Parametric Hierarchy Recovery in Layout Extracted Netlists

John Lee, UCLA

Puneet Gupta, UCLA

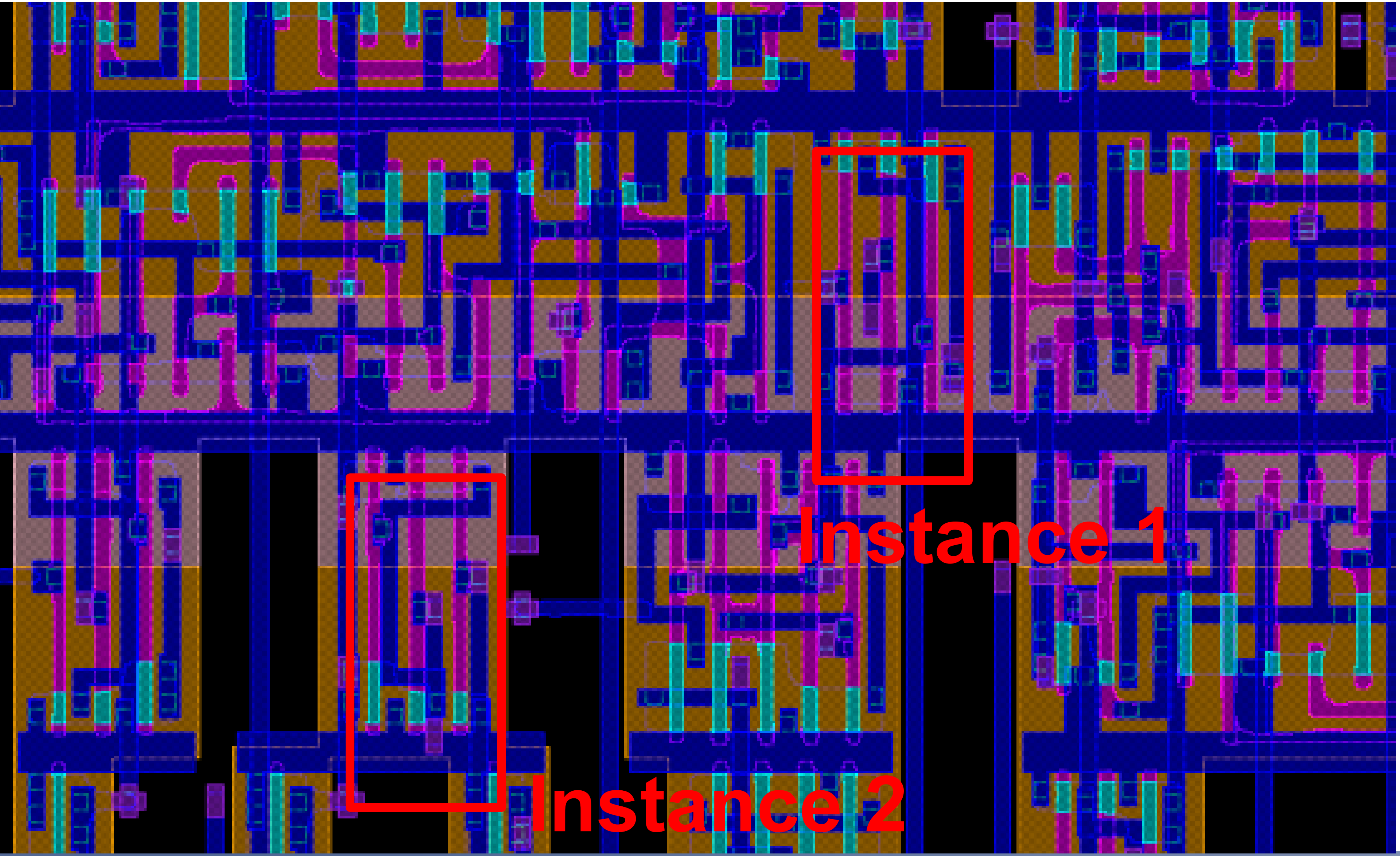
Fedor G. Pikus, Google (Formerly at Mentor)

This work was supported in part by the NSF Award 811832 and by the SRC Task 1816.

