ECE 260C, Spring 2025
On Manufacturability

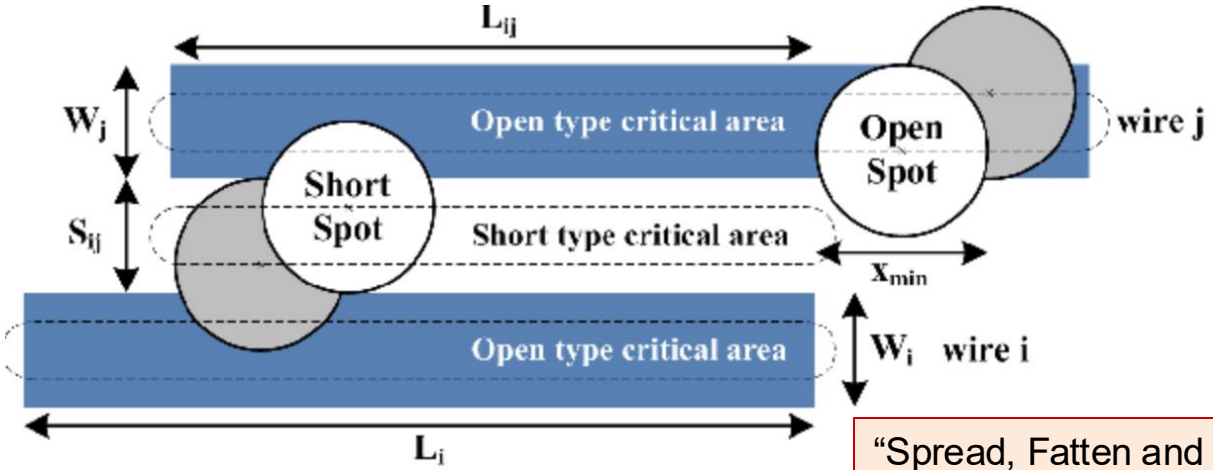
Andrew B. Kahng

"PPAC" = ?

- PPAC is why the industry continues "Moore's Law scaling"
  - Must see sufficient benefit from move to the next technology node
- Power, Performance, Area
  - Dynamic/Static power, SAIF/VCD, …
  - Incremental STA-driven sizing + buffering + synthesis operations
  - Better floorplans, better P&R tools, better "Opt" (remove overdesign)
- What about Cost?
  - Yield = fraction of chips produced that are "good" (sellable)
  - Cost of chip: (cost of fabricating, testing a wafer) / (yield * #chips/wafer)
    - Higher yield is good (**Manufacturability**); higher #chips/wafer also good (Area)
  - "good" means passing both functional and parametric criteria
    - Functional: works correctly
    - Parametric: is sellable (performs at spec, not too leaky, won't fail in a week, …)

Functional Yield

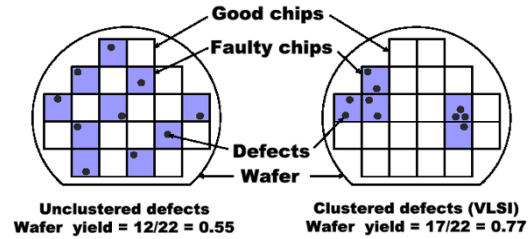
- No physical faults: “opens”, “shorts” etc.
  - DFT, ATPG, ATE, … spans EDA and Test supplier industries
- “Critical area” = vulnerability to particular defect sizes see the dashed outlines
- How much of my layout will suffer an “open” if an open spot of a particular size were to randomly appear?
  - Wider wires → layout is less vulnerable
- How much of my layout area will suffer a “short” if extra material of a particular size were to randomly appear?
  - Wider spaces → layout is less vulnerable



“Spread, Fatten and Fill”

