Physical Verification Flow for Hierarchical Analog IC Design Constraints

Volker Meyer zu Bexten
Markus Tristl

Göran Jerke

Hartmut Marquardt
Dina Medhat

SPONSORED BY THE
Federal Ministry of Education and Research

Portions of this contribution have been funded as part of the *ResCar 2.0* project (project label 01 M 3195) within the research program *ICT 2020* by the German Federal Ministry of Education and Research (BMBF).

RES 2.0 CAR

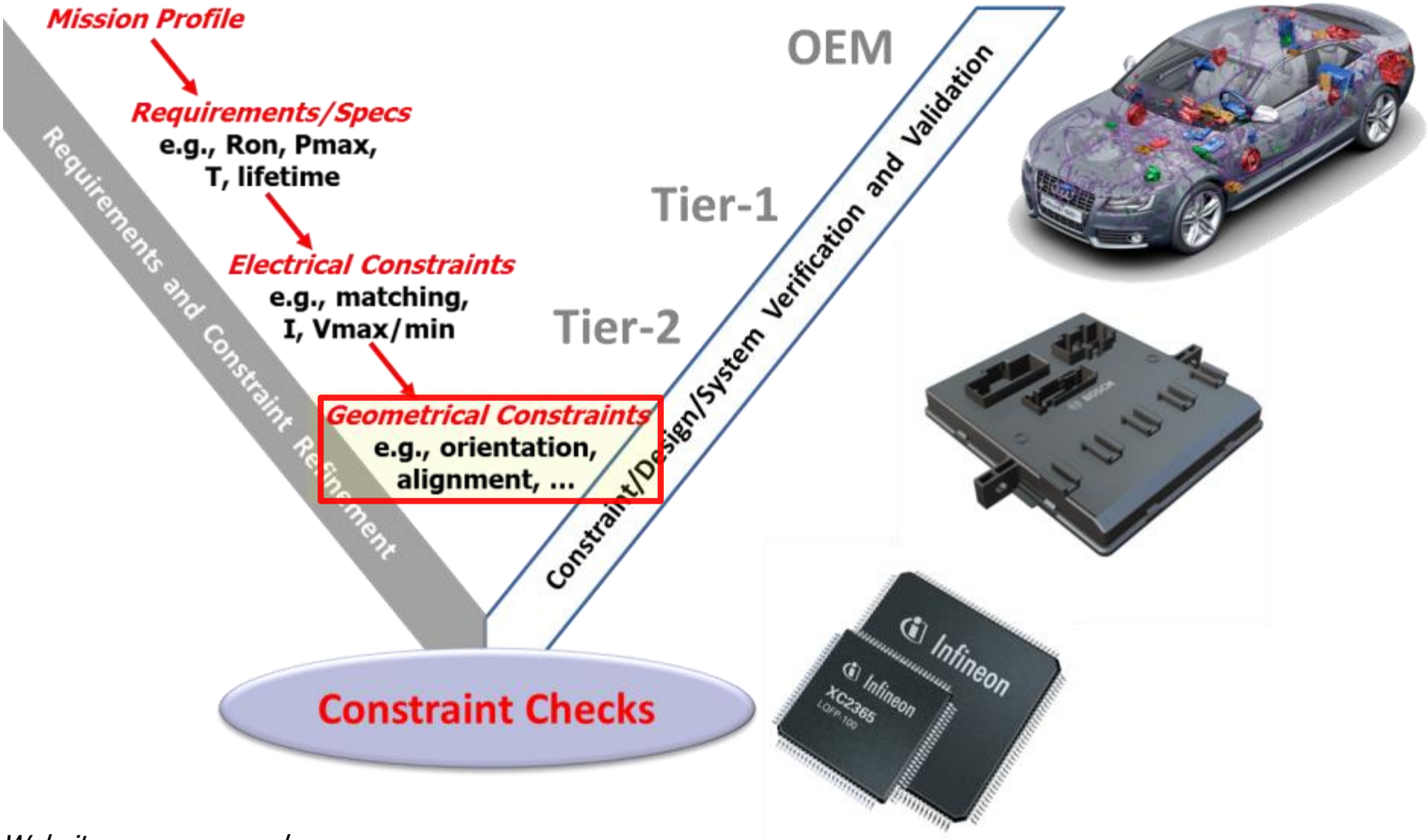
Contents

- Introduction
- Previous Work
- Verification Flow
- Constraint Derivation
- Constraint Types
- Constraint Verification
- Experimental Results
- Summary & Conclusions

Motivation

- Design of automotive ICs must satisfy challenging industry requirements for functional safety, functional robustness, and reliability
- Consistent derivation and verification of hierarchical design constraints is crucial for high-quality AMS ICs
- Standards like ISO 26262 and automotive design chain optimization require:
  - Documentation of requirements and mission profiles
  - Derivation of design constraints from requirement level
  - Rigorous enforcement of constraint compliance
  - Mask-level and independent verification