Hierarchy Degradation

- Layout-Dependent Device Parameters
  - Stress, Lithography, Parasitics
- Added to each transistor
  - Breaks the hierarchy
- Verification and analysis tools suffer from the lack of hierarchy:
  - Increases runtimes
  - Difficult to find common source of multiple errors
    - May have millions of errors from a single standard cell
    - Errors may be related to a repeating hierarchical block
  - Difficult to analyze
    - Too much information

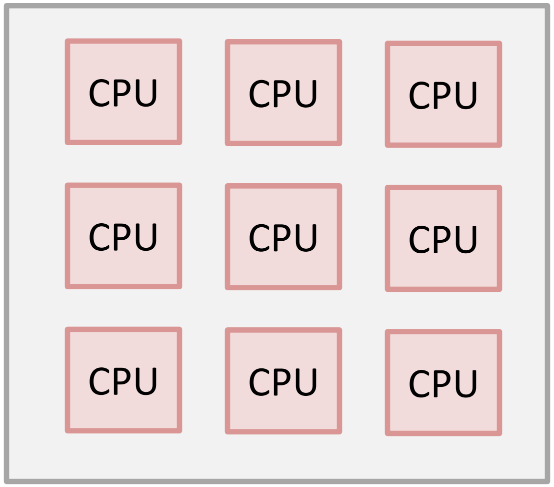
Hierarchy Degradation



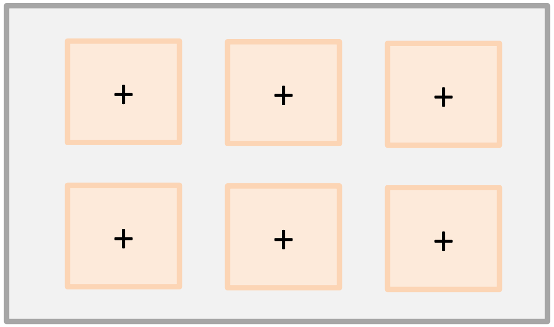
Redundant layout dependent contexts

Context dependence may be ***redundant***

- Examples:
  - Multicore designs

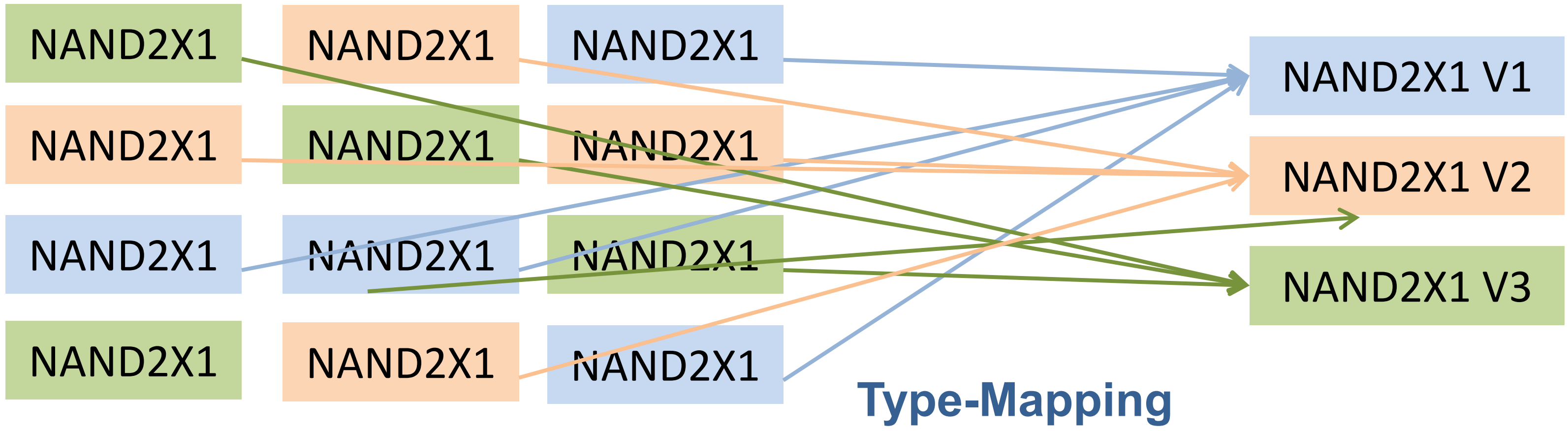


  - DSP designs



Hierarchy recovery using Type-Mapping

- Create type-variants to recover hierarchy
  - Tolerance based mapping:
    - Map device parameters together if they are within a tolerance
  - Variants on the original hierarchical blocks
  - Capture the different layout contexts in a design



Hierarchy Degradation Example

Hierarchical

```
.SUBCKT sc2
M1 1 2 3 nch L=45e-9 W=90e-9
M2 1 2 3 nch L=45e-9 W=90e-9
.ENDS

.SUBCKT sc 1 2 3 4
M1 1 2 3 nch L=45e-9 W=90e-9
M2 2 3 4 pch L=45e-9 W=90e-9
M3 4 5 1 pch L=45e-9 W=90e-9
M4 5 6 2 pch L=46e-9 W=90e-9
X1 1 sc2
X2 1 sc2
X3 1 sc2
.ENDS
```