Enormous Engineering Efforts …

- Modeling of defects
  - E.g., clustering
- Cleanroom technologies
  - E.g., factory layout, airflow, robots, …
  - "Class 1 cleanroom" etc. link
- "Design for Manufacturability"
  - E.g., to first order, make chip smaller (~cubic yield loss w/area)
  - Better layouts: "Spread, fatten, fill", avoid risky patterns, follow "recommended rules" aka "DFM rules", …
  - Redundancy / error tolerance
    - Doubled vias and contacts
    - Error detection/correction in memories
  - Better OPC / mask flows

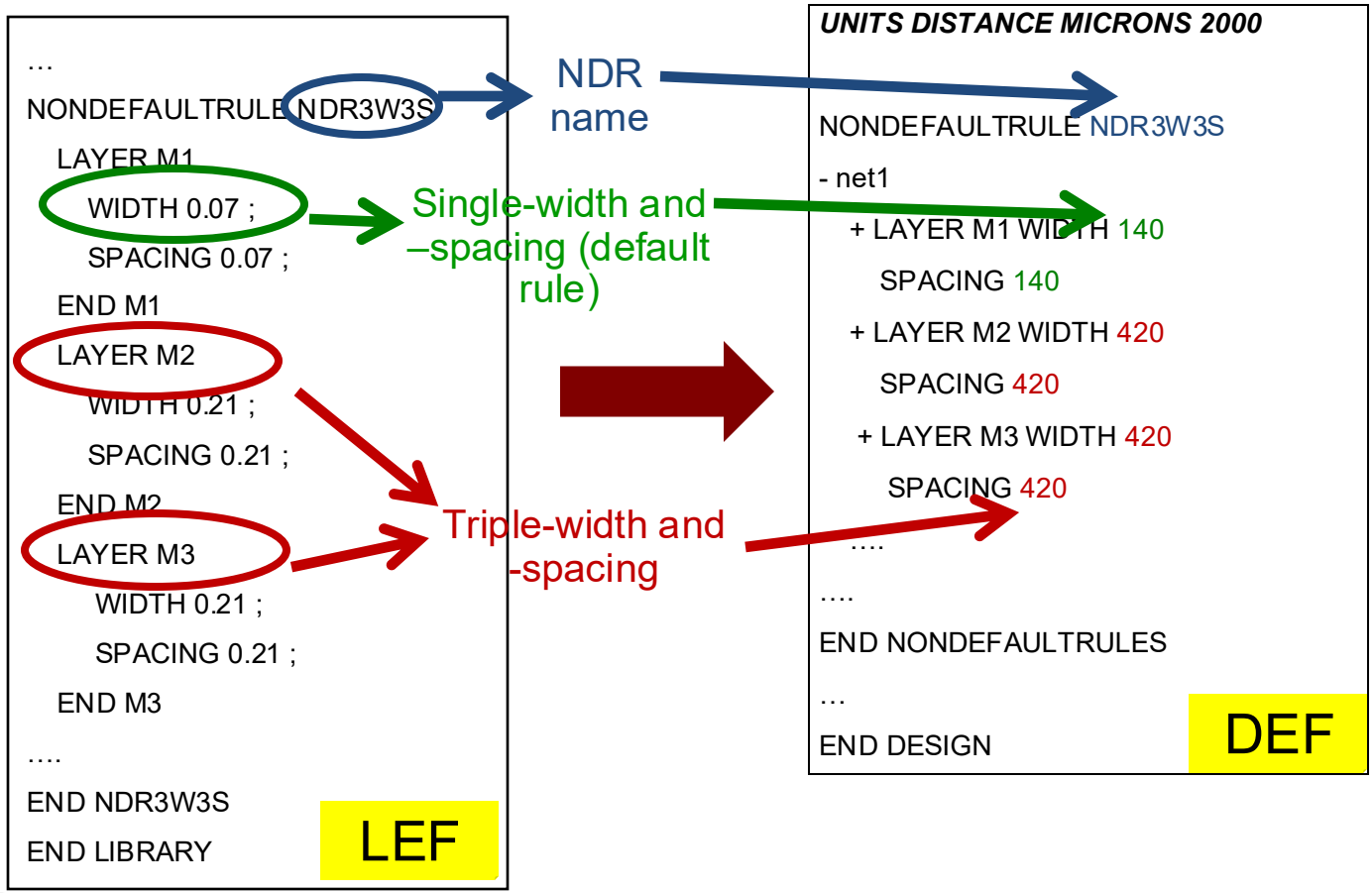


