Parallel Fault Backtracing for Calculation of Fault Coverage

R. Ubar, **S. Devadze**, J. Raik and A. Jutman

Tallinn University of Technology
Department of Computer Engineering
ESTONIA

ASPDAC 2008, January 21-24, Seoul, Korea

Outline

- Introduction and Motivation
- Description of the proposed method
- Experimental results
- Conclusions

Motivation

- **Fault simulation that used to build fault coverage table can take huge amount of time**

- **Fault simulation is widely used in digital circuit design flow:**
  - ❑ Built-in Self Test
  - ❑ Fault diagnosis
  - ❑ Test pattern generation
  - ❑ …

- **Acceleration of fault simulation will speed-up all abovementioned tasksы**

Introduction: previous work

- M.Abramovici, P.R. Menon, D.T.Miller, “Critical Path Tracing – an Alternative To Fault Simulation”, 1983, DAC

  Approximate fault simulation using critical path tracing

- K. Antriech, M. Schulz, “Accelerated Fault Simulation and Fault Grading in Combinational Circuits”, 1987, CAD

  Reducing number of fanout stems should be processed for fault simulation

- B. Underwood, J. Ferguson, “The Parallel-Test-Detect Fault Simulation Algorithm”, 1989, ITC

  Dominator gate concept, early cut-off of fault evaluation

- F. Maamari, J. Rajski, “A Method of Fault Simulation Based on Stem Regions”, 1990, CAD

  Stem regions and exit lines, reduces fault simulation area

- L.Wu, D.M.H. Walker, “A Fast Algorithm for Critical Path Tracing in VLSI Digital Circuits”, 2005, DFT

  Exact, linear-time critical path tracing

Introduction

Proposed fault analysis method:

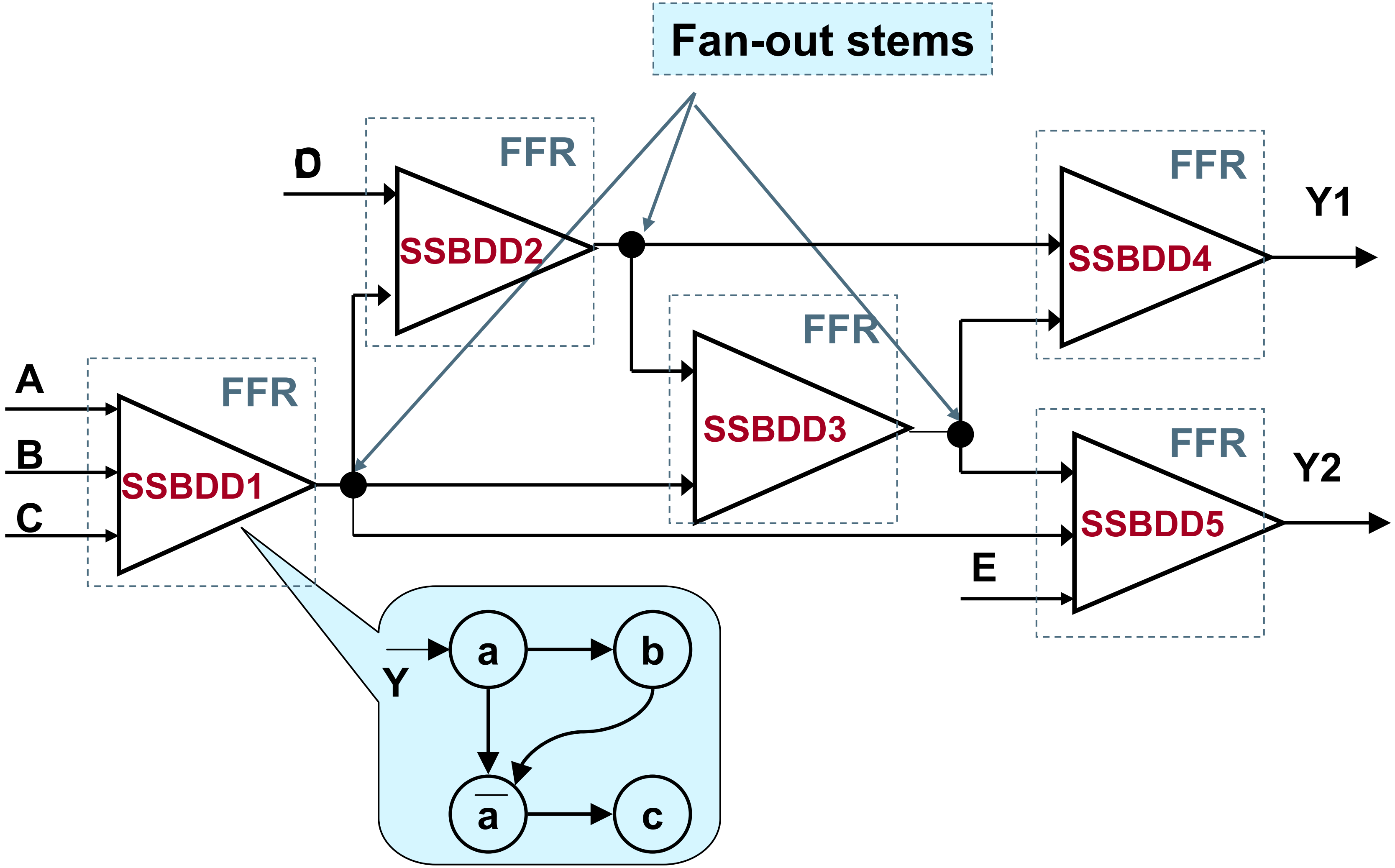
- Uses single stuck-at fault model
- Intended for use with combinational circuits
- Works on higher abstraction level than gate-level
- Describes circuit using special class of BDDs (SSBDD)
- Based on Critical Path Tracing technique

