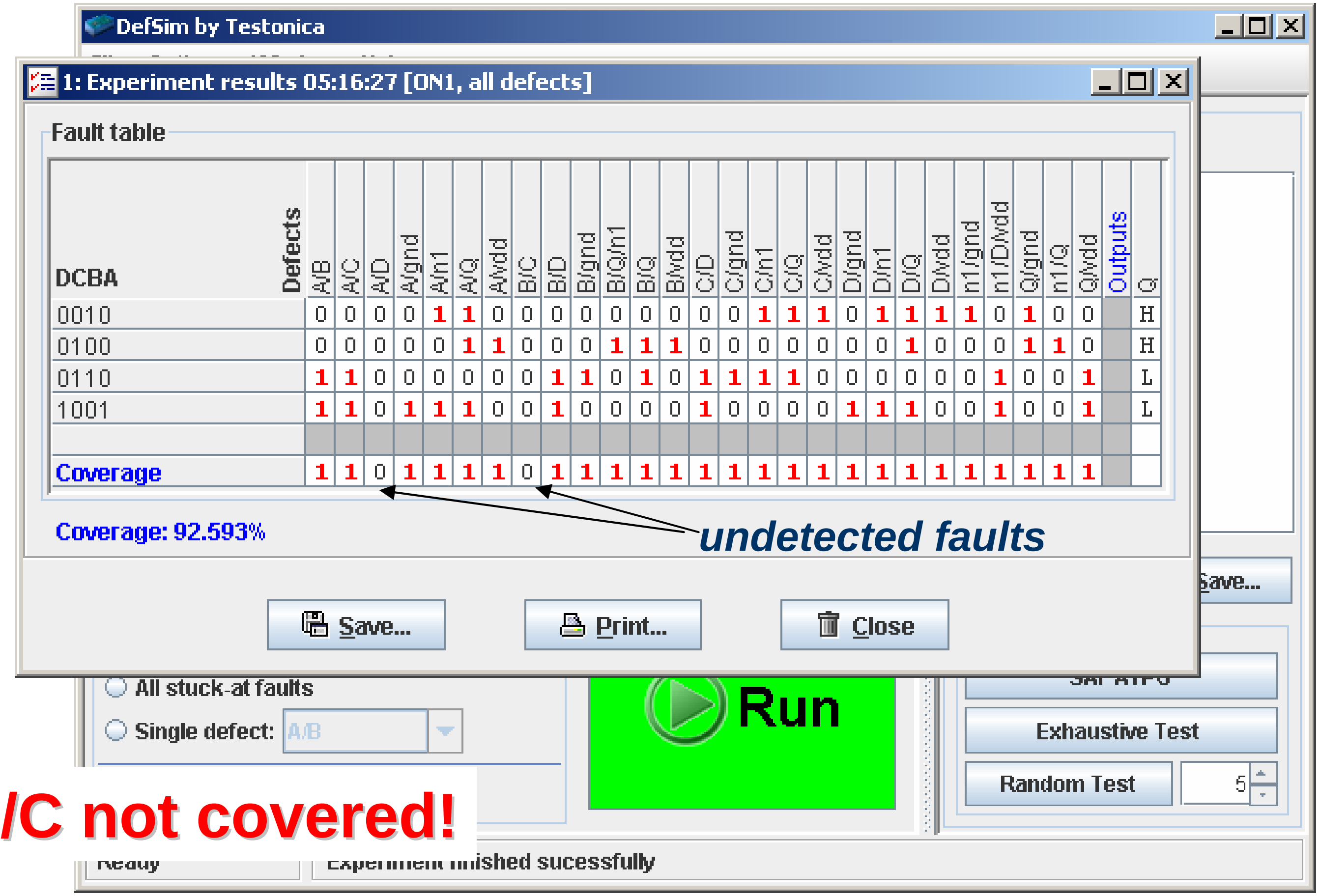
Ready

Experiment finished sucessfully

Basic experiments in the classroom

Evaluating stuck-at fault model

DCBA
0010
0100
0110
1001



DCBA / Defects	A/B	A/C	A/D	A/gnd	A/n1	A/Q	A/vdd	B/C	B/D	B/gnd	B/Q/n1	B/Q	B/vdd	C/D	C/gnd	C/n1	C/Q	C/vdd	D/gnd	D/n1	D/Q	D/vdd	n1/gnd	n1/D/vdd	Q/gnd	n1/Q	Q/vdd	Outputs	Q
0010	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0	1	0	0		H
0100	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0		H
0110	1	1	0	0	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	1	0	0	1		L
1001	1	1	0	1	1	1	0	0	1	0	0	0	0	1	0	0	0	0	1	1	1	0	0	1	0	0	1		L
Coverage	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