Flat

```
.SUBCKT sc2
MM1 1 2 3 nch L=45e-9 W=90e-9
.ENDS

.SUBCKT sc 1 2 3 4 5
MM1 1 2 3 nch L=45e-9 W=90e-9
MM2 2 3 4 pch L=45e-9 W=90e-9
MM3 4 5 1 pch L=45e-9 W=90e-9
MM4 5 6 2 pch L=45e-9 W=90e-9
MX1/M2 3 4 6 nch L=52e-9 W=90e-9
MX2/M2 3 4 6 nch L=50e-9 W=90e-9
MX3/M2 3 4 6 nch L=50e-9 W=90e-9
XX1 1 sc2
XX2 1 sc2
XX3 1 sc2
.ENDS
```

Hierarchy Degradation Example

Flat

```
.SUBCKT sc2
MM1 1 2 3 nch L=45e-9 W=90e-9
.ENDS

.SUBCKT sc 1 2 3 4 5
MM1 1 2 3 nch L=45e-9 W=90e-9
MM2 2 3 4 pch L=45e-9 W=90e-9
MM3 4 5 1 pch L=45e-9 W=90e-9
MM4 5 6 2 pch L=45e-9 W=90e-9
MX1/M2 3 4 6 nch L=52e-9 W=90e-9
MX2/M2 3 4 6 nch L=50e-9 W=90e-9
MX3/M2 3 4 6 nch L=50e-9 W=90e-9
XX1 1 sc2
XX2 1 sc2
XX3 1 sc2
.ENDS
```

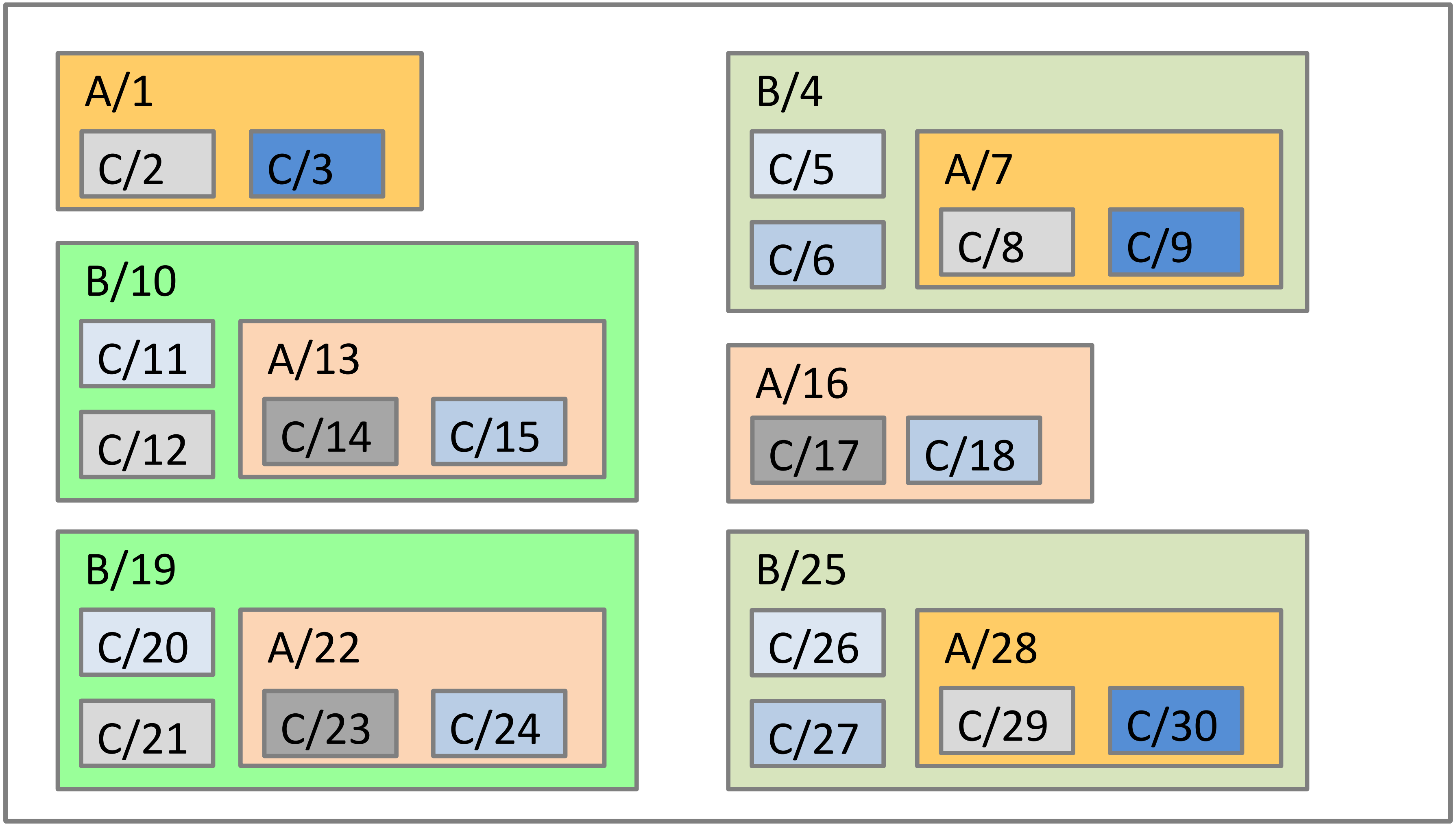
Type-Mapped

```
.SUBCKT sc2_v0
M1 1 2 3 nch L=45e-9 W=90e-9
M2 3 4 6 nch L=50e-9 W=90e-9
.ENDS

.SUBCKT sc2_v1
M1 1 2 3 nch L=45e-9 W=90e-9
M2 3 4 6 nch L=52e-9 W=90e-9
.ENDS

.SUBCKT sc 1 2 3 4 5
MM1 1 2 3 nch L=45e-9 W=90e-9
MM2 2 3 4 pch L=45e-9 W=90e-9
MM3 4 5 1 pch L=45e-9 W=90e-9
MM4 5 6 2 pch L=45e-9 W=90e-9
XX1 1 sc2_v1
XX2 1 sc2_v0
XX3 1 sc2_v0
.ENDS
```