Parallel computations for *N* patterns (*N* – width of computer word)

Extends critical path tracing beyond Fan-out Free Regions (FFRs)

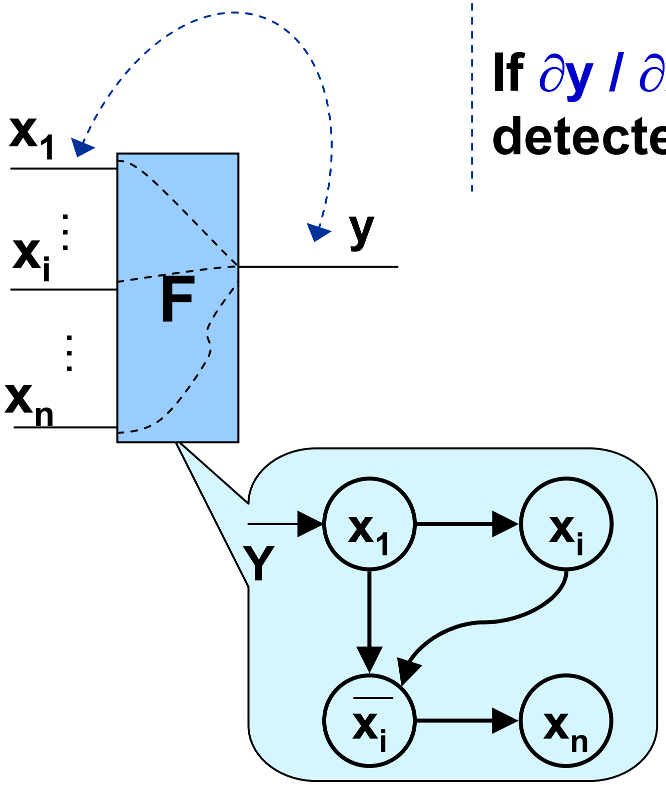
Uses calculation of parallel Boolean derivatives

Circuit representation



Critical Path Tracing inside FFR

Fan-out free region (FFR)



