5. Testide analüüs

5.1. Rikete simuleerimise meetodid

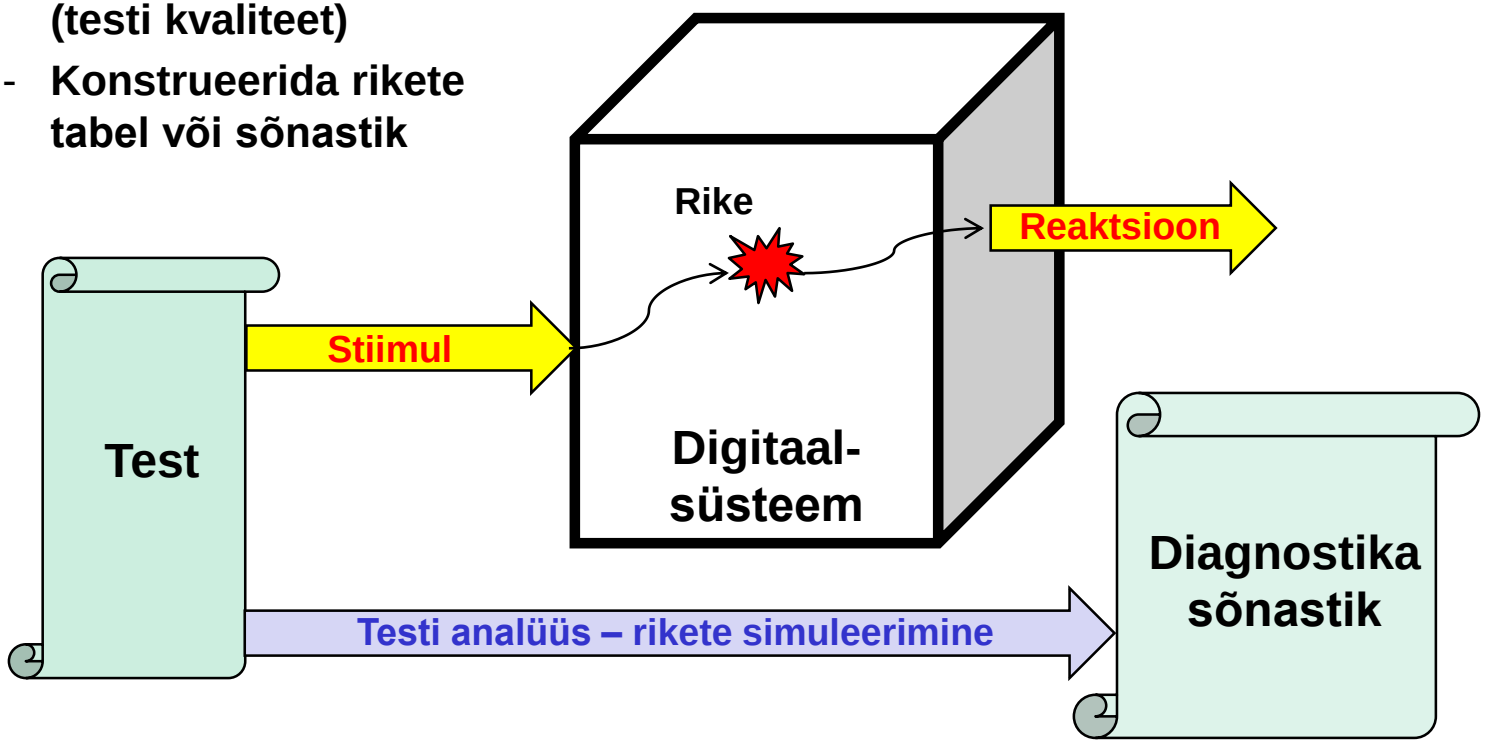
5.2. Testide analüüs deduktiivsete meetoditega

5.3. Rikete simuleerimine ja analüüs mäluga skeemides

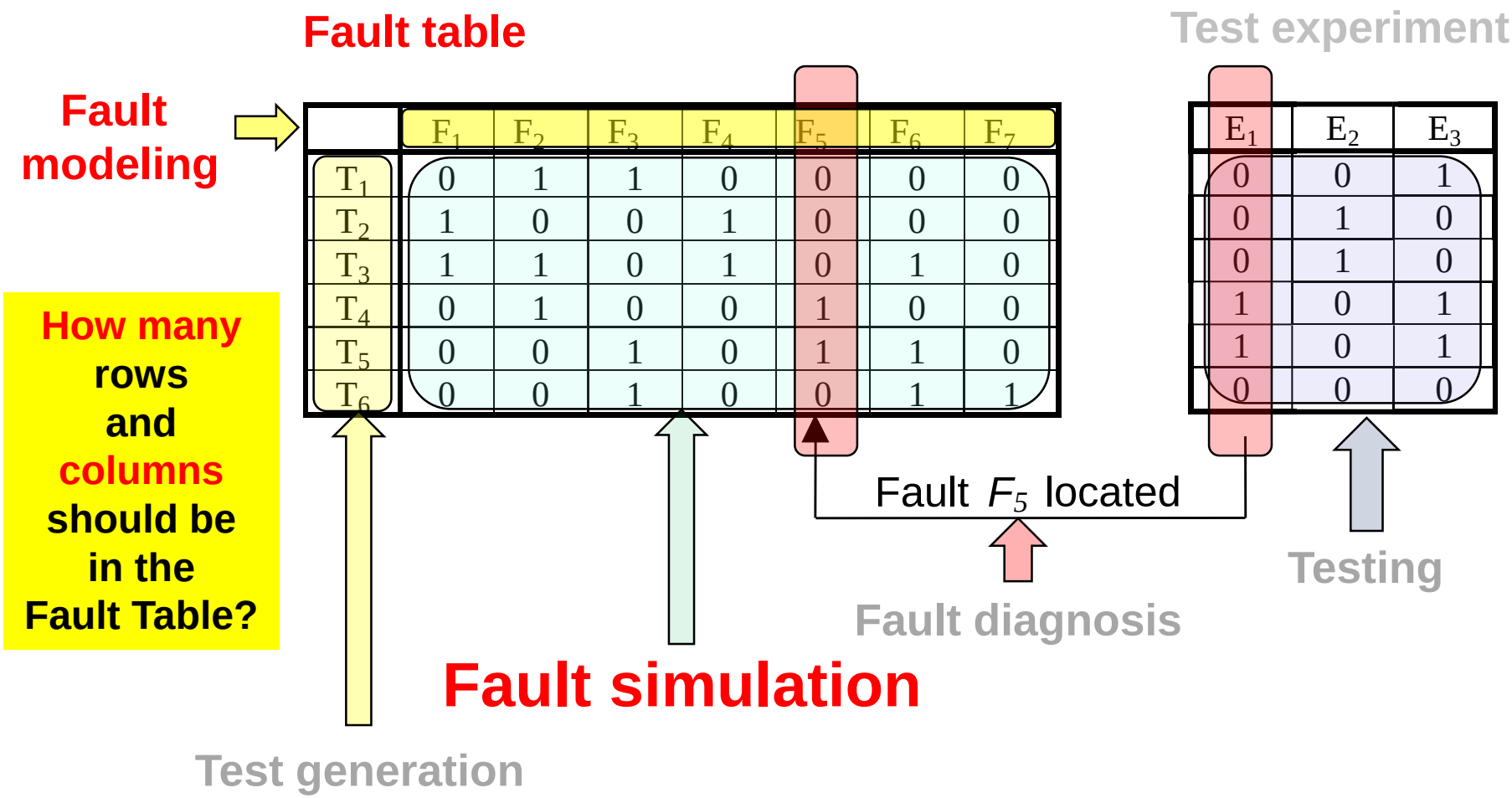
Testide analüüs

Rikete simuleerimise eesmärgid:

- **Arvutada rikete kate (testi kvaliteet)**
- **Konstrueerida rikete tabel või sõnastik**



Faults in Digital Circuits



	F_1	F_2	F_3	F_4	F_5	F_6	F_7
T_1	0	1	1	0	0	0	0
T_2	1	0	0	1	0	0	0
T_3	1	1	0	1	0	1	0
T_4	0	1	0	0	1	0	0
T_5	0	0	1	0	1	1	0
T_6	0	0	1	0	0	1	1

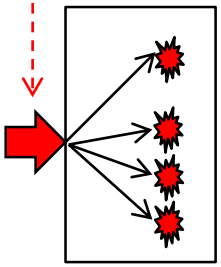
E_1	E_2	E_3
0	0	1
0	1	0
0	1	0
1	0	1
1	0	1
0	0	0

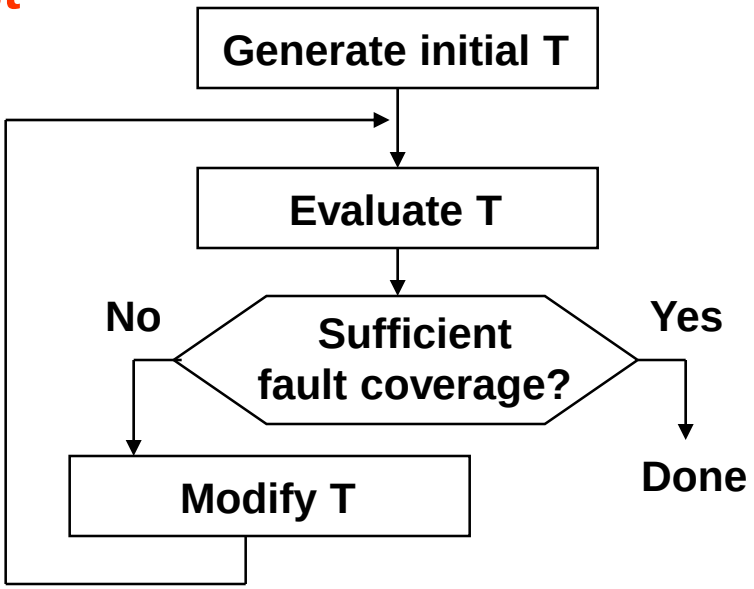
Fault simulation

Goals:

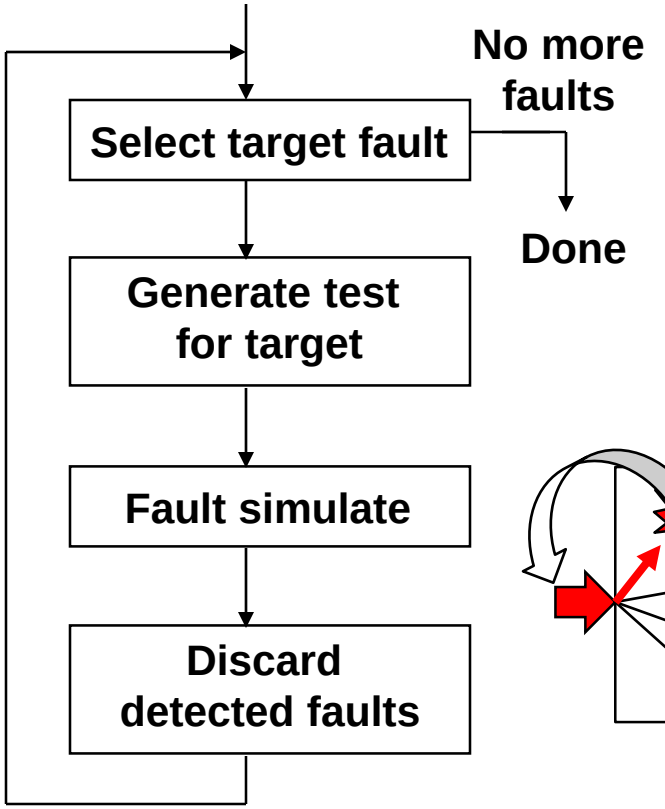
- Evaluation (grading) of a test T (fault coverage)
- Guiding the test generation process
- Constructing fault tables (dictionaries)
- Fault diagnosis

Random test generation





Deterministic test generation



Impact of Scan-Path: Fault Simulation

Fault simulation is needed for: test generation, fault diagnosis, test quality analysis, design verification, fault tolerance evaluation…

Fault simulation techniques:

Serial fault simulation

Parallel fault simulation

Analytical fault reasoning:

Deductive fault simulation

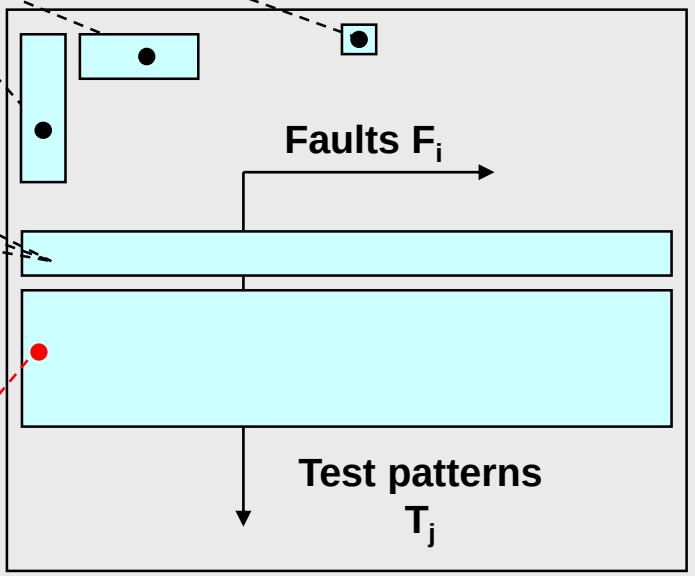
Concurrent fault simulation

Critical path analysis

These methods are not possible for sequential circuits

Comparison of methods:

Fault table



Entry (i,j) = 1 if F_i is detectable by T_j

Entry (i,j) = 0 if F_i is not detectable by T_j

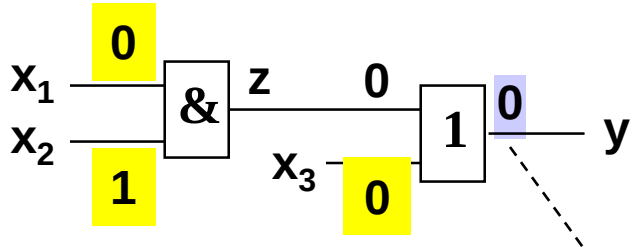
Parallel critical path analysis

ETS'2007 – for SAF
DATE'2010 – for larger class of faults
DATE'2015 – in the multicore environment

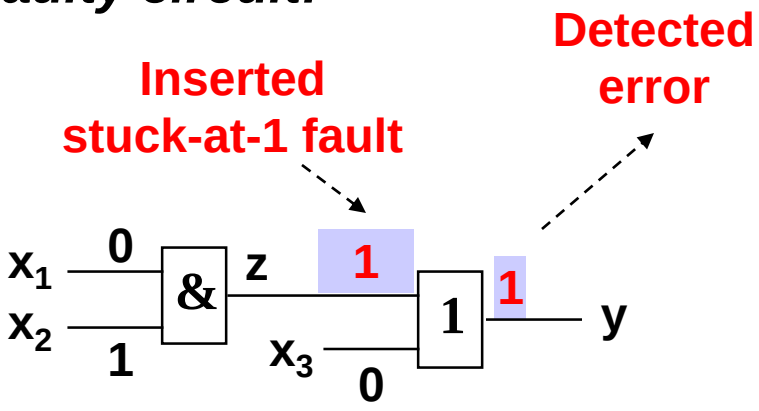
Single and Parallel SAF Fault Simulation

Single pattern

Fault-free circuit:

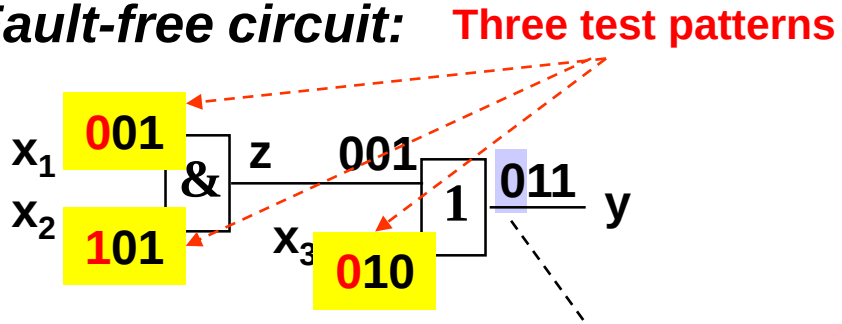


Faulty circuit:

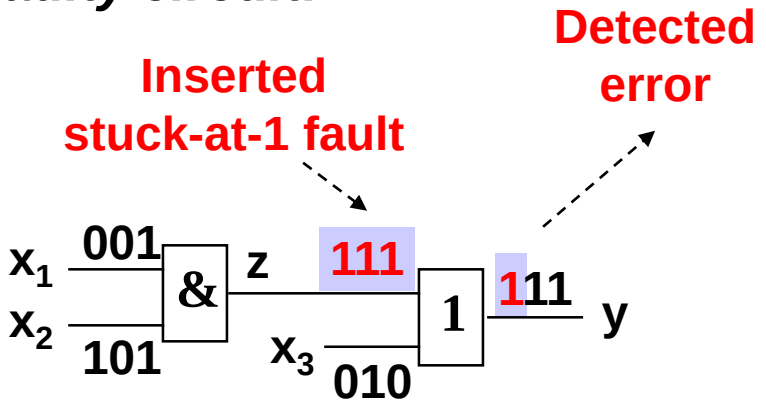


Parallel patterns

Fault-free circuit:



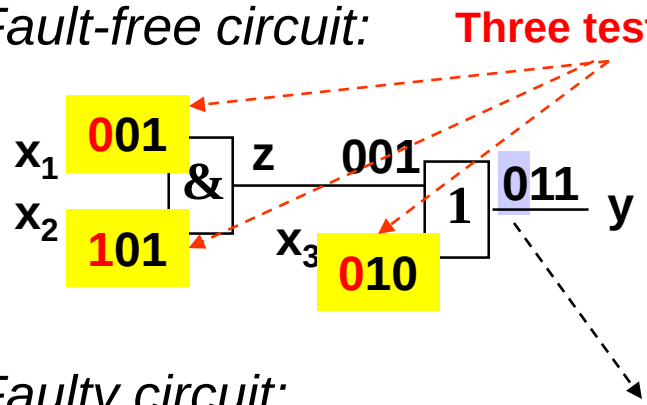
Faulty circuit:



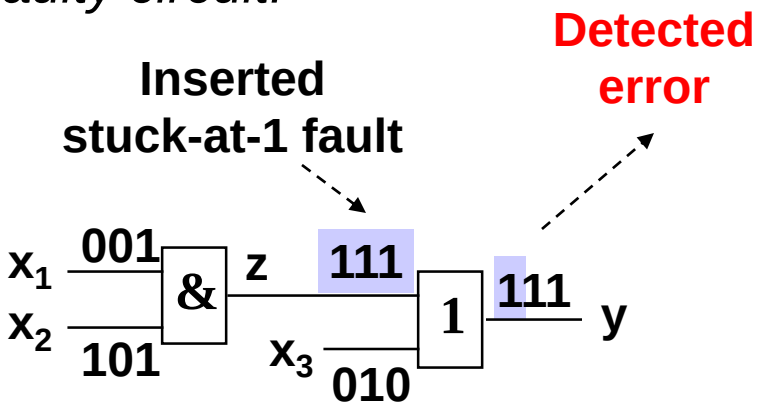
Parallel Fault & Test Pattern Simulation

