Special Session: „Massive Statistical Process Variation: A Grand Challenge for Testing Nanoelectronic Circuits“

Introduction

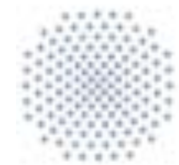
B. Becker, University of Freiburg

S. Hellebrand, University of Paderborn

I. Polian, University of Passau

W. Vermeiren, Fraunhofer IIS-EAS Dresden

H.-J. Wunderlich, University of Stuttgart



Nanoscale Integration

- Potential for integrating highly complex innovative products into single chip (SoC) or package (SiP)
- Parameter variations
  cf. Borkar, IEEE Micro 2005

Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation

AS TECHNOLOGY SCALES, VARIABILITY IN TRANSISTOR PERFORMANCE WILL CONTINUE TO INCREASE, MAKING TRANSISTORS LESS AND LESS RELIABLE. THIS CREATES SEVERAL CHALLENGES IN BUILDING RELIABLE SYSTEMS, FROM THE UNPREDICTIBILITY OF DELAY TO INCREASING LEAKAGE CURRENT. FINDING SOLUTIONS TO THESE CHALLENGES WILL REQUIRE A CONCERTED EFFORT ON THE PART OF ALL THE PLAYERS IN A SYSTEM DESIGN.

Shekhar Borkar
Intel Corp.

VLSI performance has increased by five orders of magnitude in the last three decades, made possible by continued technology scaling. This trend will continue, providing an integration capacity of billions of transistors; however, power, energy, variability, and reliability are barriers to future scaling.

Die size, chip yields, and design productivity have so far limited transistor integration in a VLSI design. Now the focus has shifted to energy consumption, power dissipation, and power delivery.[1] Transistor subthreshold leakage continues to increase, and those of us in this industry have devised leakage avoidance, tolerance, and control techniques for circuits.[2] As technology scales further we will face new challenges, such as variability,[3] single-event upsets (soft errors), and device (transistor performance) degradation—these effects manifesting as inherent unreliability of the components, posing design and test challenges. This article discusses these effects and proposes microarchitecture, circuit, and testing research that focuses on designing with many unreliable components (transistors) to yield reliable system designs.

This problem is not new; we design systems to account for reliability issues. For example, error-correcting codes are commonly used in memories to detect and correct soft errors. Careful designing and testing for frequency binning copes with variability in transistor performance. What is new is that as technology scaling continues, the impact of these issues increases, and we need to devise techniques to effectively deal with them.

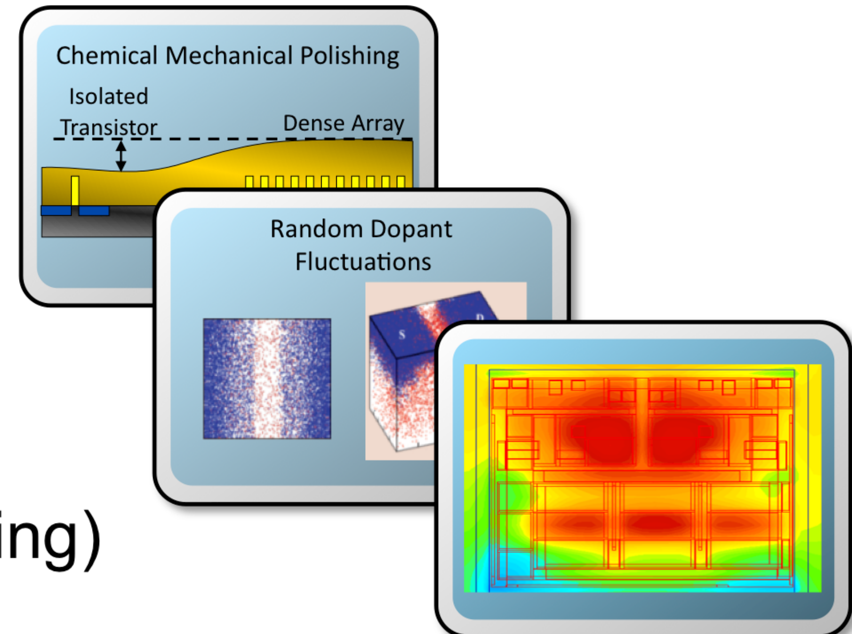
Sources of variations

There are three major sources that cause variations in transistor behavior. The first source is

10 Published by the IEEE Computer Society 0272-1732/06/$20.00 © 2006 IEEE

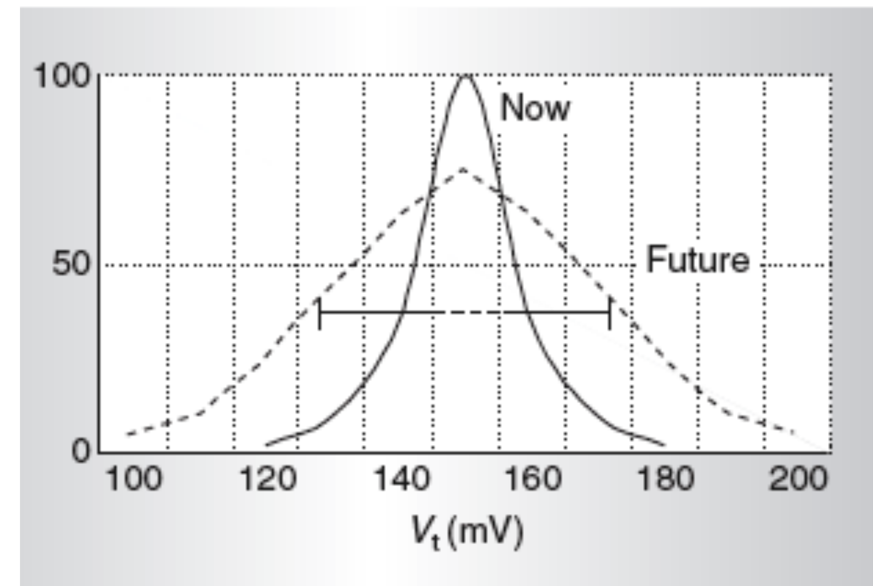
Parameter Variations

- Static variations
  - Systematic
  - Random
- Dynamic variations
- Variations over time (ageing)



Example: Random Dopant Fluctuations

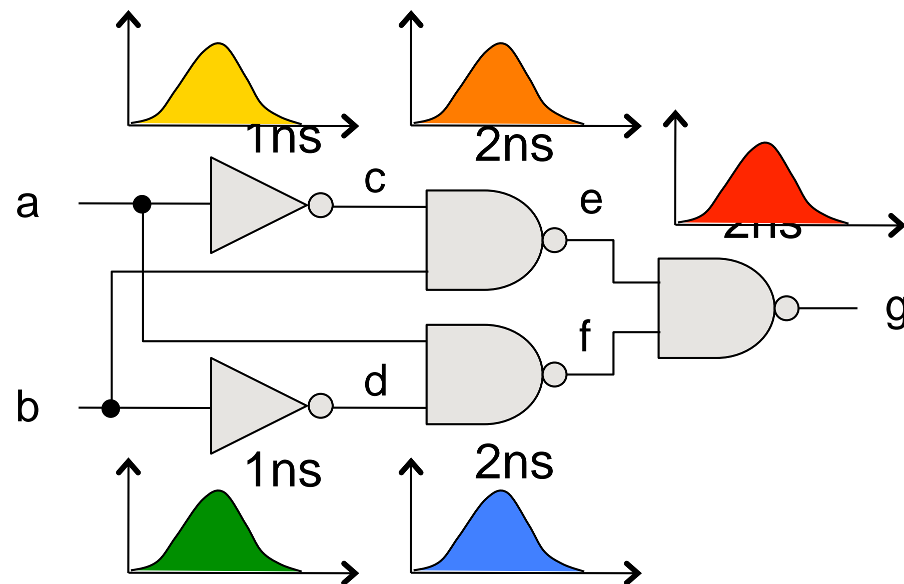
- Threshold voltage V_{th}
  - Determined by the concentration of dopant atoms in the channel
  - Only a few dopant atoms in nano scale transitors
  - Law of large numbers is no longer valid, quantum effects must be considered



[Borkar, IEEE Micro 2005]

Consequences

Most parameter variations result in timing variations

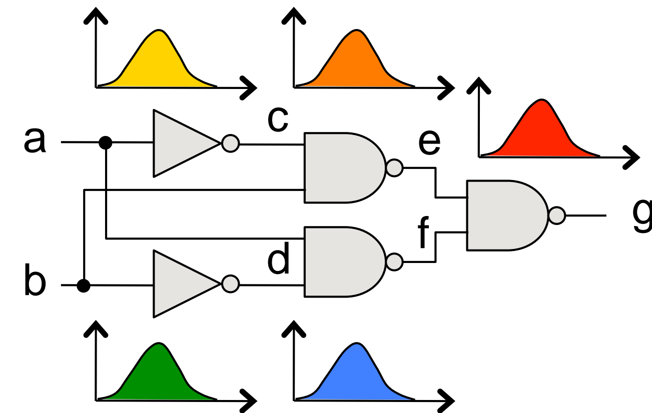


Traditional view: nominal or worst case delay

Now: probability density functions (PDF) for delay

Variation-Aware and Robust Design

- Statistical timing analysis
  - Monte Carlo
  - Path-based
  - Block-based



- Fault-tolerant and self-calibrating architectures
  - Voltage or frequency scaling
  - Body bias
- More and more commercial EDA support

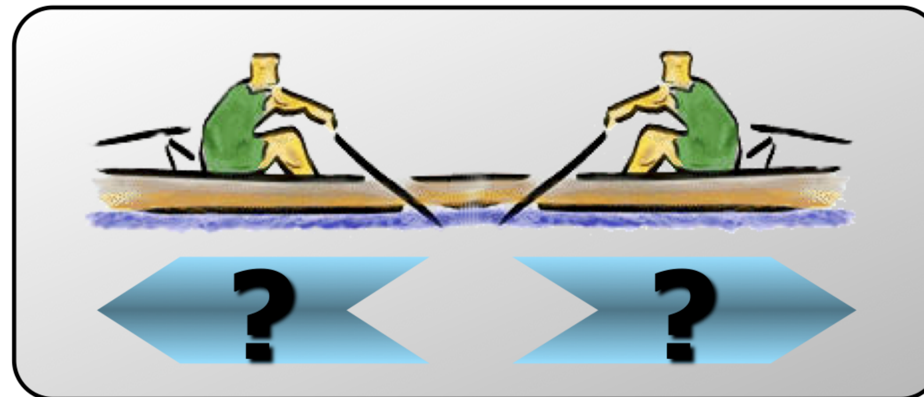
Tester und Designer in the Same Boat?

