Parametric Hierarchy Recovery in Layout Extracted Netlists (Type Mapping)

John Lee

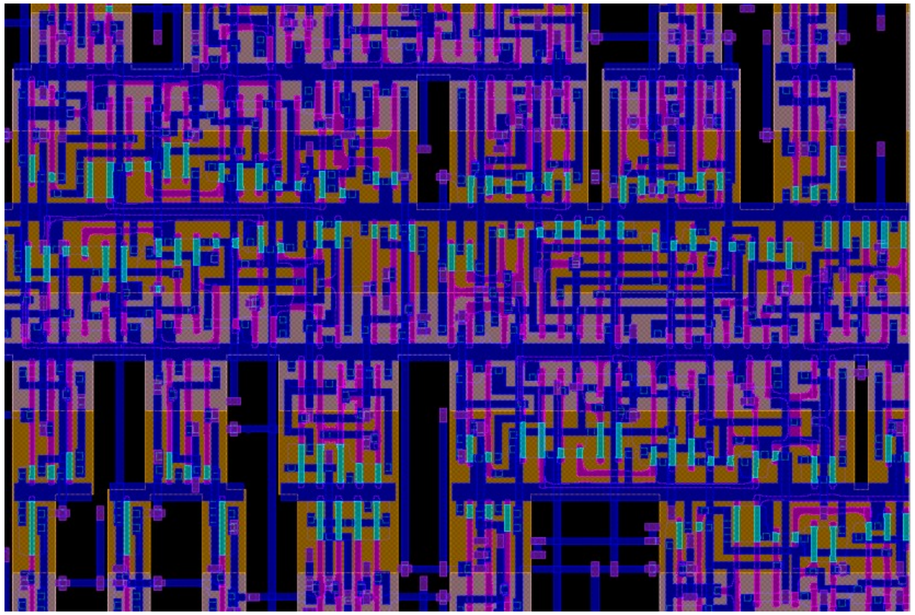
Puneet Gupta

DFM&Y 2011

UCLA

Hierarchy Degradation

- **Layout-Dependent Device Parameters**
  - Stress effects
  - Lithography effects
  - Parasitics



- Added to each transistor
  - Breaks the hierarchy

```
M0 5 8 5 VSS NMOS L=4.85294e-08 W=3.40000e-07 $X=225 $Y=160 $D=77
   sca=2.70597e+01 scb=1.90764e-02 scc=3.46602e-03 spa=1.94753e-07
   spa1=1.86534e-07 spa2=1.77552e-07 spa3=1.80107e-07 sap=1.81200e-07
   sa4=1.67593e-07 spba=2.04926e-07 spba1=2.07364e-07 sapb=2.35994e-07
   enx=3.99926e-06 enx1=3.38623e-06 eny=8.84817e-08 eny1=1.56155e-08
   eny2=2.79212e-08 rex=1.65402e-06 rey=1.85106e-07
```

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
```

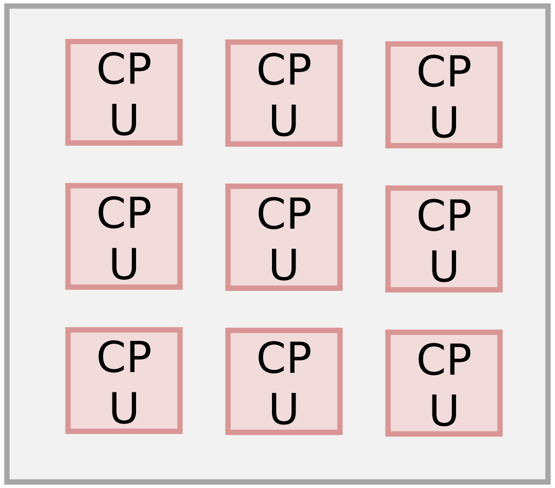
Difficulties from flattened netlists

- 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

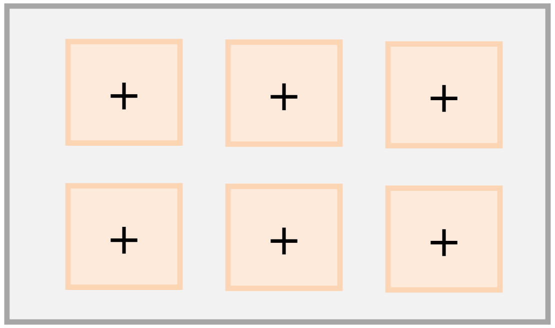
Redundant layout dependent contexts

Most designs have redundant contexts:

- Examples:
  - Multicore designs

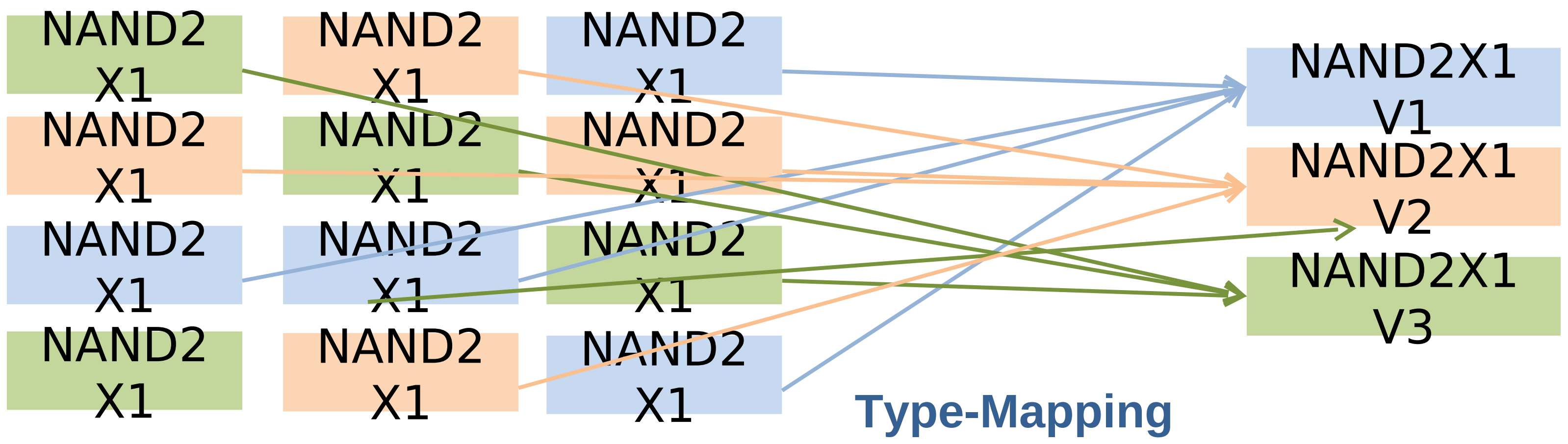


  - DSP designs



Hierarchy recovery using Type-Mapping

- Create <u>type-variants</u> to recover hierarchy
  - Tolerance based clustering:
    - Cluster device parameters 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
```