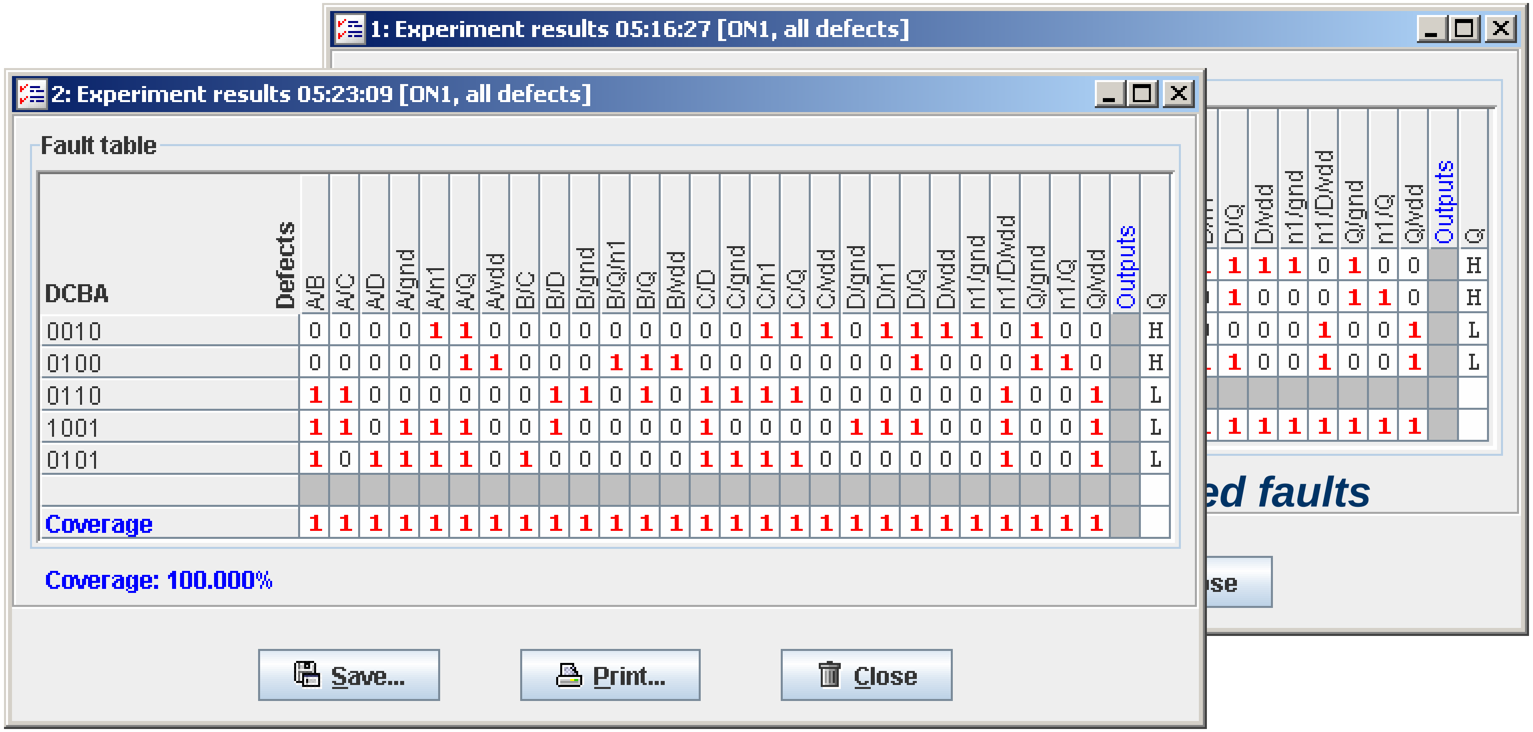
Coverage: 92.593%

undetected faults

A/D and B/C not covered!

Basic experiments in the classroom

Evaluating stuck-at fault model



1: Experiment results 05:16:27 [ON1, all defects]

D/Q	D/vdd	n1/gnd	n1/D/vdd	Q/gnd	n1/Q	Q/vdd	Outputs	Q
1	1	1	0	1	0	0		H
1	0	0	0	1	1	0		H
0	0	0	1	0	0	1		L
1	0	0	1	0	0	1		L
1	1	1	1	1	1	1		

ed faults

2: Experiment results 05:23:09 [ON1, all defects]

Fault table

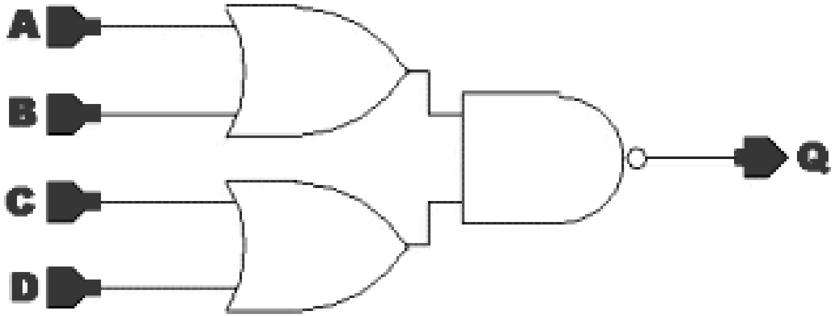
Defects DCBA	A/B	A/C	A/D	A/gnd	A/n1	A/Q	A/vdd	B/C	B/D	B/gnd	B/Q/n1	B/Q	B/vdd	C/D	C/gnd	C/n1	C/Q	C/vdd	D/gnd	D/n1	D/Q	D/vdd	n1/gnd	n1/D/vdd	Q/gnd	n1/Q	Q/vdd	Outputs	Q
0010	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0	1	0	0		H
0100	0	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0		H
0110	1	1	0	0	0	0	0	0	1	1	0	1	0	1	1	1	1	0	0	0	0	0	0	1	0	0	1		L
1001	1	1	0	1	1	1	0	0	1	0	0	0	0	1	0	0	0	0	1	1	1	0	0	1	0	0	1		L
0101	1	0	1	1	1	1	0	1	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	1		L
Coverage	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

Coverage: 100.000%

Either 0101 or 1010 vector need to be added

Basic experiments in the classroom

Bridging faults: wired-AND or wired-OR?