Designer:

- Minimize the probability of observing a timing fault

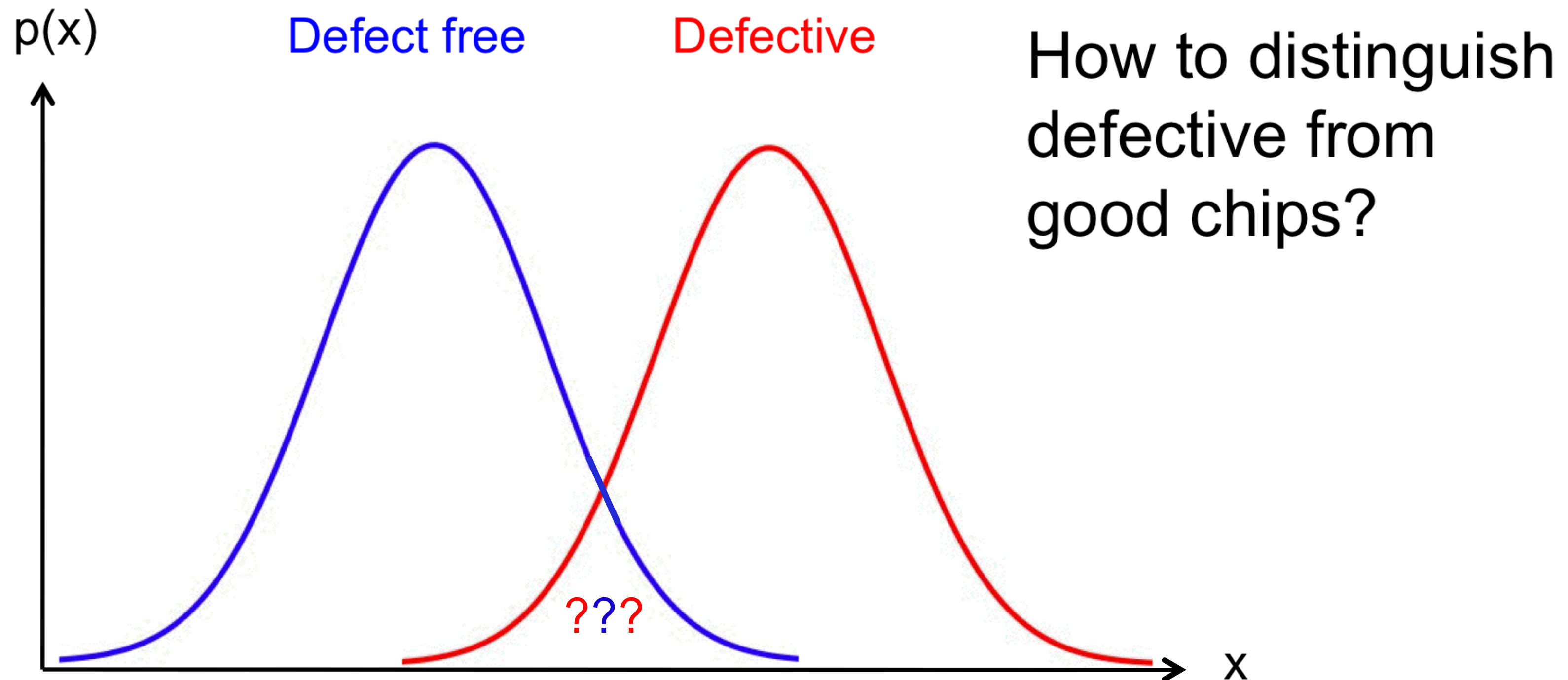
Tester:

- Make sure that any timing fault can be observed



Fundamental paradigm change is necessary

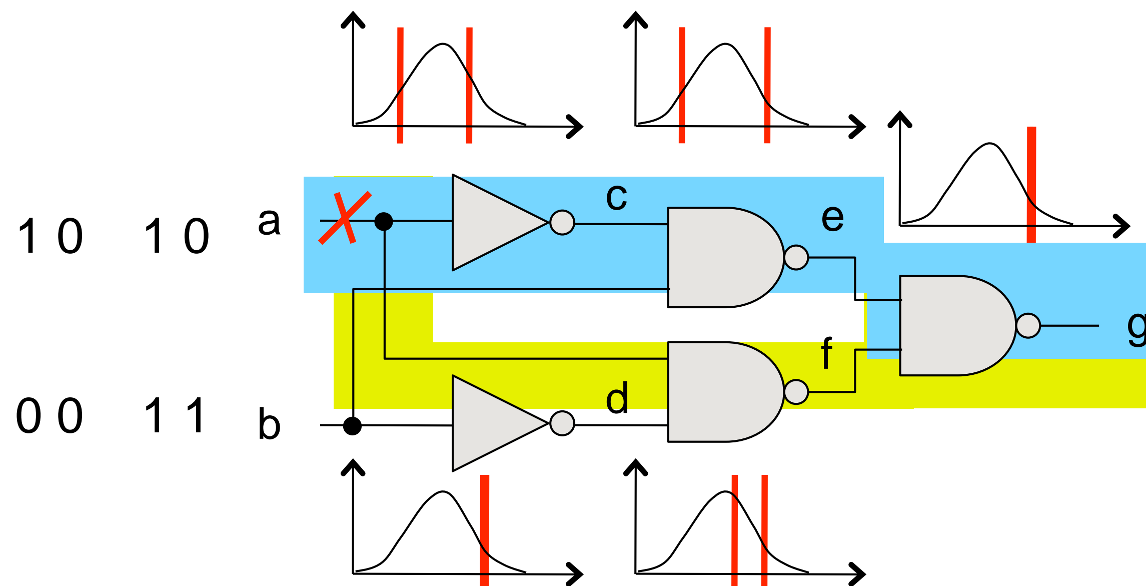
Challenges of Variation-Aware Testing (1)



How to distinguish defective from good chips?

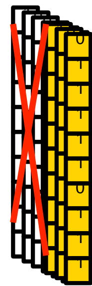
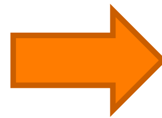
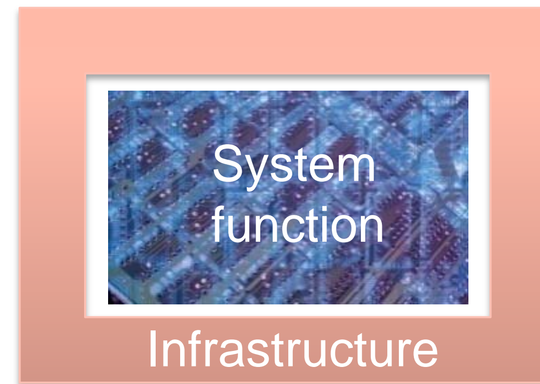
Challenges of Variation-Aware Testing (2)

Test must work for different parameter configurations



Challenges of Variation-Aware Testing (3)

- Larger test sets

- Robust infrastructure tolerates certain defects
  - Test set can be optimized
- How robust is the system during operation?

Special Session Overview

- Introduction
- Variation-Aware Fault Modeling
- Statistical Test Methods
- Automatic Test Pattern Generation (ATPG) in Statistical Testing
- Robustness Analysis and Quality Binning

Special Session: „Massive Statistical Process Variation: A Grand Challenge for Testing Nanoelectronic Circuits“

Variation-Aware Fault Modeling

B. Becker, University of Freiburg

S. Hellebrand, University of Paderborn

I. Polian, University of Passau

W. Vermeiren, Fraunhofer IIS-EAS Dresden

H.-J. Wunderlich, University of Stuttgart

Philosophy: Defect-Based Test meets Variations

- Obtain **accurate low-level models** of defective and defect-free components under process variations.
- Put **massive computational effort** to increase the accuracy of the models.
  - This characterization is run **once for a component** (e.g., a library cell) in a given manufacturing technology.
- Provide **compact representation** of this information to be used in higher-level algorithms and tools.