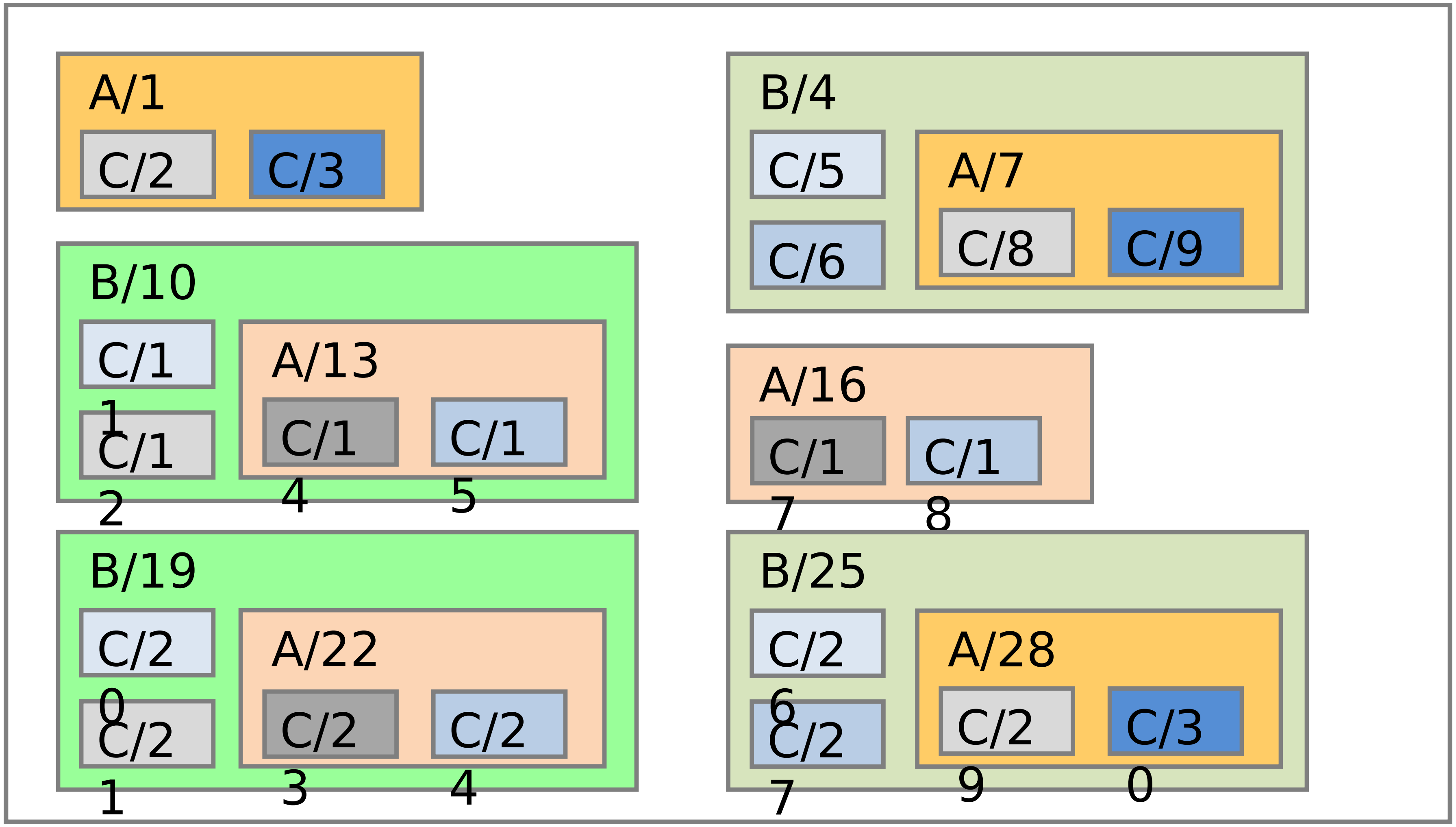
Type-Mapped

```
.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
```

Hierarchical Type-Mapping



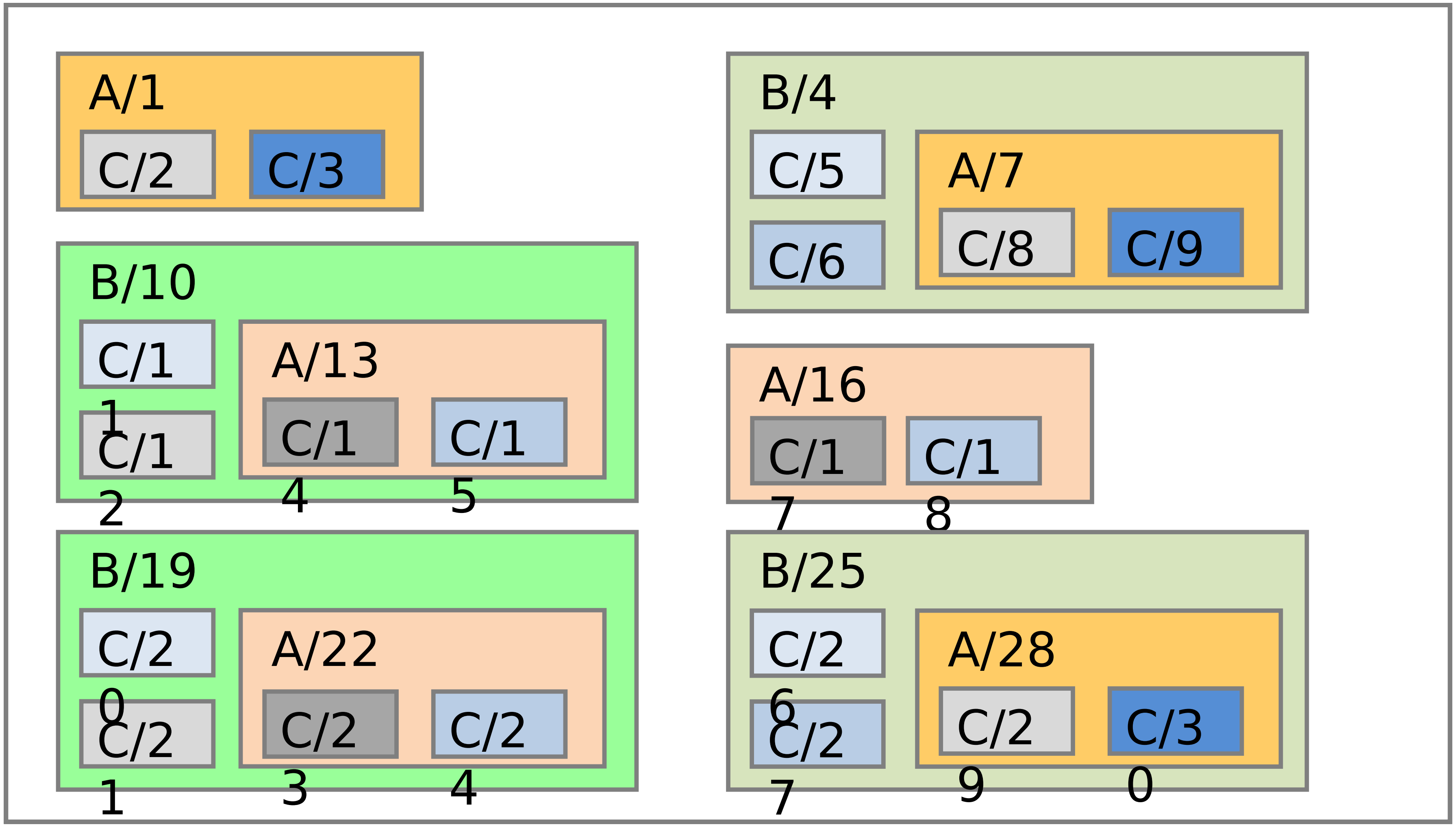
Type-Map "C": Induced clusters can be merged