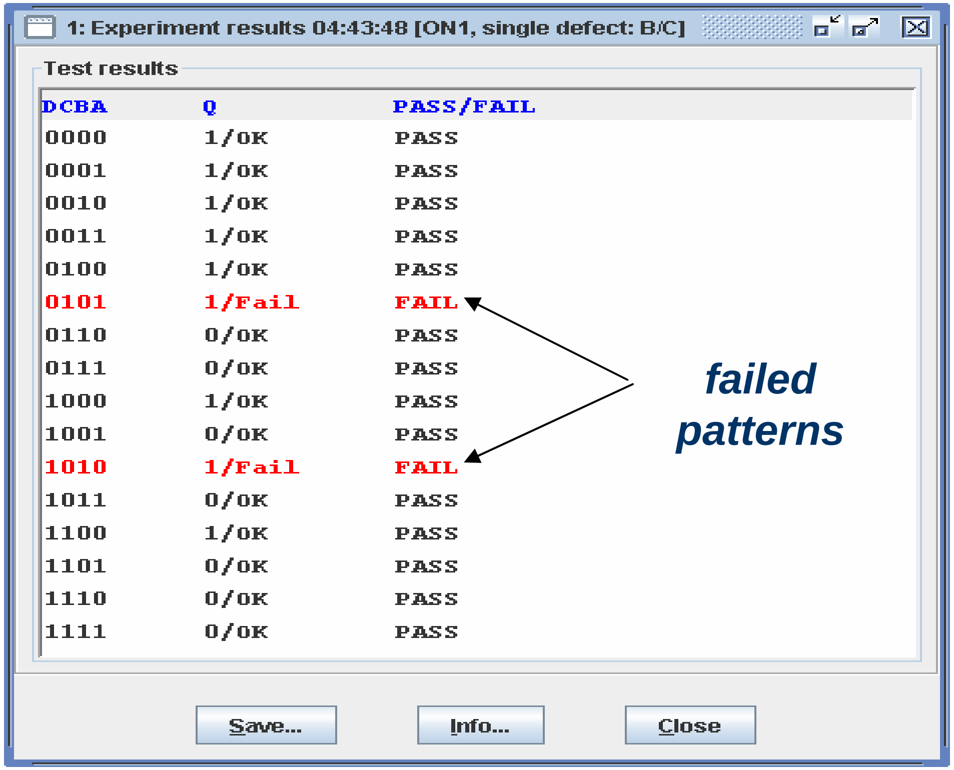


Complex gate ON1, defect: B/C

Test	wired AND	wired OR
	DCBA	*DCBA*
1	1010	1100
2	0101	0100
3		0011
4		0010

Basic experiments in the classroom

Bridging faults: wired-AND or wired-OR?



1: Experiment results 04:43:48 [ON1, single defect: B/C]

Test results

```
DCBA      Q           PASS/FAIL
0000      1/OK        PASS
0001      1/OK        PASS
0010      1/OK        PASS
0011      1/OK        PASS
0100      1/OK        PASS
0101      1/Fail      FAIL
0110      0/OK        PASS
0111      0/OK        PASS
1000      1/OK        PASS
1001      0/OK        PASS
1010      1/Fail      FAIL
1011      0/OK        PASS
1100      1/OK        PASS
1101      0/OK        PASS
1110      0/OK        PASS
1111      0/OK        PASS
```

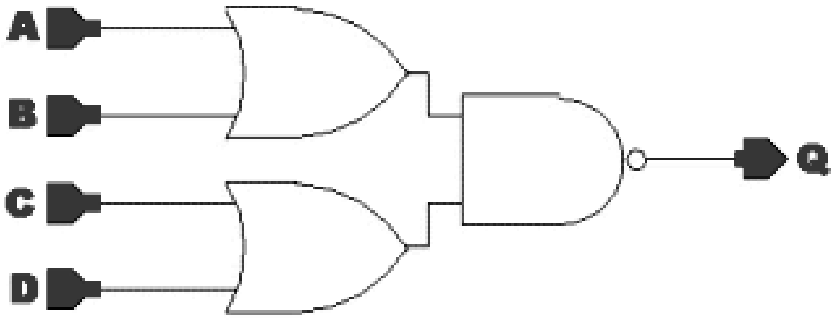
failed patterns

Test	wired AND	wired OR
	DCBA	***DCBA***
1	**1010**	**1100**
2	**0101**	**0100**
3		**0011**
4		**0010**

Wired AND model describes this defect

Basic experiments in the classroom

Bridging faults: further study

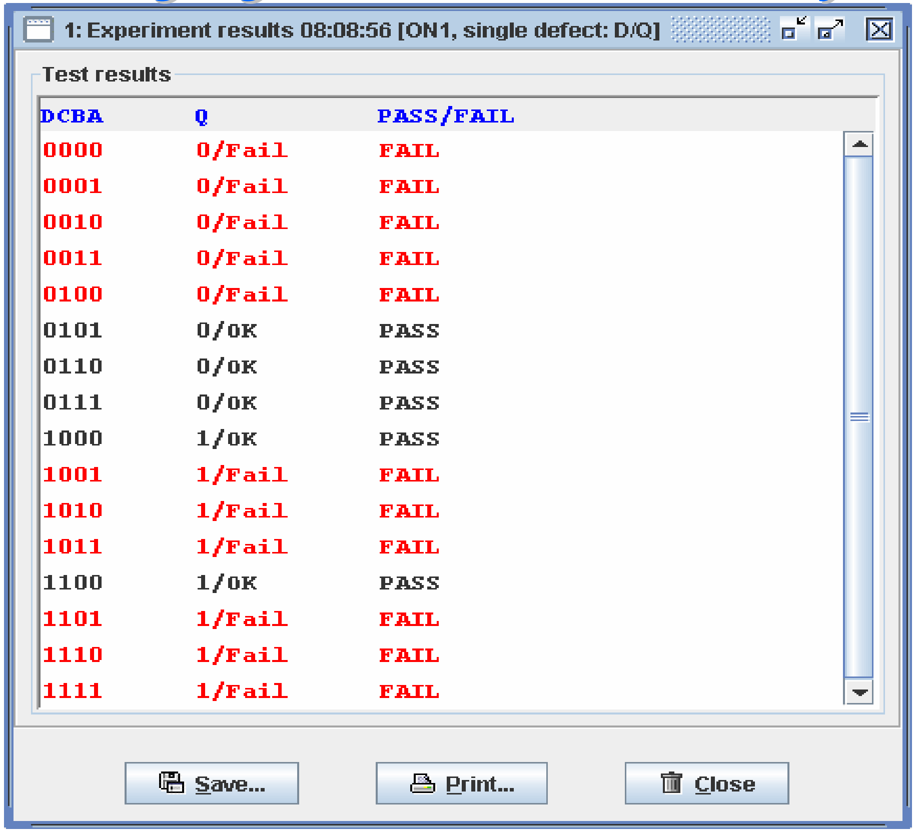


Complex gate ON1, defect: D/Q

Test	wired AND	wired OR
	DCBA	***DCBA***
1	0000	0001
2	0001	0010
3	0010	0011
4	0011	1001
5	0100	1010
6	1001	1011
7	1010	1101
8	1011	1110
9		1111

Basic experiments in the classroom

Bridging faults: further study



1: Experiment results 08:08:56 [ON1, single defect: D/Q]