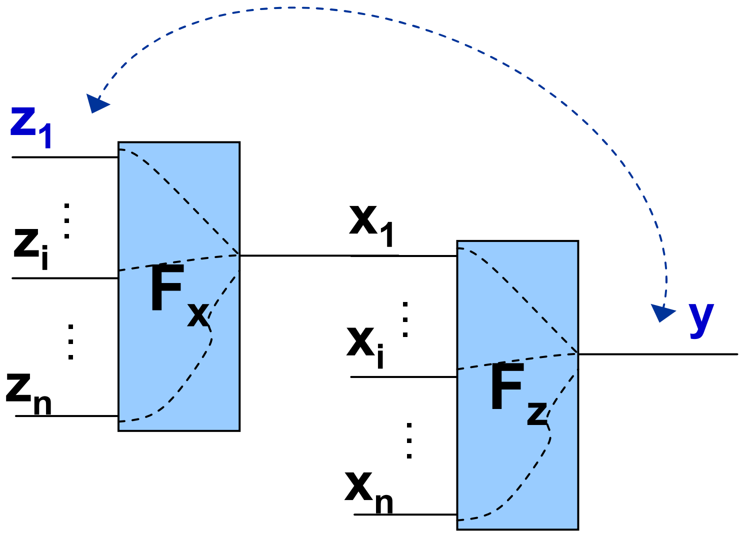
Boolean derivative: $\partial y / \partial x_1$

If $\partial y / \partial x_1 = 1$ – fault at x_1 is detected at output y

Using of special Structurally Synthesized Binary Decision Diagrams (SSBDD) we can rapidly calculate parallel Boolean derivatives (critical path tracing on SSBDD)

Extending Critical Path Tracing

Two consecutive Fan-out Free Regions

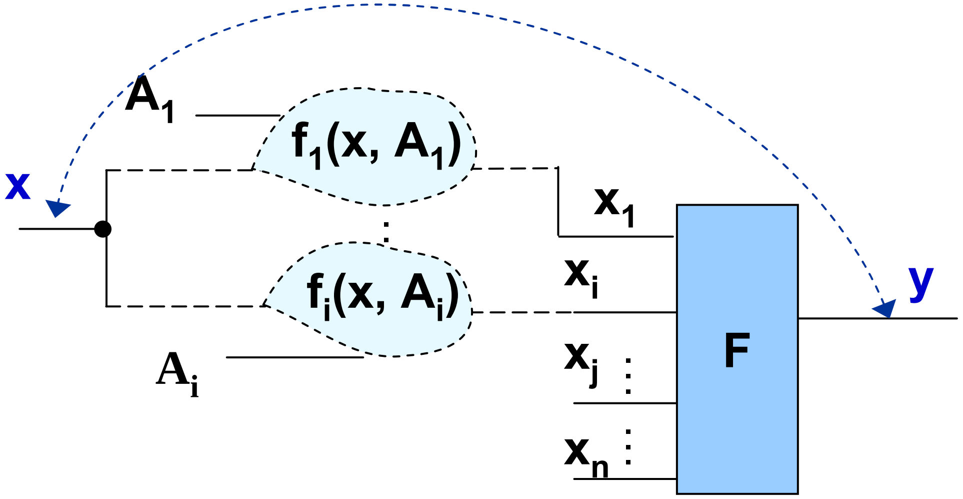


Sensitivity of y to fault at z_1:

$$\partial y / \partial z_1 = (\partial y / \partial x_1) \wedge (\partial x_1 / \partial z_1)$$

Extending Critical Path Tracing (2)