Type-Mapping Steps

- Start at the top of the hierarchy

1. Cluster the clusters that are induced by higher level parent types
2. Merge clusters that meet tolerance constraints
3. Add free instances to existing clusters, if possible
    - Otherwise, create new clusters

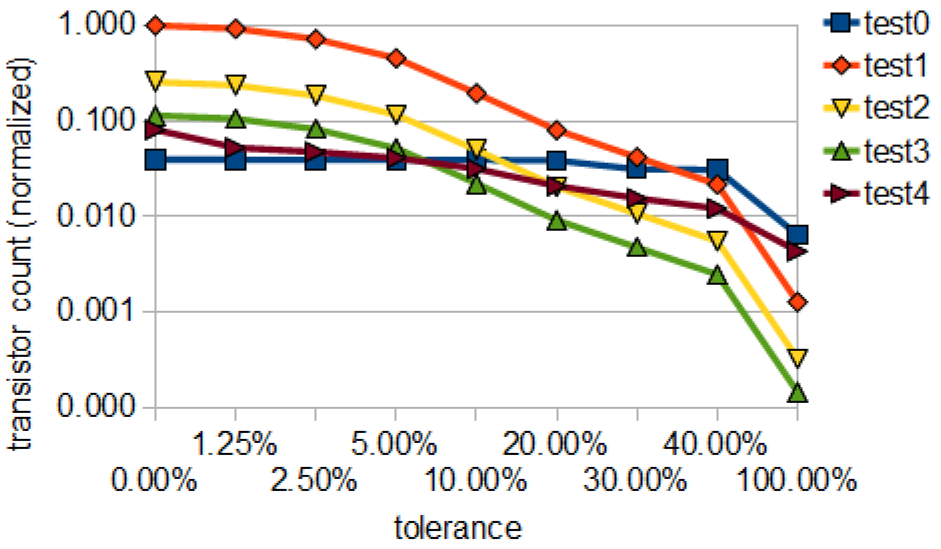
Hierarchical Type-Mapping

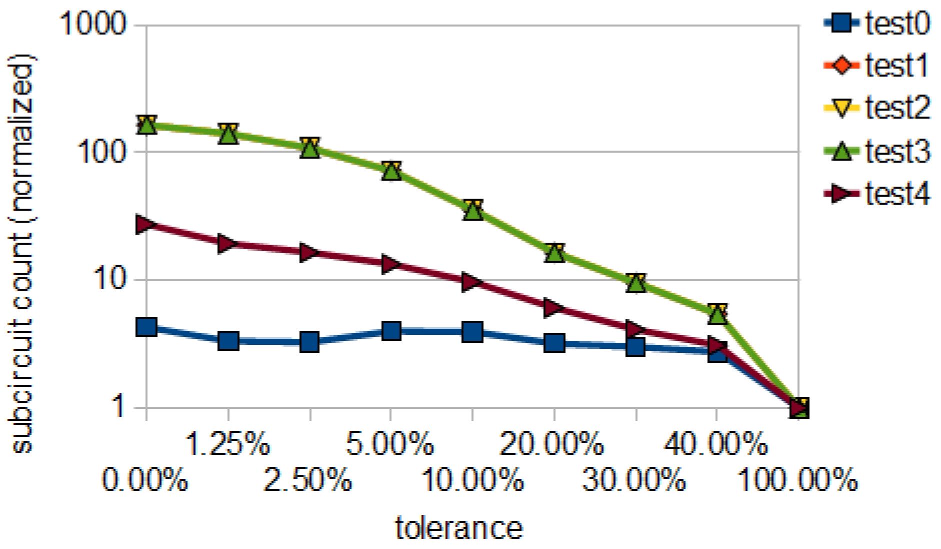


Type-Map "C": Induced clusters can be merged

Applications: Verification

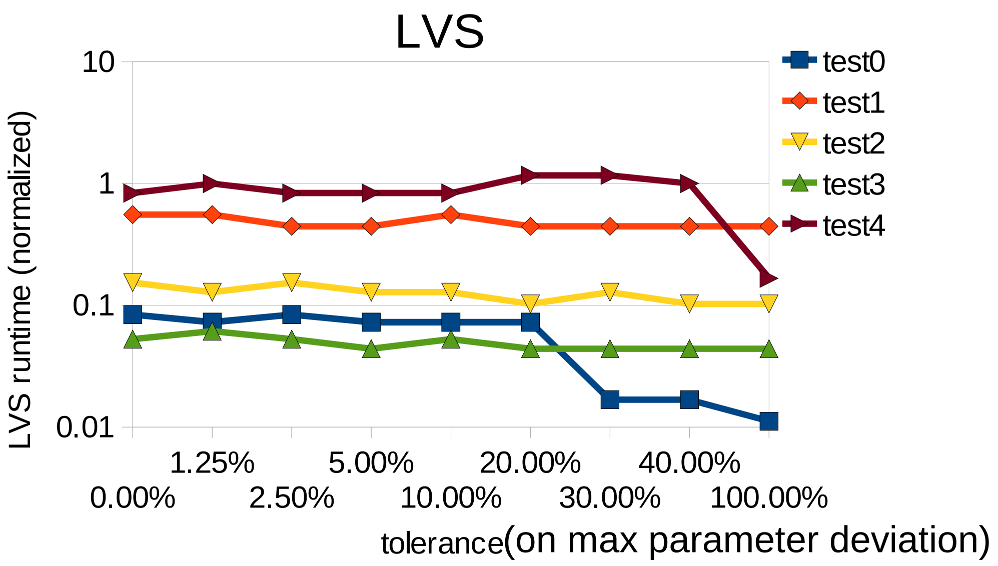
- 5 real Industrial designs
  - Test0: 17M transistors
  - Test1: 834K transistors
  - Test2: 3.3M transistors (4 test1 cores)
  - Test3: 7.5M transistors (9 test1 cores)
  - Test4: 2.4M transistors