Test results

```
DCBA         Q              PASS/FAIL
0000         0/Fail         FAIL
0001         0/Fail         FAIL
0010         0/Fail         FAIL
0011         0/Fail         FAIL
0100         0/Fail         FAIL
0101         0/OK           PASS
0110         0/OK           PASS
0111         0/OK           PASS
1000         1/OK           PASS
1001         1/Fail         FAIL
1010         1/Fail         FAIL
1011         1/Fail         FAIL
1100         1/OK           PASS
1101         1/Fail         FAIL
1110         1/Fail         FAIL
1111         1/Fail         FAIL
```

Test	wired AND	wired OR
	DCBA	*DCBA*
1	0000	0001
2	0001	0010
3	0010	0011
4	0011	1001
5	0100	1010
6	1001	1011
7	1010	1101
8	1011	1110
9		1111

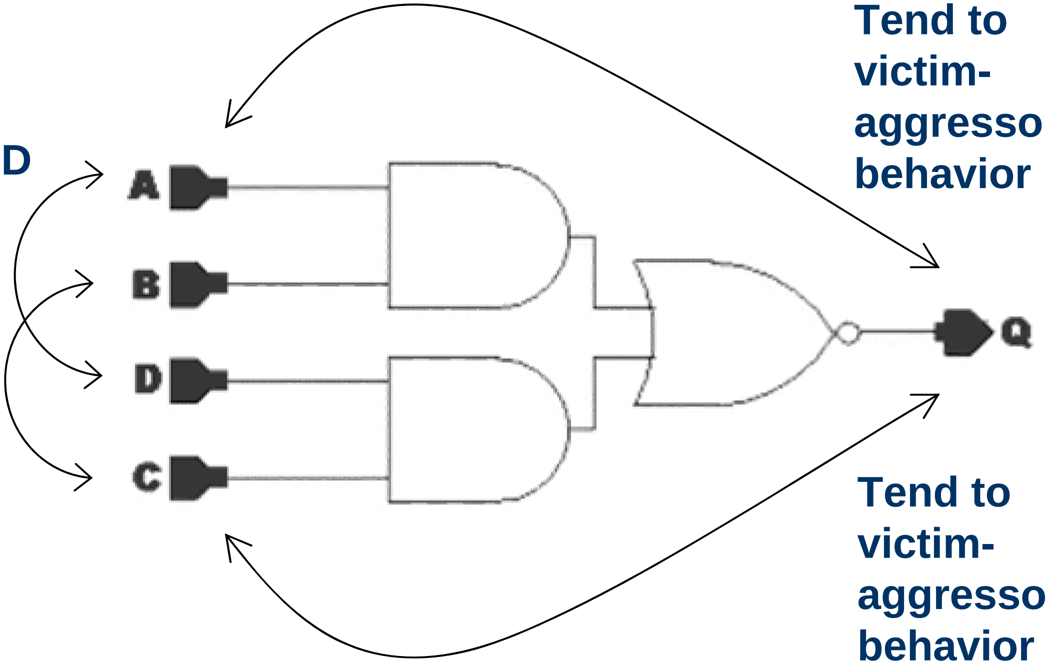
Neither wired AND nor wired OR!?

Victim-aggressor behavior: D is a stronger driver!