The ResCar 2.0 approach

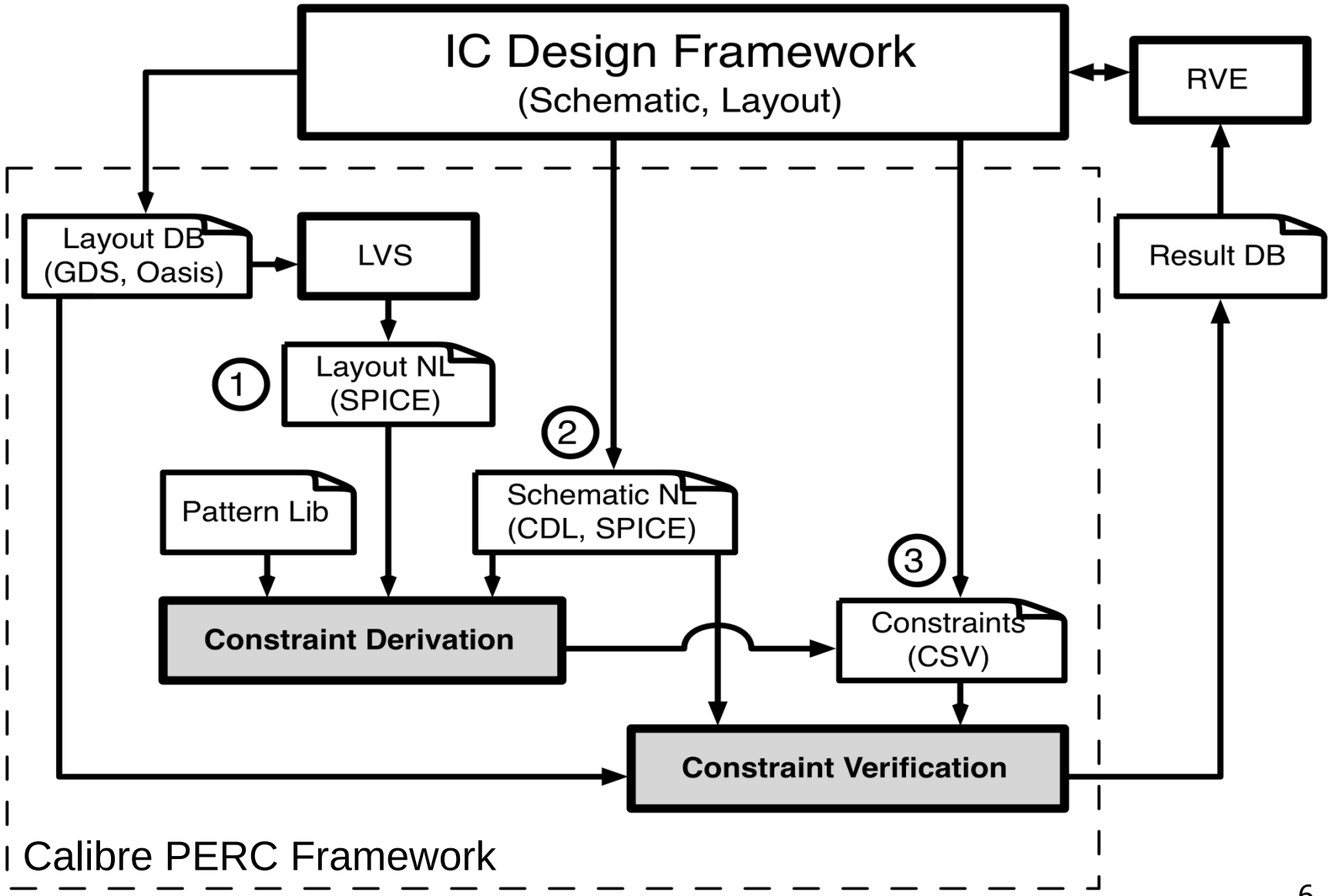


Website: www.rescar.de

Previous Work

- The *constraint-driven* design paradigm [cf. Chang et al. 1996] has its roots in efforts to establish analog layout synthesis in the 1990s
- Several parts are implemented in commercial tools (e.g. by Cadence Design Systems, Mentor Graphics, Synopsys)
- Major components:
  A. Constraint input/derivation
  B. Support of layout implementation
  C. Constraint compliance verification
- This presentation covers improvements in A) and C)

Verification Flow



Constraint Derivation

- Constraint members are found by pattern matching

patterns in netlist syntax

```
* p type current mirror
.SUBCKT cmp in out pwr
M0 in in pwr pwr P
M1 out in pwr pwr P
.ENDS
```

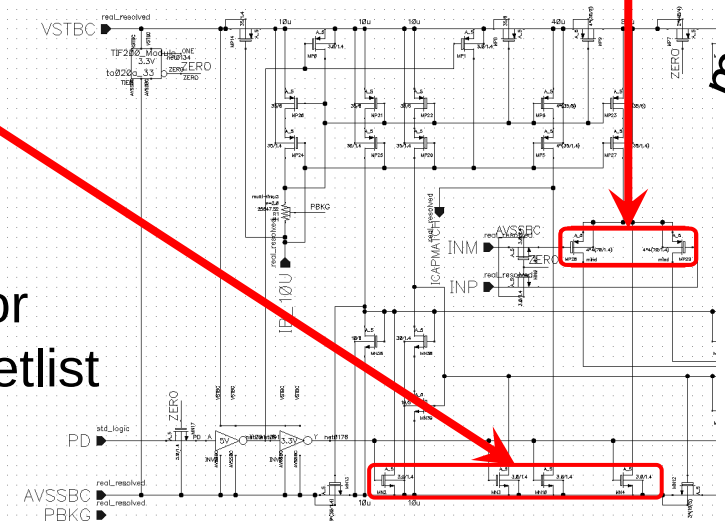
...

```
* n type differential pair
.SUBCKT dpn in1 in2 out1 out2 a gnd
M0 out1 in1 a gnd N
M1 out2 in2 a gnd N
.ENDS
```