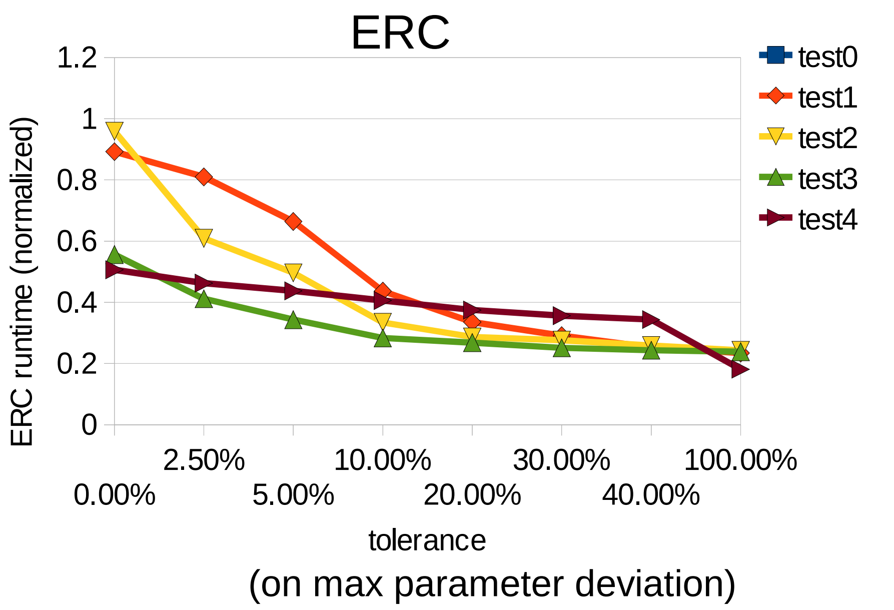




Applications: Verification

- 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

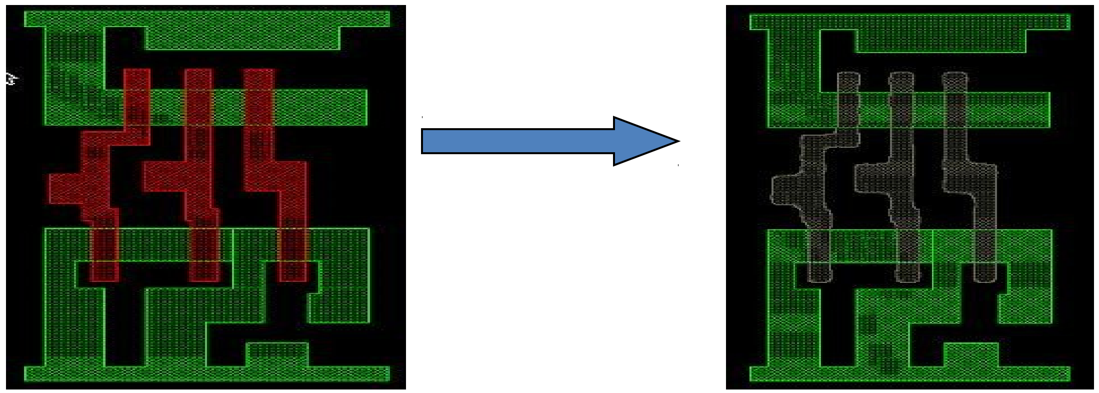




Lithography effects and Timing Mismatch

What designer sees

What silicon shows



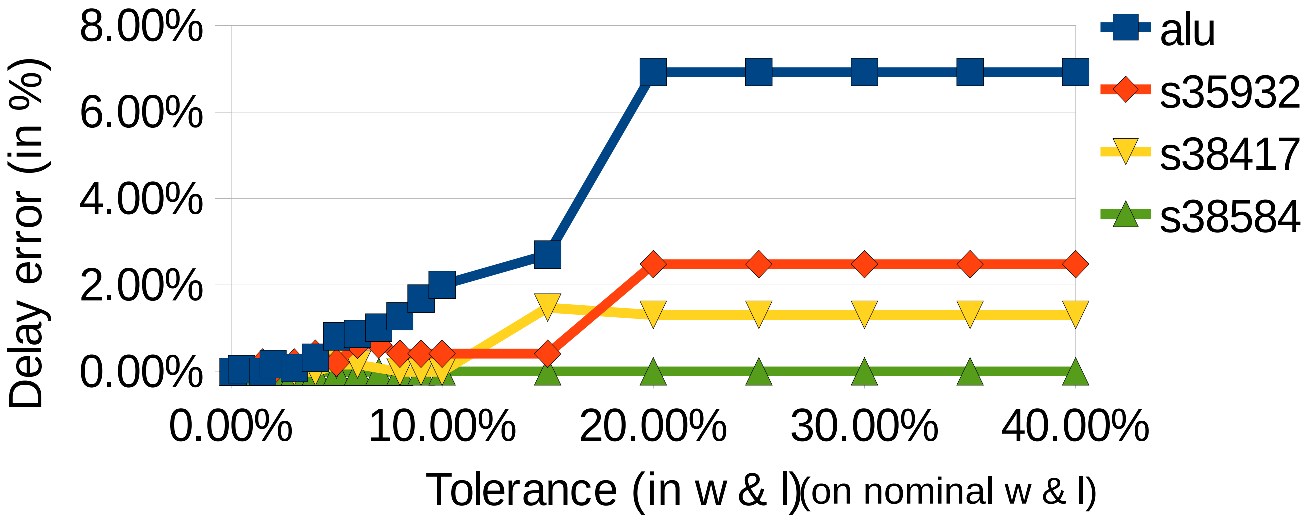
1mW, 200MHz

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

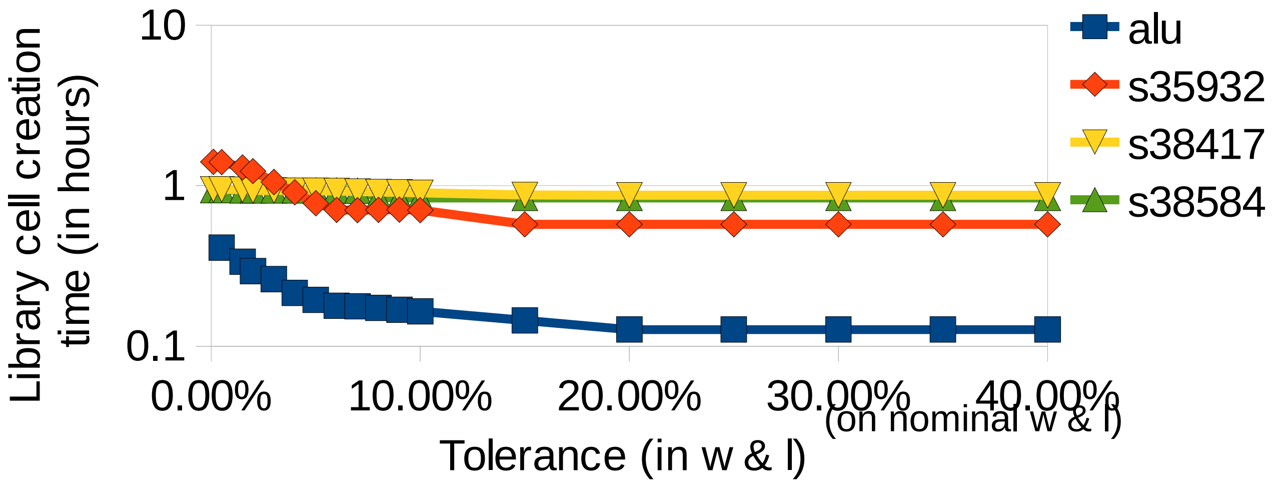
- ISCAS `89 designs + Open Cores ALU + 45nm Library
- Create type variants of standard cells
  - Post-lithography extracted netlist
  - Variations in L & W, tolerance a function of the nominal values
  - Map layout extracted cells onto standard cells
    - E.g. NAND_X1_VARIANT1, NAND_X1_VARIANT2, etc.



*estimated using Synopsys NanoTime