Basic experiments in the classroom

Behavior of shorts in DefSim

Tend to wired-AND behavior



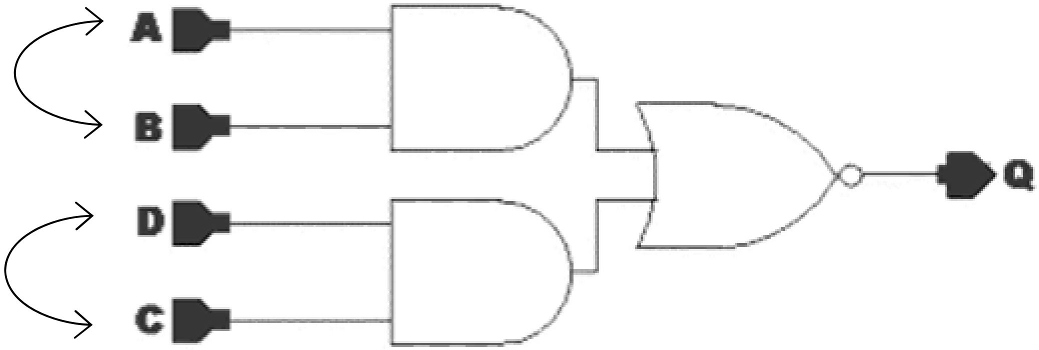
Tend to victim-aggressor behavior

Tend to victim-aggressor behavior

Basic experiments in the classroom

I_{DDQ} versus voltage testing

Impossible to generate test vectors for A/B and D/C

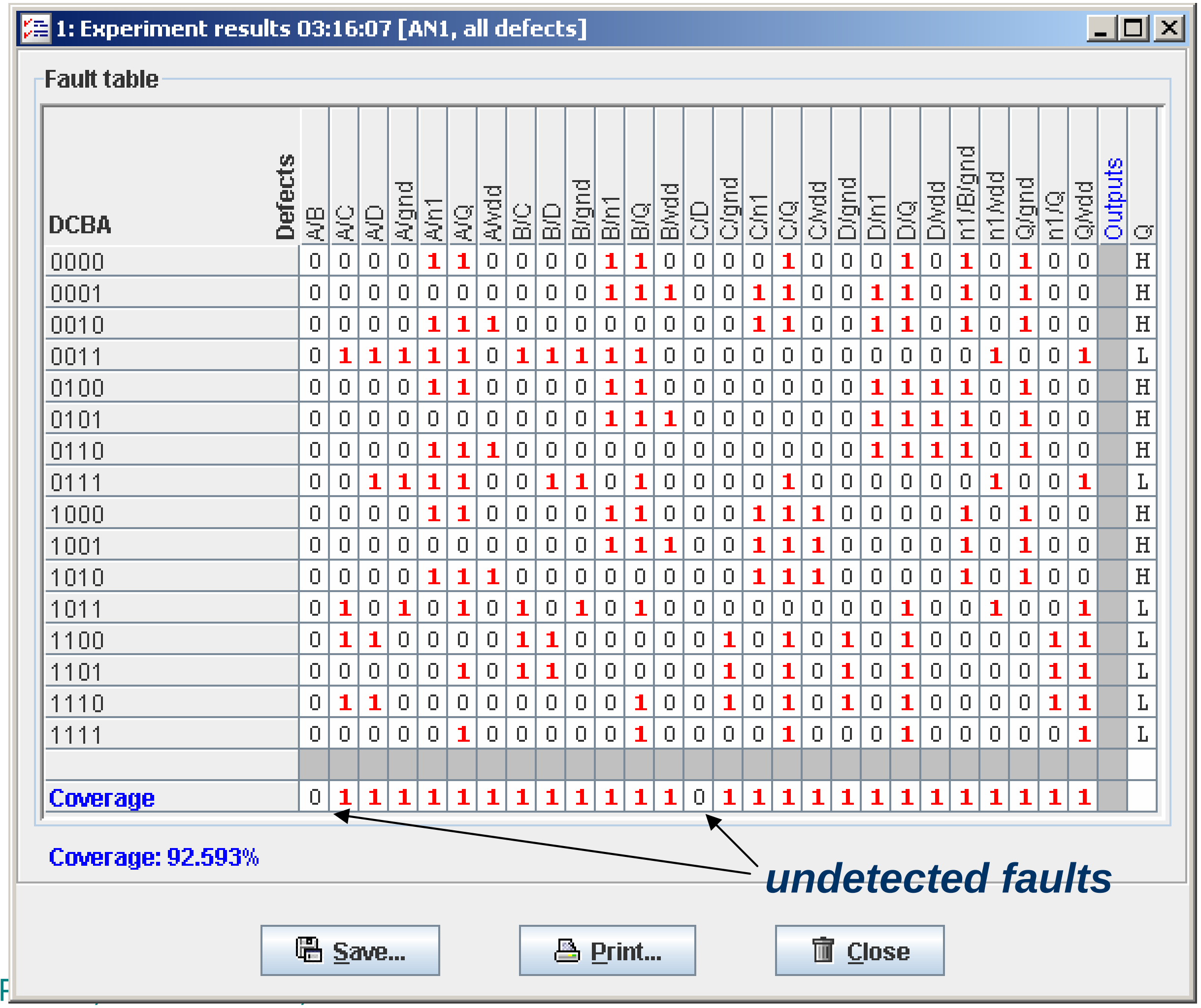


Complex gate AN1

Basic experiments in the classroom

I_{DDQ} versus voltage testing

Impossible to generate test vectors for A/B and D/C



1: Experiment results 03:16:07 [AN1, all defects]

Fault table

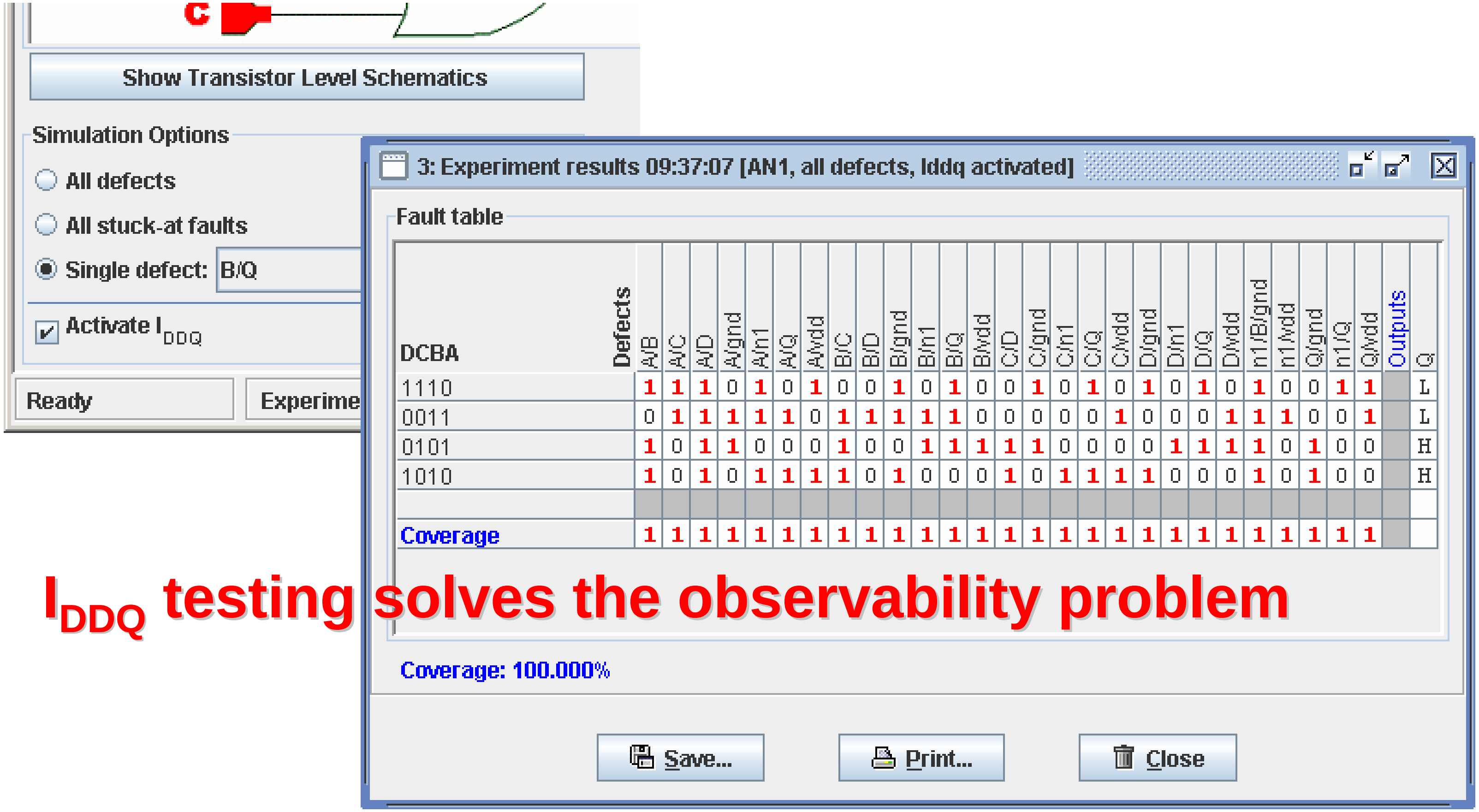
DCBA Defects	A/B	A/C	A/D	A/gnd	A/n1	A/Q	A/vdd	B/C	B/D	B/gnd	B/n1	B/Q	B/vdd	C/D	C/gnd	C/n1	C/Q	C/vdd	D/gnd	D/n1	D/Q	D/vdd	n1/B/gnd	n1/vdd	Q/gnd	n1/Q	Q/vdd	Outputs	Q
0000	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0		H
0001	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	0	0	1	1	0	1	0	1	0	0		H
0010	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	1	0	1	0	0		H
0011	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1		L
0100	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0		H
0101	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1	1	1	0	1	0	0		H
0110	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	0	0		H
0111	0	0	1	1	1	1	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1		L
1000	0	0	0	0	1	1	0	0	0	0	1	1	0	0	0	1	1	1	0	0	0	0	1	0	1	0	0		H
1001	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1	0	0	0	0	1	0	1	0	0		H
1010	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	0	1	0	0		H
1011	0	1	0	1	0	1	0	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1		L
1100	0	1	1	0	0	0	0	1	1	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	1	1		L
1101	0	0	0	0	0	1	0	1	1	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	0	1	1		L
1110	0	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	1	0	1	0	1	0	0	0	0	1	1		L
1111	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1		L
Coverage	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1		