Parallel patterns

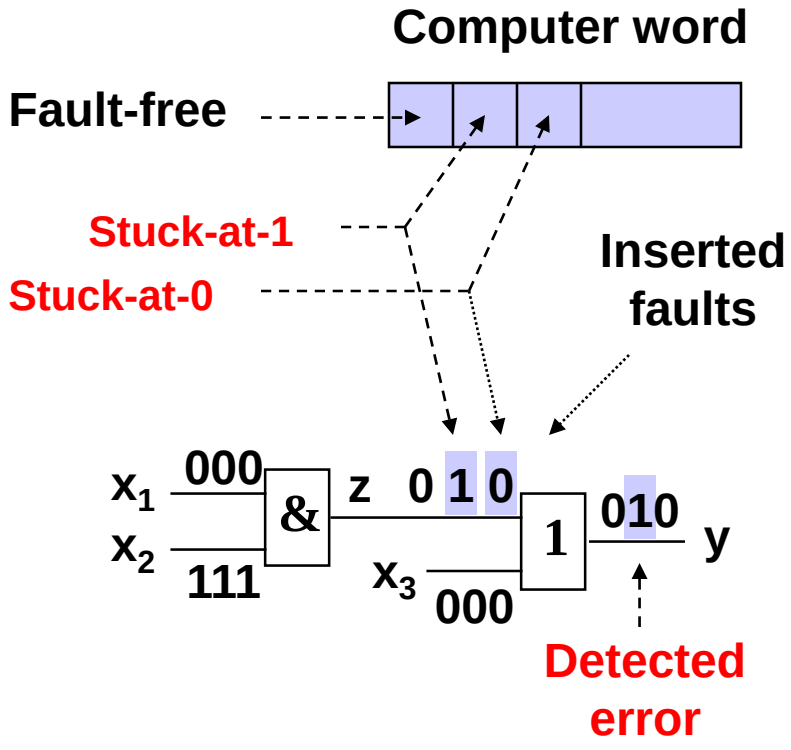
Fault-free circuit:



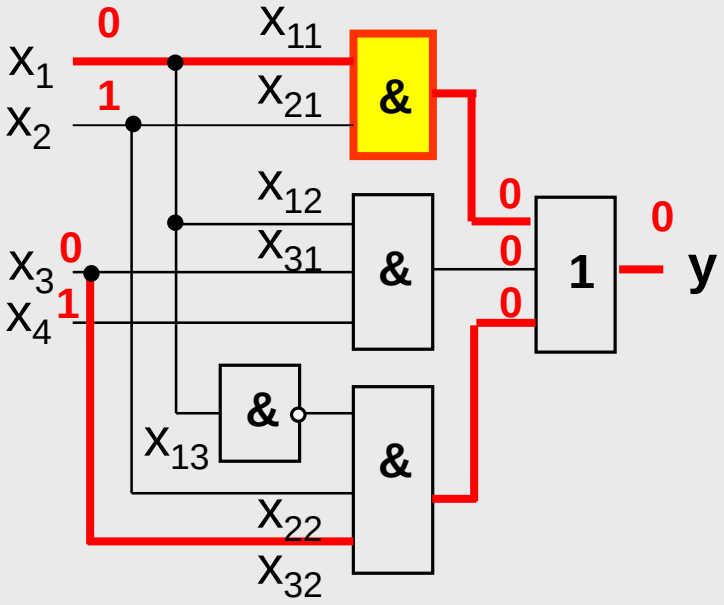
Faulty circuit:

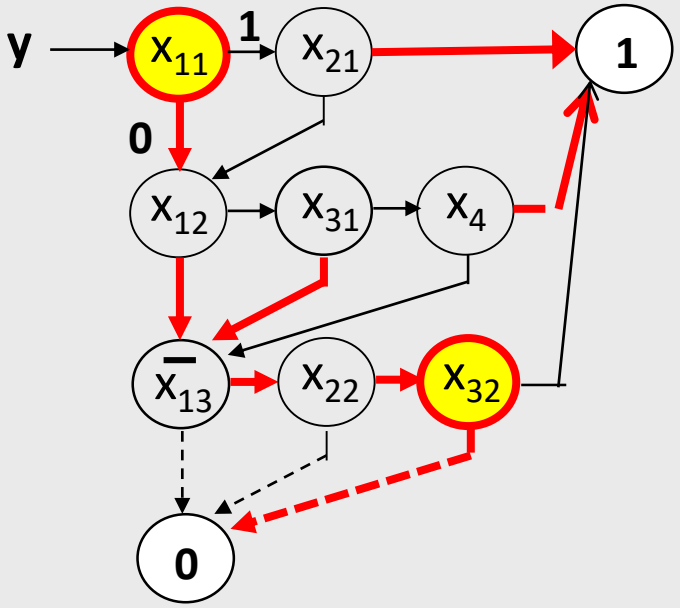


Parallel faults



Fault Analysis with SSBDDs

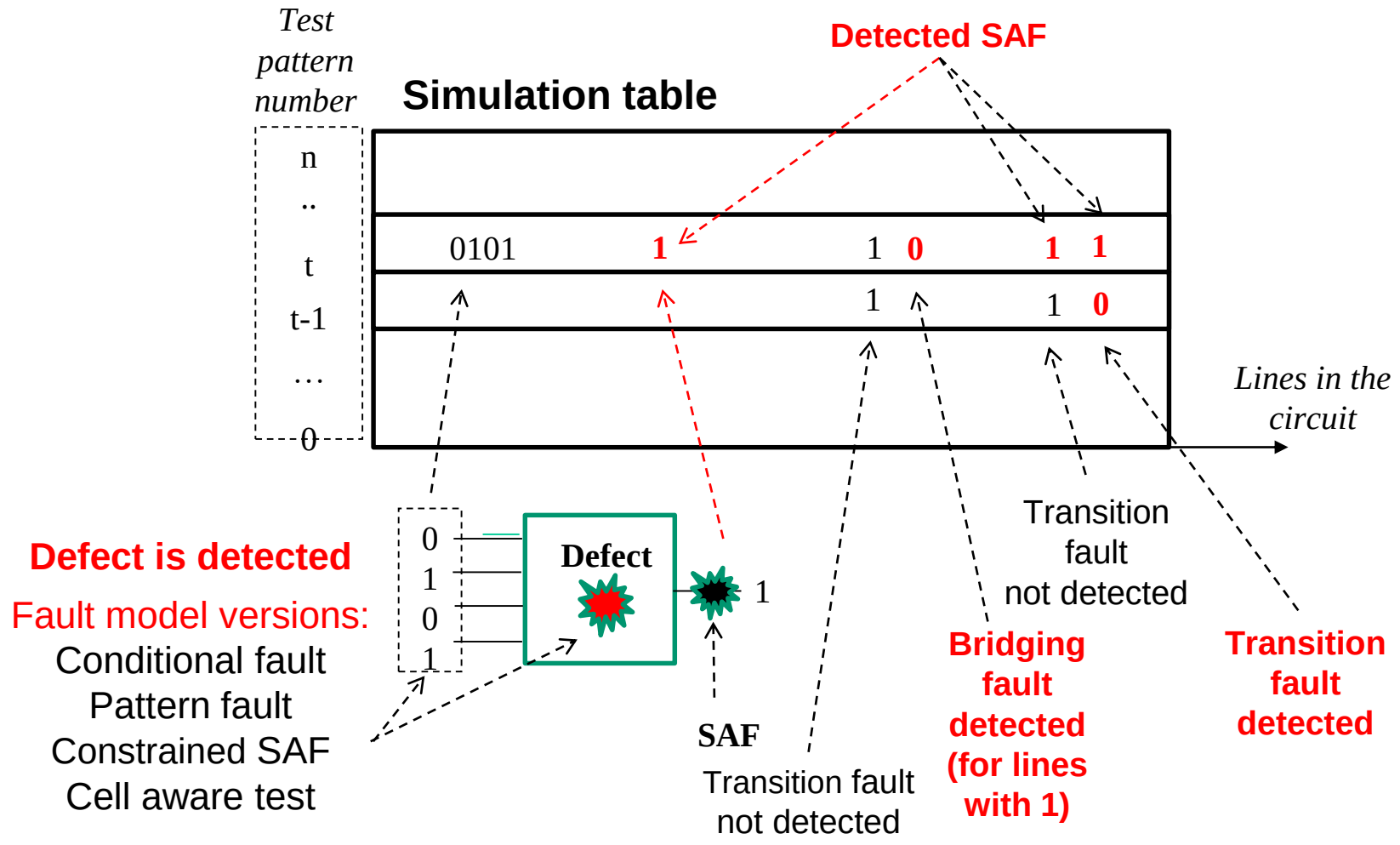




Algorithm:

1. Determine the activated path to find the fault candidates
2. Analyze the detectability of the each candidate fault (each node represents a subset of real faults)

Simulation of Diferent Classes of Faults



5. Testide analüüs

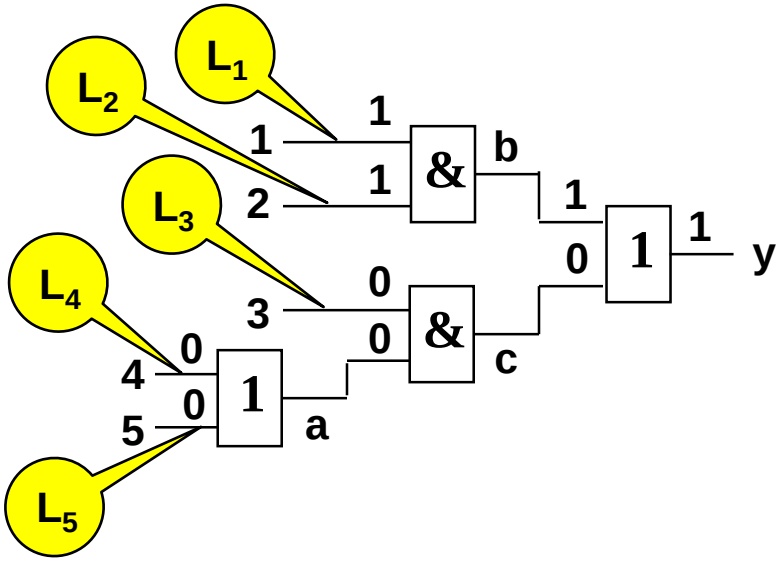
5.1. Rikete simuleerimise meetodid

5.2. Testide analüüs deduktiivsete meetoditega

5.3. Rikete simuleerimine ja analüüs mäluga skeemides

Deductive Fault Simulation

Gate-level fault list propagation



L_a – faults causing erroneous signal on the node a

Fault list calculation:

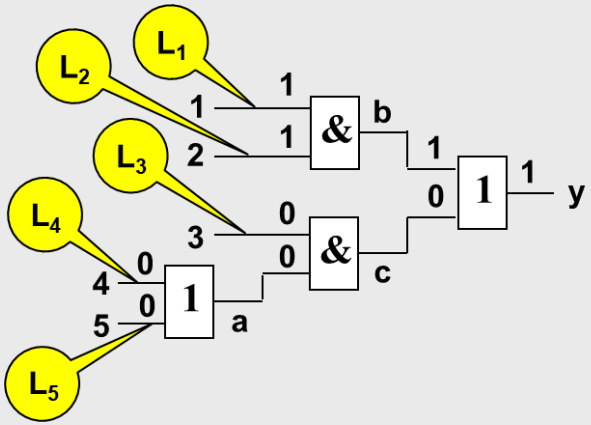
$$L_a = L_4 \cup L_5$$
$$L_b = L_1 \cup L_2$$
$$L_c = L_3 \cap L_a$$
$$L_y = L_b - L_c$$

$$L_y = (L_1 \cup L_2) - (L_3 \cap (L_4 \cup L_5))$$

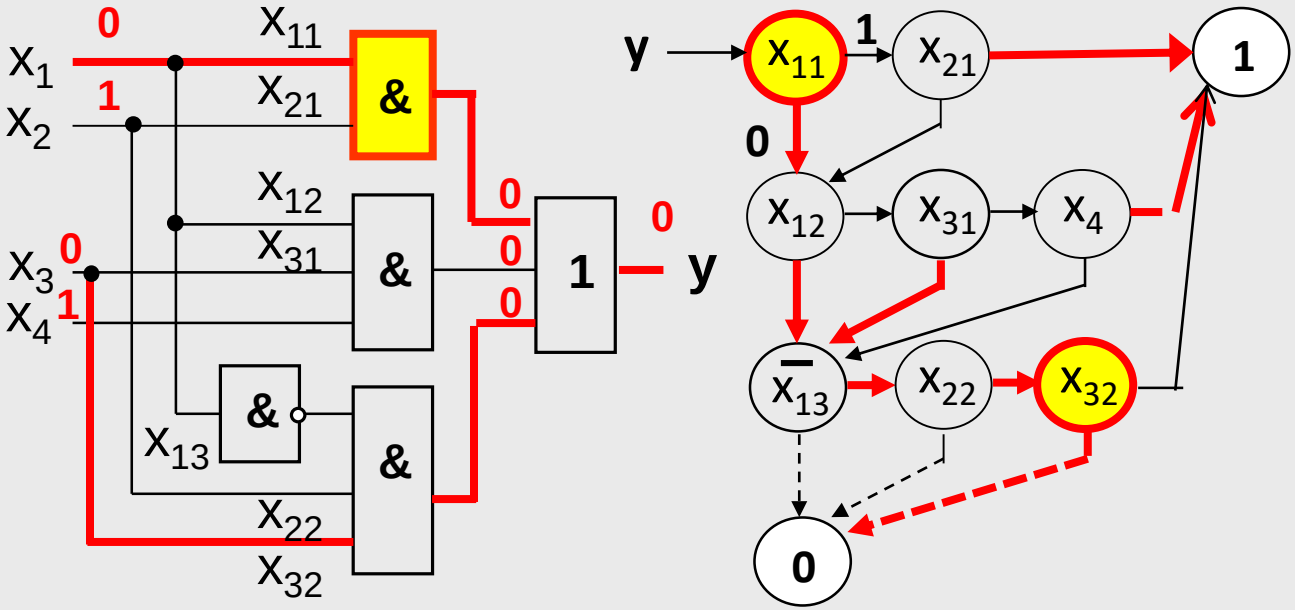
Library of formulas for gates

L_y – faults causing erroneous signal on the output node y

Fault Analysis with SSBDDs



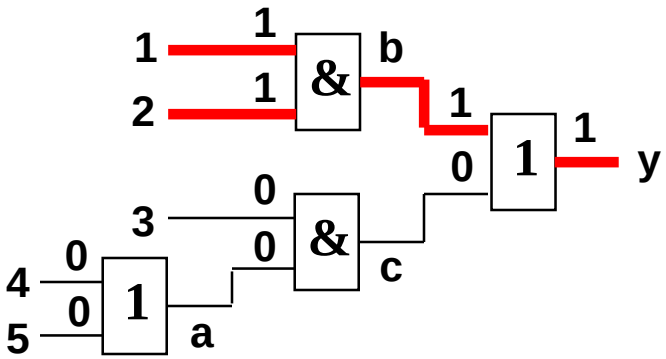
The components to be analyzed may be also fan-out free regions (FFR), and the fault propagation can be calculated with SSBDDs



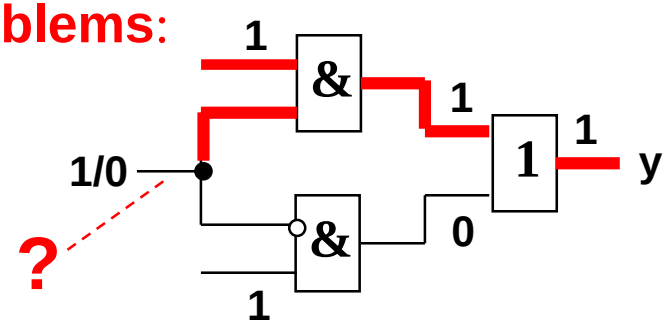
Algorithm:

1. Determine the activated path to find the fault candidates
2. Analyze the detectability of the each candidate fault (each node represents a subset of real faults)

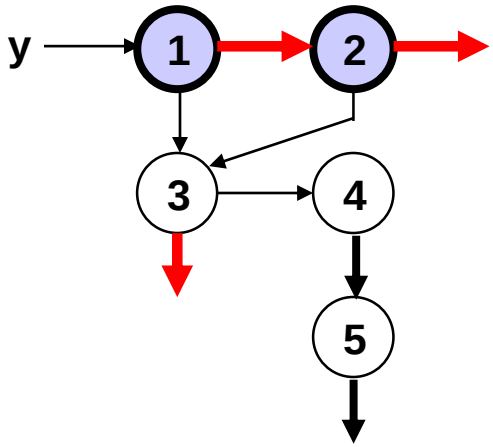
Critical Path Tracing

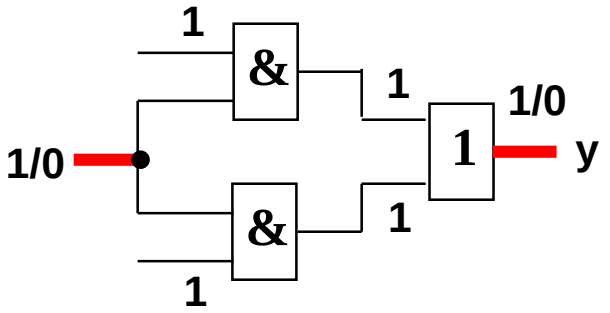


Problems:



The critical path breaks on the fan-out

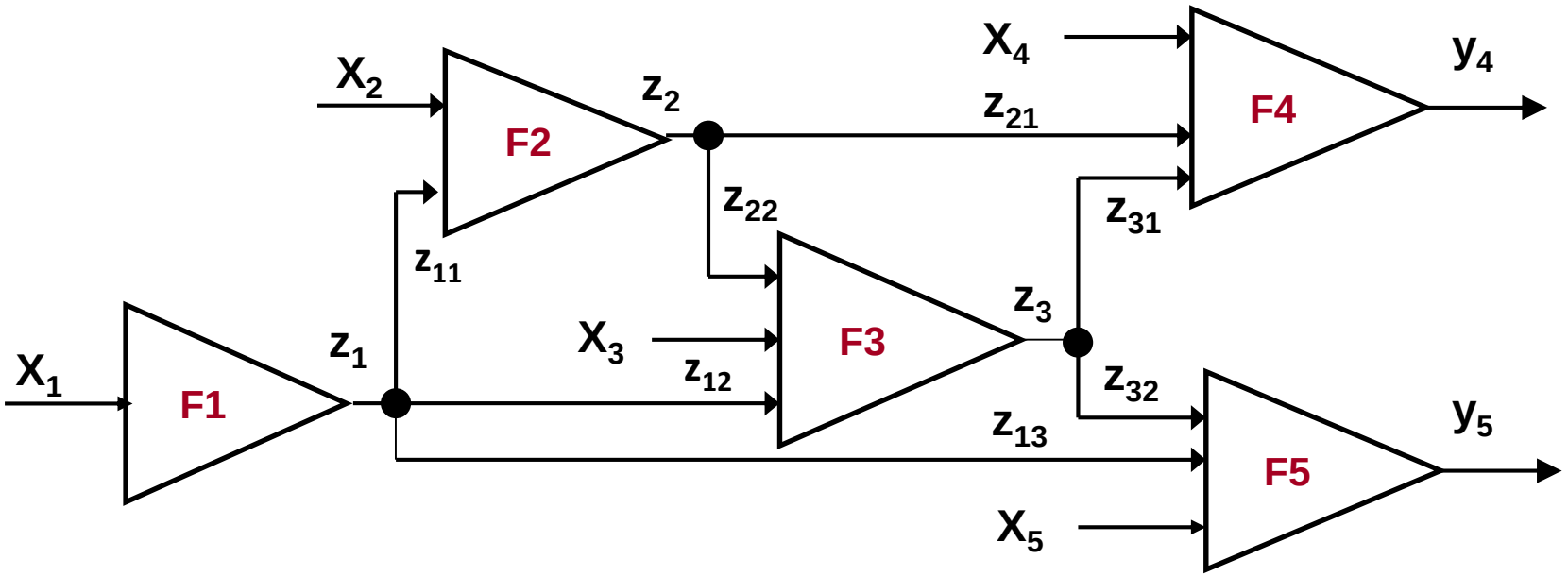




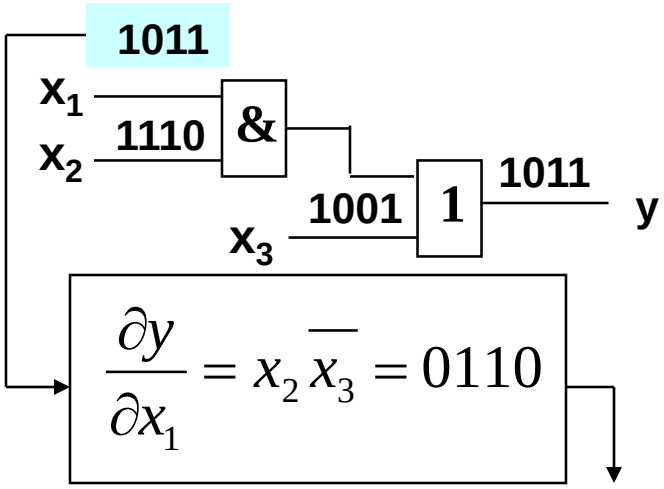
The critical path is not continuous

Parallel Critical Path Tracing

Problem with fan-out points:



Parallel Critical Path Tracing

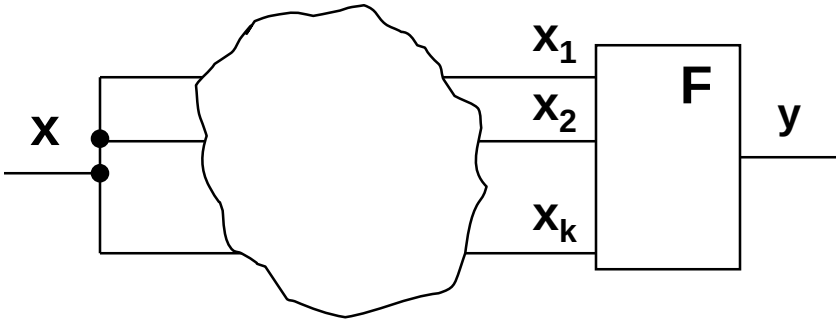


Detected faults vector: **- 10 -**

T1: No faults detected
T2: $x_1 \equiv 1$ detected
T3: $x_1 \equiv 0$ detected
T4: No faults detected

Handling of fanout points:

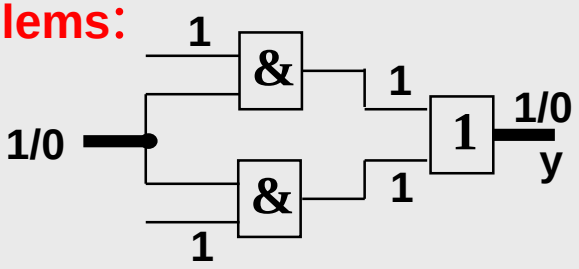
- Fault simulation
- Boolean differential calculus



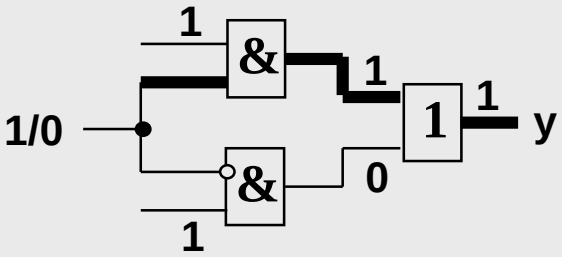
$$\frac{\partial y}{\partial x} = y \oplus F((x_1 \oplus \frac{\partial x_1}{\partial x}),\ldots,(x_k \oplus \frac{\partial x_k}{\partial x}))$$

Parallel Critical Path Tracing

Problems:



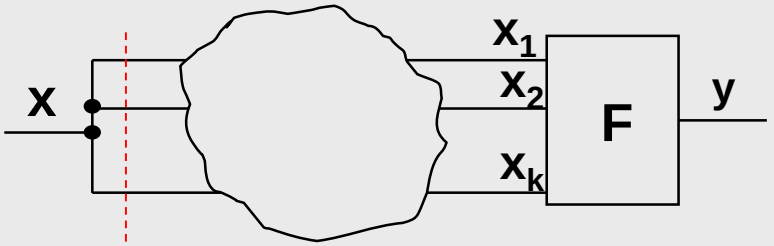
The critical path is not continuous



The critical path breaks on the fan-out

Solution:

Secret of solving the reconvergent fan-out problem:



In this FFR-part trad. backtracing is used

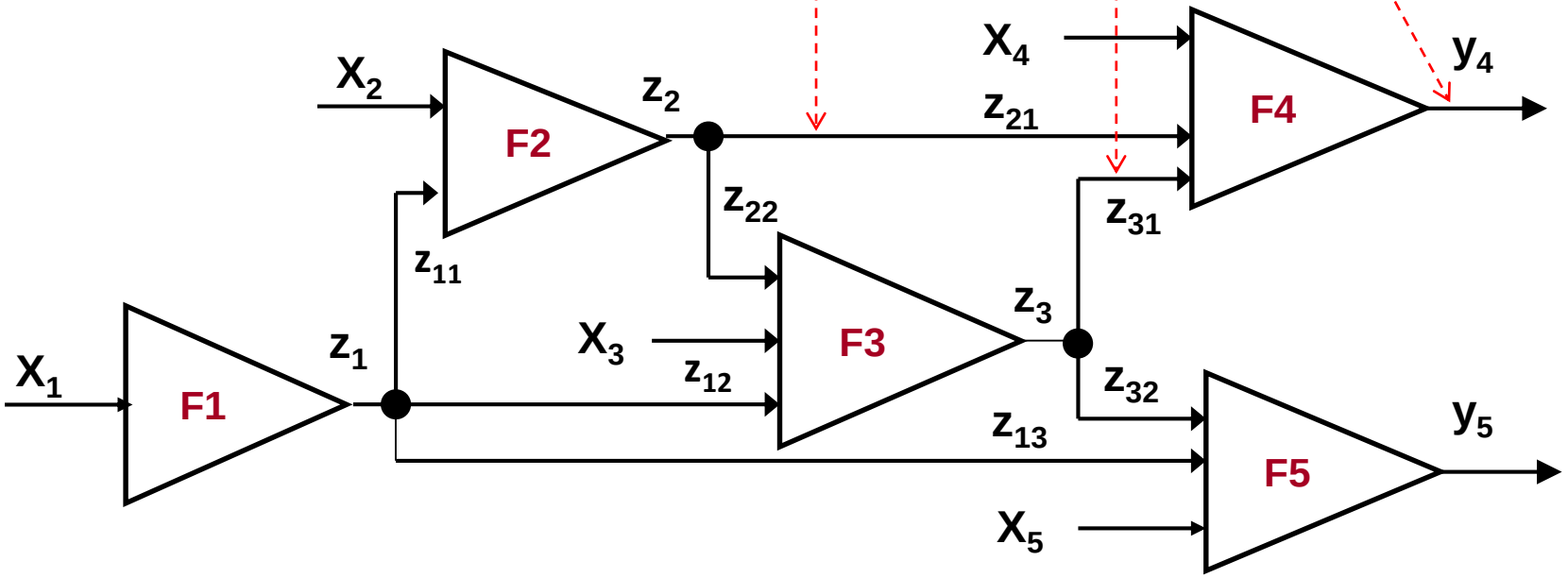
$$\frac{\partial y}{\partial x} = y \oplus F((x_1 \oplus \frac{\partial x_1}{\partial x}),\ldots,(x_k \oplus \frac{\partial x_k}{\partial x}))$$

The result is exact critical path tracing beyond the convergent fan-outs

Parallel Critical Path Tracing

$$\frac{\partial y_4}{\partial z_2} = F_4(x_4, (z_{21} \oplus \frac{\partial z_{21}}{\partial z_2}), (z_{31} \oplus \frac{\partial z_{31}}{\partial z_2})) \oplus y_4$$

Problem with fan-out points:



5. Testide analüüs

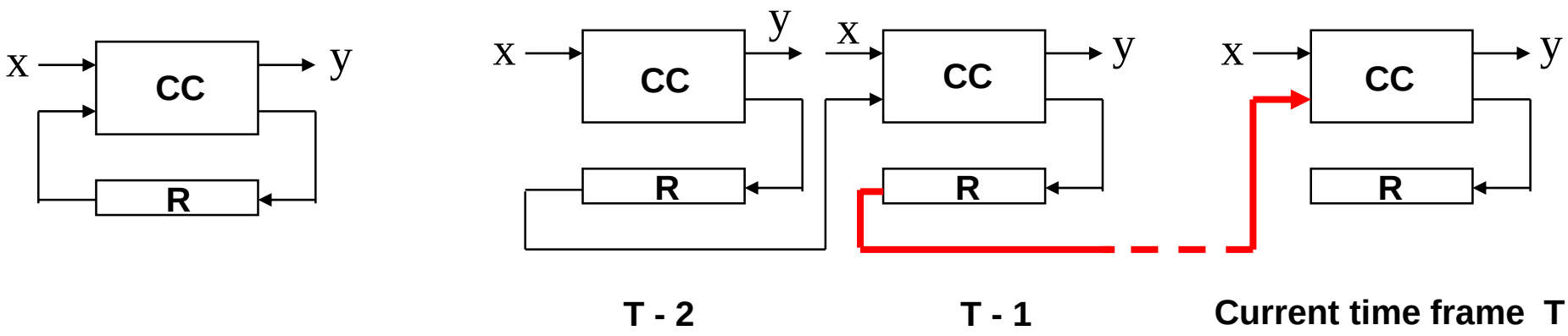
5.1. Rikete simuleerimise meetodid

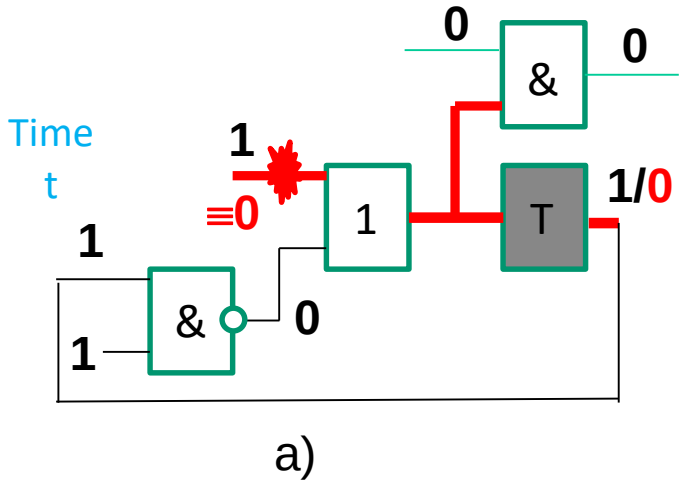
5.2. Testide analüüs deduktiivsete meetoditega