Note: hard to cleanly separate timing, power, reliability, yield …

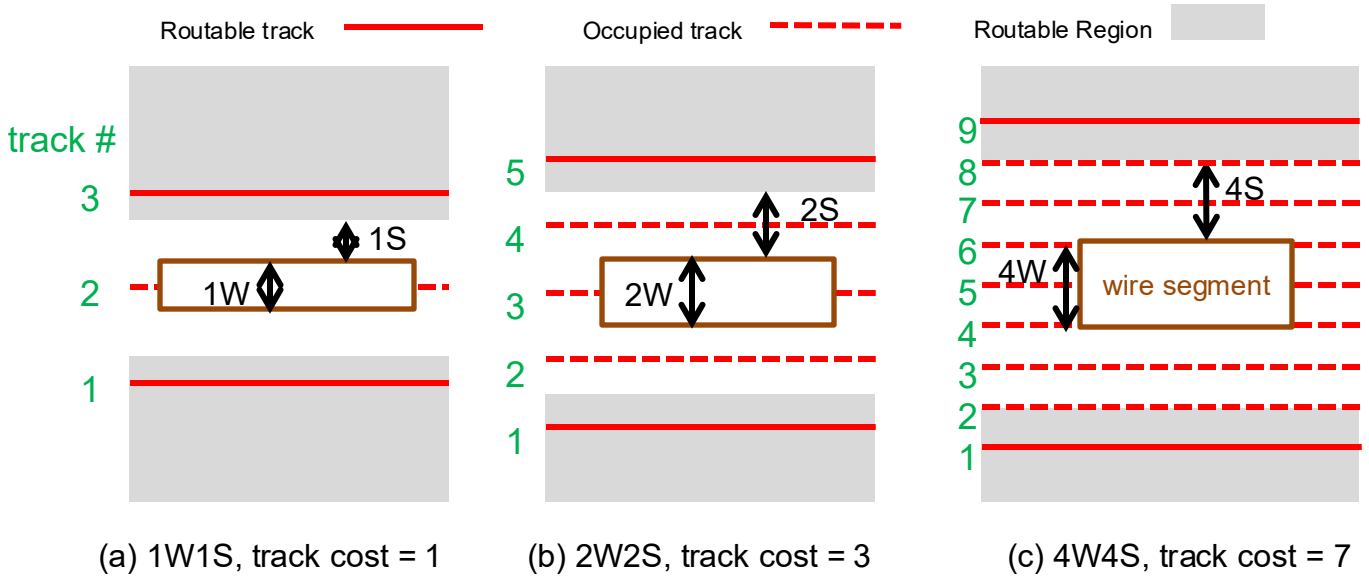
DFM → DTCO (design-technology co-optimization) → STCO link link link

NDR, Redundancy Levers

Specification of NDRs in LEF, DEF



Track Impact of NDRs



(a) 1W1S, track cost = 1 (b) 2W2S, track cost = 3 (c) 4W4S, track cost = 7

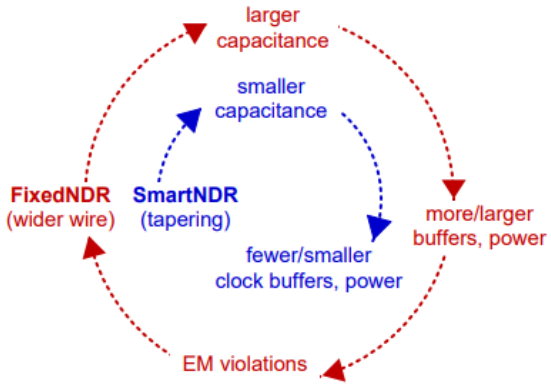
- Track cost = Number of routing tracks unavailable to the detailed router because of increase in the required spacing to neighboring wires when applying NDRs
- **Key: Router centers each wire segment on a track gridline**