Coverage: 92.593%

undetected faults

Basic experiments in the classroom

I_{DDQ} versus voltage testing



Show Transistor Level Schematics

Simulation Options

○ All defects

○ All stuck-at faults

● Single defect: B/Q

☑ Activate I_{DDQ}

Ready

Experime

3: Experiment results 09:37:07 [AN1, all defects, Iddq activated]

Fault table

DCBA Defects	A/B	A/C	A/D	A/gnd	A/n1	A/Q	A/vdd	B/C	B/D	B/gnd	B/n1	B/Q	B/vdd	C/D	C/gnd	C/n1	C/Q	C/vdd	D/gnd	D/n1	D/Q	D/vdd	n1/B/gnd	n1/vdd	Q/gnd	n1/Q	Q/vdd	Outputs	Q
1110	1	1	1	0	1	0	1	0	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	0	1	1		L
0011	0	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	0	1	0	0	0	1	1	1	0	0	1		L
0101	1	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	0	1	1	1	1	0	1	0	0		H
1010	1	0	1	0	1	1	1	1	0	1	0	0	0	1	0	1	1	1	1	0	0	0	1	0	1	0	0		H
Coverage	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		

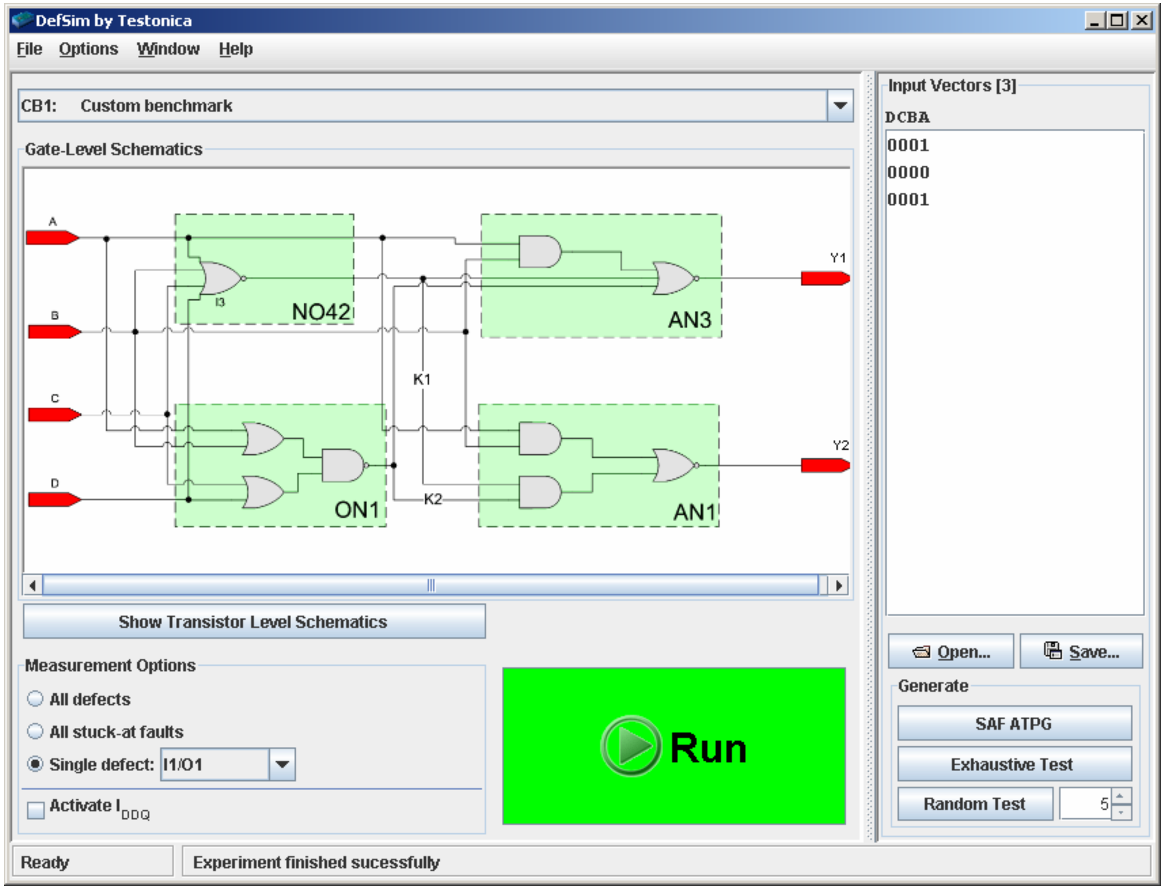
Coverage: 100.000%

I_{DDQ} testing solves the observability problem

Basic experiments in the classroom

Memory effects due to shorts

DCBA
0001
0000
0001



CB1: Custom benchmark

Input Vectors [3]

```
DCBA
0001
0000
0001
```

○ All defects
○ All stuck-at faults
● Single defect: I1/O1
□ Activate I_{DDQ}

Ready

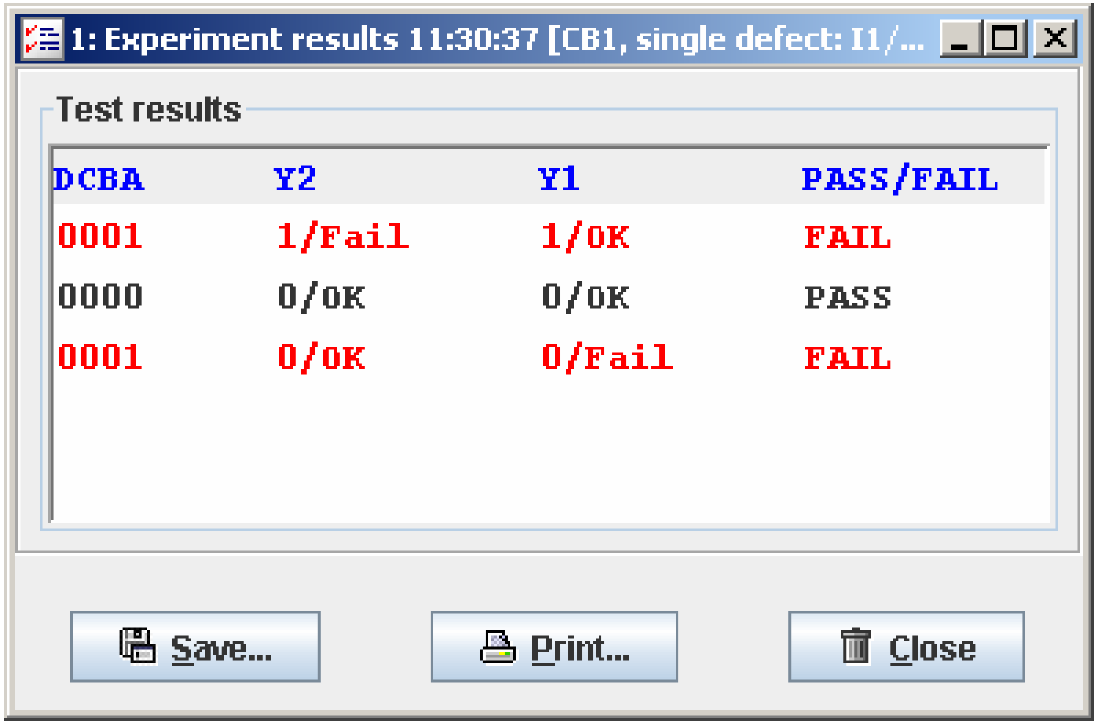
Experiment finished sucessfully

Artur Jutman, Warsaw, Poland, November 16, 2007

Basic experiments in the classroom

Memory effects due to shorts

DCBA
0001
0000
0001



1: Experiment results 11:30:37 [CB1, single defect: I1/...