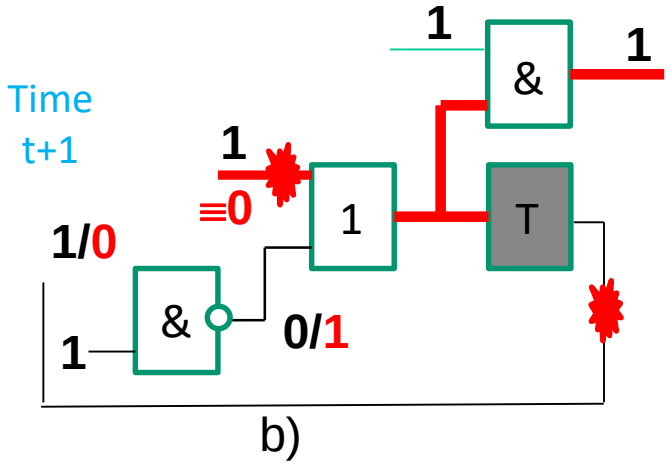
5.3. Rikete simuleerimine ja analüüs mäluga skeemides

Fault Simulation in Sequential Circuits

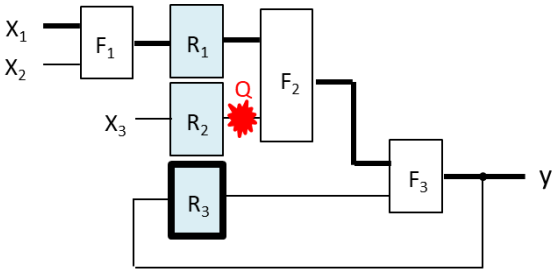
Time frame model:



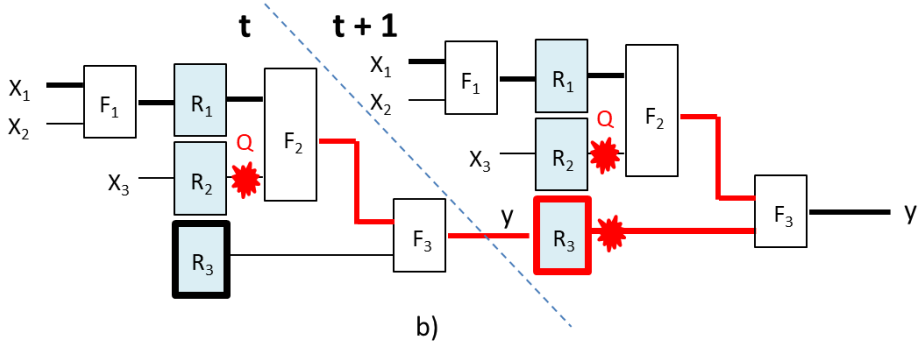




DFT-ased Critical Path fault Tracing



a)



b)

Critical path tracing problem:

•Why it is not possible in sequential circuits?

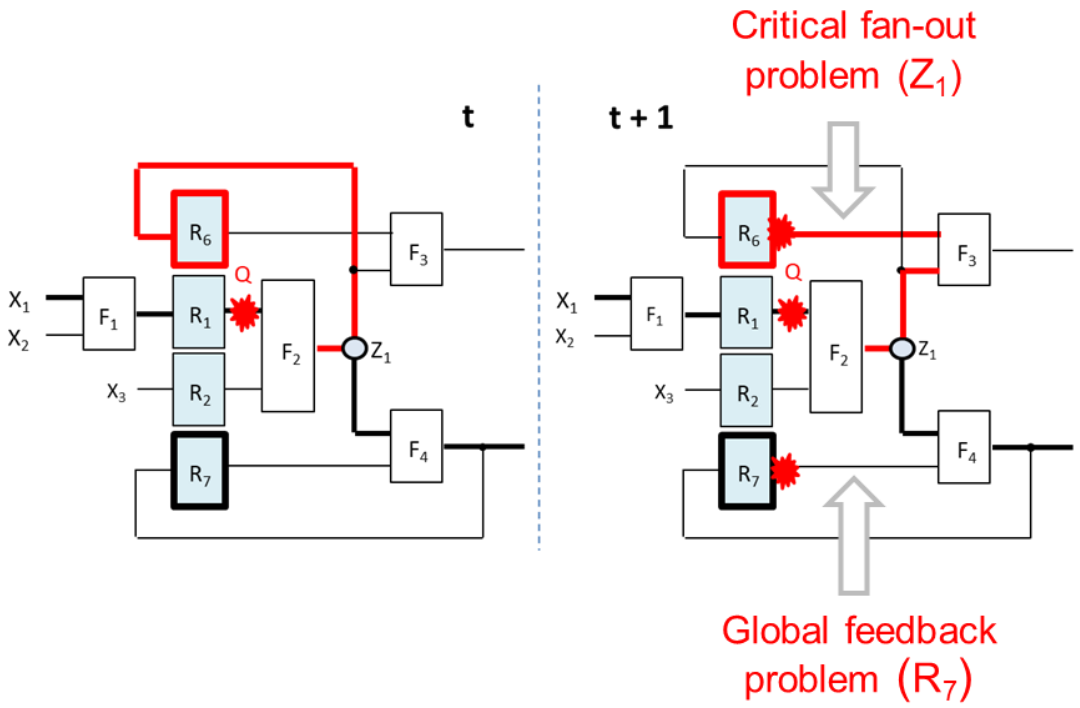
•Because the critical path backtracing is based on reasoning of correct signals in the circuit

Example:

- Two time-frames for simulating two patterns.
- Fault Q propagates from time frame ***t*** to ***t* + 1**, causing the wrong value in R3
- Hence, backtracing in frame ***t* + 1** is not any more possible

- **The sequential circuit has to be modified by cutting a subset of global feedbacks for inserting MISR**
- **For the remaining part of local feedbacks the critical path tracing method was updated for path tracing over diferent time frames.**

DFT-based fault simulation

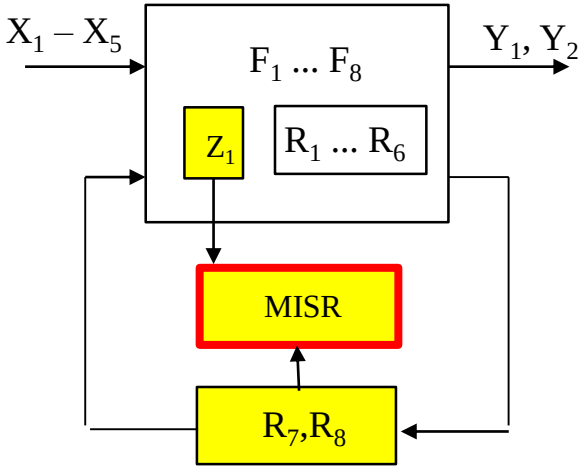


Another critical path tracing problem:

A fault may corrupt the signals in the next time frame

due to the fan-out,

from which the signals may converge via the paths over diferent time frames

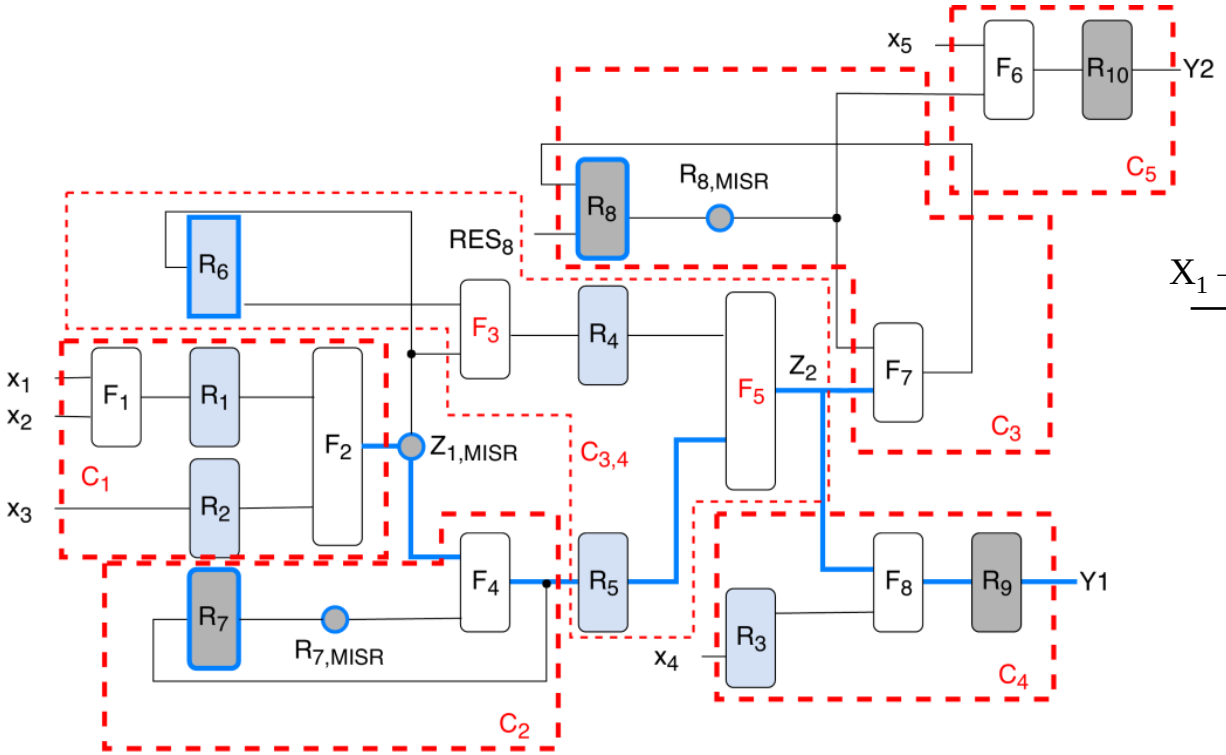


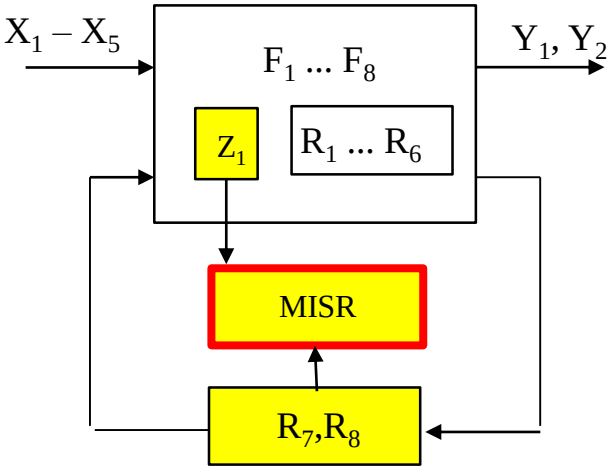
Solution: The global feedbacks, and the critical fan-outs should be made observable using MISR