Type-Mapping Steps

- Start at the top of the hierarchy
    1. Map the types that are induced by higher level parent types
    2. Merge types that meet tolerance constraints
    3. Add free instances to existing types, if possible
        - Otherwise, create new types

Hierarchical Type-Mapping



A/1

C/2 C/3

B/4

C/5

C/6

A/7

C/8 C/9

B/10

C/11

C/12

A/13

C/14 C/15

A/16

C/17 C/18

B/19

C/20

C/21

A/22

C/23 C/24

B/25

C/26

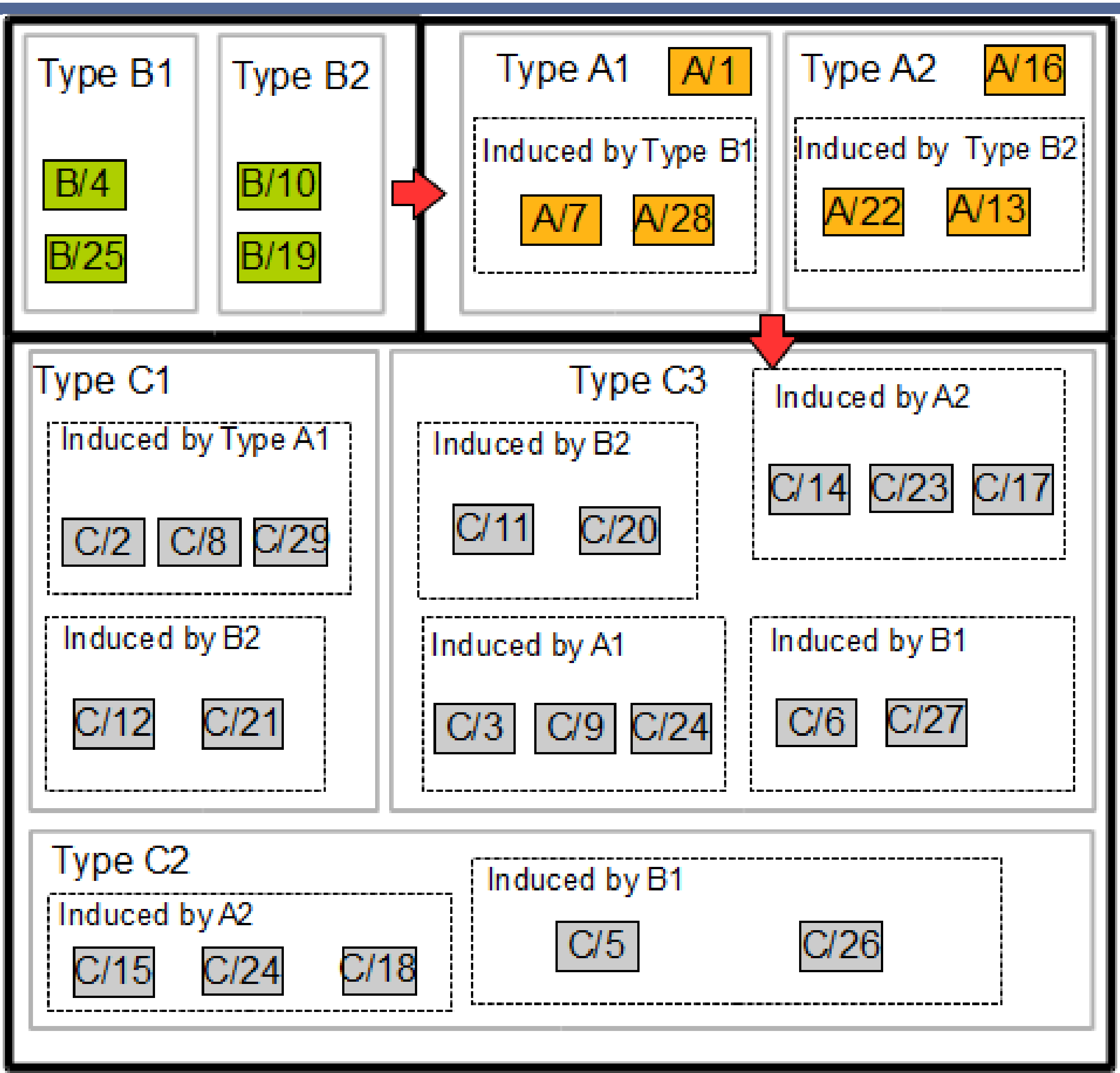
C/27

A/28

C/29 C/30

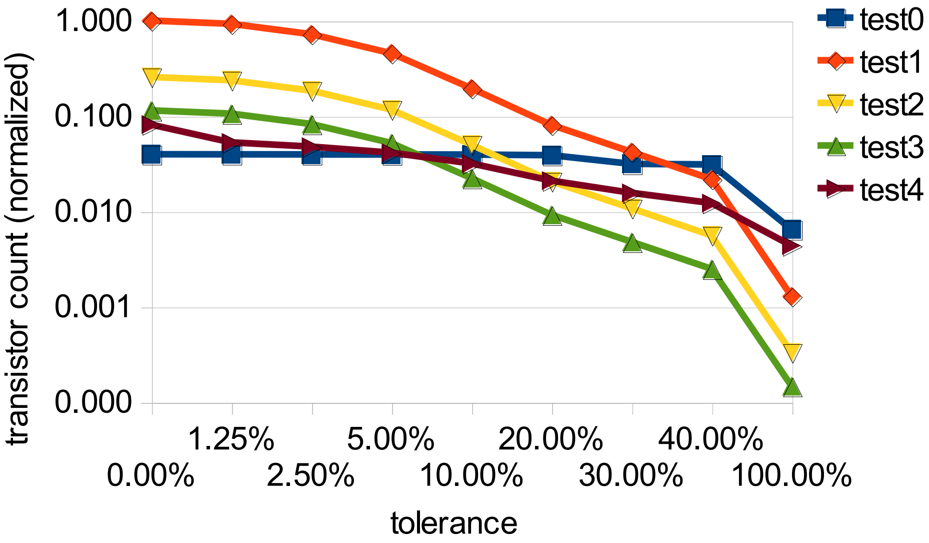
Type-Map “C”: Induced types can be merged