match

match

schematic or
extracted netlist



Constraint Types

- Alignment

  Edges or center points must align

- Symmetry

  Symmetry required for device seeds and context

- Matched Device Orientation

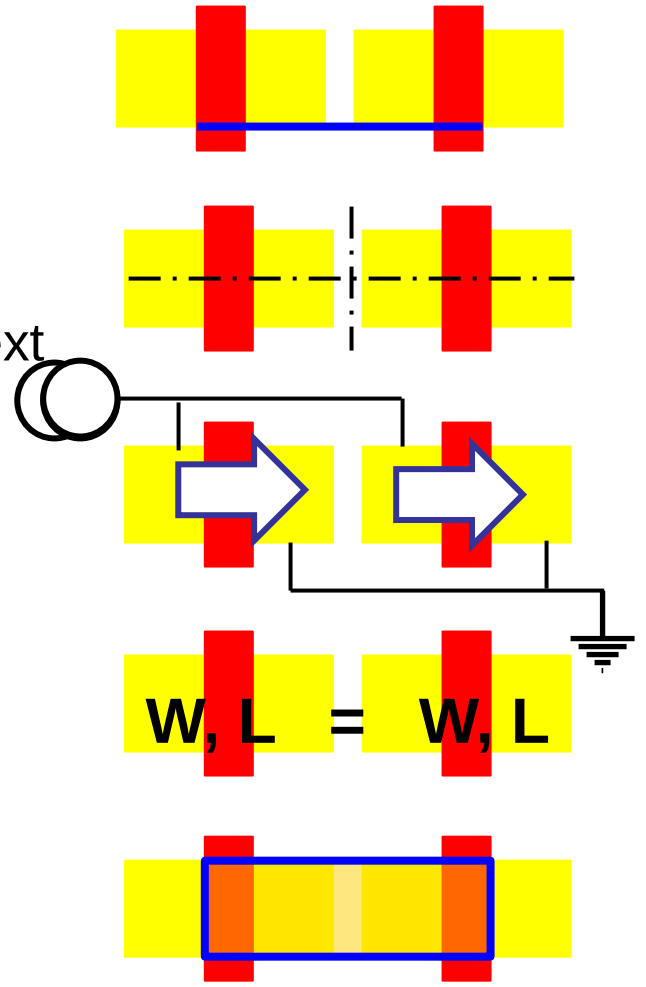
  Current vectors must match

- Matched Device Parameters

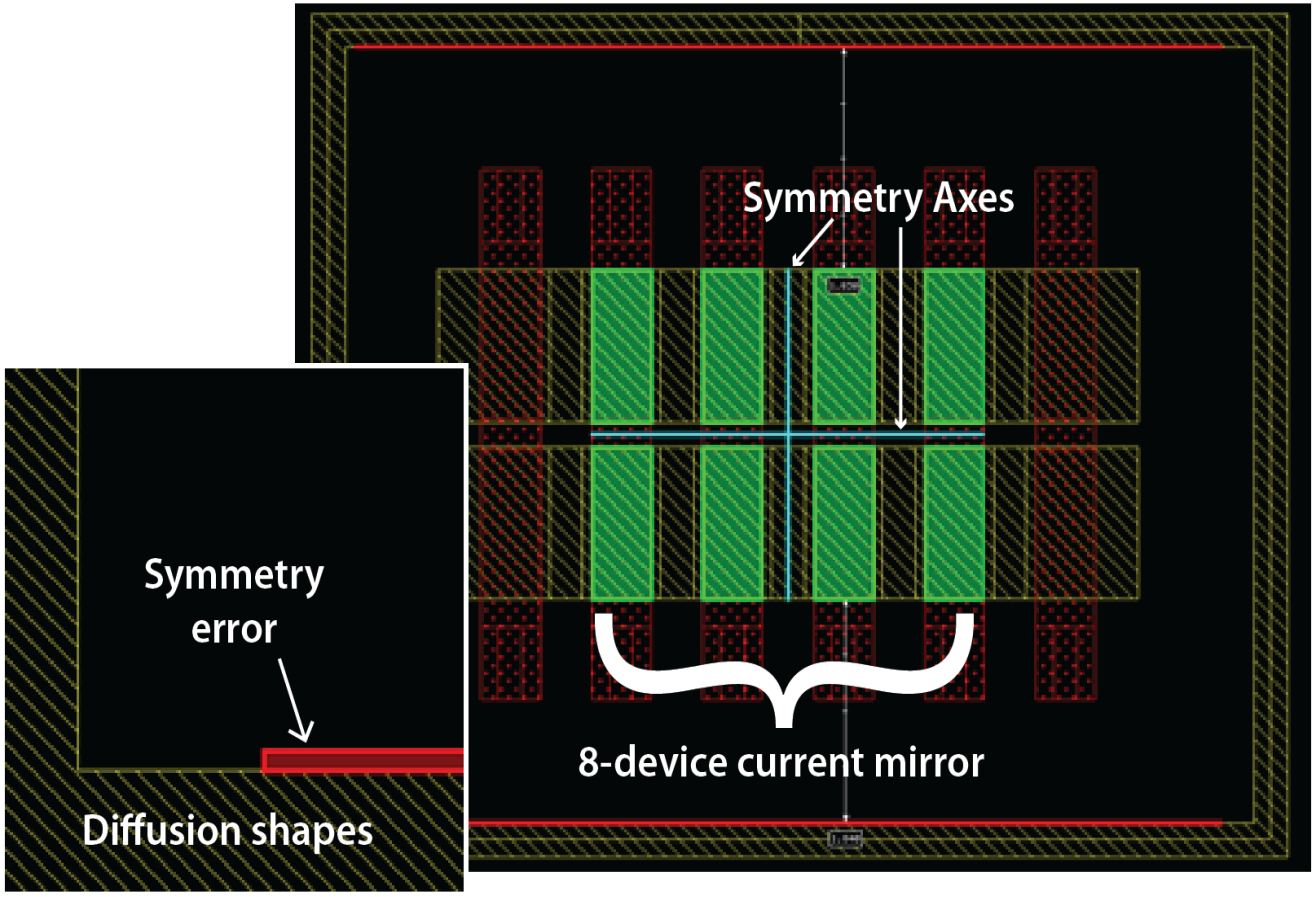
  All layout extracted parameters must match