Use of NDRs in Clock Routing

- Eliminate noise
  - Coupling capacitance induces delay uncertainty → increase spacing
- Prevent EM violation
  - Large buffers driving minimum-width wires violate current density limit → **wider** wires have improved EM/IR robustness
- Reduce variation
  - Smaller geometries and more resistive interconnects (and, multiple patterning steps) induce large variations of parasitics and delays → **wider** wires have less variation
- Reduce power (complementary: reduce skew)
  - Wide wires have larger capacitance and increase dynamic power → motivates “Smart NDRs” (e.g., by wire tapering) to reduce capacitance

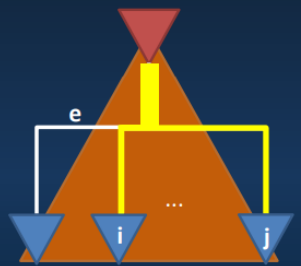
Smart NDRs for Clock Power Reduction link link

- Vicious vs. Virtuous cycles: Fixed NDRs vs. Smart NDRs



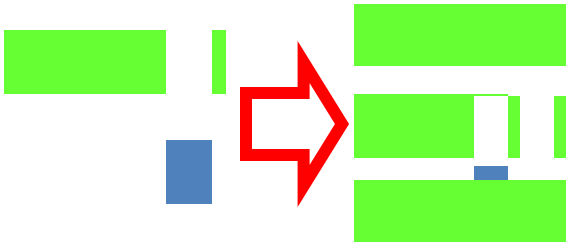
- Can optimize wire segment widths in a clock subnet

Minimize total capacitance
Subject to
$Delay_i < MaxDelay$
$Skew_{i,j} < MaxClockSkew$
$I_e < MaxEMCurrentLimit_e$
$w_e \geq w_{desc(e)}$
(*desc(e)* ≡ edges downstream from *e*)



Via Doubling For Redundancy

- Single-cut via →
  Double-cut via ("via doubling")
  - During Routing?
  - or After Routing?

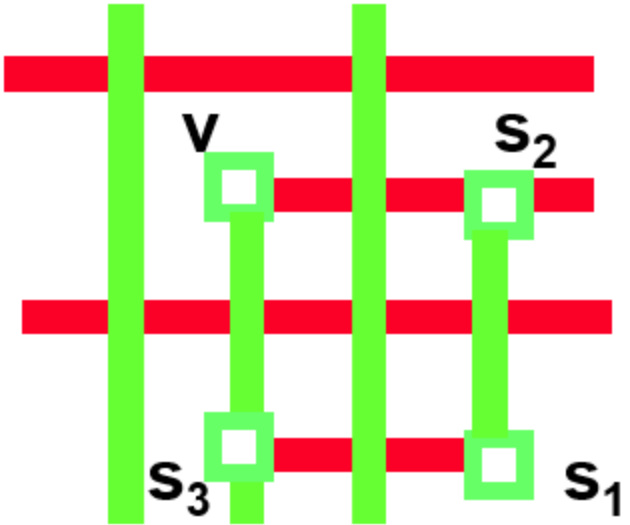


- Why redundant vias?
  - Improve yield: redundant via is "insurance" for via opens
  - Improve timing yield: the resistance of partially blocked vias increases
  - Reservoirs of metal for electromigration reliability
  - Etc.