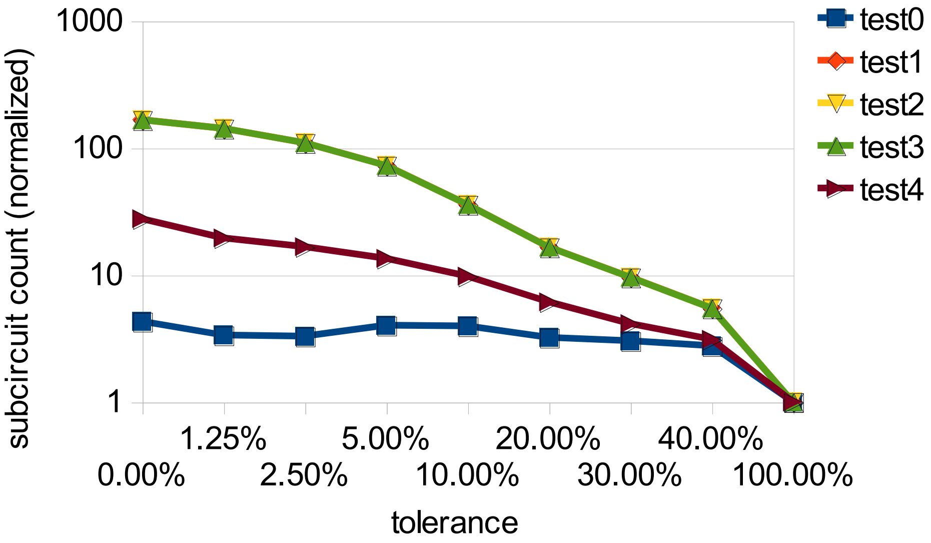
Hierarchical Type-Mapping



Applications: Netlist Size

- 5 real Industrial designs
  - Test0: 17M transistors
  - Test1: 834K transistors
  - Test2: 3.3M transistors (4 test1 cores)
  - Test3: 7.5M transistors (9 test1 cores)
  - Test4: 2.4M transistors



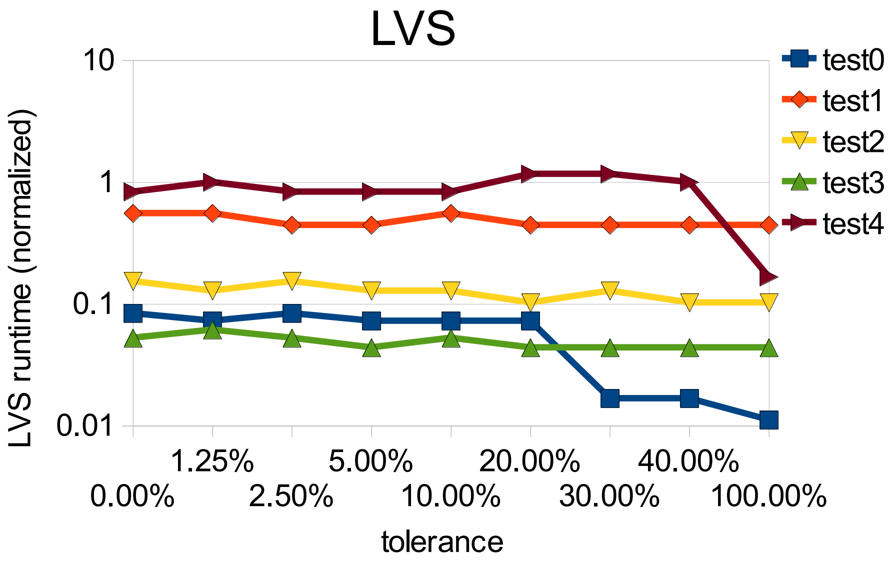


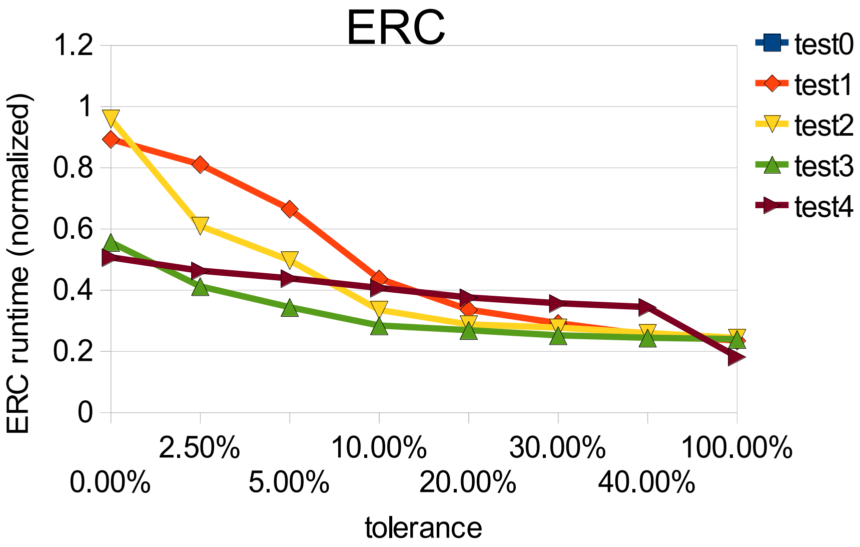
*Tolerance is with respect t to max parameter deviation

*Subcircuit and transistor counts are normalized relative to the “pushed” layout extracted by the commercial tool

