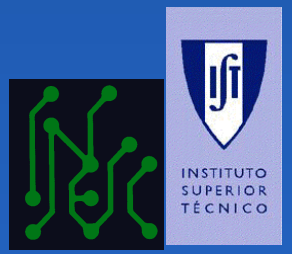
RTL-based Functional Test Generation for High Defects Coverage in Digital SOCs

M.B. Santos, F.M. Gonçalves, I.C. Teixeira, J.P. Teixeira

IST / INESC

Lisboa, Portugal

ETW 2000 May, 23-26, 2000



Purpose

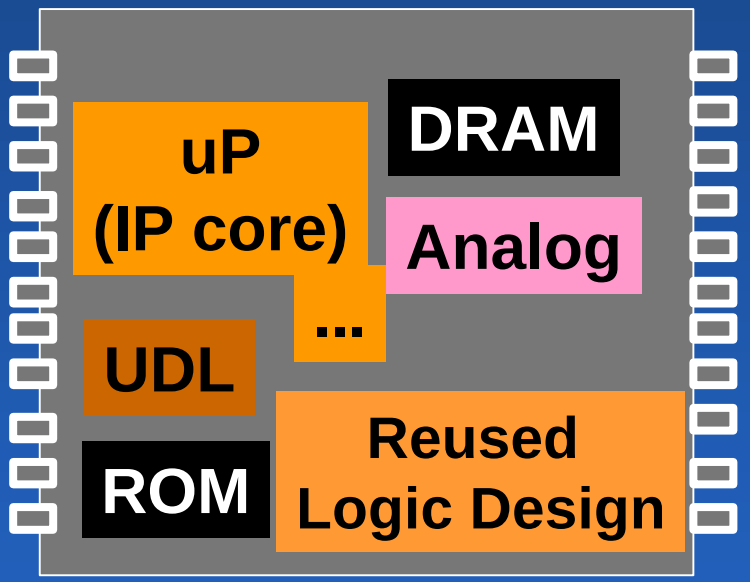
- **Propose a RTL-based test generation methodology for digital SOCs to be used for:**
  - **design validation**
  - **identification of “dark corners” of the functionality**
  - **generate a functional test which can be reused in structural testing (low energy, high defects coverage)**

Outline

- **Introduction**
- **RTL fault models**
- **Methodology**
- **Experimental Set-up**
- **Results**
- **Conclusions**

Introduction

- **HDL - concurrent D&T**
- **Test Reuse**
  - DFT
  - Design validation
  - Production test (HQ)
  - Lifetime test
  - Product / technology evolution



Introduction

Abstraction LEVEL	Fault Models
RTL	?
logic	LSA
Layout	BRI, Line-Open

- **Purpose: Defect detection, low cost, test reuse**
- **Classic approach: LSA based ATPG / test validation**
- **Can RTL functional test be reused in production test, to increase Defects Coverage ?**

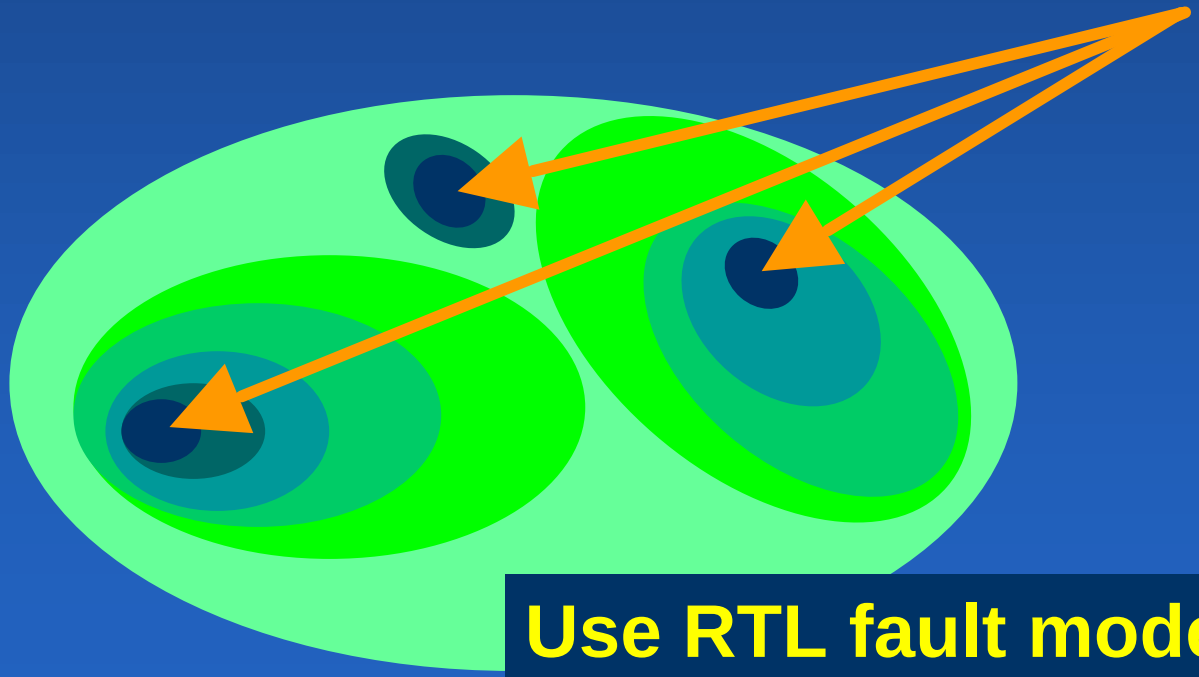
RTL Fault Models

RTL Fault Model	[1]	[2]	[3]	[4]	[5]	[6]
LSA type	X		X	X	X	**X**
Logic/ Arithmetic Operations		X	X		X	
Constant/ Variable Switch		X				
Null Statements	X	X			X	**X**
IF/ELSE	X	X	X	X	X	**X**
CASE	X	X	X	X		**X**
FOR	X	X		X		**X**

[1] Teresa Riesgo, UPM, 96
[2] Ch. Robach et al., ITC 96
[3] Haynes, Johnson, ITC 99
[4] D. Schiuto et al., ITC 98
[5] Ward, Armstrong, DAC 90
[6] our work

Methodology

Deeply embedded functionality:



- **“dark corners”**
- **few vector sequences in the input space able to exercise it**

Use RTL fault models to unveil them

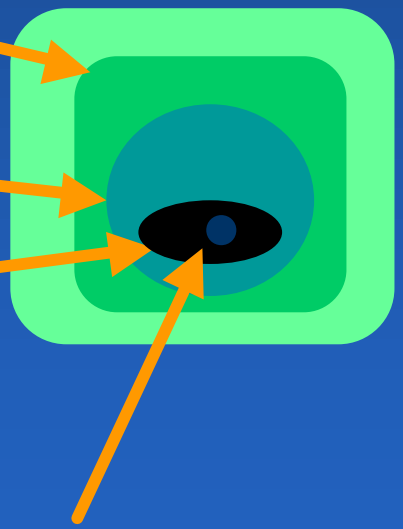
Methodology

- Approach:
  - identify “dark corners” - RTL conditions (e.g., IF, CASE) that highly restrict observability / controllability
  - dark corners must be:
    - **targeted by N vectors RTL fault activation / detection, or**
    - **reported as design errors**
- Method:
  - Partial definition of test vectors - ***masks*** - which are the key to “illuminate” each dark corner