DFT-based fault simulation

Example: A circuit as a Network with two global feedbaks to be cut, and the registers $R_{7,MISR}$ and $R_{8,MISR}$ should be made observable.

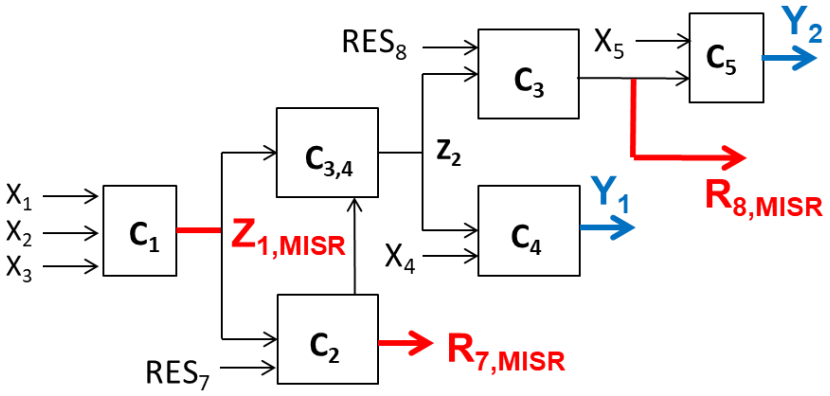
The fanout $R_{8,MISR}$ should be made observable as well.





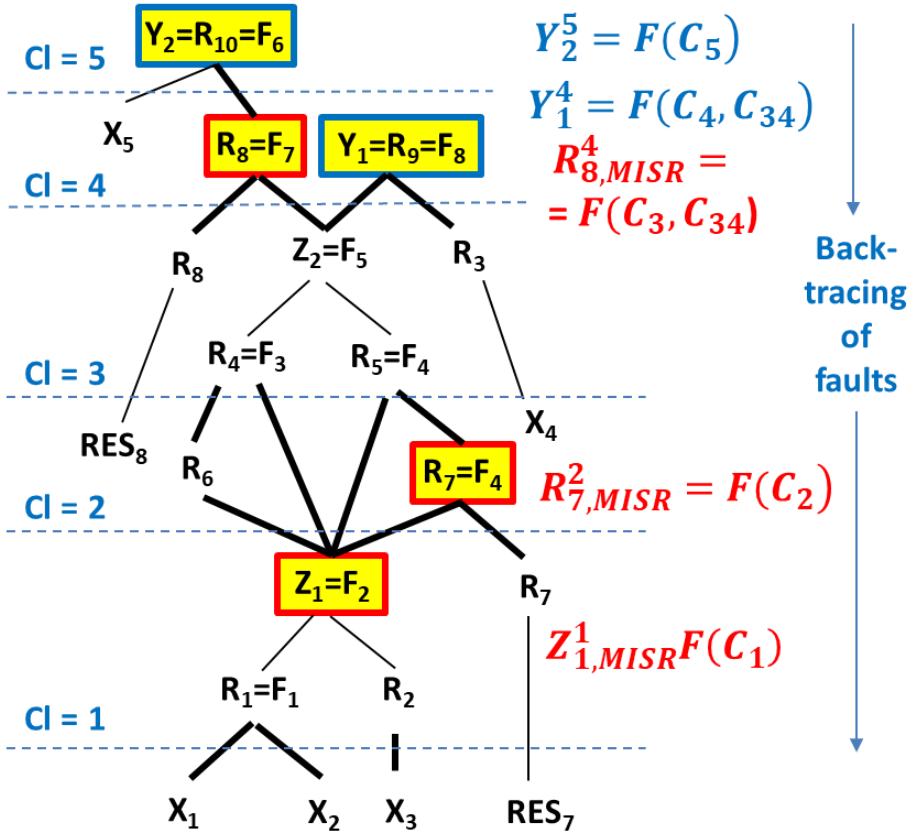
2a. DFT-based fault simulation

Solution: The full circuit is partitioned into the network of combinational subcircuits which can be backtraced (up to the observation points) in diferent time frames



Observation points: outputs **Y1** and **Y2**, and the nodes $R_{7,MISR}$, $R_{8,MISR}$, $Z_{1,MISR}$, connected to MISR

Example: Critical path fault tracing over 5 time frames similarly as in combinational circuits.



DFT-based fault simulation

Experiments: We compared the speed-up achieved by the **proposed method** in comparison with single fault simulation method (**FS**) in sequential circuits

Circuts	Number of faults F	SAF simulation time, s			Gain in speed, G
		Log Sim LS	FaultSim FS	**Proposed method**	
8a	112034	0.155	17365	30.0	579
8b	83940	0.152	12759	24.7	517
8be	99330	0.168	16687	62.1	269
8bk	86878	0.159	13814	25.2	548
8bs	100820	0.154	15526	173.4	90
8c	122386	0.159	19459	35.9	542
8d	123012	0.161	19804	35.5	558
8de	136876	0.164	22447	81.3	276

The benchmark family (pipe-lined signal processors) was created in the institut during the Project CEBE (2007-2015)

FS = LS * F

Gain: G = FS / Proposed method

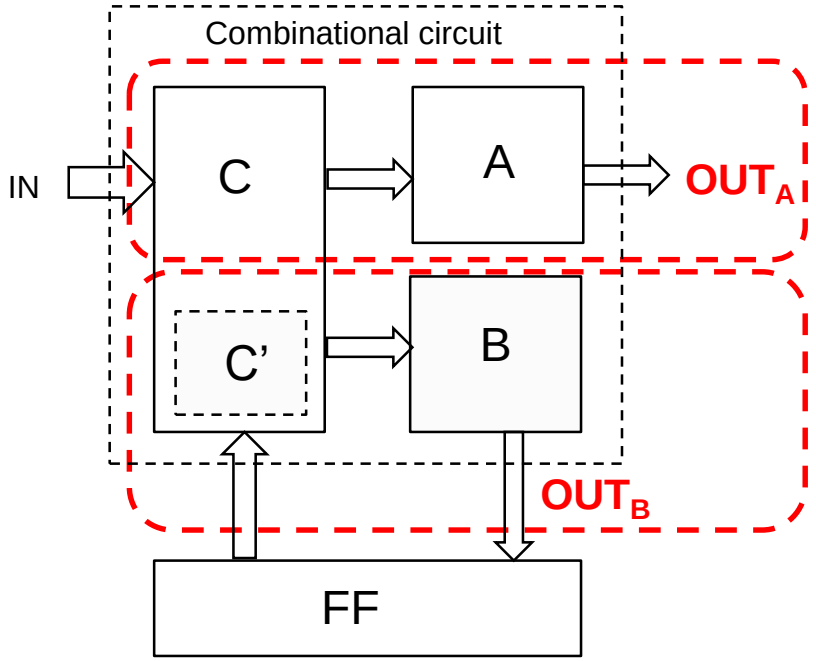
Span: 90 – 579 times

Fault dropping was not used to make the comparison fear with the critical path tracing approach, which produces all detected faults for all patterns simulated

The gain depends on the complexity of the circuit in terms of the (nested) reconvergent fan-outs

Joint parallel-sequential fault simulation

Exact parallel critical path backtracing of one group of faults is combined with sequential fault simulation of the remaining group of faults



Combinational part, where fast parallel critical path tracing to **OUT_A** is possible

Sequential part, where critical path tracing to **OUT_A** in a single clock is not possible

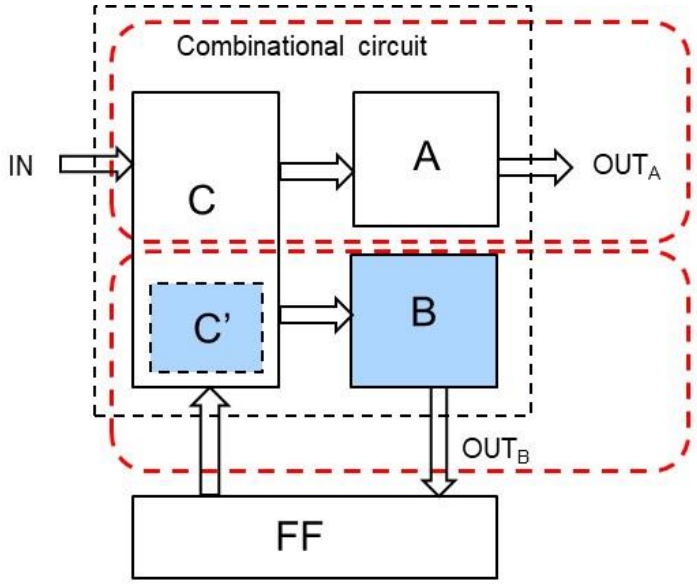
These faults have to be simulated one-by-one in sequence