Applications: Verification

- 5 real Industrial designs (800k – 17M transistors)
- Comparison with Calibre: LVS & ERC
  - Runtime improvement vs. spread tolerance
    - Tolerance is a function of ***maximum parameter deviation***
    - 1.0 – runtime for layout extracted netlist

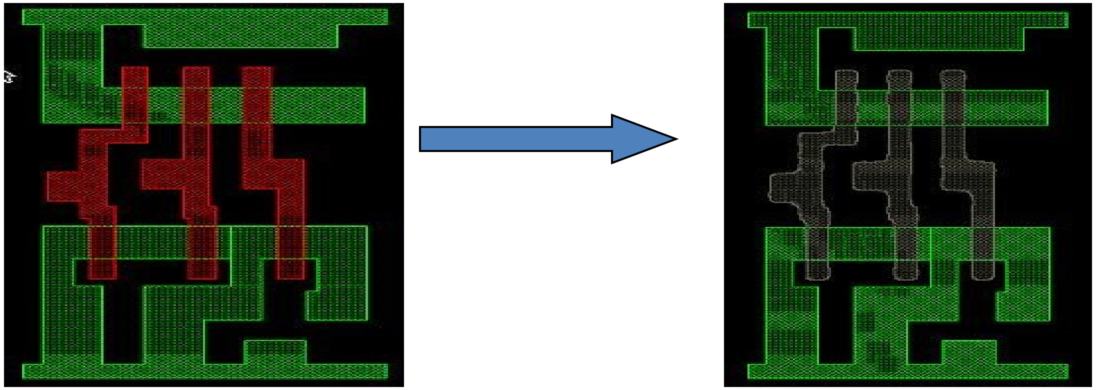




*Tolerance is with respect t to max parameter deviation

* LVS and ERC performance are normalized relative to the “pushed” layout extracted by the commercial tool

Lithography effects and Timing Mismatch



What designer sees

What silicon shows

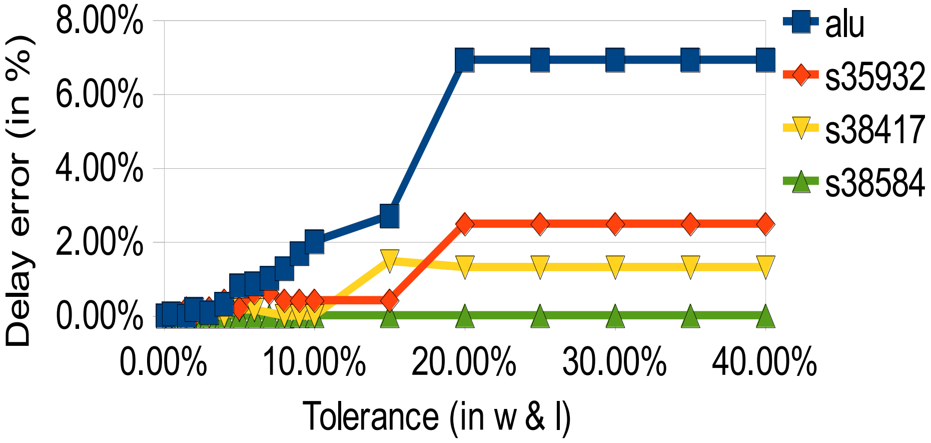
1mW, 200MHz

1.3mW, 180MHz

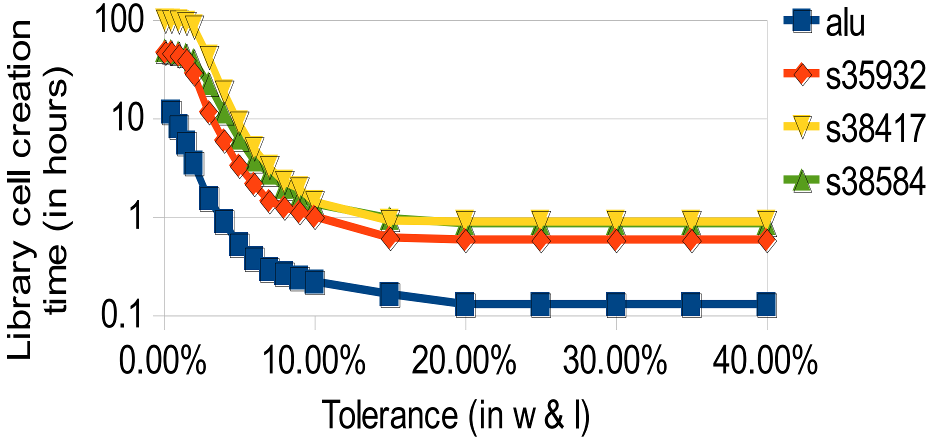
- Static analysis flows based on standard cell abstraction
    - One cell is 2-100 transistors
    - Timing/power views stored in pre-characterized “.lib” files
- State of art 45nm logic designs have 10M+ cells and 50M+ transistors →Hierarchy preservation essential