Methodology

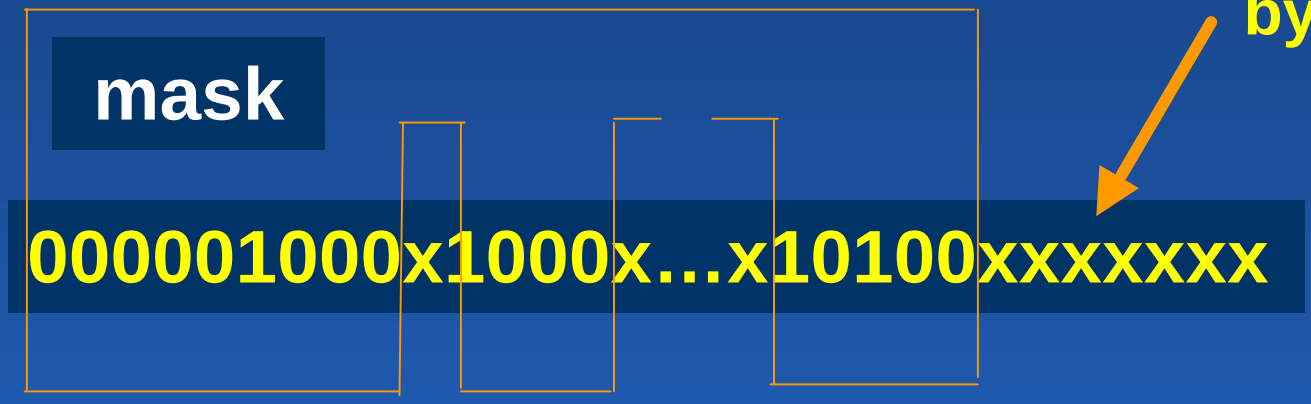
- **Example:**

```
if ( {pdb2[23:16], pdb2[7:5]} == 11'b00000100_101  )
00000100xxxxxx...xx101xxxxxxxxx
        ……..
         if (pdb2[15])==0  /* if W =1 */
000001000xxxxxx...x101xxxxxxxxx
                  ……..
                  if (  pdb2[4:2] == 3'b000 )
000001000xxxxxx...x101000xxxxxx
                           ……..
                           if (pdb2[13:10] == 4'b1000 )
000001000x1000x...x101000xxxxxx
```



Methodology

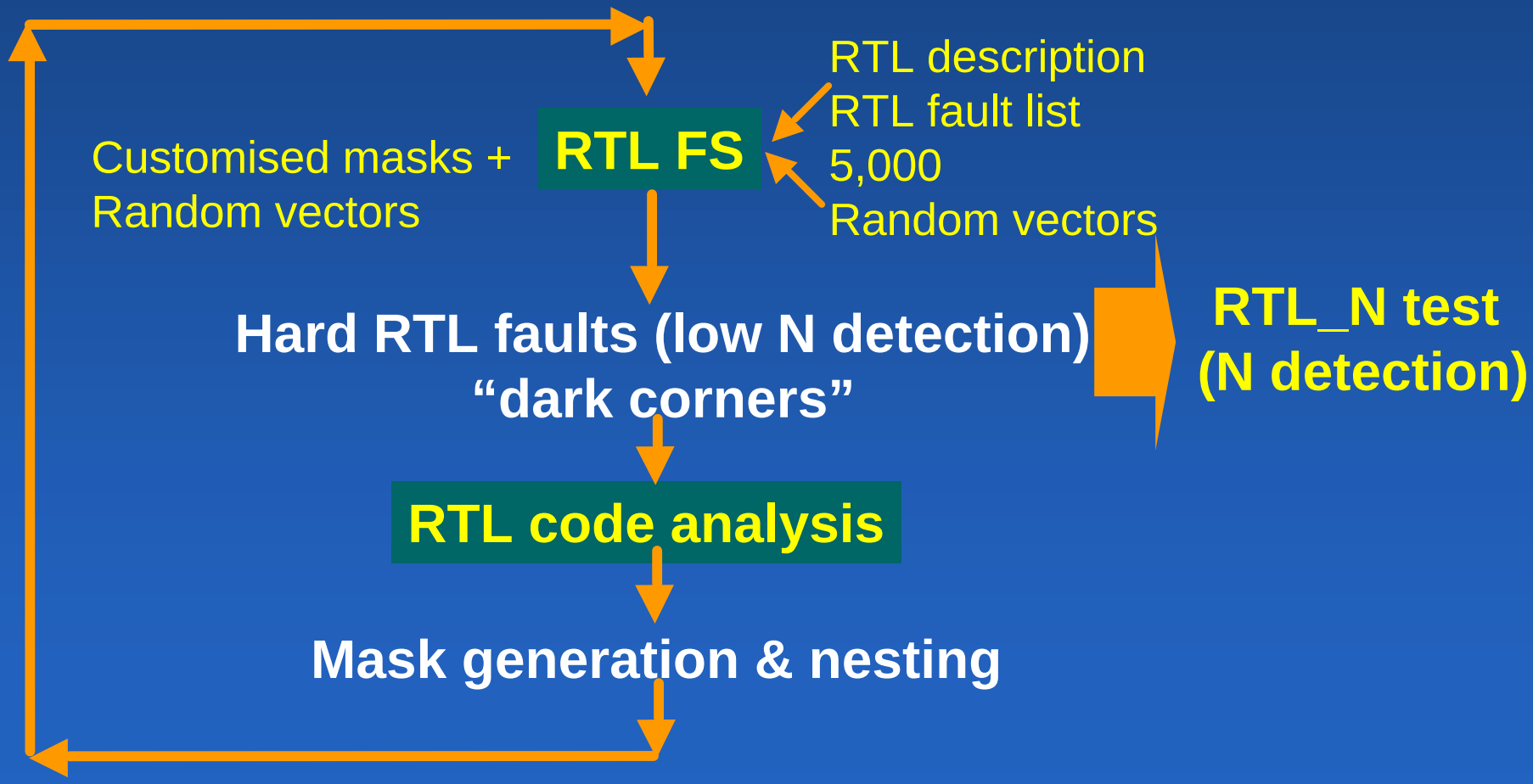
Customized mask vectors: x randomly substituted by 0s and 1s



m_i fixed positional bits
(m_i << n, n input bit words)

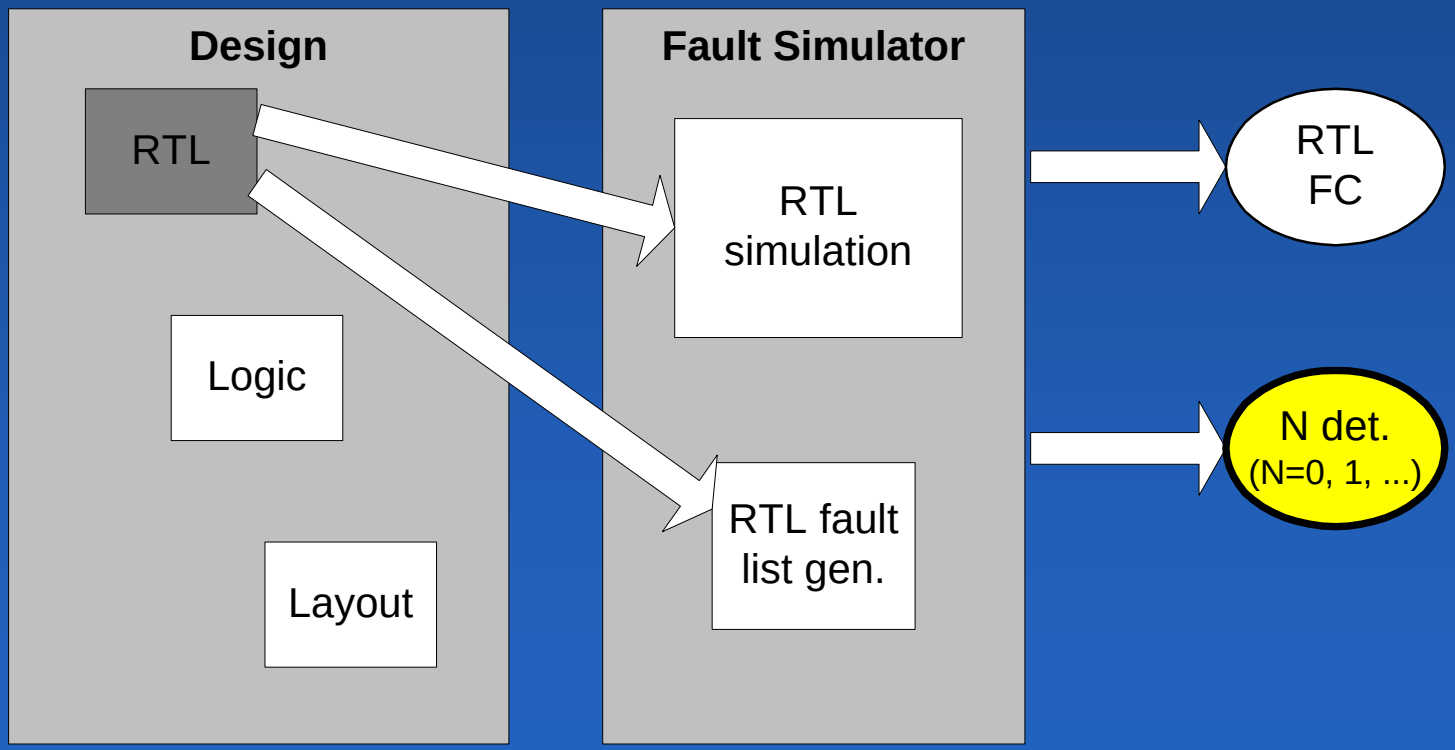
RTL_N test pattern: N detection of RTL faults, complete branch coverage