- Cluster

  No foreign devices may touch the hull



Symmetry



Layout: Infineon Technologies

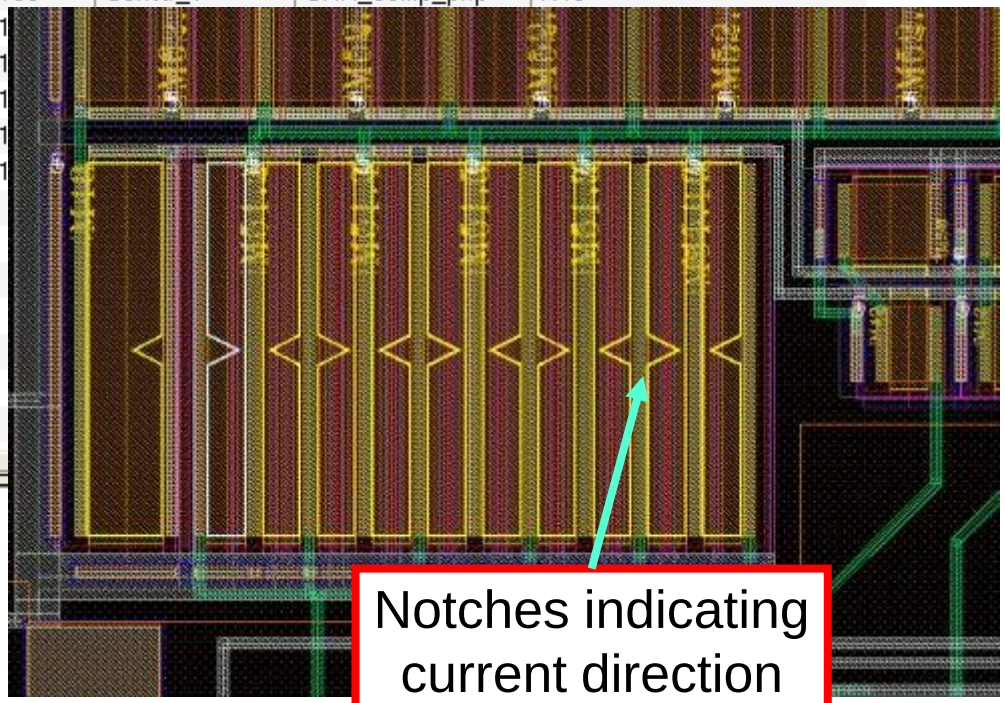
Matched Device Orientation

PERC Results | LDL: assertions.rdb | LDL: results.rdb ×

Show All | CAN_comp_pnp, 49 Results (in 3 of 3 Checks)