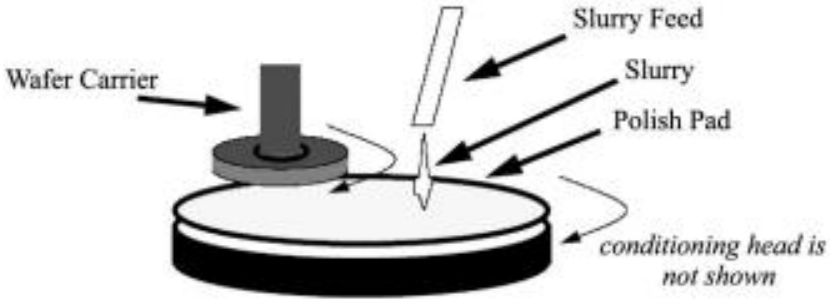
- Observation:
  - The first 70-80% of vias can be easily doubled even for very congested designs

Short-Loop Paths to Add Redundancy

- Short-loop paths
  - Create detours to double vias which cannot otherwise be doubled with an adjacent single via (figure: via "v" is made redundant with a short loop)
  - Up to 97.5% via doubling coverage by IBM Boeblingen in early 2000's
    - May lead to antenna / timing issues



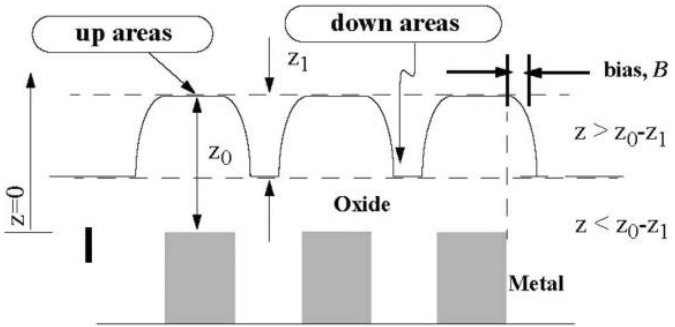
CMP Fill



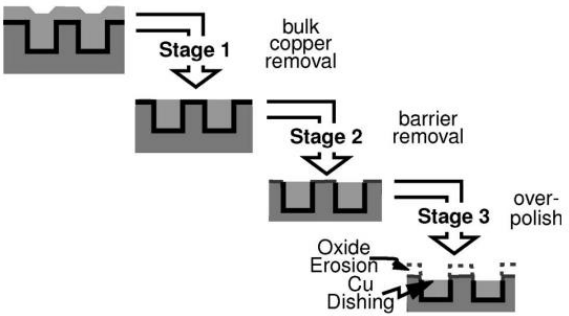
https://vlsicad.ucsd.edu/Publications/Journals/j40.pdf
https://vlsicad.ucsd.edu/Publications/Journals/j84.pdf

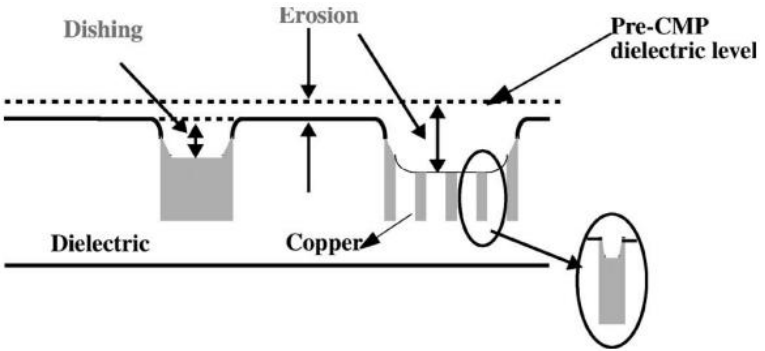
Planarization = Flat Wafer Topography

- Planarize oxide
  - Starting topography depends on layout pattern
  - Figure: D. Ouma, MIT 1998



- Planarize copper
  - Multiple materials
  - **Erosion** of dielectric, **dishing** of wide (soft) metal → "slotting" needed
  - Figures: T. Tugbawa, MIT 2002