Reconvergent fan-out



$y = F(x_1, \ldots, x_i, x_j, \ldots x_n)$

$x_1 = f_1(x, X_1)$

…

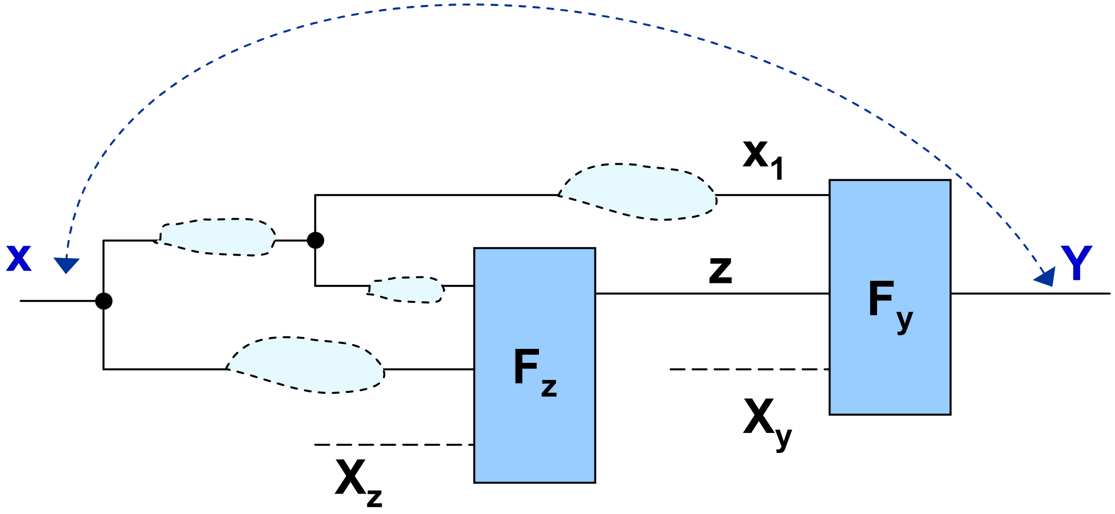
$x_i = f_i\,(x, X_i)$

$$\partial y \,/\, \partial x = y \oplus F(x_1 \oplus (\partial x_1 / \,\partial x)), \ldots, (x_i \oplus (\partial x_i \,/\, \partial x)), x_j\,, \ldots, x)$$

Where: $\partial x_1 / \,\partial x$ and $\partial x_i \,/\, \partial x$ are Boolean derivatives were calculating during critical path tracing inside fan-out free regions f_1 and f_2

Extending Critical Path Tracing (3)

Nested reconvergencies



$y = F_y(x, z, X_y)$

$z = F_z(x, X_z)$

$$\partial y / \partial x = y \oplus F_y(x_1 \oplus (\partial x_1 / \partial x), z \oplus (\partial F_z / \partial x), X_y)$$

Fault simulation algorithm (steps)