Methodology



Experimental Set-up

- RTL fault simulation



Experimental Set-up

- Requirements for methodology validation:
  - Realistic Fault Simulator
    - Multi-level: RTL + logic
    - compatible with commercial EDA

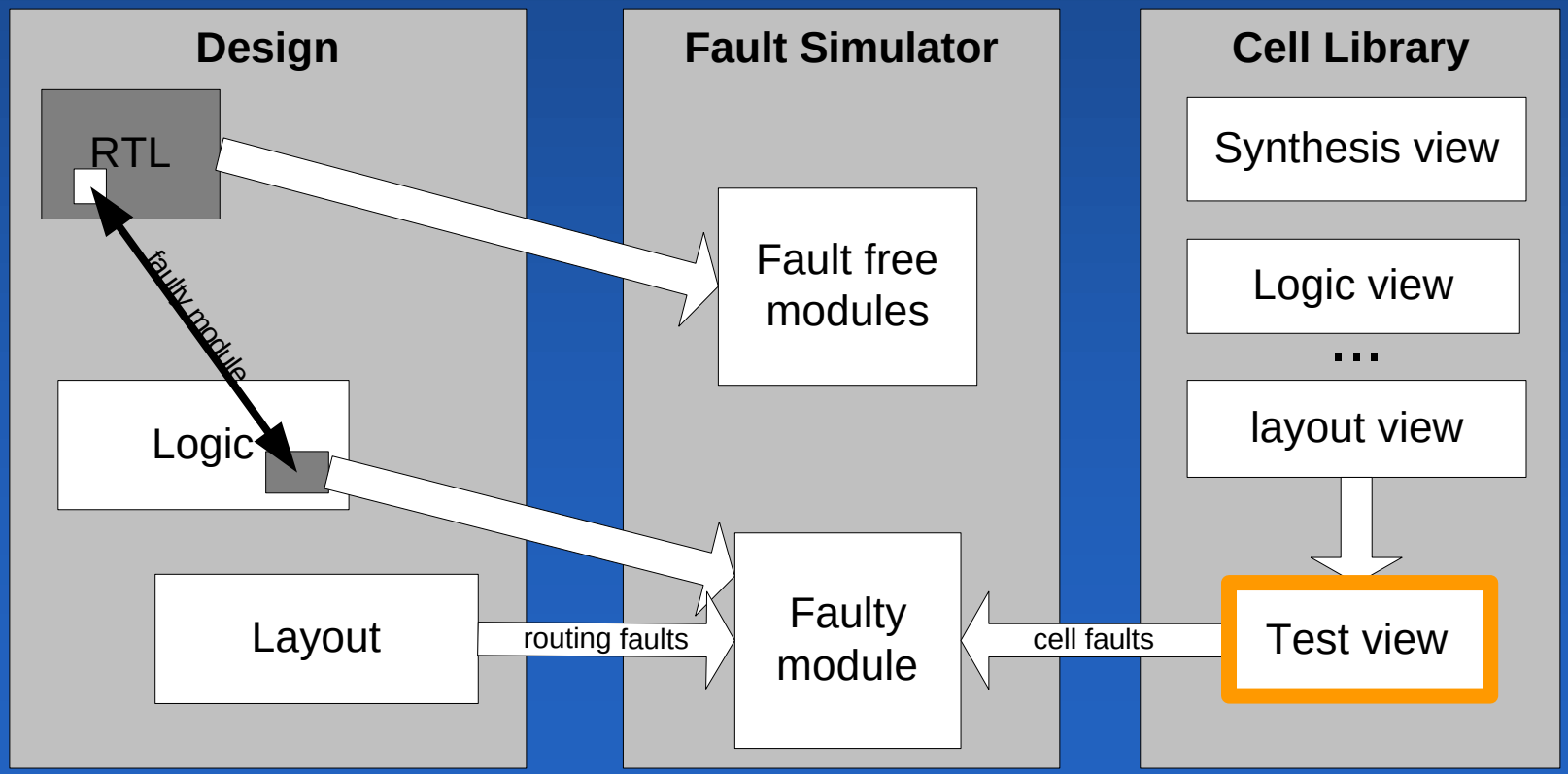
VeriDOS

  - Realistic Fault Models
    - Accurate fault extraction

lobs

Experimental Set-up

- Multi-level realistic fault simulation

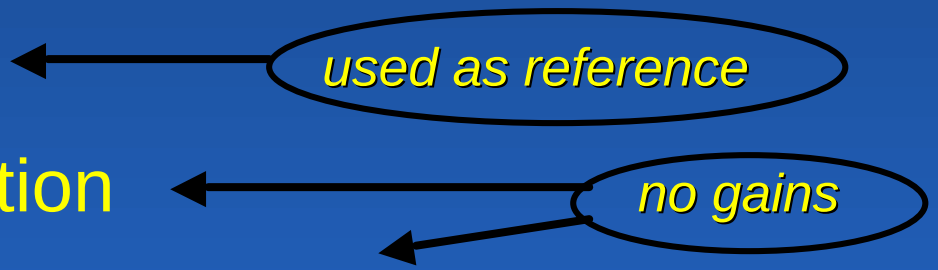


Experimental Set-up

- Case studies:
  - ITC99 benchmarks - CMUDSP and TORCH
  - CMUDSP: Add. Gen. Unit Controller (AGU)*, Prog. Control Unit Controller (PCU);
  - TORCH: co-processor control (cp0control), multiply or add PPSum module
- *Different structural synthesis:
  - area optimization (AREA)
  - critical path optimization (TIME)

Experimental Set-up

- Realistic Fault Simulation *with...*
- Vector Generation:
  - LSA + random
  - RTL faults detection
  - LSA + RTL faults not detected by LSA vectors
  - **N-detection: RTL faults multiple detection**