Check / Cell	Results
Check cluster	6
Check morient	11
Check symmetry	32

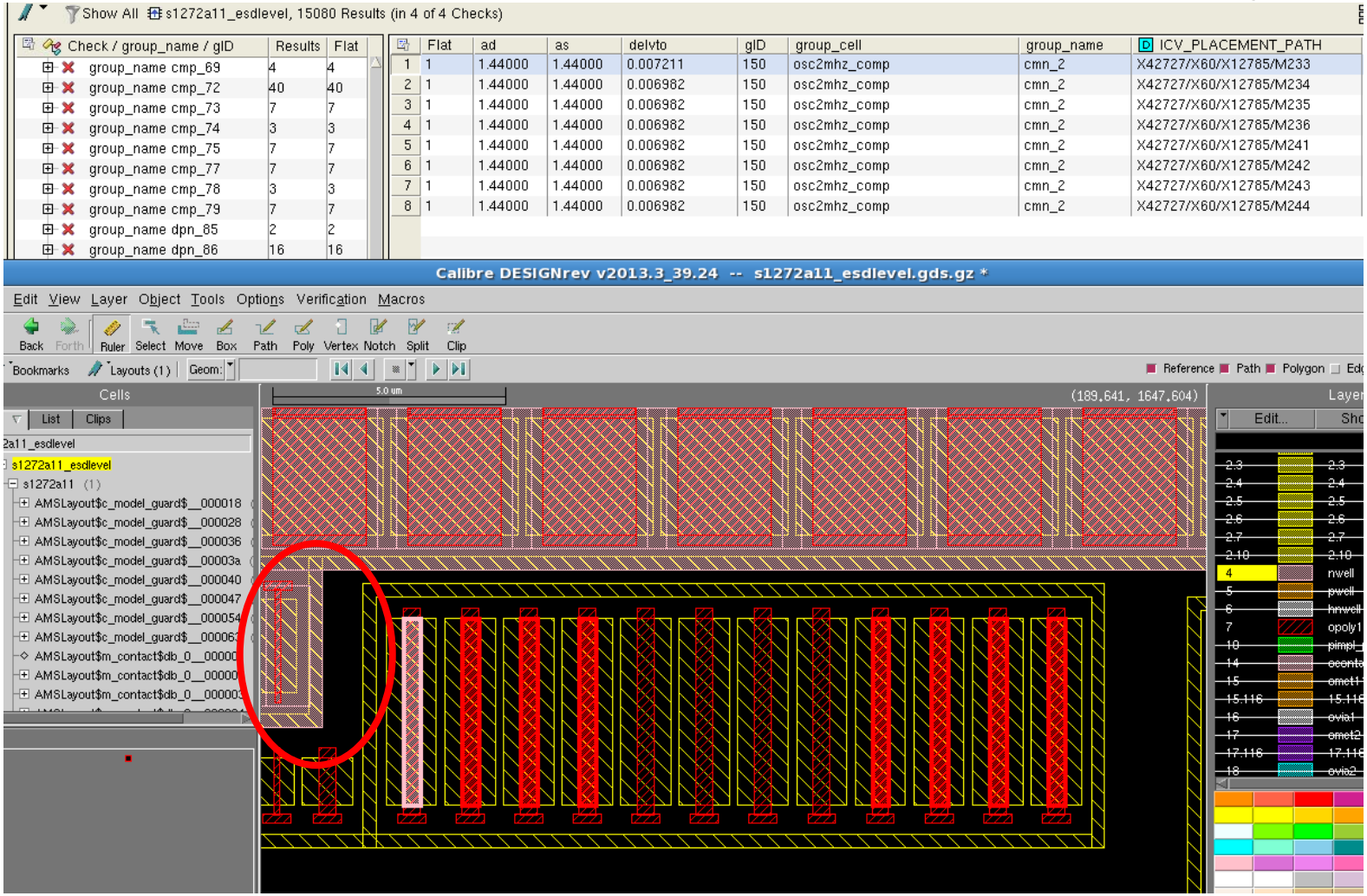
	angle	angle_inc	delta	group_name	cell	ICV_PLACEMENT_PA...
1	270	0	180	Constr_1	CAN_comp_pnp	X34
2	90	0	180	Constr_1	CAN_comp_pnp	X36
3	270	0	180	Constr_1	CAN_comp_pnp	X38
4	90	0	180	Constr_1	CAN_comp_pnp	X40
5	270	0	180	Constr_1	CAN_comp_pnp	X41
6	90	0	180	Constr_1	CAN_comp_pnp	X43
7	270	0	1			
8	90	0	1			
9	270	0	1			
10	90	0	1			
11	270	0	1			