- **Topological pre-analysis**

  Constructs reconvergency and calculation models of the circuit

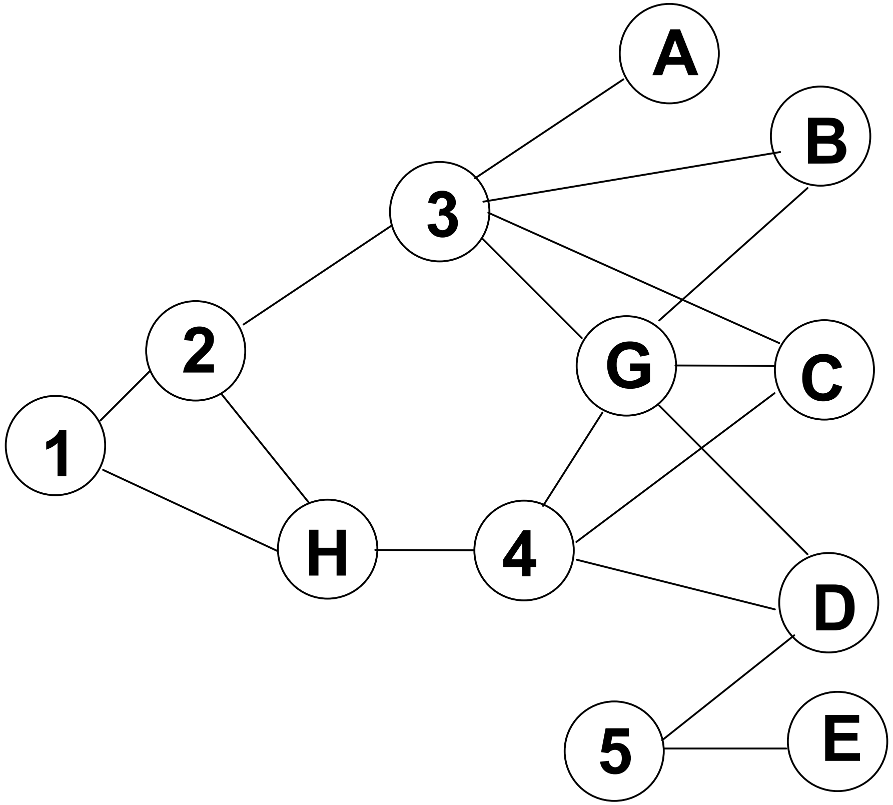
- **Parallel simulation**

  Calculates the values of all variables for given set of patterns

- **Fault simulation**

  Performs fault backtracking on the created calculation model

Topological model

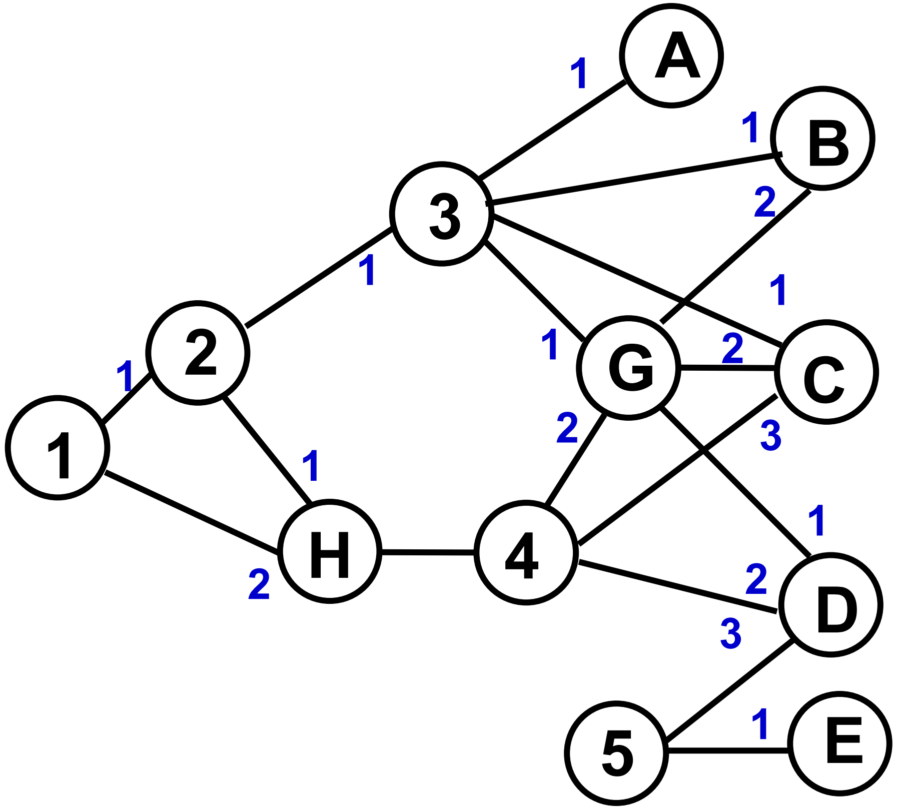


Primary outputs:
A, B, C, D, E

Fan-out stems:
1, 2, 3, 4, 5, G

Internal fan-in gates:
H, G

Creating formulas / calculation steps



Simulation from node 2:

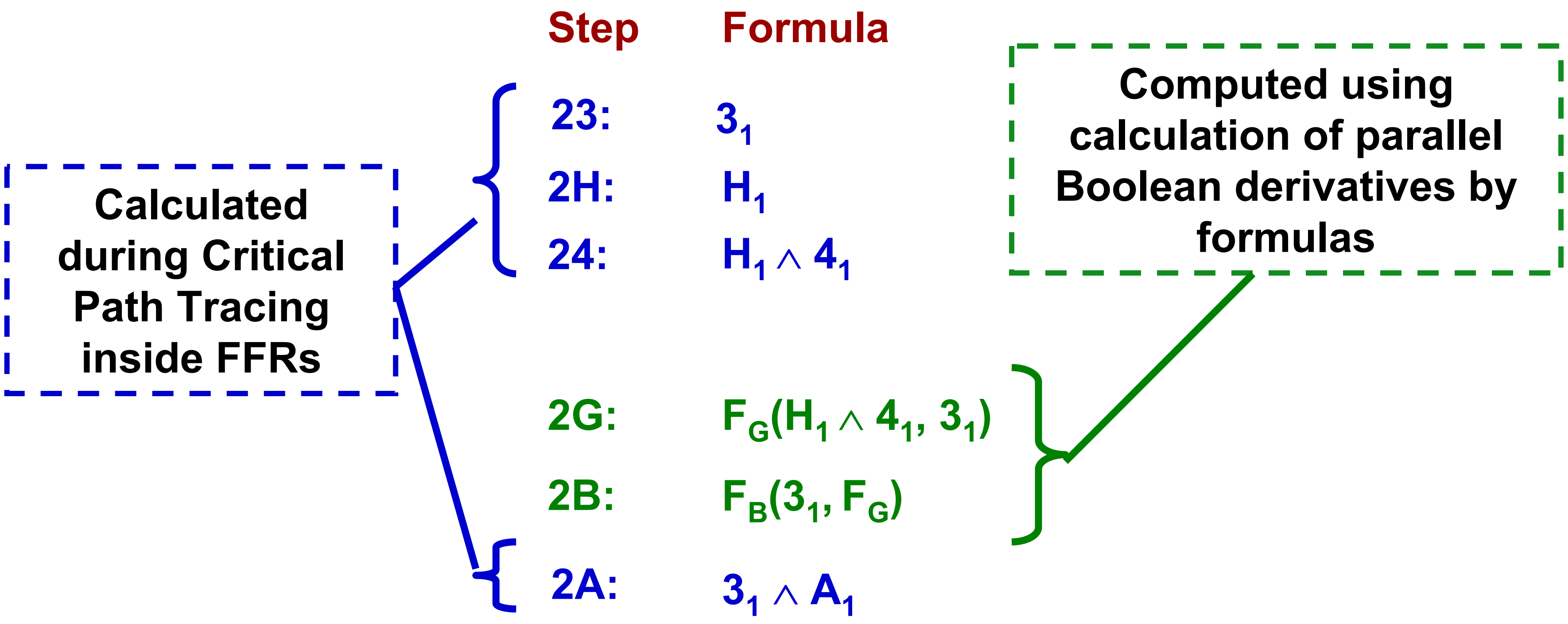
Step	Action / Whole formula
23:	3_1
2H:	H_1
24:	$H_1 \wedge 4_1$
2G:	$F_G(H_1 \wedge 4_1, 3_1)$
2B:	$F_B(3_1, F_G)$
2A:	$3_1 \wedge A_1$

……..

$2A \vee 2B \vee 2C \vee 2D$

Calculation model

Simulation from node 2:



Step	Formula
23:	3_1
2H:	H_1
24:	$H_1 \wedge 4_1$
2G:	$F_G(H_1 \wedge 4_1, 3_1)$
2B:	$F_B(3_1, F_G)$
2A:	$3_1 \wedge A_1$