Results

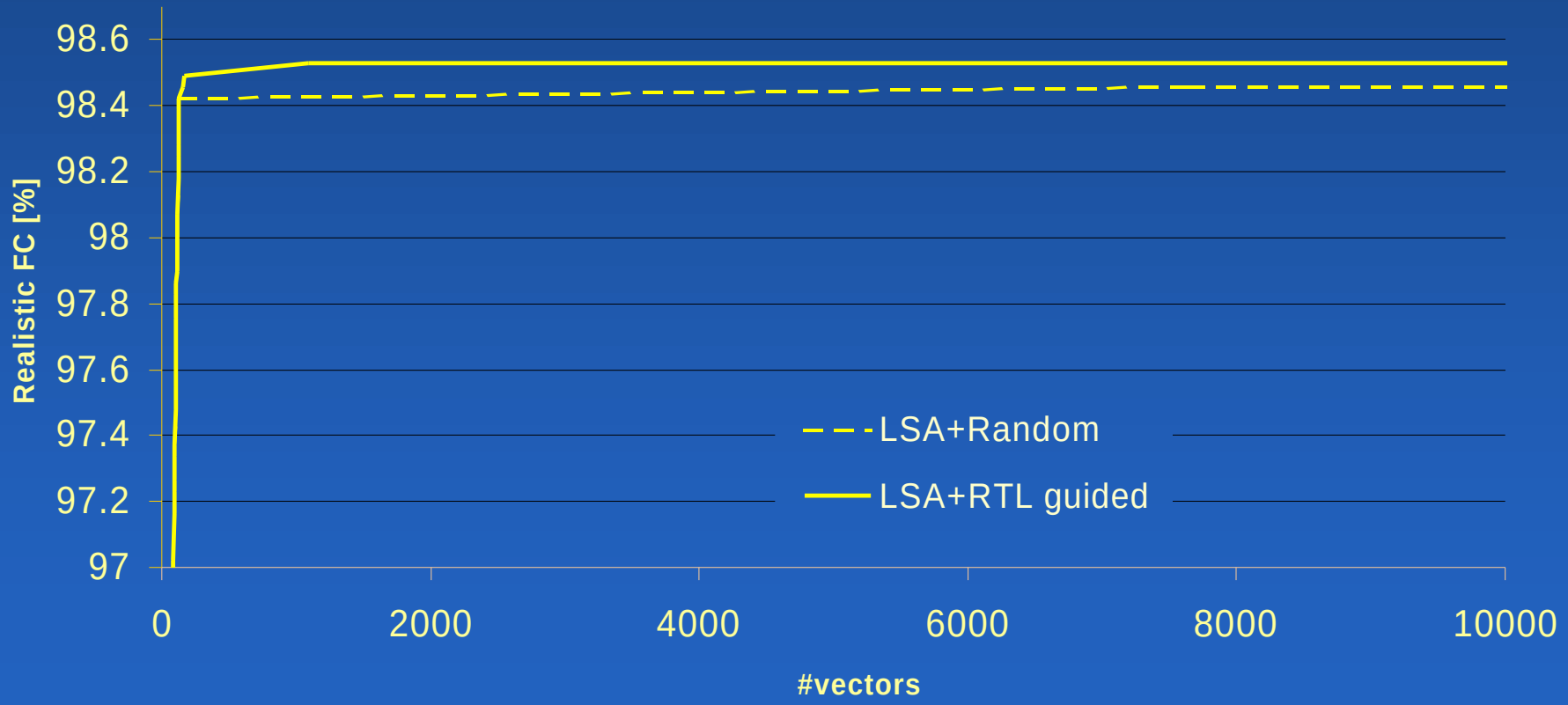
Circuit: CMUDSP Module: AGU control (AREA)

- RTL
  - # lines = 4000
  - # faults= 600
  - RTL FC = 95%
  - undetected:
    - bit sa - 2.2%
    - case dead - 1.7%

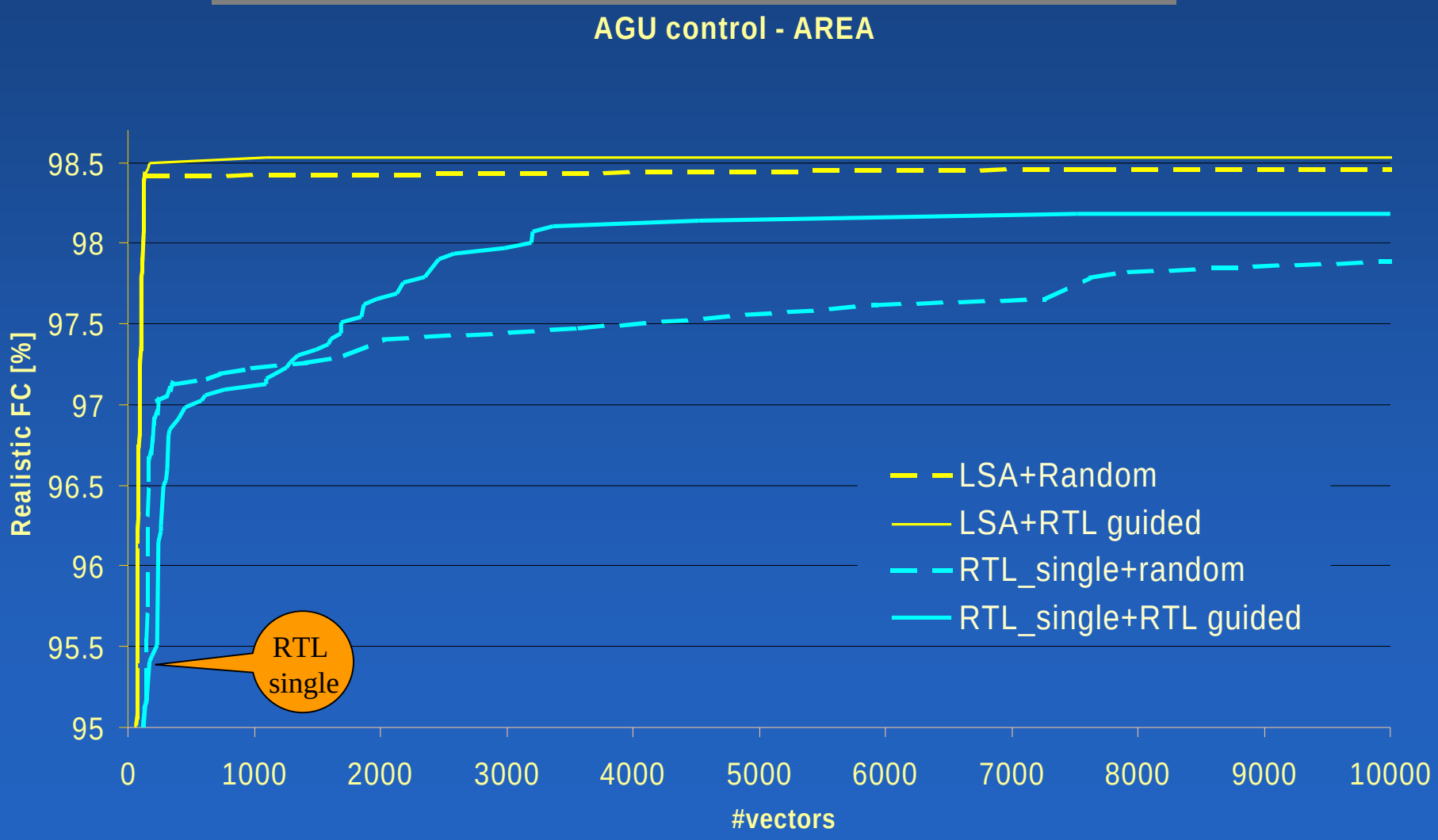
14 masks

- Logic
  - # gates = 466
  - # LSA vectors = 129
- Layout
  - # faults = 2853
  - undetected:
    - BRI AY - 0.6%
    - BRI AB - 0.5%