  → **Histogram data base (HDB).**

Approach

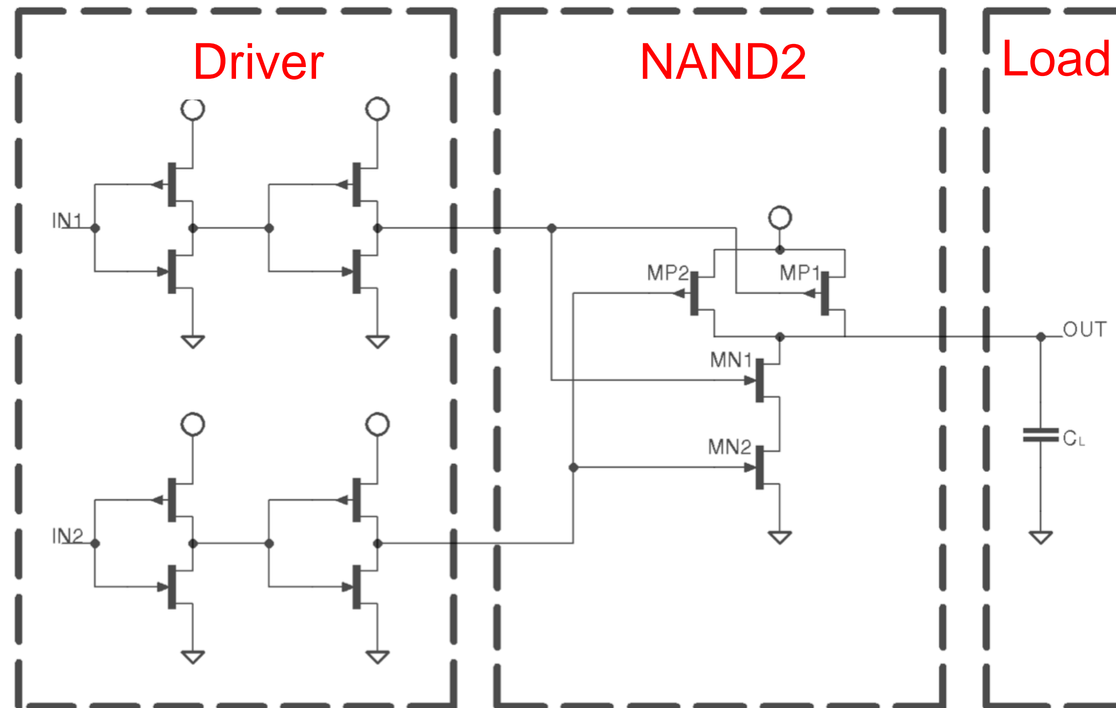
- Primitive-library characterization by **Monte-Carlo electrical simulations.**
  - Tool aFSIM run on a 32-node high-performance cluster.
    - Technology: Nangate 45nm Open Cell Library.
  - Variation of 14 parameters modeled by Gaussian distribution.
    - LINT, VTH0, K1, U0, XJ, TOX, L for n and p transistors.
    - σ and μ set based on industrial input.
  - For each primitive cell, **10,000 sets of parameters** are generated and the delay of the cell is recorded.
  - This is repeated for a number of **defects** in the cell.

Analysis Steps

- Gate embedding.
- Generation of a realistic defect list.
- Input stimuli selection.
- Electrical fault simulation.
- Histogram generation (to be stored in HDB).

- Illustration: NAND2 gate.

Gate Embedding



- Use a **transistor-level representation** of the gate.
- Add realistic **driver** @ inputs, capacitive **load** @ outputs.

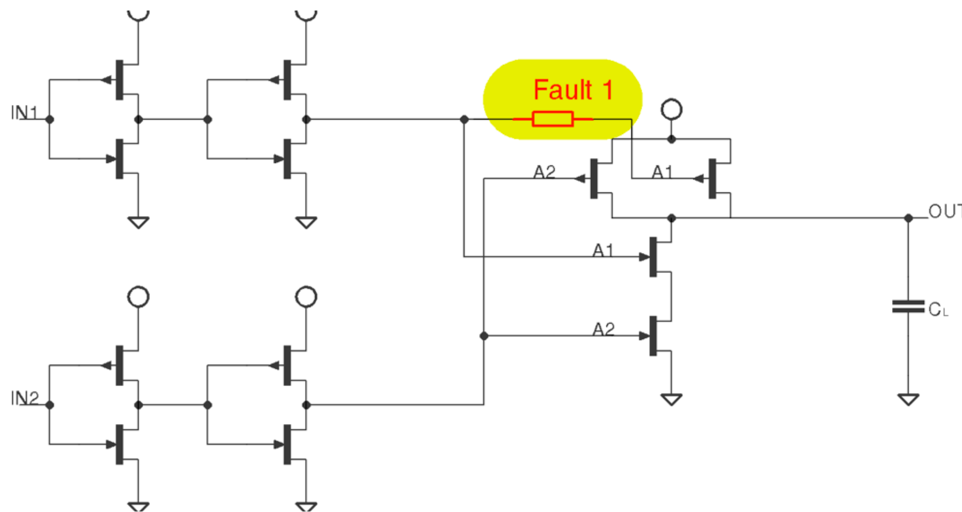
Realistic Defect List Construction

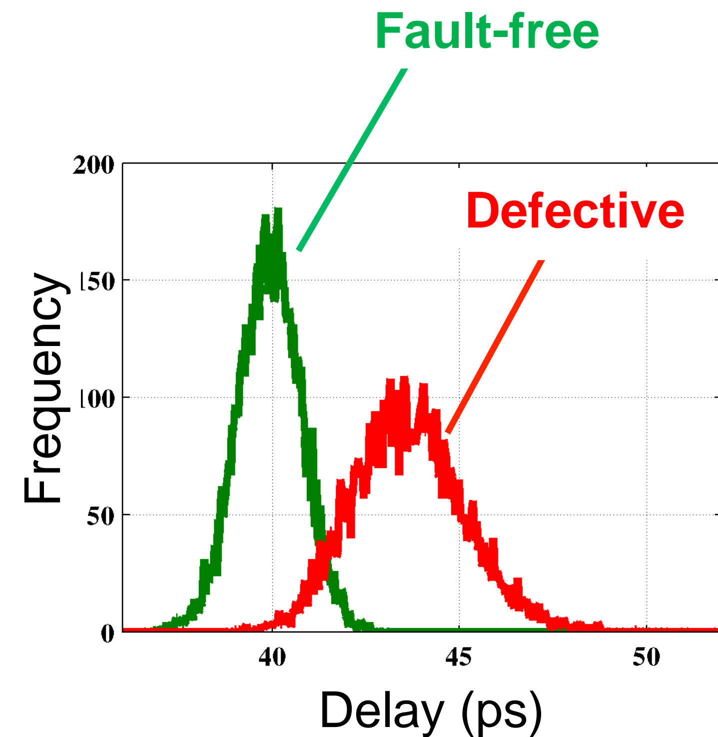
- Realistic resistive opens and shorts.
  - A number of different resistance values.
  - Implemented by fault injection in transistor-level net-list.
- NAND2: 11 opens, 13 shorts, 10 resistance values
  ➔ 240 modeled defects.

Electrical Fault Simulation

- **Automatic distribution** of the simulations by aFSIM.
- 20 ns simulated, input signal change @ 10 ns.
- NAND2 gate: 14,400,000 simulations.
  - 6 test sequences.
  - Computation time ~ **10 days** on a 32-CPU Cluster.
  - Raw data generated: ~ **250 Mbyte.**

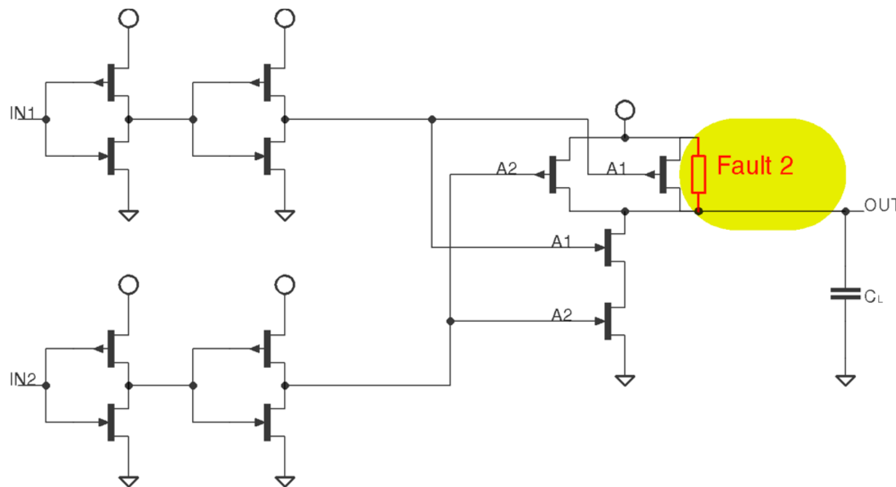
Example: Fault 1 in NAND2

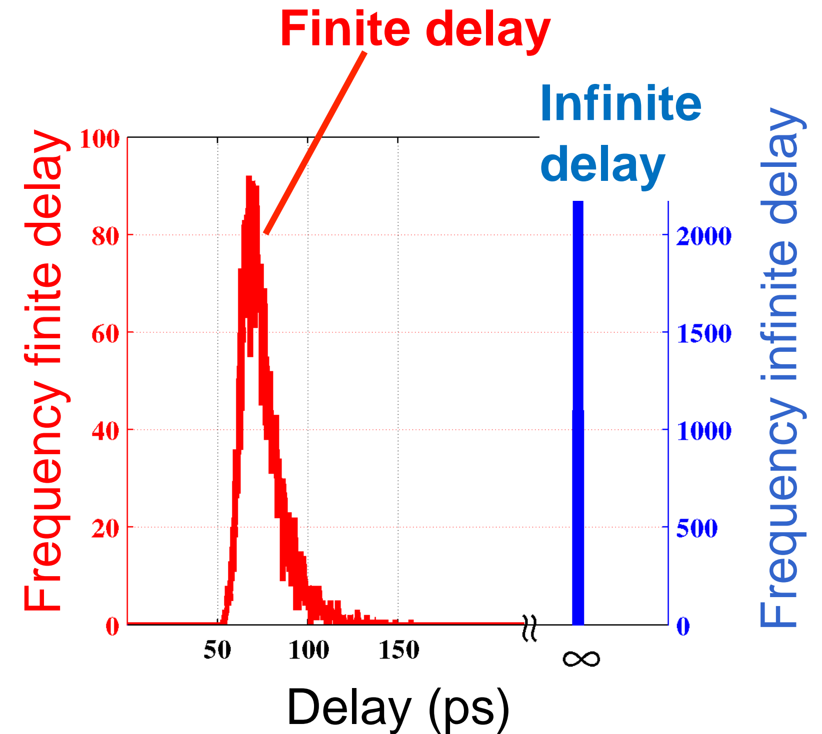




- 500-kΩ resistive open at the gate of pMOSFET MP1.
- Delay histograms of the fault-free and defective cell.

Example: Fault 2 in NAND2





- 7,5-kΩ drain-source resistive short at MP1.
- Finite and infinite extra delay observed.

Histogram Data Base (HDB)

- Provides low-level data to statistical test methods.
- Contains histograms indexed by
  - the primitive cell,
  - the defect,
  - the input sequence.
- Further information is **abstracted away.**
- Resolves **intellectual-property issues.**
  - Customer requires only the HDB and no proprietary manufacturing technology parameters.