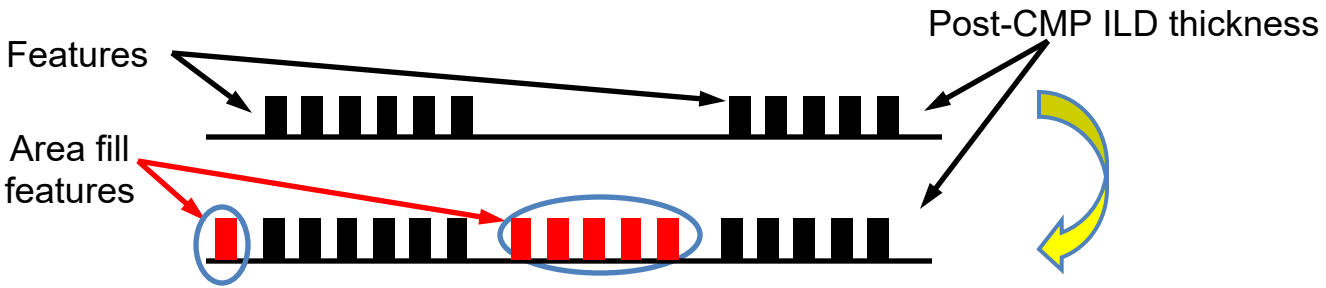




Manufacturability: Layout Density Control ("Fill")

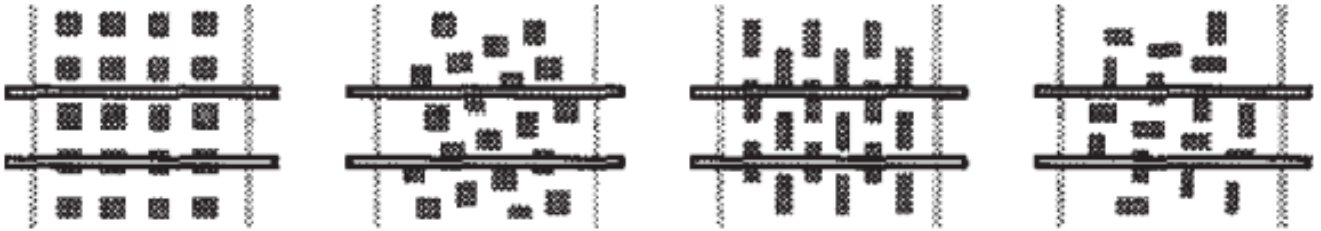
- Area fill or "dummy fill": electrically inactive; floating or grounded
- Area fill insertion (and, slotting)
  - Decreases local density variation (→ area density design rules)
  - Decreases post-CMP interlayer dielectric (ILD) erosion, conductor dishing
    - CMP: Chemical-Mechanical Polishing (Planarization)
  - Cf. "Filling and Slotting: Analysis and Algorithms", ISPD-98

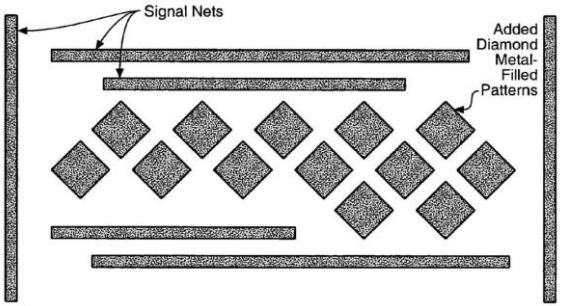
Must satisfy min, max area density limits in all windows of given size and "step" (e.g., density between 20-80% in 100um x 100um windows, with step size = 25um)

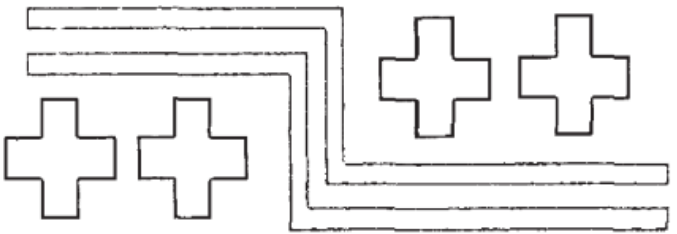


Other Examples https://vlsicad.ucsd.edu/Publications/Journals/j84.pdf

- Top (l-r): traditional, staggered, alternating, basket-weave why?
- Bottom: diamond and plus-sign patterns







OpenROAD: “max”, “fixed pattern” (types)

https://openroad.readthedocs.io/en/latest/main/src/fin/README.html#example-scripts

- “fin” = layout finishing
- Metal fill can be part of output gds (foundry can insert fill as well)

The rules `json` file controls fill and you can see an example here.