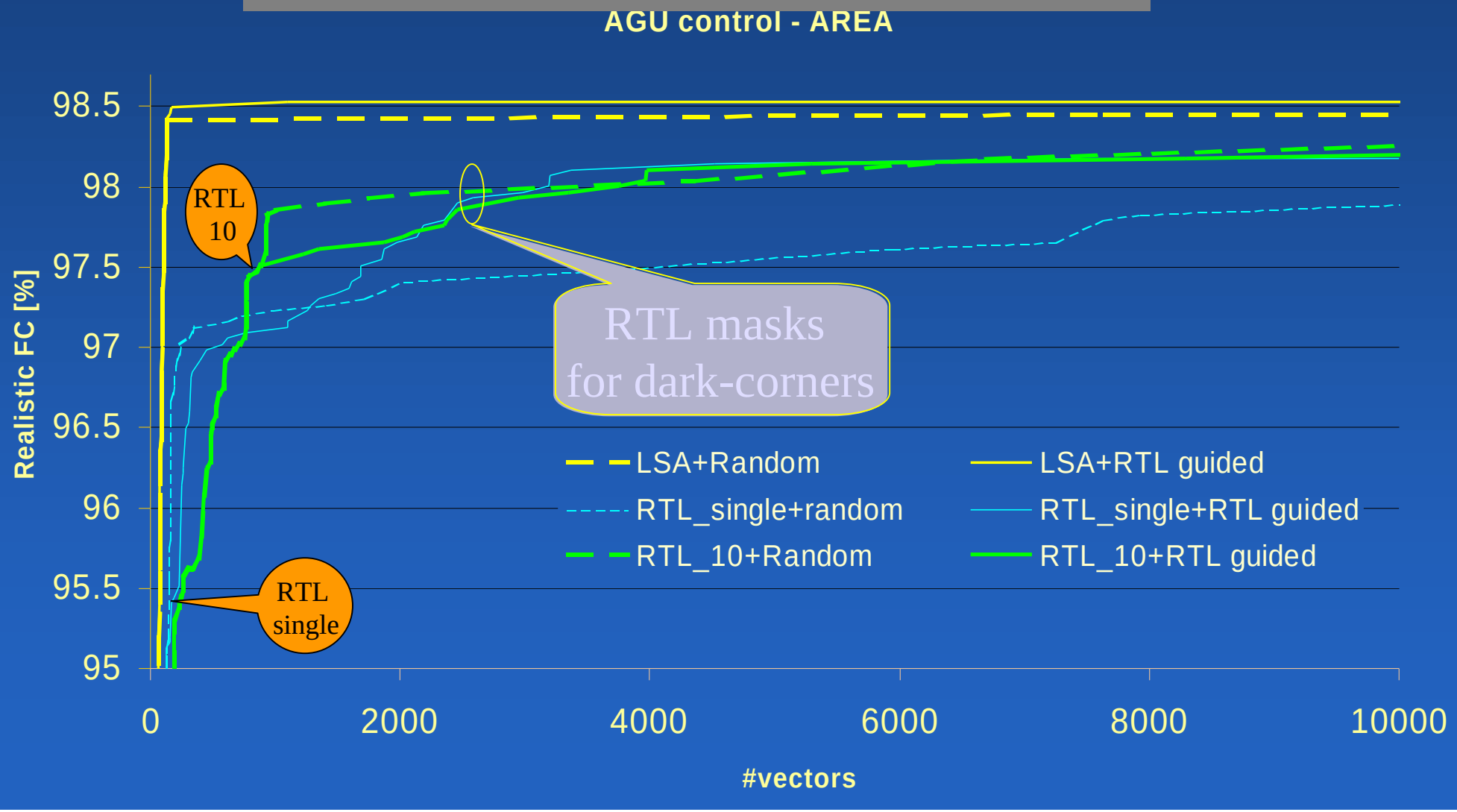
Results


AGU control - AREA
98.6
98.4
98.2
98
97.8
97.6
97.4
97.2
97
Realistic FC [%]
LSA+Random
LSA+RTL guided
0
2000
4000
6000
8000
10000
#vectors

Results


AGU control - AREA
Realistic FC [%]
98.5
98
97.5
97
96.5
96
95.5
95
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
#vectors
LSA+Random
LSA+RTL guided
RTL_single+random
RTL_single+RTL guided
RTL single

Results


AGU control - AREA
Realistic FC [%]
98.5
98
97.5
97
96.5
96
95.5
95
0
2000
4000
6000
8000
10000
#vectors
RTL 10
RTL single
RTL masks for dark-corners
LSA+Random
LSA+RTL guided
RTL_single+random
RTL_single+RTL guided
RTL_10+Random
RTL_10+RTL guided

Results

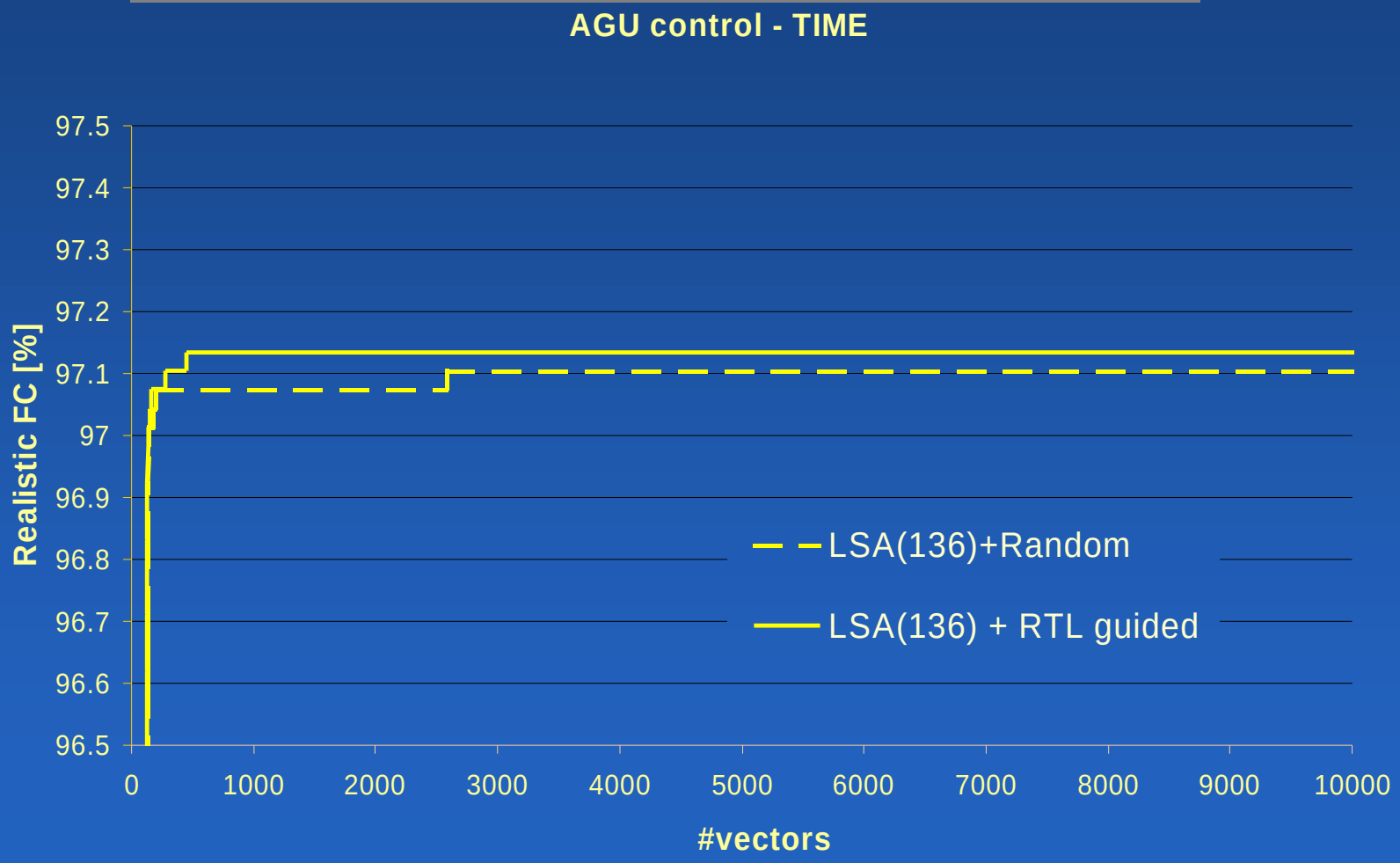
Circuit: CMUDSP Module: AGU control (TIME)

- RTL
  - # lines = 4000
  - # faults= 600
  - RTL FC = 95%
  - undetected:
    - bit sa - 2.2%
    - case dead - 1.7%