Special Session Overview

- Introduction
- Variation-Aware Fault Modeling
- Statistical Test Methods
- Automatic Test Pattern Generation (ATPG) in Statistical Testing
- Robustness Analysis and Quality Binning

Special Session: „Massive Statistical Process Variation: A Grand Challenge for Testing Nanoelectronic Circuits“

Statistical Test Methods

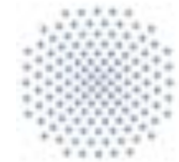
B. Becker, University of Freiburg

S. Hellebrand, University of Paderborn

I. Polian, University of Passau

W. Vermeiren, Fraunhofer IIS-EAS Dresden

H.-J. Wunderlich, University of Stuttgart

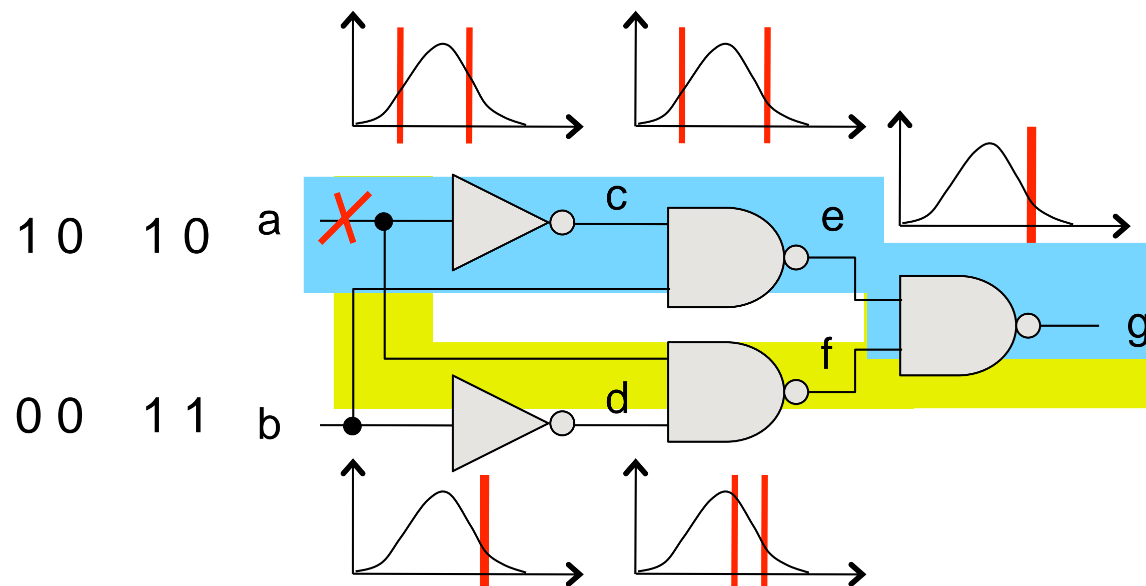


Outline

- **Variation-aware fault simulation**
- The theory
- The practice

Back to the Introductory Example

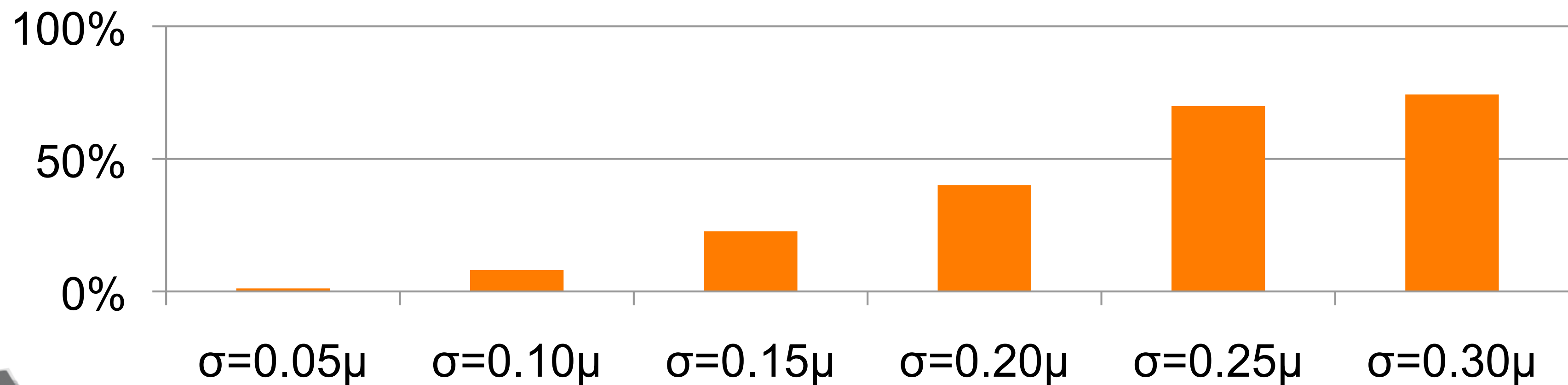
Test must work for different parameter configurations



Robust test not possible

Are Variations a Real Test Problem?

- Results of Monte Carlo Simulation (c880)
  - Gate delays have normal distribution $N(\mu,\sigma^2)$
  - Single fault of fixed size
  - Apply best single test pattern pair for each fault location
- Percentage of faults where detection is unreliable:



Outline

- Variation-aware fault simulation
- **The theory**
- The practice

Evaluating Fault Coverage (1)

- The standard concept describes the portion of faults detected by a test set:

D	delay size
$f(D)$	density function of the delay size
$FC(D)$	fault coverage of delay fault of size D
$FC = \int_{0 \le D} FC(D) \cdot f(D) dD$	Fault Coverage

Evaluating Fault Coverage (2)

- Fault coverage under variations:

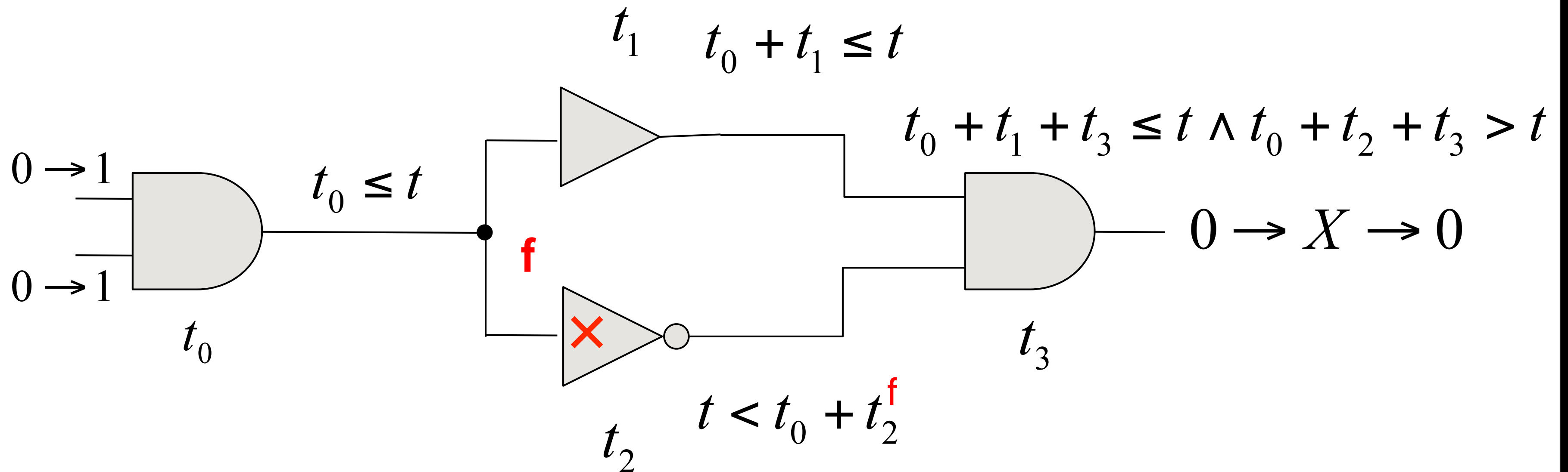
$FC_{(p_1,\ldots,p_n)}(D)$	Fault coverage of delay faults of size D in a circuit with parameters $p_1,\ldots,p_n$
$f(p_1,\ldots,p_n)$	density function of parameters

- Circuit coverage:

$$FC(D) = \int_{(p_1,\ldots,p_n)} FC_{(p_1,\ldots,p_n)}(D) f(p_1,\ldots,p_n)\, dp_1 \ldots dp_n$$