Application: Timing + Slack

- Type-variant library characterization too slow
  - 11 to 97 hrs
- Use slacks to adjust tolerances:
  - Clock period T and a gate with slack Y
    - ***Delay can increase by Y/T before path is critical***
    - Increase the tolerance by $\frac{Y}{\alpha T} \rightarrow$ reduce variant library size



*estimated using Synopsys NanoTime

Summary

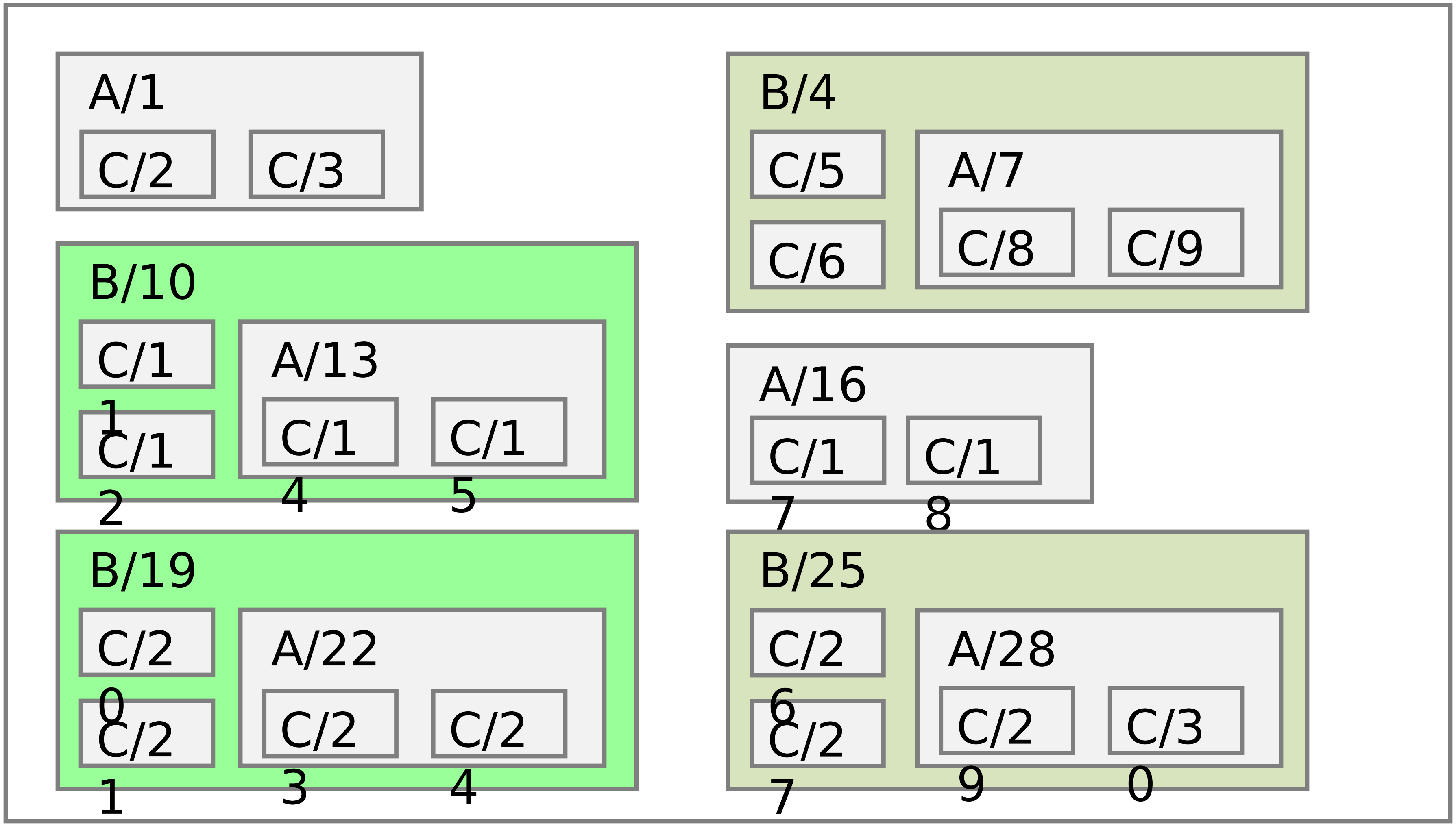
- Created a method to recover hierarchy in post-layout designs
  - Creates type-variants of the original cells / blocks
  - Uses hierarchical type-mapping methods to ensure the mapping is correct across all levels
- With no parametric error
  - 16% - 96% runtime reduction for LVS
  - 4% - 49% runtime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