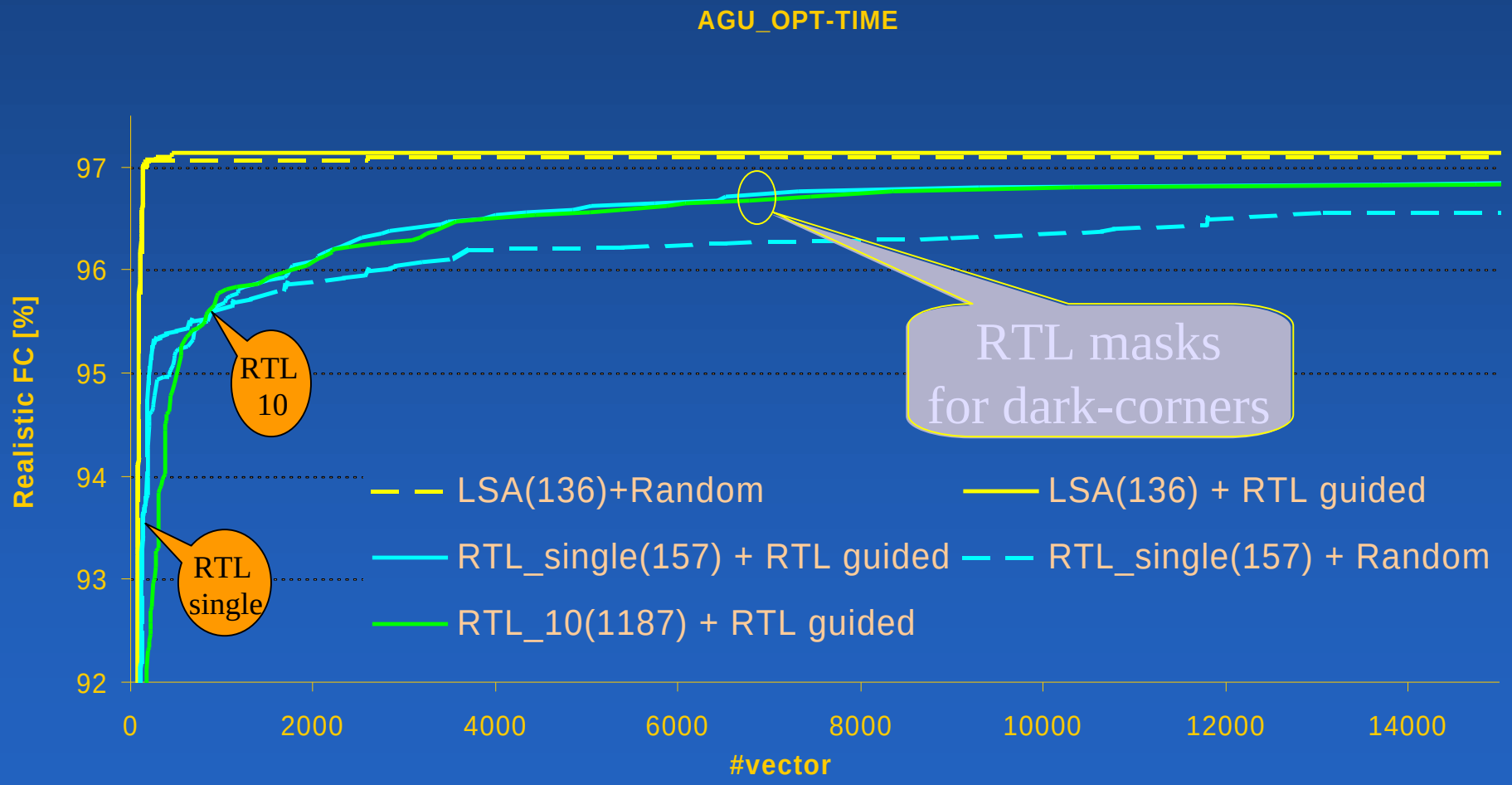
14 masks

- Logic
  - # gates = 580
  - # LSA vectors = 136
- Layout
  - # faults = 3315
  - undetected:
    - BRI AB - 1.5%
    - LOP AY, LOP VDD-X, BRI AY - 0.5%

Results

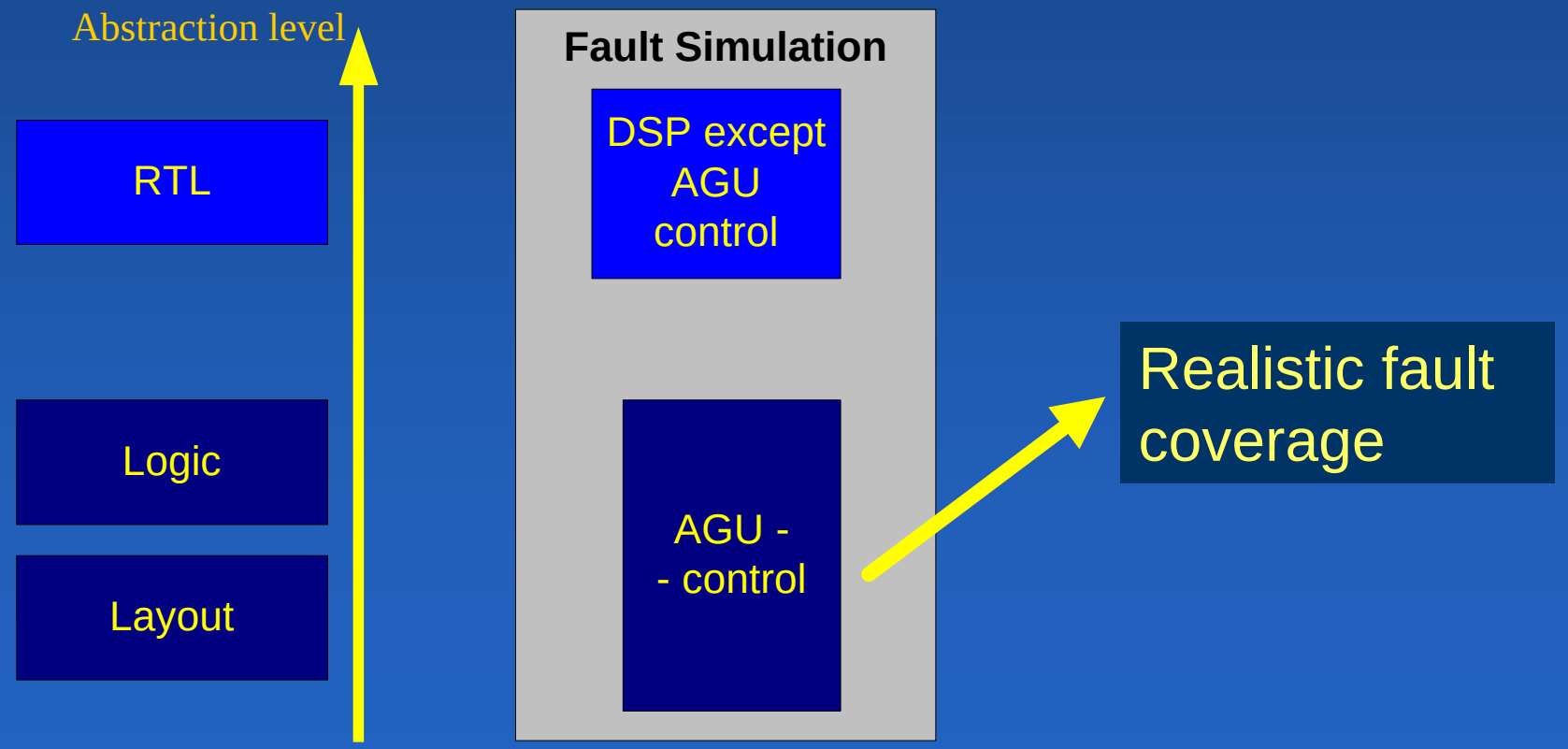

AGU control - TIME
Realistic FC [%]
97.5
97.4
97.3
97.2
97.1
97
96.9
96.8
96.7
96.6
96.5
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
#vectors
LSA(136)+Random
LSA(136) + RTL guided