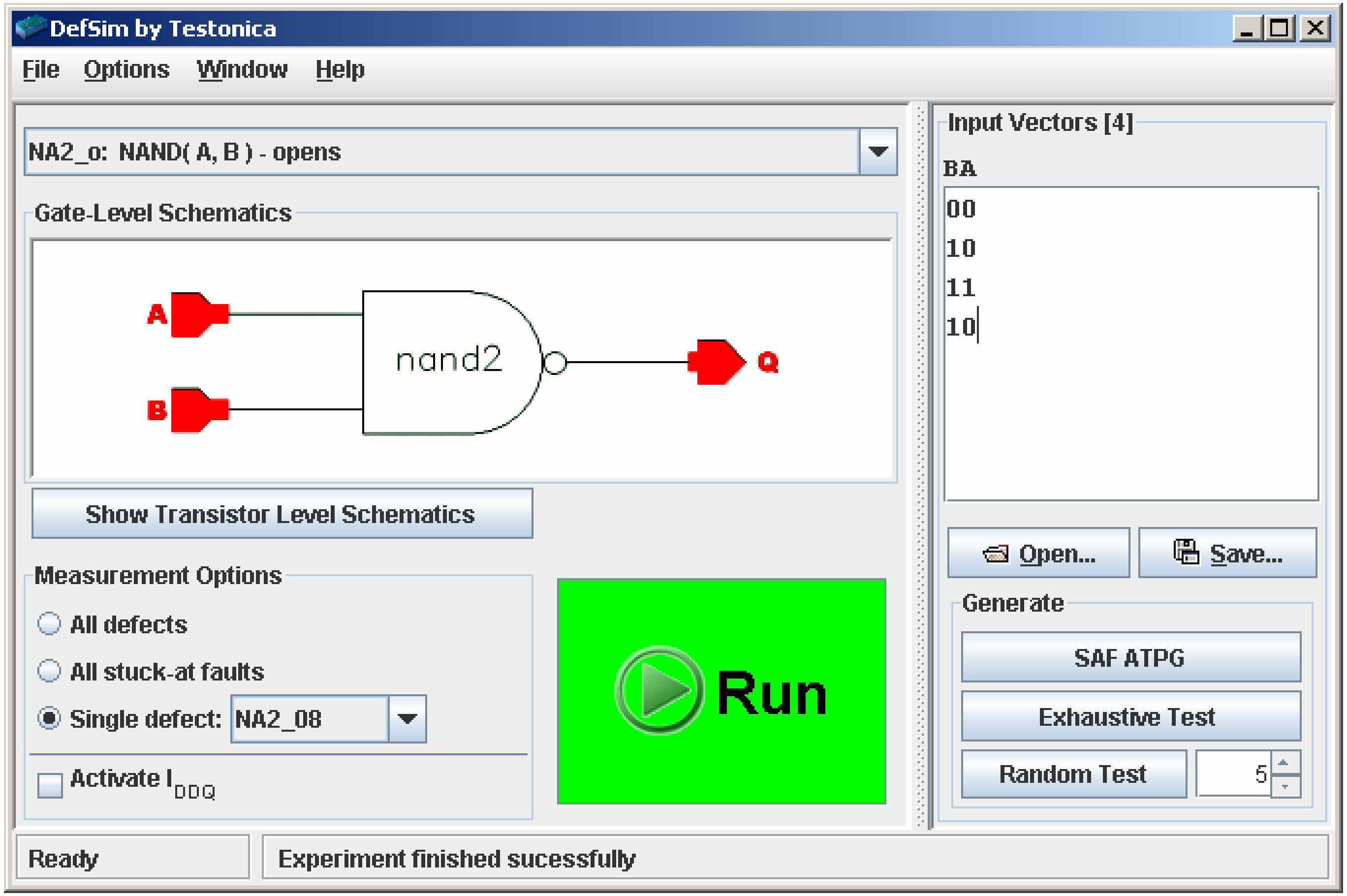
Test results

DCBA	Y2	Y1	PASS/FAIL
0001	1/Fail	1/OK	FAIL
0000	0/OK	0/OK	PASS
0001	0/OK	0/Fail	FAIL

Basic experiments in the classroom

Memory effects due to opens

BA
00
10
11
10



NA2_o: NAND(A, B) - opens

○ All defects
○ All stuck-at faults
● Single defect: NA2_08
□ Activate I_{DDQ}

Input Vectors [4]

```
BA
00
10
11
10
```

Random Test: 5

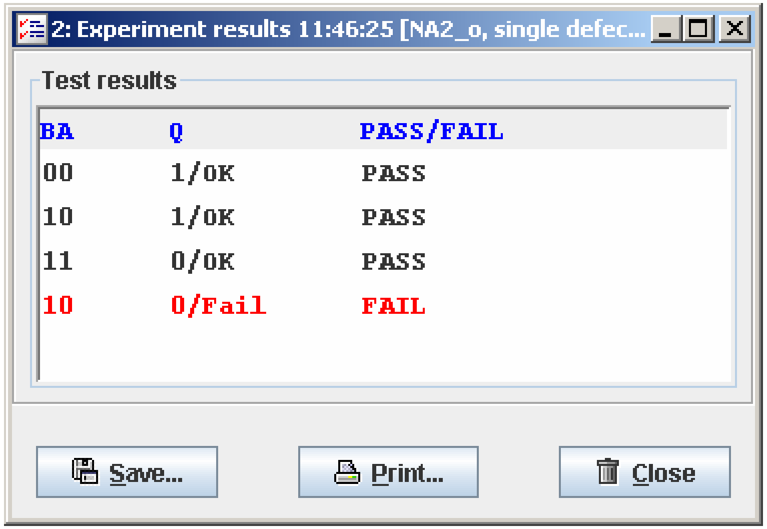
Ready

Experiment finished sucessfully

Artur Jutman, Warsaw, Poland, November 16, 2007

Basic experiments in the classroom

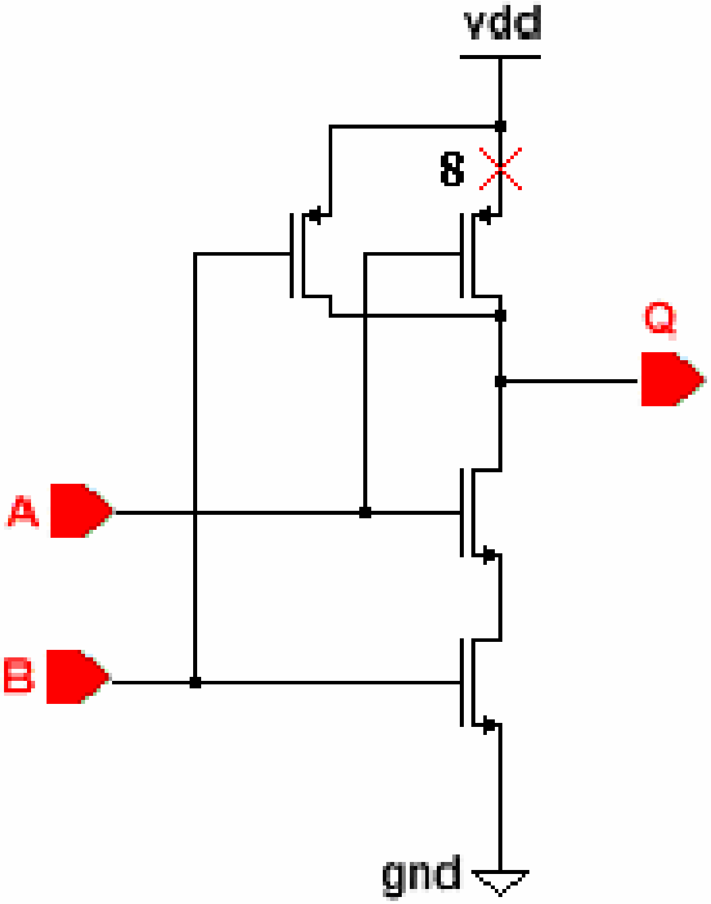
Memory effects due to opens



2: Experiment results 11:46:25 [NA2_o, single defec...

Test results

BA	Q	PASS/FAIL
00	1/OK	PASS
10	1/OK	PASS
11	0/OK	PASS
10	0/Fail	FAIL



Future work

Current version is only the first step; our plans are

- Looking into *other* defect types – increasing diversity
- Implementing *didactically* interesting defects
- Development of new DefSim IC versions for *research* needs
- Updating DefSim IC using upcoming CMOS technologies
- Development of new teaching lab scenarios

Conclusions

- The educational integrated circuit *DefSim* was designed and manufactured.
- This chip is dedicated to development of students' skills in fault simulation and test pattern generation for digital circuits.
- It allows applying both voltage (i.e. logical) and current test methods and offers comparing of their efficiencies on basic digital circuit examples.
- Operation of DefSim was verified experimentally.
- “Plug and Play” DefSim bundle represents a unique and easy to handle educational and research environment.

Thank your for your attention!

Try DefSim at http://www.defsim.com

Read more about DefSim at www.testonica.com

Read more about REASON project at http://reason.imio.pw.edu.pl/