The schema for the `json` is:

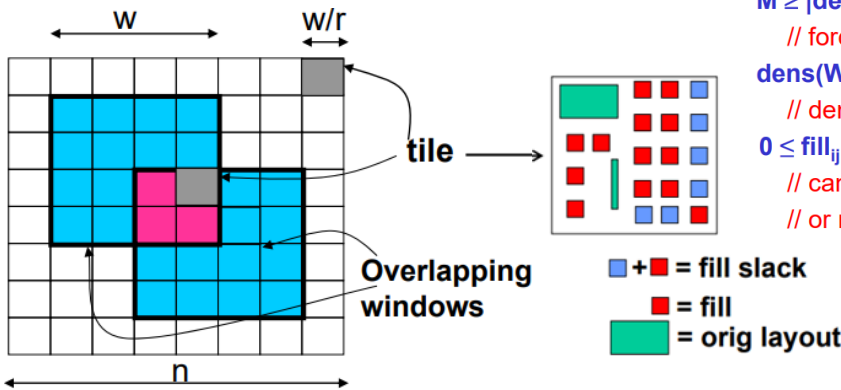
```
{
  "layers": {
    "<group_name>": {
      "layers": "<list of integer gds layers>",
      "names": "<list of name strings>",
      "opc": {
        "datatype":  "<list of integer gds datatypes>",
        "width":   "<list of widths in microns>",
        "height":   "<list of heightsin microns>",
        "space_to_fill": "<real: spacing between fills in microns>",
        "space_to_non_fill": "<real: spacing to non-fill shapes in microns>",
        "space_line_end": "<real: spacing to end of line in microns>"
      },
      "non-opc": {
        "datatype":  "<list of integer gds datatypes>",
        "width":   "<list of widths in microns>",
        "height":   "<list of heightsin microns>",
        "space_to_fill": "<real: spacing between fills in microns>",
        "space_to_non_fill": "<real: spacing to non-fill shapes in microns>"
      }
    }, ...
  }
}
```

SKY130 M5 example: fixed pattern = fixed density of 42.5% (how did I see this?)

```
63          "met5" : {
64              "layer":                59,
65              "name":                 "met5",
66              "dir":                  "H",
67              "datatype":              0,
68              "space_to_outline":      70,
69              "non-opc": {
70                  "datatype":         0,
71                  "width":            [3.00],
72                  "height":           [3.00],
73                  "space_to_fill":    1.6,
74                  "space_to_non_fill":    3.0
75              }
```

"Smart Fill" https://vlsicad.ucsd.edu/Publications/Journals/j40.pdf

- Note: **many** ways that "dummy fill" can be smarter
  - Less data volume (GDS, OASIS compressibility)? Friendly to IP reuse? Smooth density gradient? Grounded fill for PI/SI? Timing-aware,-fixing? Control both area and perimeter of metal layout? (see papers at https://vlsicad.ucsd.edu)
- One key type of "smarter": don't add a maximum amount of fill !!!
  - Capacitance = delay and power; data volume considerations as well
- "Optimal" fill: minimize "**M**" in the linear program
  https://vlsicad.ucsd.edu/courses/cse101-w18/slides-w18/cse101-w18-Lecture13-linearprog.pdf



$M \geq |dens(W_i) - dens(W_j)|$ **for all windows i,j**
// force minimum variation of post-fill window densities

dens(W) = density(orig layout + fill) in all tiles of W
// density of a window = original + added fill

$0 \leq fill_{ij} \leq slack_{ij}$ **for all tiles** T_{ij}
// cannot put more fill than will fit (i.e., more than "slack")
// or negative amount of fill into any window

"Fill slack" = maximum amount of fill that can possibly be added into a given "file"