Layout: Robert Bosch

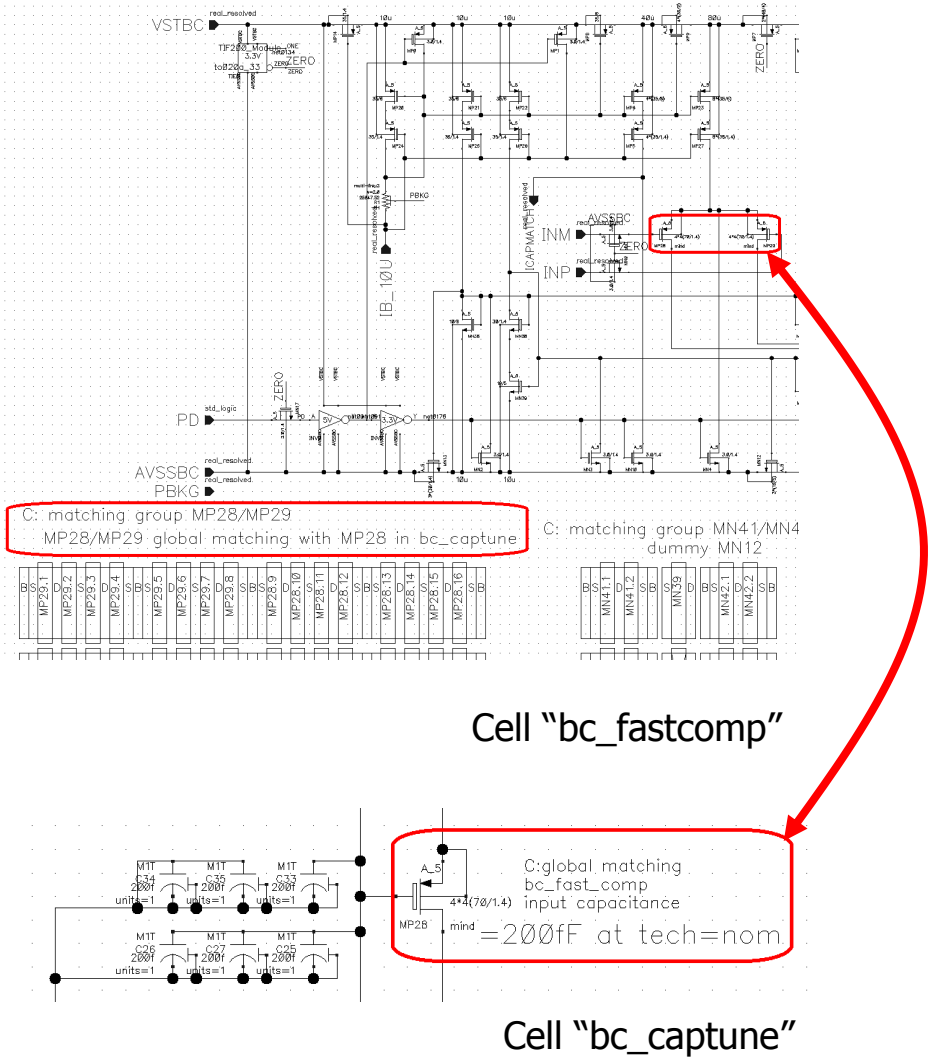
Matched Device Parameters

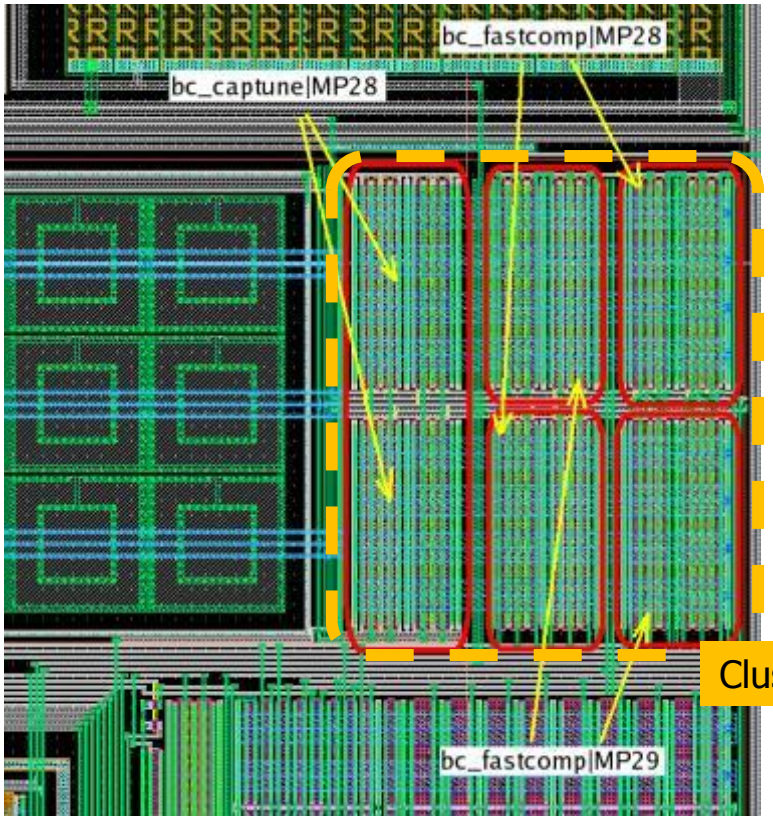
Unmatched “delvto”: inconsistent well proximity



Layout: Infineon Technologies

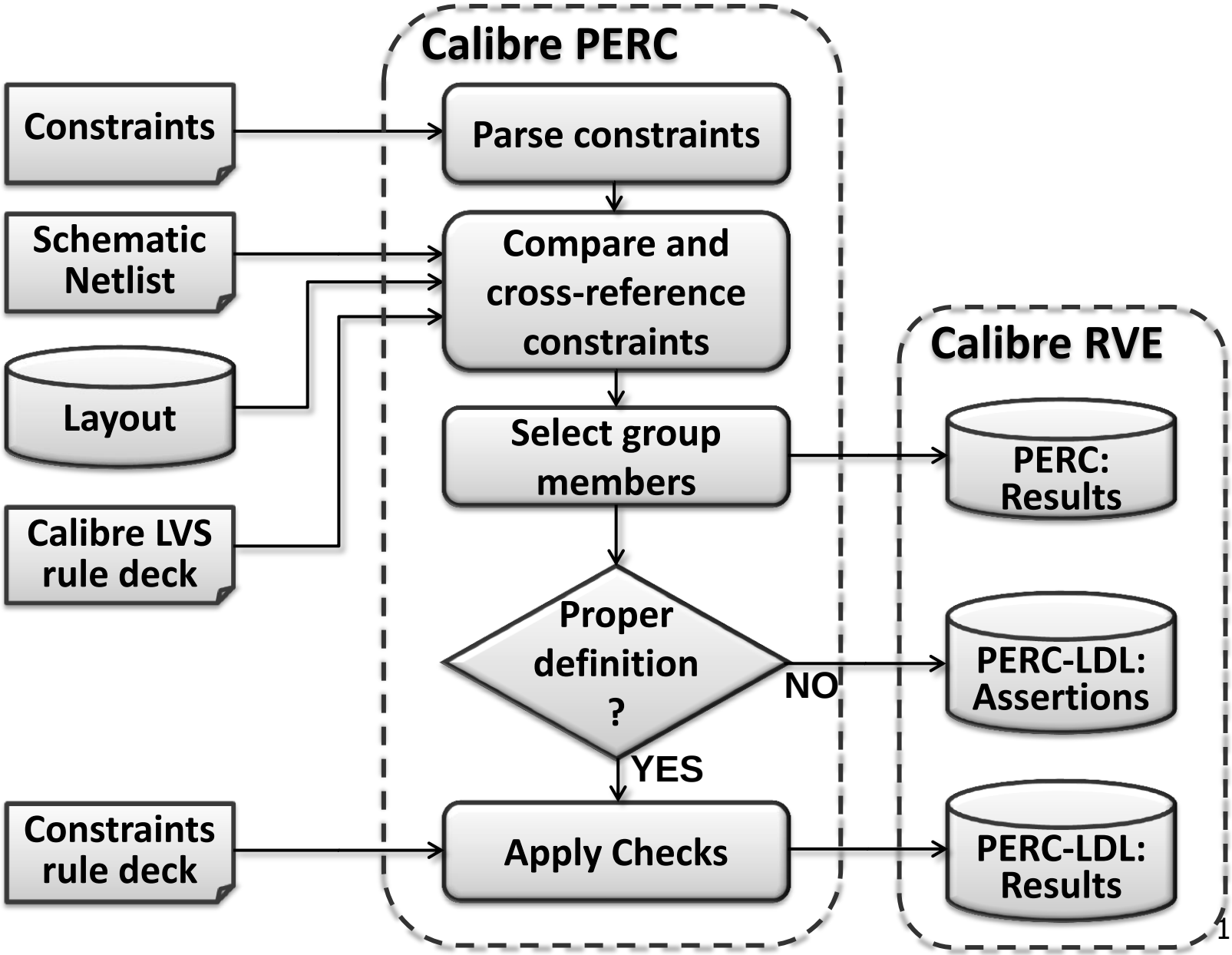
Cluster (Hierarchical Constraint)