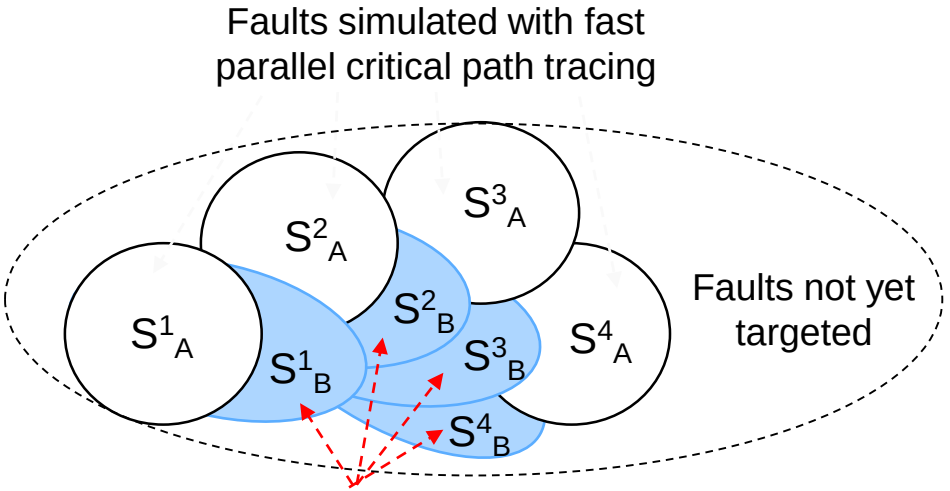
The faults in C may fall into the both parts

Share of the faults, falling into the parts, depends on the test patterns

Joint parallel-sequential fault simulation

The fault groups are separated on the simulation flow algorithmically, no DFT modification of the circuit is needed





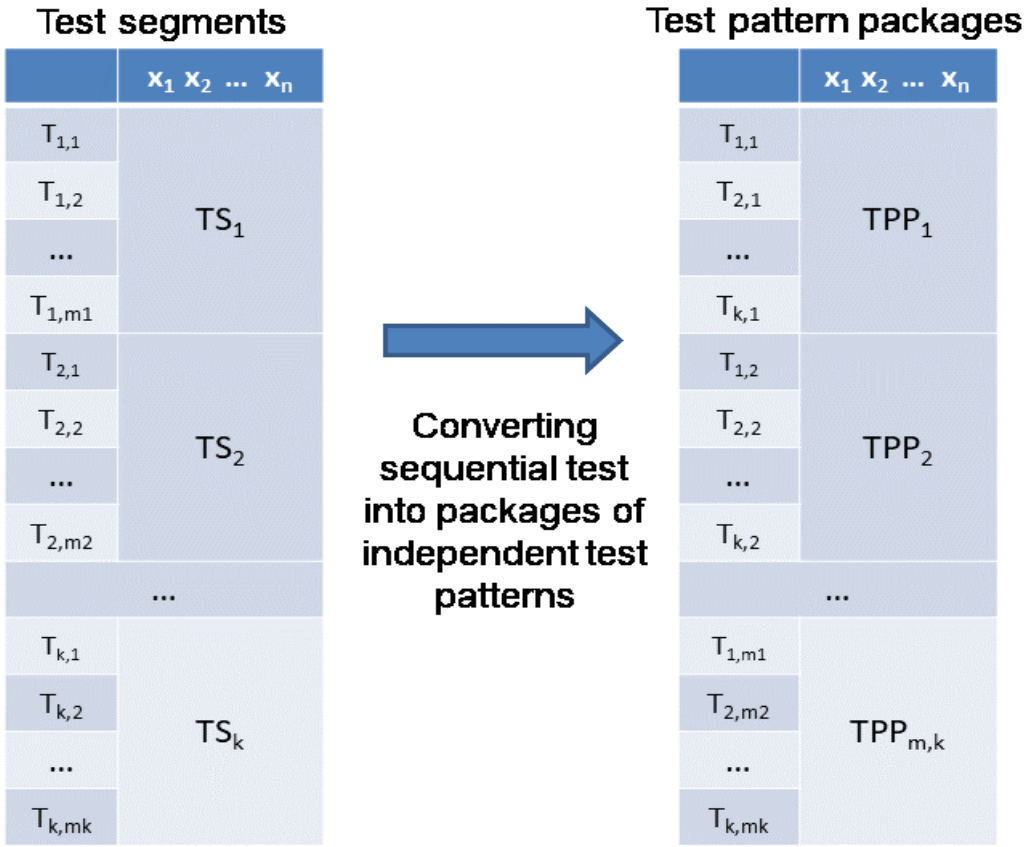
The current state of fault simulation can be expressed after the *k*-th pattern of the test sequence as:

$$S(A) = S(A_k) = S(A_{k-1}) \cup \left(S_A^k - S(B_{k-1})\right)$$

$$S(B) = S(B_k) = S(B_{k-1}) \cup \left(S_B^k - S(A_k)\right)$$

Joint parallel-sequential fault simulation

Before the fault simulation the set of test sequences is to be converted into the ordered independent pattern packages



Independent test pattern packages can be analyzed by fault backtracing in parallel

Joint parallel-sequential fault simulation

Solution: Parallel fault backtracking along the critical paths

Test pattern packages

Test	x_1 x_2 … x_n
$T_{1,1}$	TPP_1
$T_{2,1}$	
…	
$T_{k,1}$	
$T_{1,2}$	TPP_2
$T_{2,2}$	
…	
$T_{k,2}$	
…	
$T_{1,m1}$	TPP_k
$T_{2,m2}$	
…	
$T_{k,mk}$	

Fault simulation with parallel critical path tracing

Fault coverage table

Test		r_1r_2 … r_p	r_1r_2 … r_p
TPP_1	$T_{1,1}$	Fault cover S^1_A	Fault cover S^1_B
	$T_{2,1}$		
	…		
	$T_{k,1}$		
TPP_2	$T_{1,2}$	Fault cover S^2_A	Fault cover S^2_B
	$T_{2,2}$		
	…		
	$T_{k,2}$		
…			
TPP_k	$T_{1,m1}$	Fault cover S^k_A	Fault cover S^k_B
	$T_{2,m2}$		
	…		
	$T_{k,mk}$		

S^1_A: A part of faults in the sequential circuit can be simulated by exact parallel fault backtracing **fast**

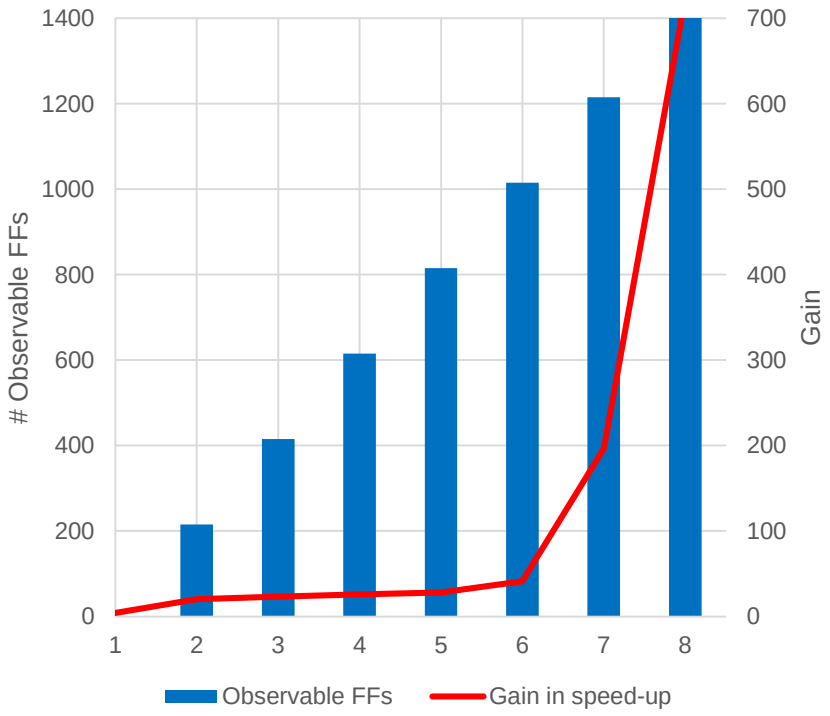
S^1_B: he rest of the faults have to be simulated by **slow** fault-by-fault simulation concept

Joint parallel-sequential fault simulation

Experimental results: The gain is between 1.9 and 4.2

#	Circuit	Faults #	Fault coverage of the simulated test sequence (%)			Time for critical path tracing (s)		Total time (s)		Gain
			s(A)	s(B)	FC	t_{CP}	t_{CL}	t_{NEW}	t_{OLD}	
1	s526	984	3.3	32.3	35.6	0.10	0.01	0.4	0.9	2.1
2	s713	1266	43.8	37.6	81.4	0.12	0.03	0.8	1.8	2.2
3	s1494	2864	37.9	20.2	58.1	0.15	0.07	1.5	6.3	4.2
4	b05	3952	2.2	32.2	34.4	0.21	0.16	4.5	13	2.9
5	s5378	9122	45.7	26.2	71.9	0.36	0.24	23	85	3.7
6	s9234	16756	2.1	30.1	32.2	0.58	0.36	91	300	3.3
7	s15850	29682	5.1	32.5	37.6	1.17	0.60	309	946	3.1
8	b15	36496	0.6	49.7	50.3	5.35	0.65	521	1036	2.0
9	s38417	69662	1.9	50.7	52.6	2.44	1.35	2516	4956	1.9
10	b17	129422	0.3	25.0	25.3	15.4	1.96	3226	12857	4.0

Joint parallel-sequential fault simulation



The possible simulation speed-up is defined by the observability of flip-flops

(investigations of the circuit b17)

- ✓ **The efficiency of the concept is ecplained by on-line classification of faults into two classes:**
  - **detectable by parallel path tracing,**
  - **to be simulated by traditional methods**
- ✓ **Speed-up of fault simulation was achieved 1.9 – 4.2 times**
- ✓ **High scalability was achieved**