Circuit coverage vs. Fault coverage

Propagating Conditions

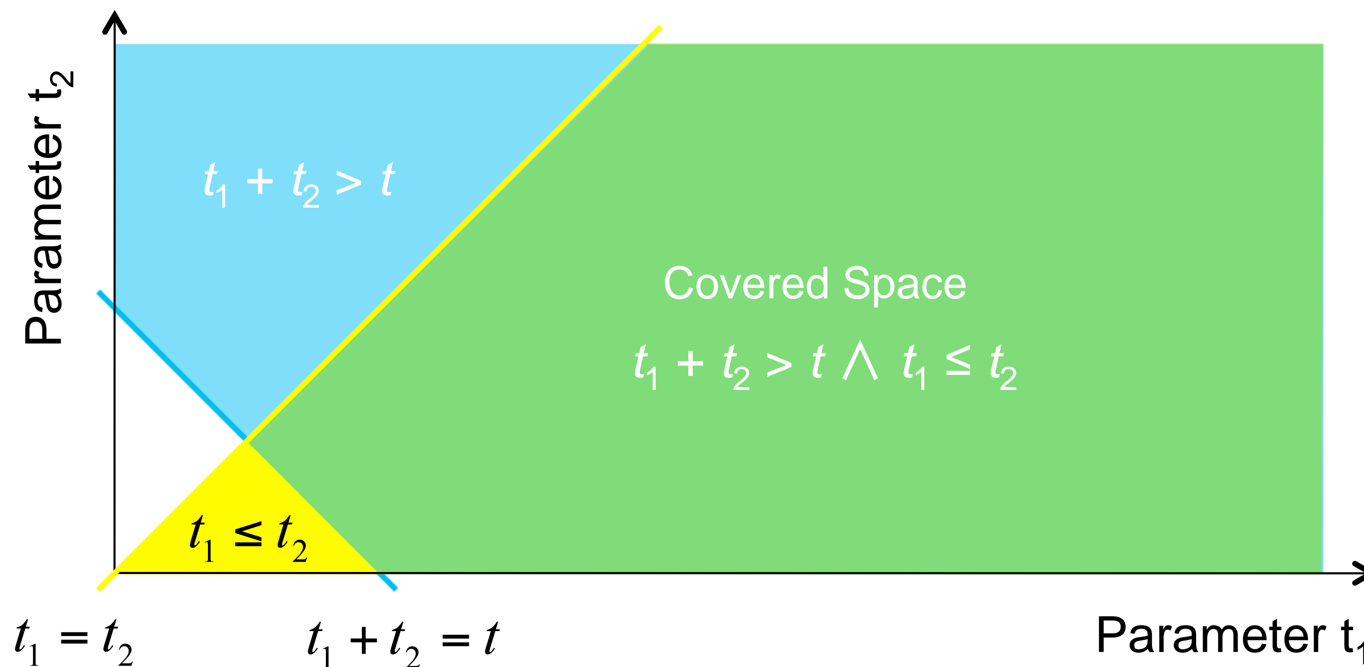


- Gate delays are symbols $t_0, \ldots, t_n$
- Condition for logic "1"
- Common variables in conditions at gate inputs indicate reconvergency

Covered Parameter Space

- Computed condition must evaluate to erroneous logic value of output:

e.g. $t_1 + t_2 > t \wedge t_1 \leq t_2 = \text{true}$



Evaluating Conditions

- Given gate delays $t_1, \ldots, t_n$ and a conjunction of inequalities
- Replace sums in inequalities with random variables $X_1, \ldots, X_k$ of normal distribution (path delays)
- Compute correlation matrix R and mean μ of $X_1, \ldots, X_k$
- Probability that condition is true

$$P_k(\mu, R) = \int_{-\infty}^{t} \ldots \int_{t}^{+\infty} \phi_k\left(x; \mu, R\right) dx_1 \ldots dx_k$$ (Solve numerically)

ϕ_k : density function of *k*-dimensional normal distribution

Evaluating conditions (example)

$$t_0 + t_3 + t_1 \le t \wedge t_0 + t_3 + t_2 > t$$

X_1 X_2

Mean vector μ Correlation Matrix R (Reconvergence!)

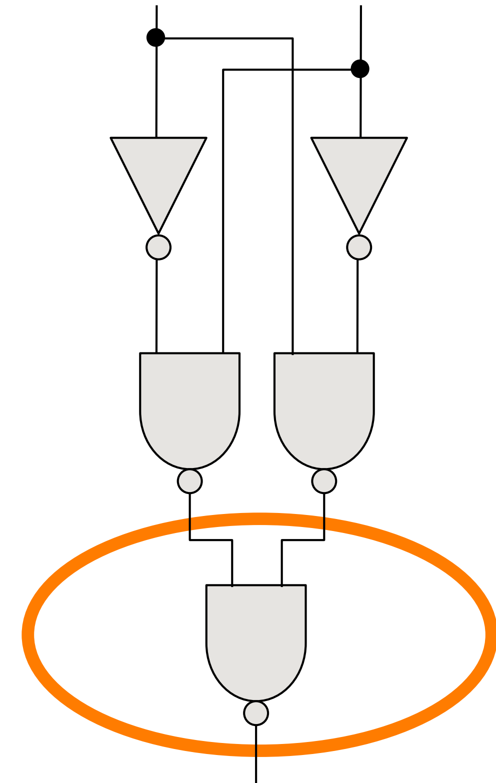
$$P_2(\mu, R) = \int_{-\infty}^{t} \int_{t}^{+\infty} \phi_2\left(x; \mu, R\right) dx_1 dx_2$$

↓

Probability that condition is true for parameter space

Reconvergencies

- Reconvergencies impact computing twofold:
  - Correlation
  - Complexity
- Statistical dependencies maintained in gate delay symbols and handled by correlation matrix.
- Number of paths increases exponentially with number of reconvergencies.

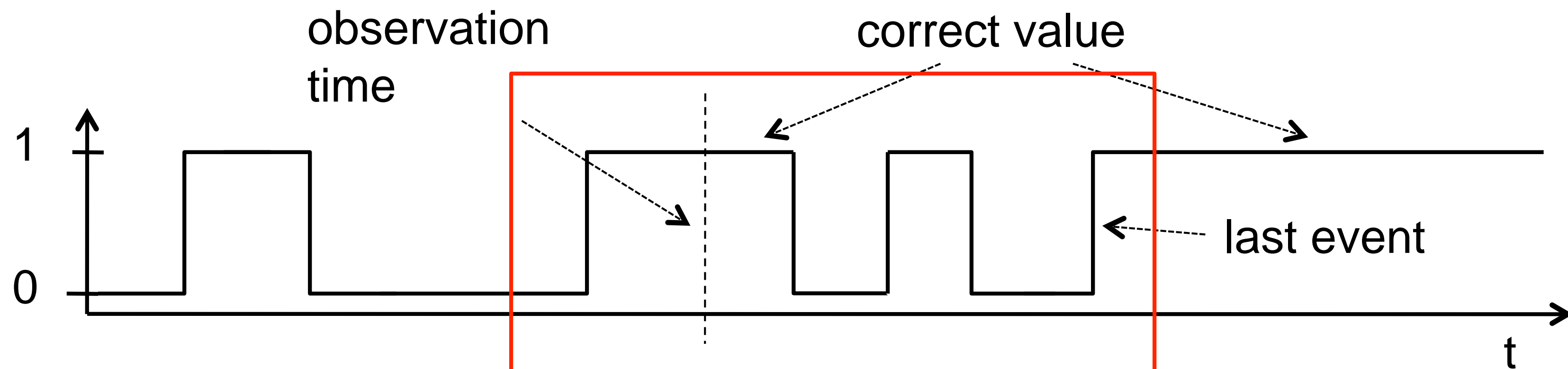


Approximation

- Introduce minimal and maximal gate delays
  - One standard is the 3 σ rule
- At each gate:
  - If the minimum arrival time + the shortest path to an output is *later* than the observation time: neglect path.
  - If the maximum arrival time + the longest path to an output is *earlier* the the obervation time: neglect path.

Fault Detection under Variations

- Latest arrival time in presence of a fault
  - determines, if the fault causes an error
  - does not determine fault detection



- For efficiency, compute only relevant part of the waveform

Outline

- Variation-aware fault simulation
- The theory
- **The practice**

Statistics is Best Practice of Test

- N-Detect
  - Test one fault by at least N patterns
  - Increase probability that patterns are appropriate for circuit under test
- Adaptive testing
  - Observe test outcomes to identify the corner of the die, wafer or lot
  - Adapt patterns to the identified corner
- Iterative pattern generation

Iterative pattern generation

Circuit Parameter B

Circuit Parameter A

1	0	0	0
1	1	0	1
0	1	1	1
1	1	1	1
1	1	1	1
1	1	1	0
0	0	0	0
1	1	1	1
1	0	1	0
1	1	1	1

Single test pattern pair

Integration with Test Generation

Configurable ATPG Pattern C → Statistical Fault Simulation → (diagram)

Pattern A
Pattern B
Pattern C

Special Session Overview

- Introduction
- Variation-Aware Fault Modeling
- Statistical Test Methods
- Automatic Test Pattern Generation (ATPG) in Statistical Testing
- Robustness Analysis and Quality Binning

Special Session: „Massive Statistical Process Variation: A Grand Challenge for Testing Nanoelectronic Circuits“

ATPG in Statistical Testing

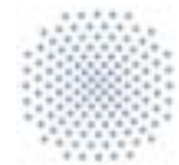
B. Becker, University of Freiburg

S. Hellebrand, University of Paderborn

I. Polian, University of Passau

W. Vermeiren, Fraunhofer IIS-EAS Dresden

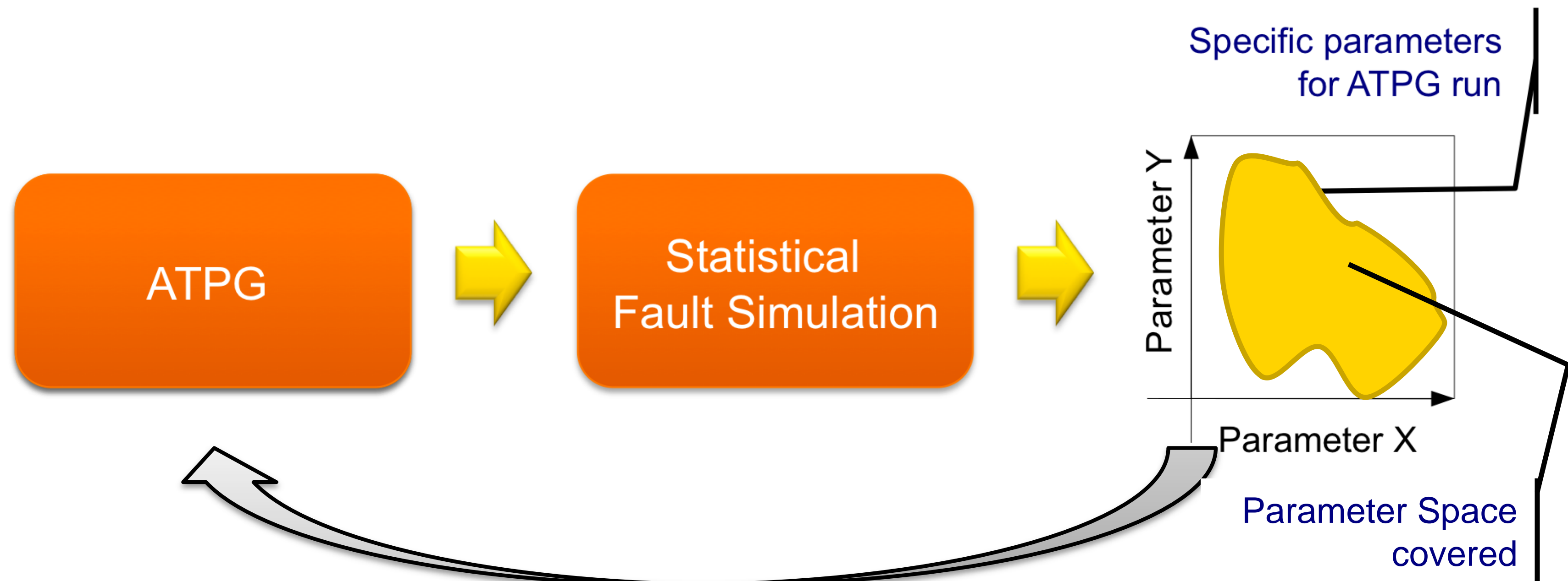
H.-J. Wunderlich, University of Stuttgart