Extra Slides

Motivation: Loss of hierarchy

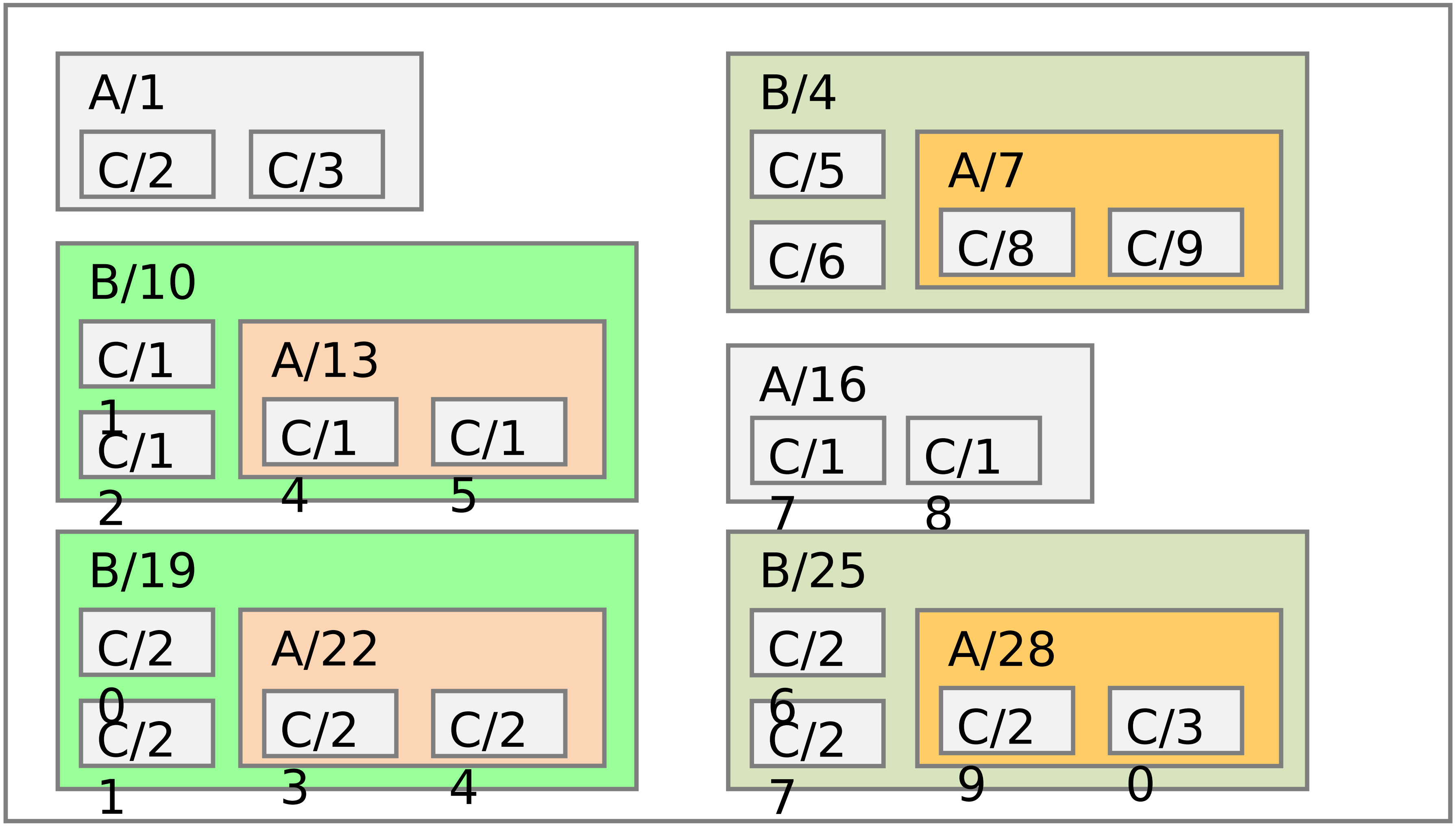
- <u>**Pre-layout**</u>
  - Lots of hierarchy:
    - Blocks, standard cells, modules
- <u>**Post-layout**</u>
  - Hierarchy flattened due to ***layout dependent*** parameters
    - Stress, annealing, etch, lithographic variability
  - Netlists 100x-200x larger
- Verification and analysis tools suffer from the lack of hierarchy:
  - Can the hierarchy be recovered?

Hierarchical Type-Mapping



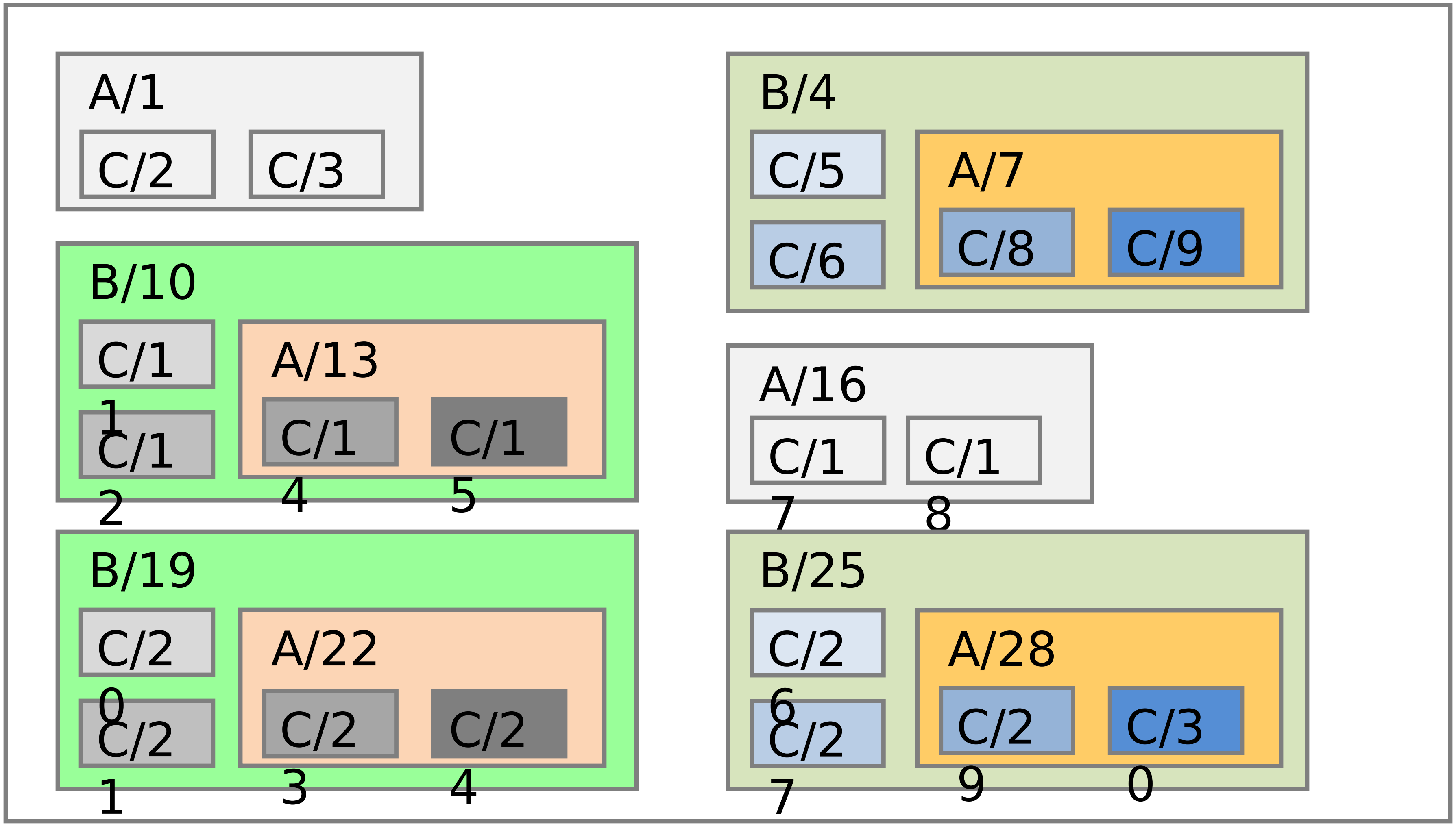
Type-Map "B"