Application: Timing

- ISCAS `89 designs + Open Cores ALU + 45nm Library
- Create type variants of standard cells
  - Post-lithography extracted netlist
  - Variations in L & W, tolerance a function of the nominal values
  - Map layout extracted cells onto standard cells using adapted kmeans++
    - E.g. NAND_X1_VARIANT1, NAND_X1_VARIANT2, etc.



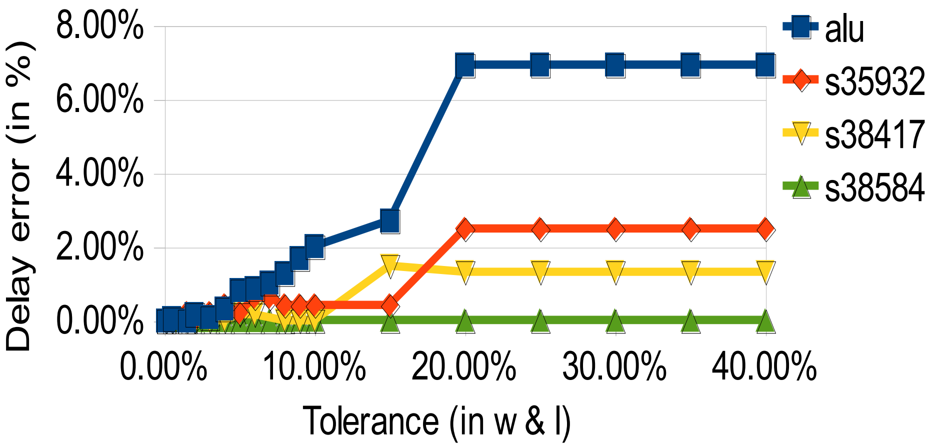
(tolerance is with respect to nominal w & l)



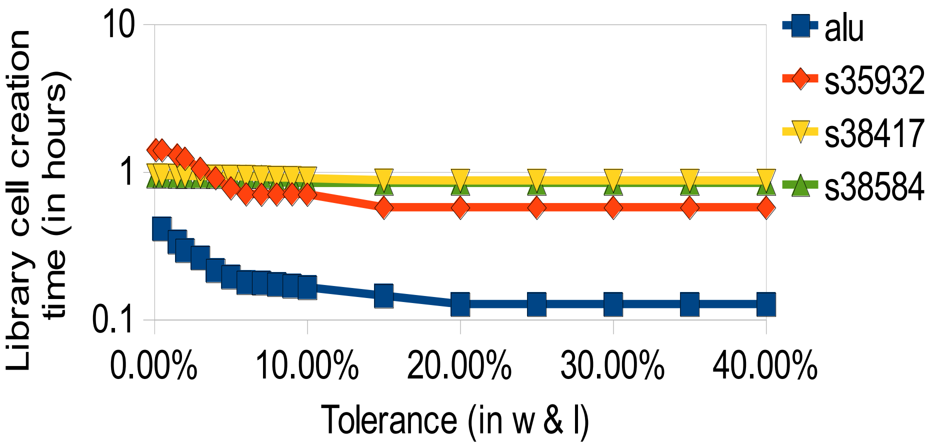
*estimated using Synopsys NanoTime

Application: Timing + Slack

- Type-variant characterization too slow
  - 11 to 97 hrs
- Use slack to adjust tolerances:
  - Clock period T and gate with slack Y
    - ***Delay can increase by Y/T before path is critical***
    - Increase the tolerance by *Y*/(*αT*) to reduce the library size
      - *α >1 used to account for modeling errors*



(tolerance is with respect to nominal w & l)



*estimated using Synopsys NanoTime

Summary

- Created a method to recover hierarchy in post-layout designs
  - Creates type-variants of the original cells / blocks
  - Uses hierarchical type-mapping methods to ensure the mapping is correct across all levels
- With no parametric error
  - 16% - 96% runtime reduction for LVS
  - 4% - 49% runtime reduction for ERC
- Tractable method for post-layout timing
- Future work will look at type-mapping parasitic information and consider application specific distance metrics
  - Timing distance, power distance