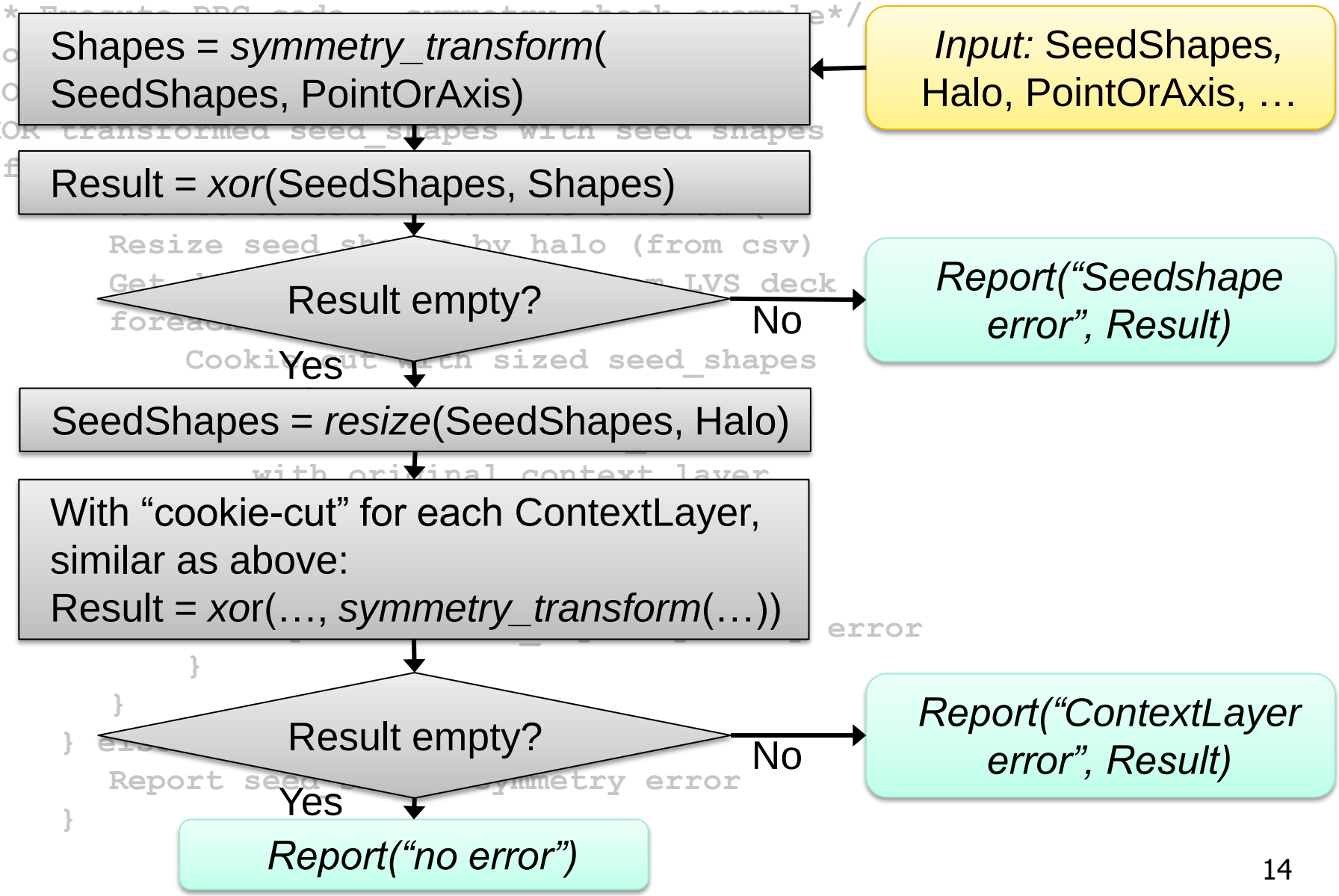


Source: Robert Bosch

Constraint Verification



Constraint Verification – DRC Code

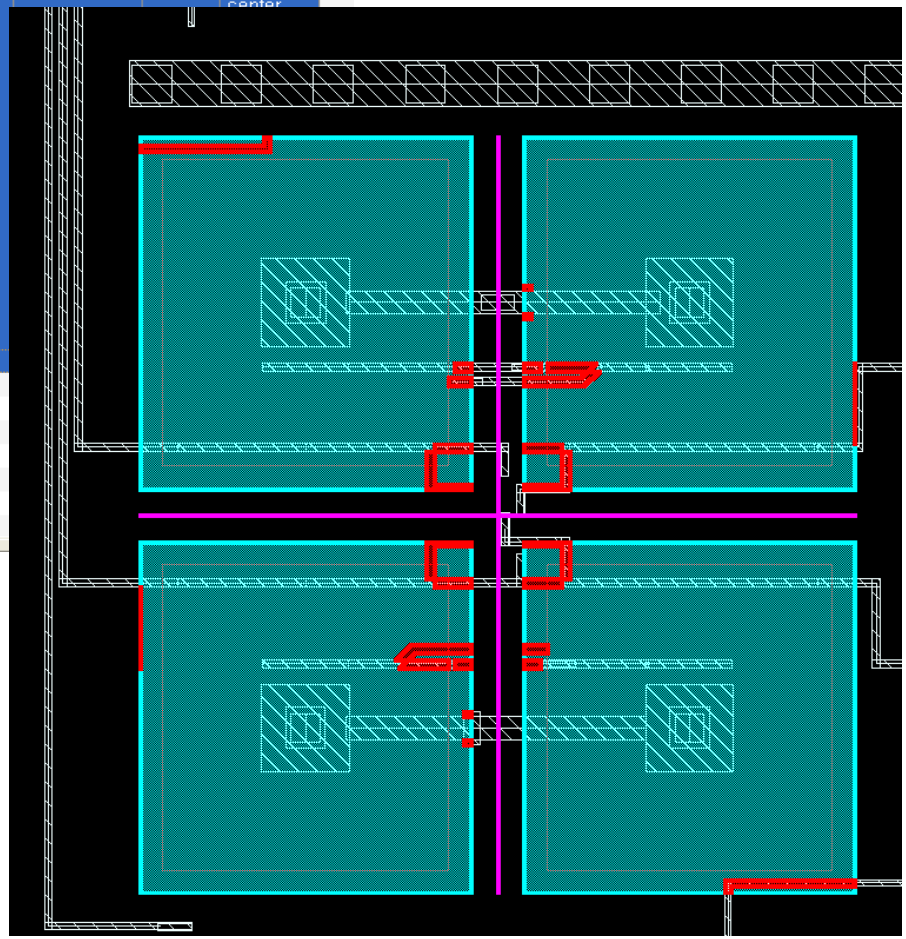


Symmetry Constraint – Result

Show All CAN_comp_pnp, 49 Results (in 3 of 3 Checks)

Check / Cell	Results
Check cluster	6
Check morient	11
Check symmetry	32

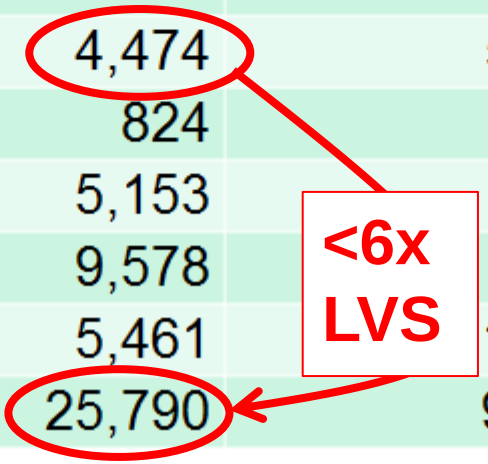
	conte...	devID	group_name	cell	ICV_PL...	is_ref	mode
8	ME1D		bjt_1	CAN_comp_pnp			center
9	ME1D		bjt_1	CAN_comp_pnp			center
10	ME1D		bjt_1	CAN_comp_pnp			center
11	ME1D		bjt_1	CAN_comp_pnp			center
12	ME1D		bjt_1	CAN_comp_pnp			
13	ME1D		bjt_1	CAN_comp_pnp			
14	ME1D		bjt_1	CAN_comp_pnp			
15	ME1D		bjt_1	CAN_comp_pnp			
16	ME1D		bjt_1	CAN_comp_pnp			
17	ME1D		bjt_1	CAN_comp_pnp			
18	ME1D		bjt_1	CAN_comp_pnp			
19	ME1D		bjt_1	CAN_comp_pnp			
20	ME1D		bjt_1	CAN_comp_pnp			
21	ME1D		bjt_1	CAN_comp_pnp			
22	ME1D		bjt_1	CAN_comp_pnp			
23	ME1D		bjt_1	CAN_comp_pnp			
24	ME1D		bjt_1	CAN_comp_pnp			
25	ME1D		bjt_1	CAN_comp_pnp			
26	ME1D		bjt_1	CAN_comp_pnp			
27			bjt_1	CAN_comp_pnp			
28			bjt_1	CAN_comp_pnp			
29		675	bjt_1	CAN_comp_pnp			
30		675	bjt_1	CAN_comp_pnp			
31		675	bjt_1	CAN_comp_pnp			
32		675	bjt_1	CAN_comp_pnp			



Source: Robert Bosch

Experimental Results

Design	A	B	C
Complexity:			
Hierarchical device count	111,194	14,467	8,583
Flat device count	5,208,564	1,354,624	201,777
Group count (schematic)	418	87	59
Constraint count (schem.)	1,255	250	256
Group count (layout)	1,270	88	280
Constraint count (layout)	3,950	270	1,120
Time (seconds elapsed):			
Constraint derivation	300	39	13
DB creation + xref (LVS)	4,474	511	52
Selection of devices	824	116	41
Assertion checks	5,153	77	38
Core constraint checks	9,578	68	61
Result processing/export	5,461	149	6
Total time	25,790	960	211



Hardware: Quad-core AMD Opteron 2384, 2.7GHz

Summary & Conclusions

Summary:

- Successful application to industrial designs
- Identified constraint violations demonstrate the need for hierarchical constraint verification

Outlook:

- Implementation of additional constraint checks
- Refinement/extension of rule-based constraint derivation