Goals

- Repeated computation of delay tests for specific points in the parameter space
- Identification of vulnerable circuit components
- Combination with robust design using information redundancy

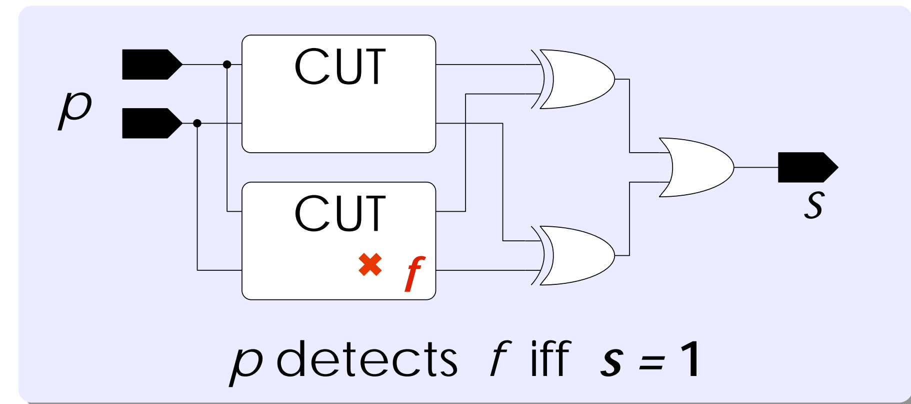
ATPG to cover the parameter space



- Requirement
  - Test patterns satisfying multiple constraints, e.g. control and sensitization of specific (multiple) paths

SAT-based ATPG

- Three basic steps
  - Construct miter
  - Express as boolean satisfiability problem (SAT)
  - Solve SAT-instance



- SAT-based ATPG outperforms structural ATPG for hard instances, in particular, on redundant faults

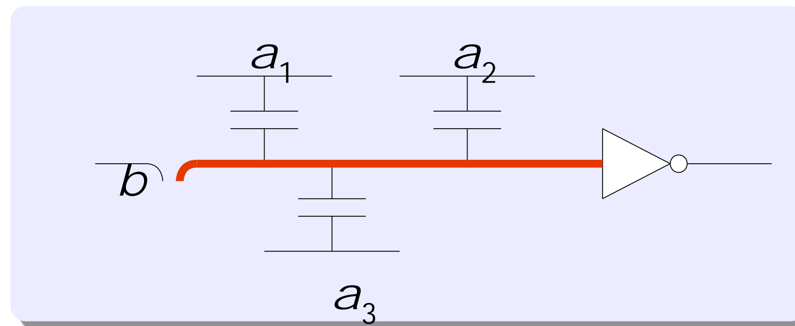
TIGUAN

Thread-parallel Integrated test pattern Generator Utilising satisfiability Analysis
[Czutro et al., in Int. Jour. Parallel Programming, 2010]

- SAT-based ATPG employing multi-threading
- Classified stuck-at faults on very large industrial designs
- Supports “Conditional Multiple Stuck-At” fault model (**CMS@**)

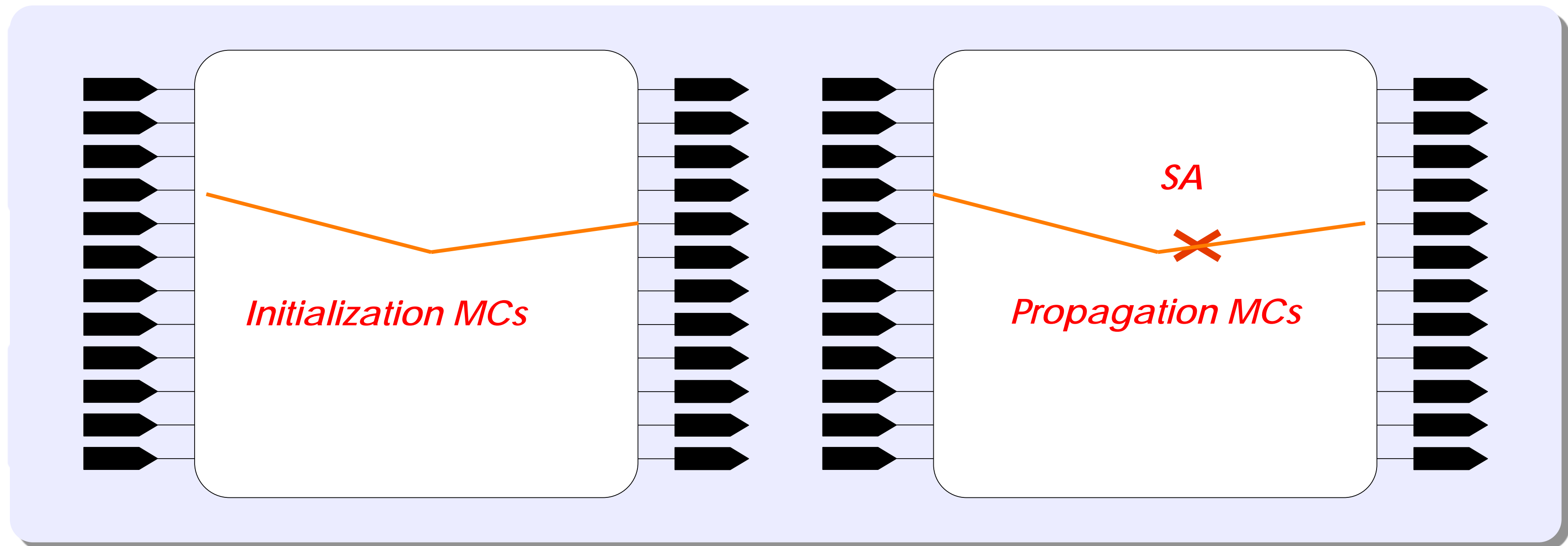
Conditional Multiple Stuck-At (**CMS@**)

- m aggressors (m ≥ 0), n victims(n ≥ 1)
  - if all aggressors satisfy a condition, all victims are s-a-0 or s-a-1
  - example (open defect): if [*a1 = 0 & a2 = 1 & a3 = 0] b s-a-0*



- ATPG for complex fault models (resistive opens, bridges, …)

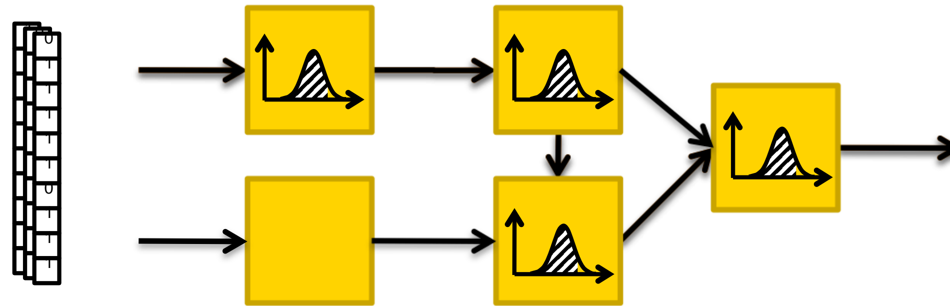
TIGUAN with multiple time frames



- CMS@ extended to multiple time frames to support
  - Delay faults
  - Sensitization of specific paths by multiple constraints (MCs): Initialization MCs, Propagation MCs

Identification of Vulnerable Components (1)

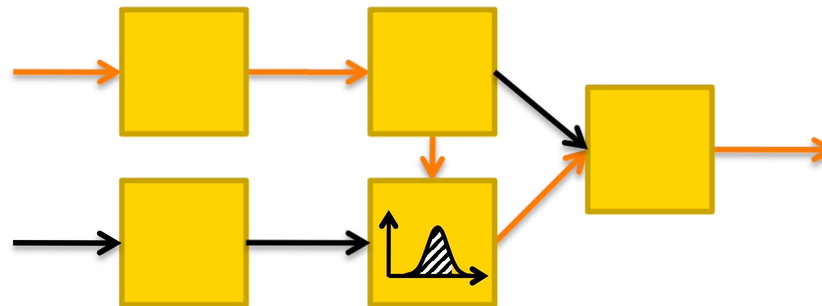
- **Relevance measures:** estimate the probability that a fault in a component will be visible at the outputs



- Consider paths through the component
  - Static path relevance: prob. of sensitization by random inputs (indep. of path length)
  - Dynamic path relevance: prob. of sensitization through „sufficiently slow“ path

Identification of Vulnerable Components (2)