Results

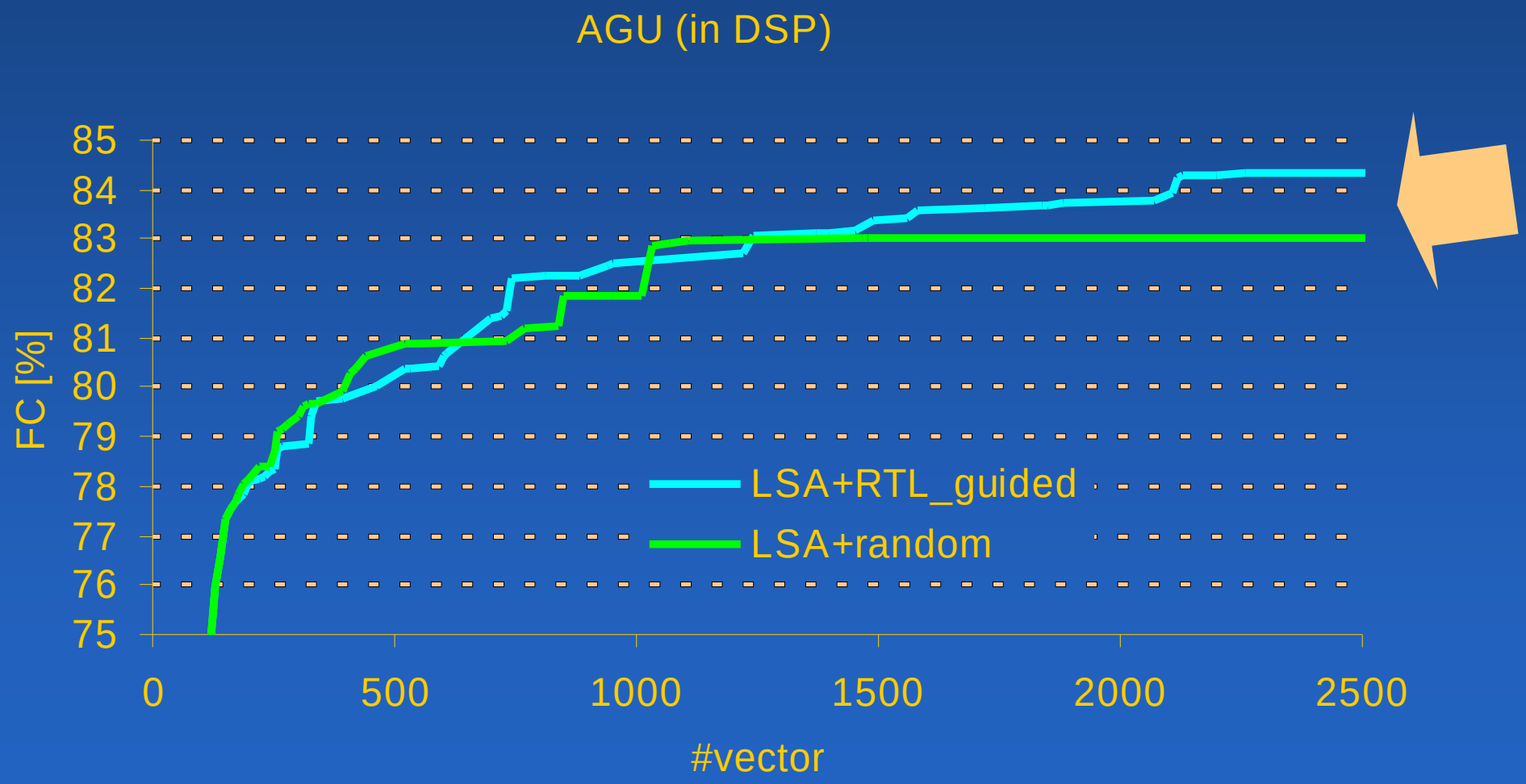

AGU_OPT-TIME
Realistic FC [%]
97
96
95
94
93
92
0
2000
4000
6000
8000
10000
12000
14000
#vector
RTL 10
RTL single
RTL masks for dark-corners
LSA(136)+Random
LSA(136) + RTL guided
RTL_single(157) + RTL guided
RTL_single(157) + Random
RTL_10(1187) + RTL guided

Results

- AGU-control fault simulation in DSP



Results


AGU (in DSP)
FC [%]
85
84
83
82
81
80
79
78
77
76
75
0
500
1000
1500
2000
2500
#vector
LSA+RTL_guided
LSA+random

Results

Circuit: CMUDSP Module: PCU control (BIST)

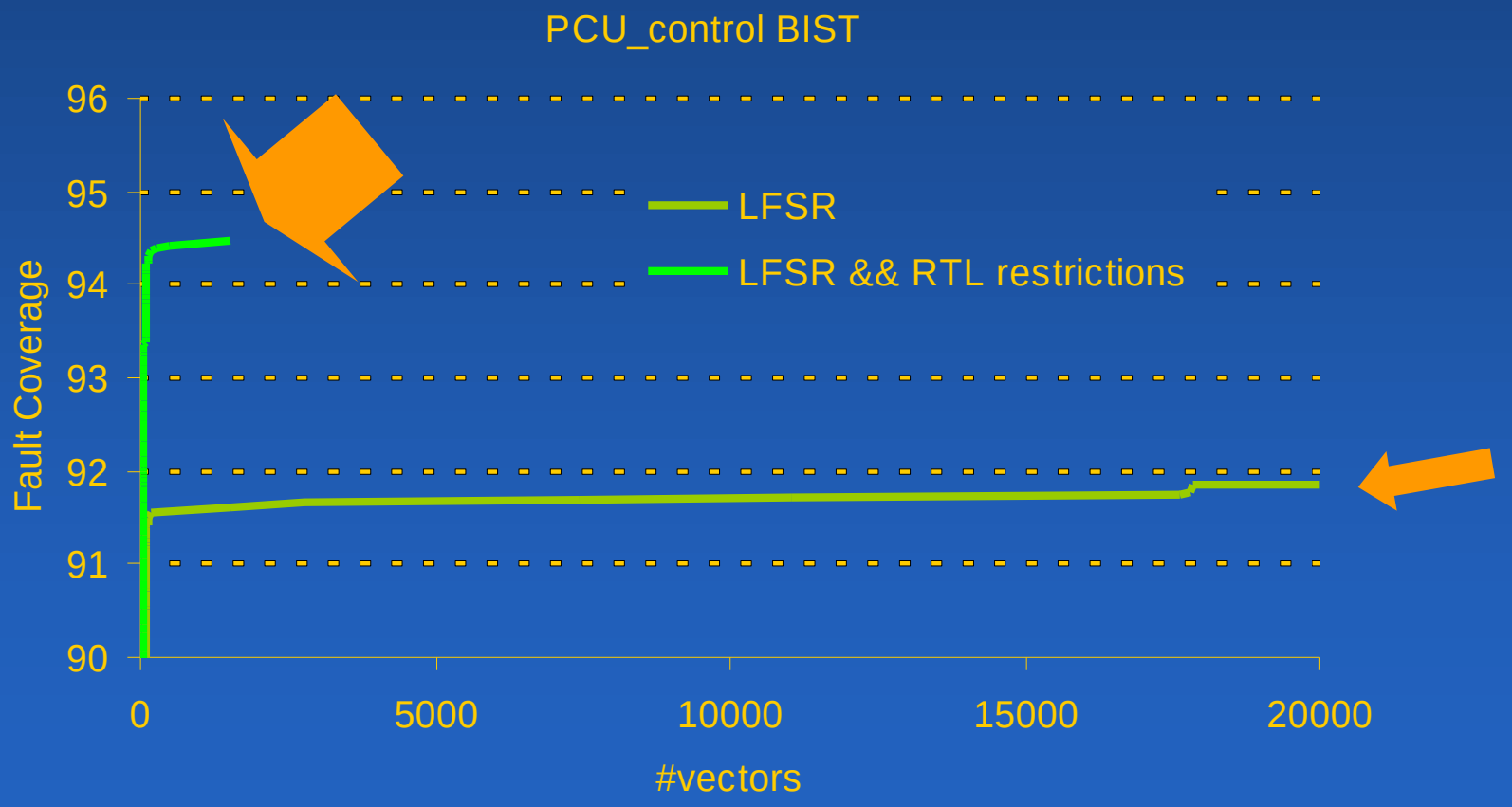
- RTL
  - # lines = 714
  - # faults= 972
  - RTL FC = 97%
  - undetected:
    - bit sa0 - 2.5%
    - if sa false - 0.5%
- Logic
  - # gates = 640
- Layout
  - # faults = 3489

6 masks:

4 x 51/350 deterministic bits +

2 x 19/350 deterministic bits

Results


PCU_control BIST
96
95
94
93
92
91
90
Fault Coverage
LFSR
LFSR && RTL restrictions
0
5000
10000
15000
20000
#vectors