Hierarchical Type-Mapping



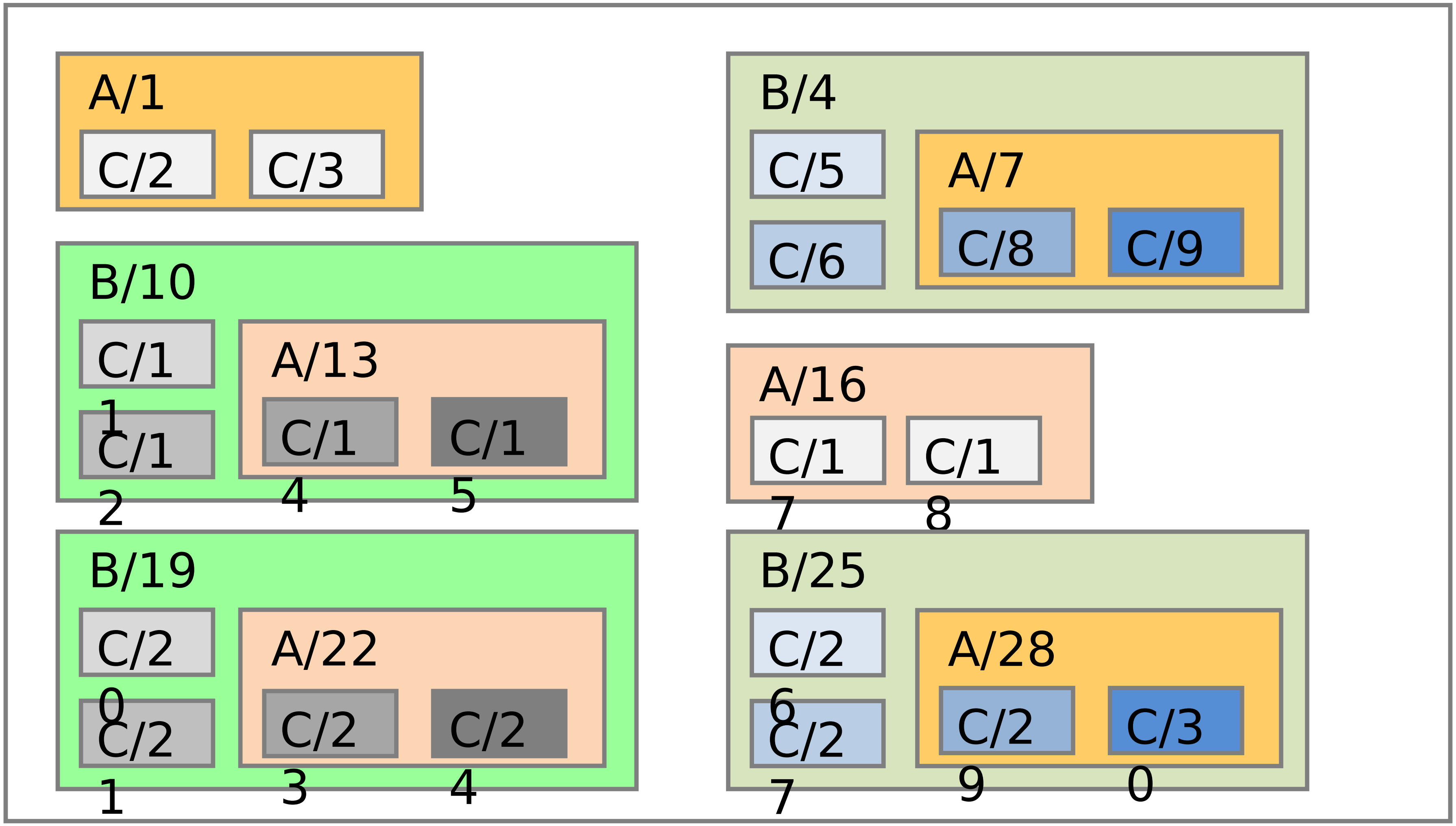
This induces clusters in "A"