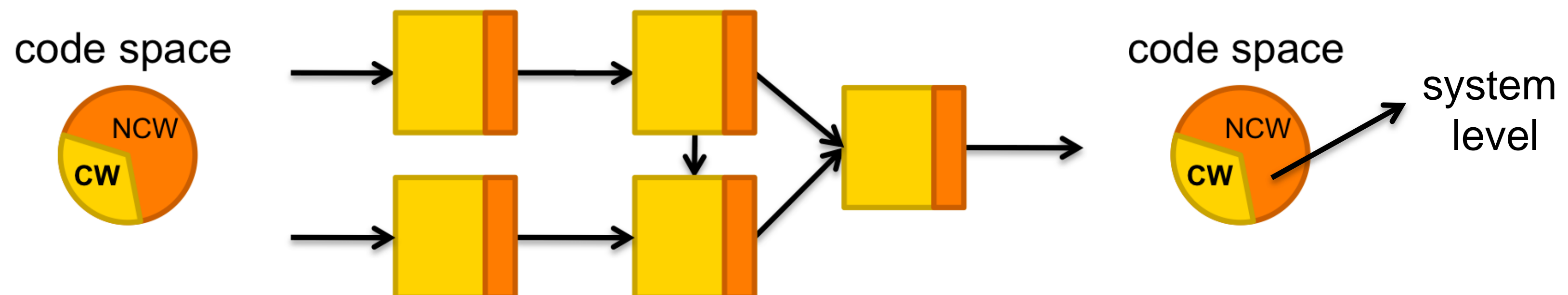
- **Relevance measures**



- Use TIGUAN to model static and dynamic path relevance
- #SAT to compute/approximate relevance measure
- **Validation** by statistical fault simulation

Refined Analysis for Robust Systems

- System with information redundancy



  - Extension of SAT-ATPG for multiple constraint delay faults and vulnerability analysis
  - Code space is taken into account
    - Only code words (CW) as inputs
    - Output: infra structure handles non code words (NCW), faulty CW lead to critical faults

Special Session Overview

- Introduction
- Variation-Aware Fault Modeling
- Statistical Test Methods
- Automatic Test Pattern Generation (ATPG) in Statistical Testing
- ➡ Robustness Analysis and Quality Binning

Special Session: „Massive Statistical Process Variation: A Grand Challenge for Testing Nanoelectronic Circuits“

Robustness Analysis and Quality Binning

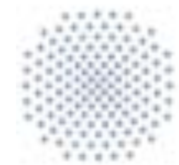
B. Becker, University of Freiburg

S. Hellebrand, University of Paderborn

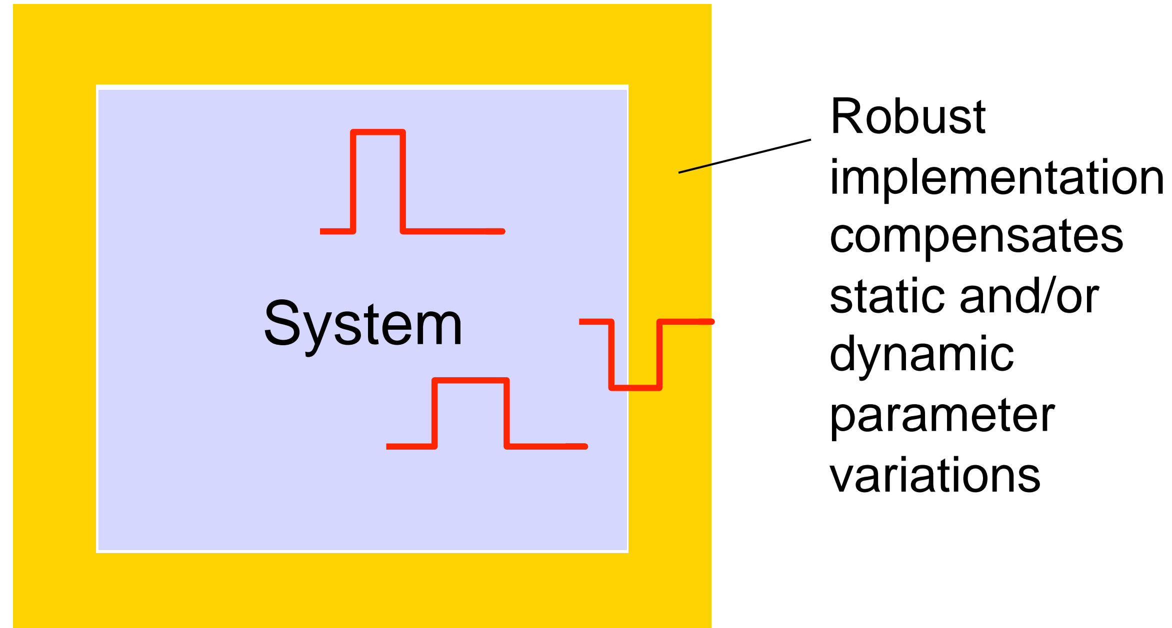
I. Polian, University of Passau

W. Vermeiren, Fraunhofer IIS-EAS Dresden

H.-J. Wunderlich, University of Stuttgart



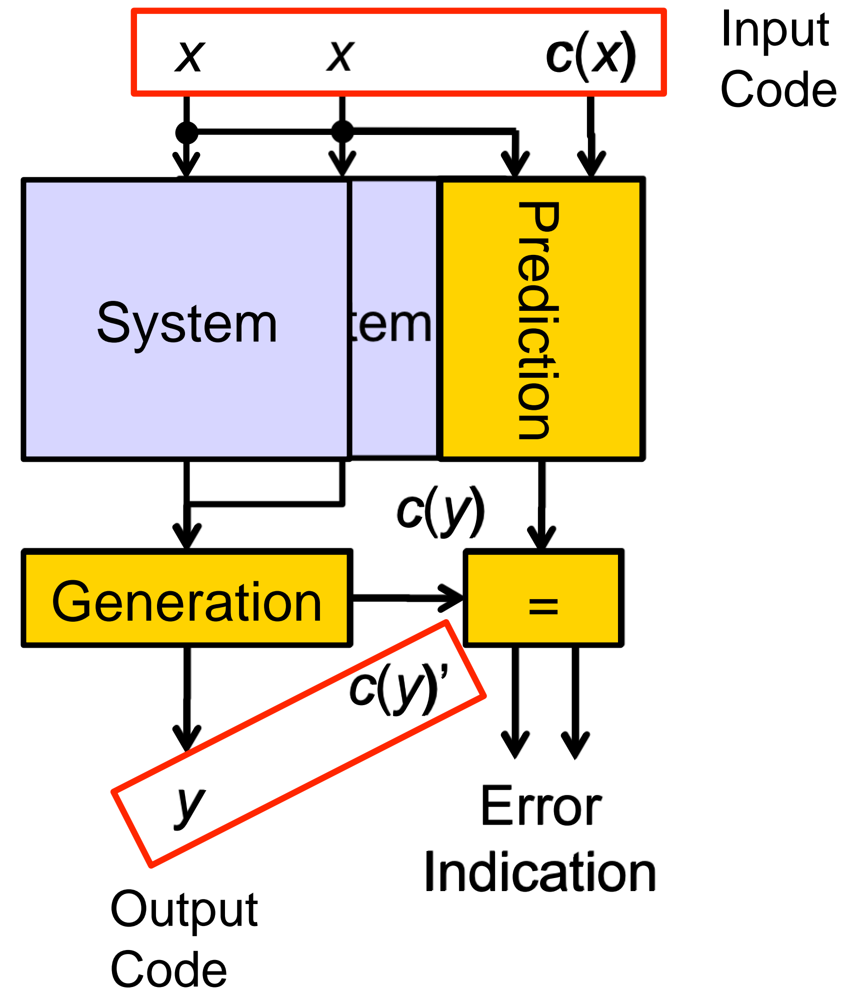
Robust Systems



- Classical fault tolerant architectures (Self-checking circuits, TMR, …)
- New self-calibrating, self-adaptive solutions

Example 1: Self-Checking Circuits

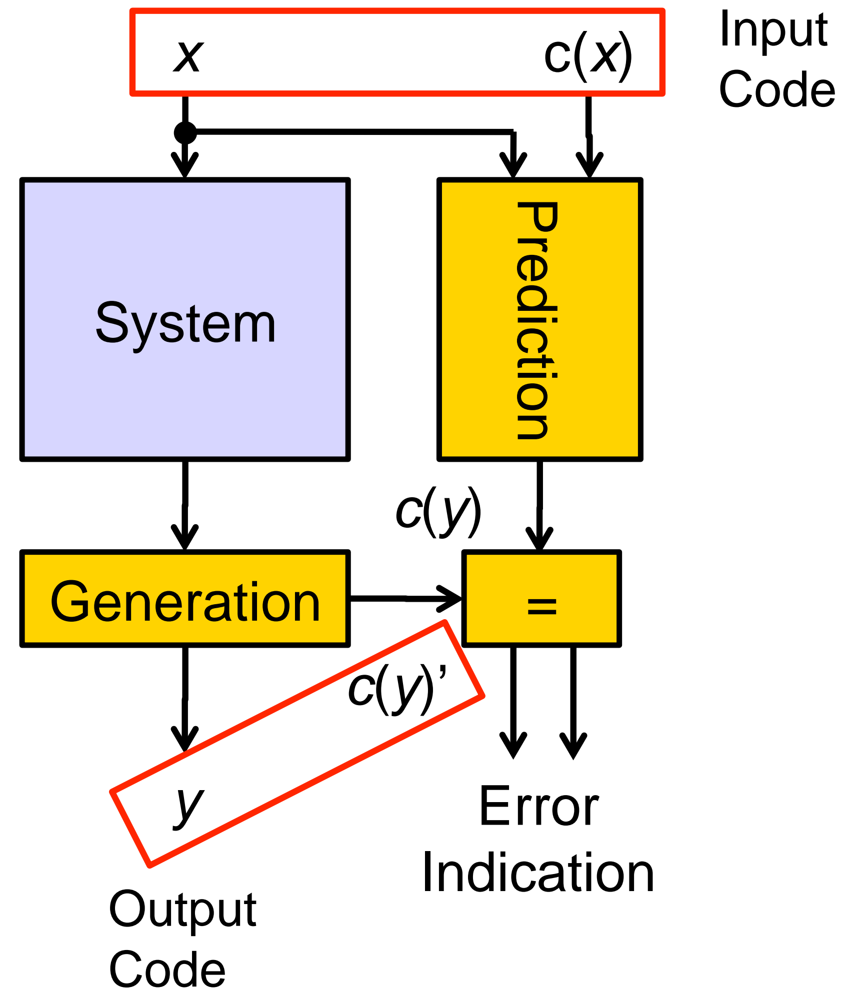
- Cost-effective solution to mitigate transient faults
- Design strategies for self-checking circuits well-known
- But: synthesis may destroy self-checking properties, e.g. by logic sharing