Calculated during Critical Path Tracing inside FFRs

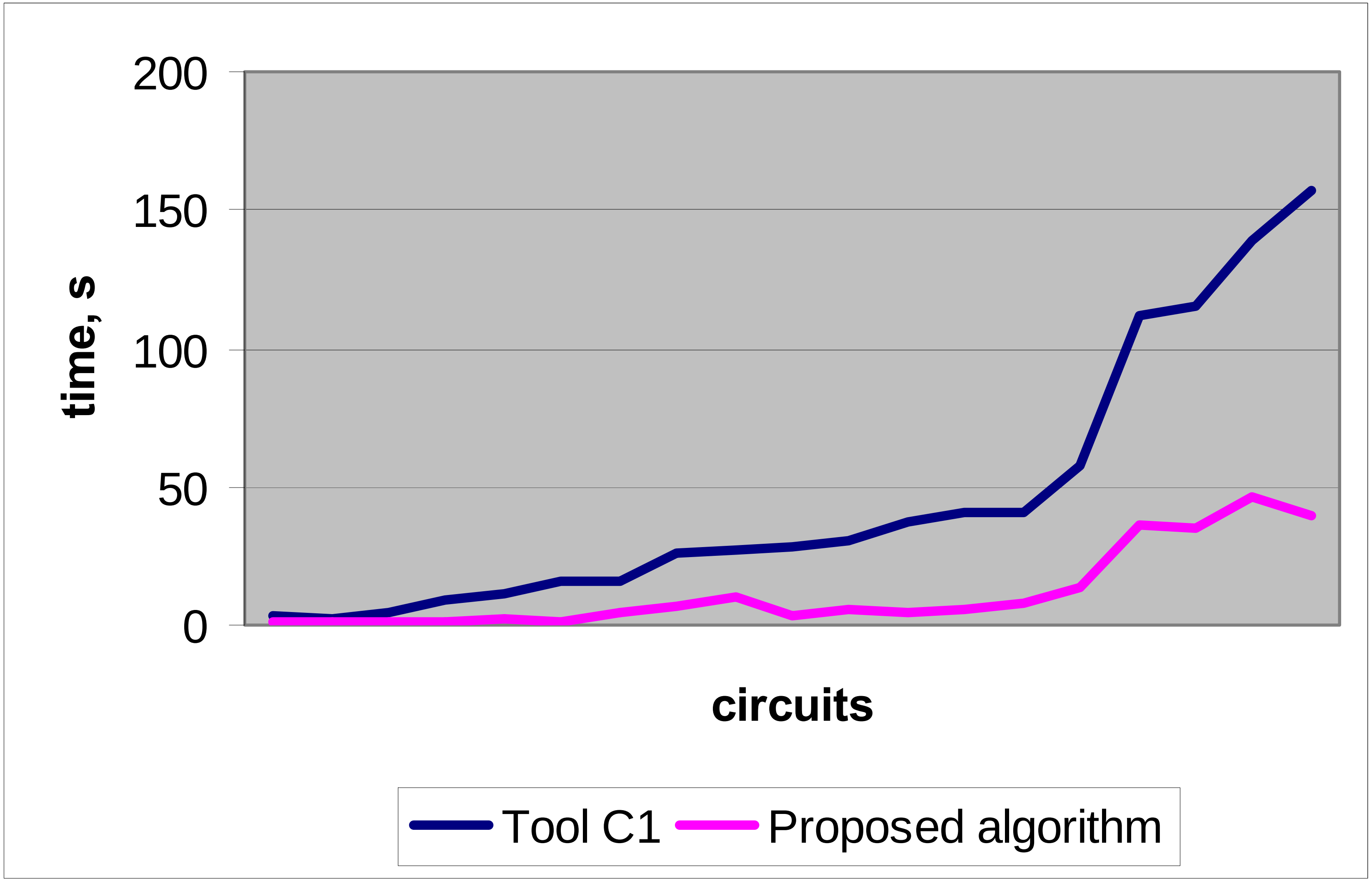
Computed using calculation of parallel Boolean derivatives by formulas

Experimental results

- ISCAS’85/ISCAS’89 combinational benchmarks
- Comparison with:
  - State-of-the-art commercial tools from CAD vendors
  - Exact Critical Path Tracing implementation by:

    L.Wu, D.M.H. Walker, “A Fast Algorithm for Critical Path Tracing in VLSI Digital Circuits”, 2005, DFT
  - Older version of the same algorithm (w/o topology optimization)
- Fault dropping mode was disabled
- 10000 patterns were simulated for each circuit

Scalability of the algorithm



Conclusions / Future Work

- New fault simulation algorithm is proposed
- Simulation is performed for network of macros (instead of gates) with gate-level accuracy
- Macros are represented by Structurally Synthesized BDDs
- Topological analysis is used to speed-up simulation
- The speed of fault simulation outperforms several commercial tools

- Further optimization still possible
- Solution for fault dropping

Thank You for Your Attention!