Hierarchical Type-Mapping



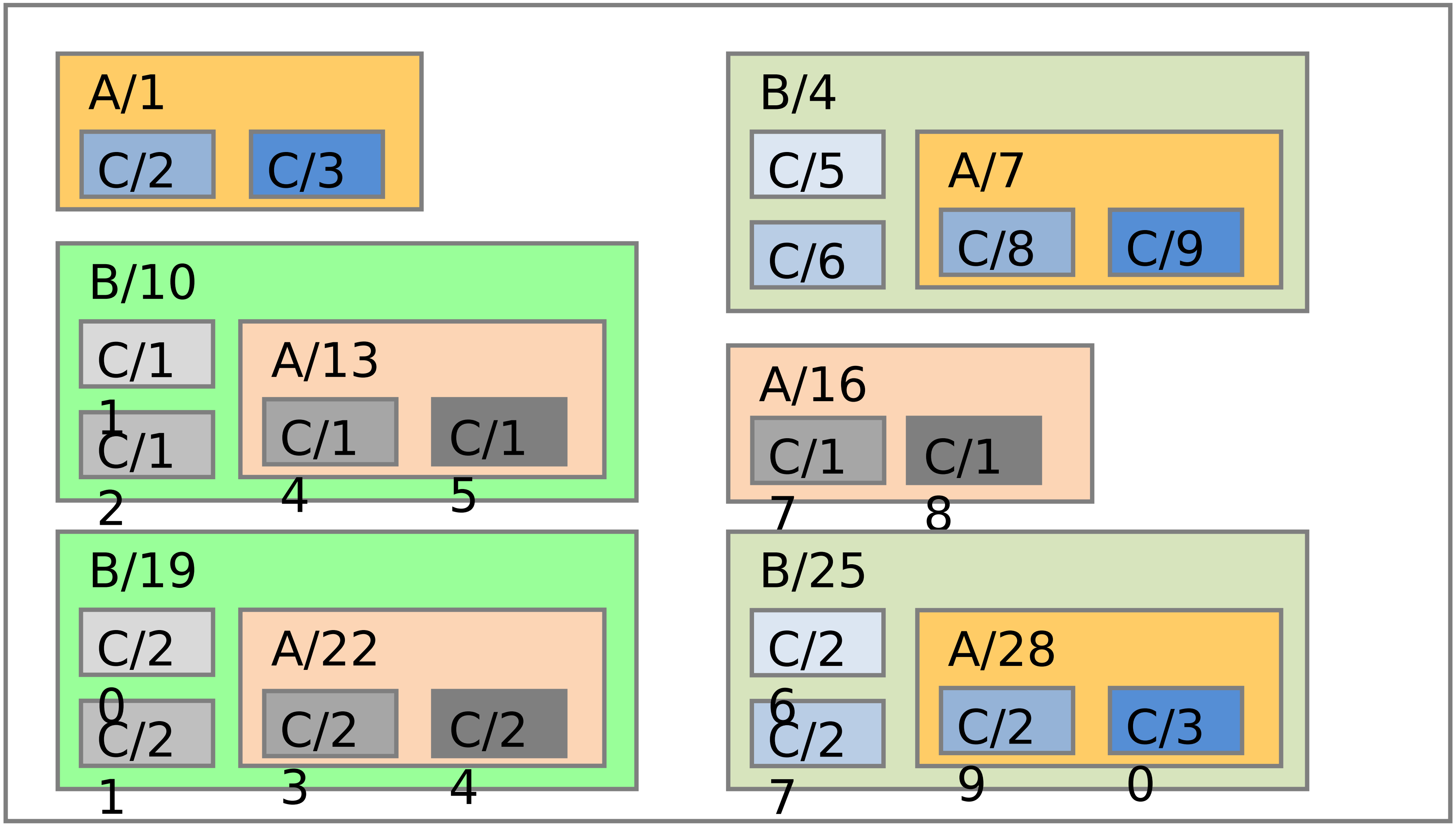
and in "C"

Hierarchical Type-Mapping



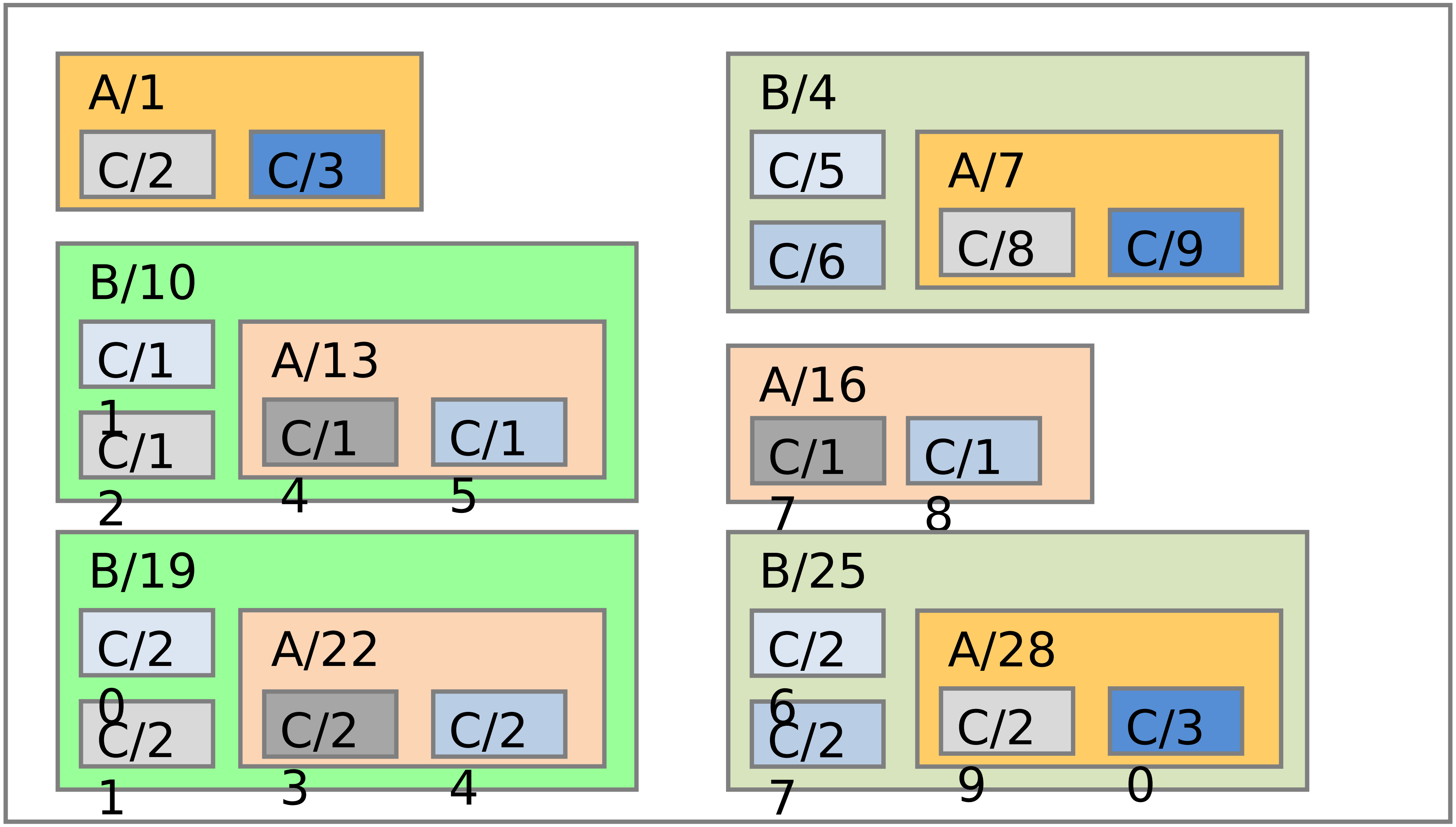
Type-Map "A"

Hierarchical Type-Mapping



This induces clusters in "C"

Hierarchical Type-Mapping



Type-Map "C": Induced clusters can be merged