Robustness Analysis

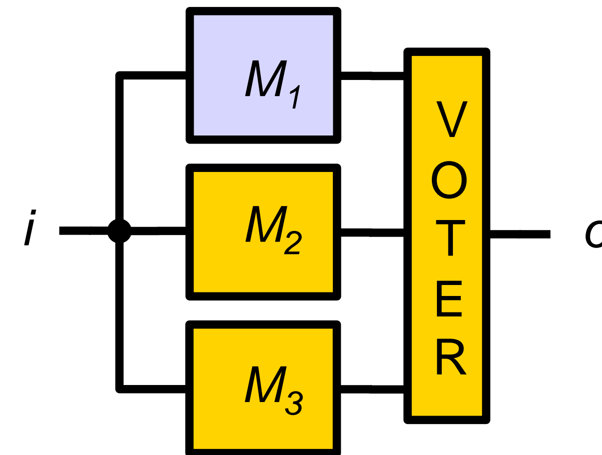
- Important for self-checking circuits: TSC property
  - Each fault is detected when it produces the first erroneous output
  - Fault accumulation must be considered
  - Analysis corresponds to ATPG problem for multiple faults with constraints

[IOLTS’08, IOLTS‘09]



Example 2: Triple Modular Redundancy

- Can compensate both permanent and transient faults
- Used both for yield and reliability improvement



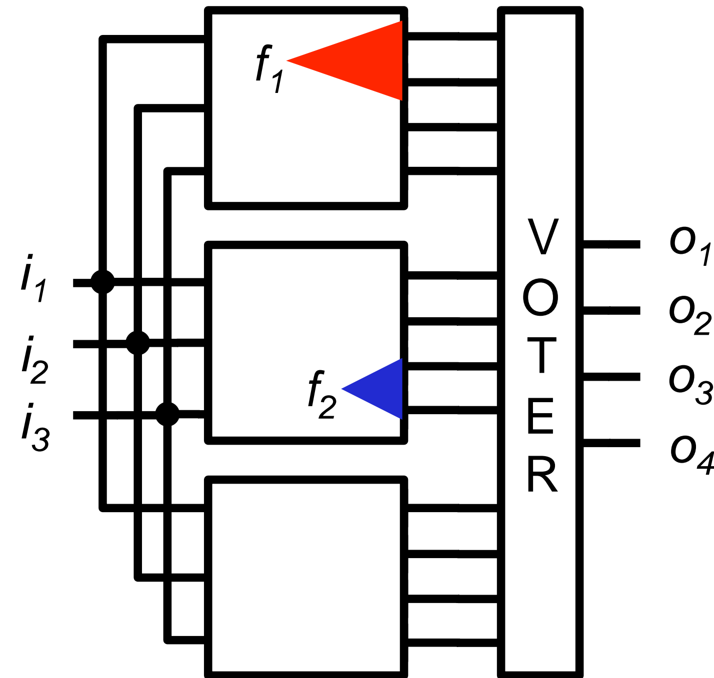
$$Yield = \sum_{i=0}^{\infty} r(i)\, p(i)$$

i faults tolerated — $r(i)$

i faults occur — $p(i)$

"Fault Tolerant" Yield

- Fault tolerance properties in the presence of compensated manufacturing defects ??
- Necessary: refined yield estimation for "fault tolerant" yield

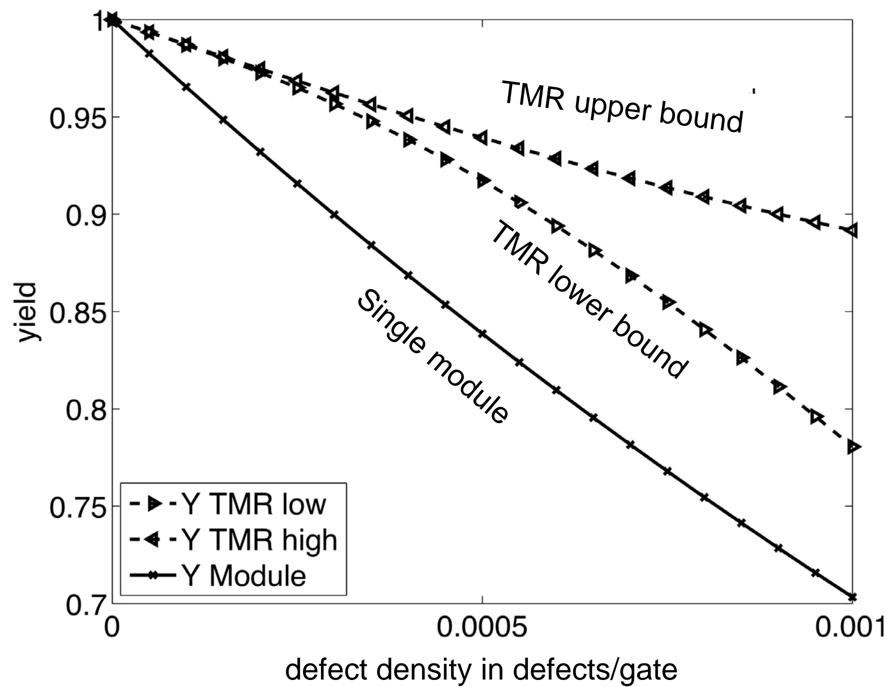


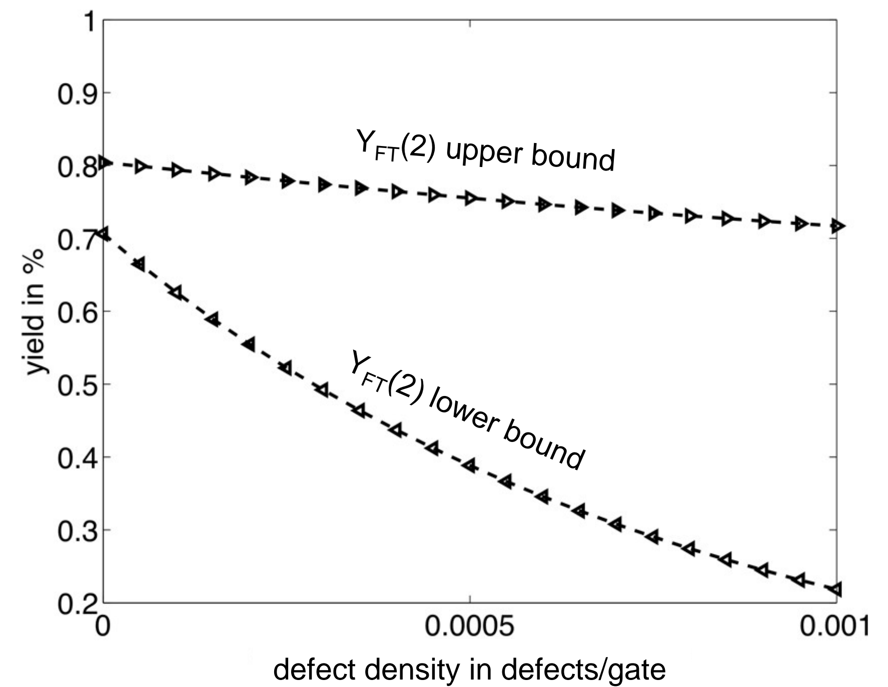
$$Y_{FT}(k) = \sum_{i=0}^{\infty} r(i+k \mid i) r(i) p(i)$$

k additional faults tolerated

[DFT'10]

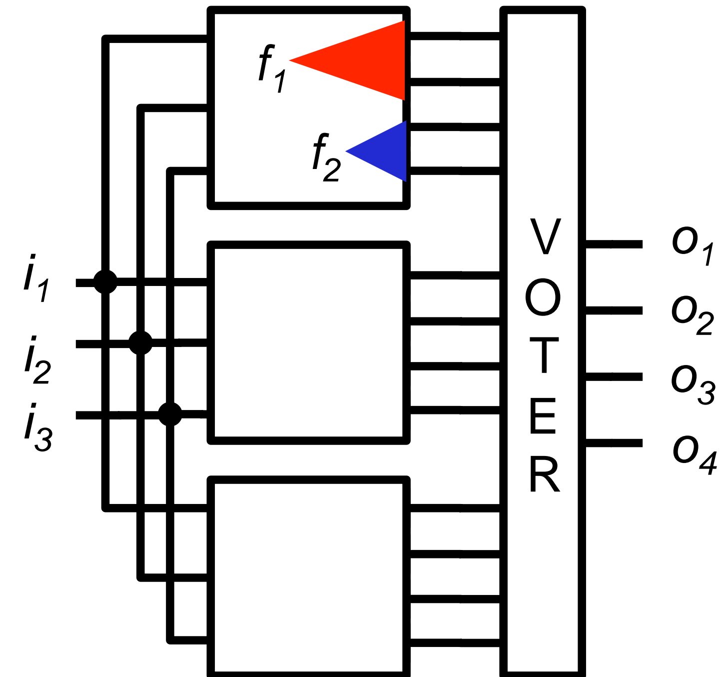
Preliminary Results





Quality Binning

- Go/NoGo is not sufficient as a result of manufacturing test
- Remaining robustness must be determined
  - "Functional" Test: Go/NoGo
  - Diagnostic Test with DfT
    - Reveals "functionally redundant" faults
    - Critical faults must be distinguished from tolerable faults



Conclusions

- Parameter variations require a paradigm change in testing
  - Variation-aware library characterization provides basis, main challenge is the reduction of the computational complexity
  - Basic statistical test algorithms have been outlined, optimized overall test flow is still challenging
  - Testing robust systems is particularly difficult, variation-aware diagnosis is needed
  - Parameter variations must be considered already at system level