Results

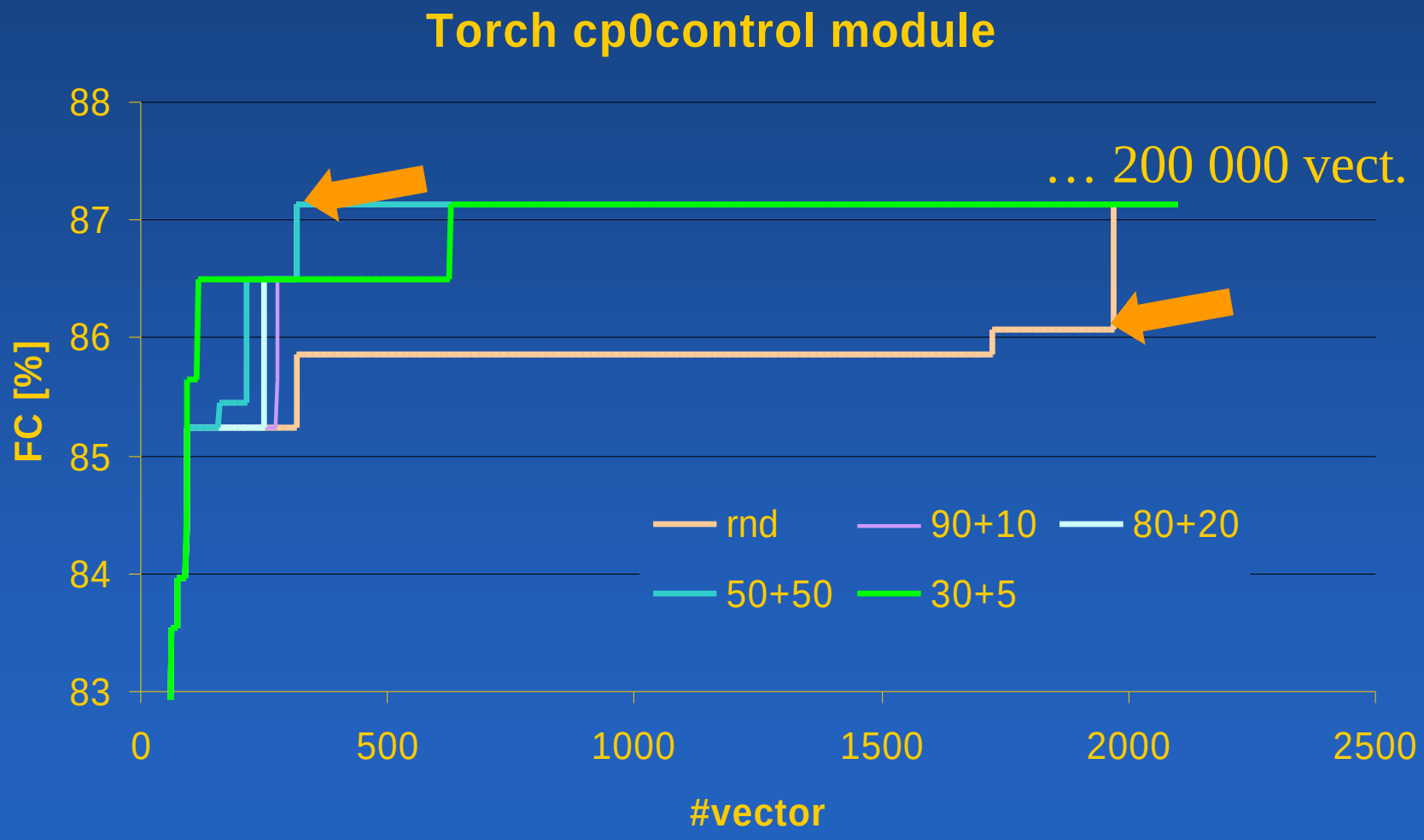
Circuit: Torch **Module: cp0 control**

- RTL
  - # lines = 503
  - # faults= 110
  - RTL FC = 100%

2 masks

- Logic
  - # gates = 100
- Layout
  - # faults = 474
  - undetected:
    - BRI AY - 6.8%
    - LOP AY - 2.5%
    - BRI YX - 1.7%
    - BRI AB - 0.6%

Results


Torch cp0control module
88
87
86
85
84
83
FC [%]
… 200 000 vect.
rnd
90+10
80+20
50+50
30+5
0
500
1000
1500
2000
2500
#vector

Results

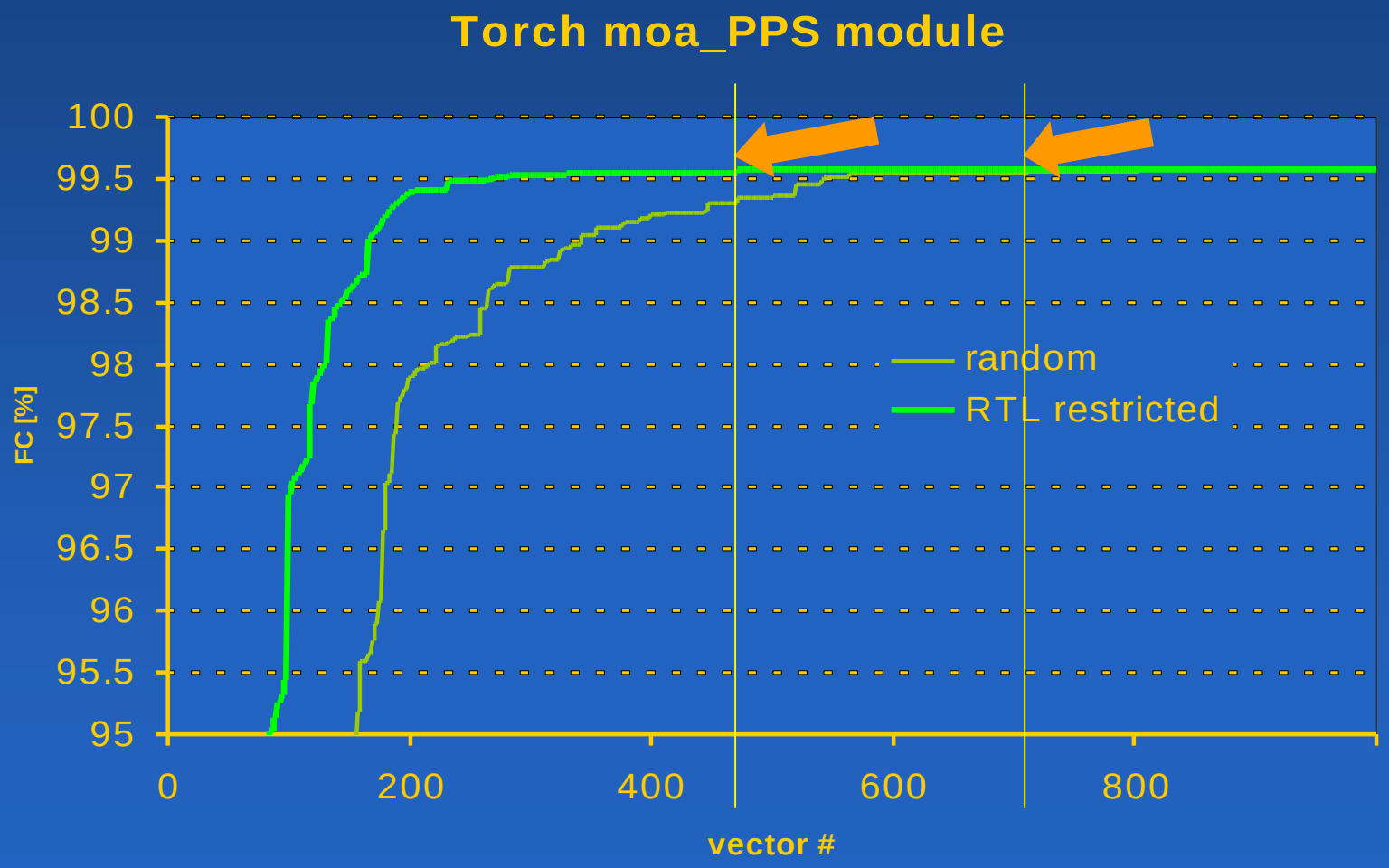
Circuit: Torch **Module: MOA PPsum**

- RTL
  - # lines = 96
  - # faults= 9 912
  - RTL FC = 99.5%
  - undetected:
    - bit sa - 0.5%

3 masks

- Logic
  - # gates = 13 200
- Layout
  - # faults = 39 682
  - undetected:
    - BRI AY - 0.1%
    - BRI YX - 0.1%
    - LOP AY - 0.1%

Results


Torch moa_PPS module
100
99.5
99
98.5
98
97.5
97
96.5
96
95.5
95
FC [%]
0
200
400
600
800
vector #
random
RTL restricted

Conclusions

- **single RTL fault detection does not improve LSA + random**
- **RTL fault models can be used for multiple purposes:**
  - **Design errors diagnosis**
  - **Dark corners identification**
  - **> useful in verification, test and diagnostic**

Conclusions

- **Multiple (N) RTL fault detection in dark corners leads to:**
  - **functional test generation for high DC**
  - **test application cost minimization**
  - **test reuse (weak structural dependence)**
  - **amenable to HQ, low energy BIST**