OpenRCX

Thanks: Dimitris Fotakis and David Overhauser

Capacitance Extraction

- What is distribution of charge on conductor surfaces and in dielectrics?
  - Solve Maxwell's equations → electric field and potential everywhere
- C = Q/V → obtain capacitance values
- Slow and accurate: 3D field solvers
- Quick and dirty: 1D (per-unit length total cap)
  or 2D (per-unit area cap + per-unit length lateral, fringing cap)
- Good in practice: 2.5D ≈ 2D + corrections for adjacent layers

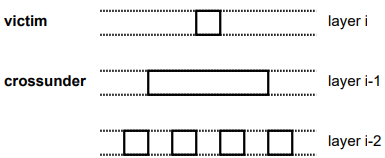
- OpenRCX is a 2.5D method

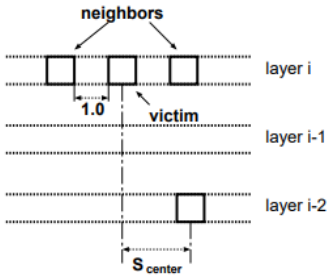
- Key challenge: building the (lookup table, interpolation) model for "corrections …"

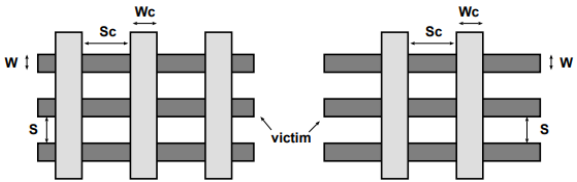
DAC 1997: “Five Foundations”

https://vlsicad.ucsd.edu/Publications/Conferences/69/c69.pdf

- Foundations (1997) for lookup tables and interpolations
  1. Shielding effects of ground, and same-layer neighbors
  2. High metal density on Layer i (e.g., with dummy fill !) decouples Layers i-1, i+1
  3. Layers i ± 2 can be treated as ground planes; no need to look beyond
  4. Left/right same-layer couplings can be treated independently, and neighbor widths have little impact
  5. Above/below crossunders can be treated independently
- Systematic design of experiments (patterns), use of field solvers







Can Go Much Deeper (e.g., Fill Impact)

https://vlsicad.ucsd.edu/Publications/Conferences/205/c205.pdf

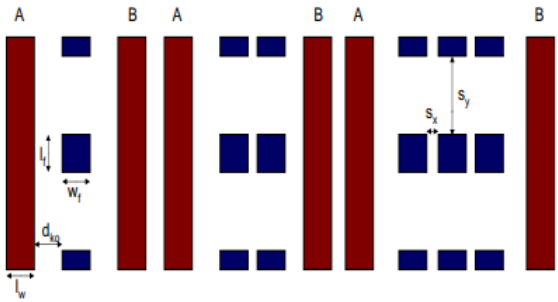


Fig. 12. Floating fill pattern examples.

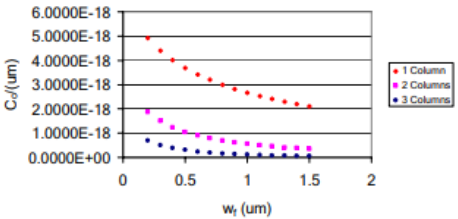


Fig. 15. Variation of coupling capacitance between active interconnects with respect to fill width.

IV. Impact of CMP Fill on Interconnect Performance

Chemical-mechanical planarization enables multilayer interconnect architectures. To enhance uniformity of post-CMP wafer topography, dummy fill is inserted to improve the uniformity of (effective) feature density. However, while dummy fill insertion improves feature density uniformity, it also changes coupling and total capacitance of functional interconnects [7], [11], [12].

In this section, we first study the impact of floating and grounded metal fill shapes on coupling (C_c) and total (C_{tot}) interconnect capacitance, in the context of various metal and fill configurations. We then motivate and outline an "intelligent fill synthesis" methodology that may be appropriate in future technology nodes.

A. Modeling of Fill Patterns

In the following discussion, the following notation is used.

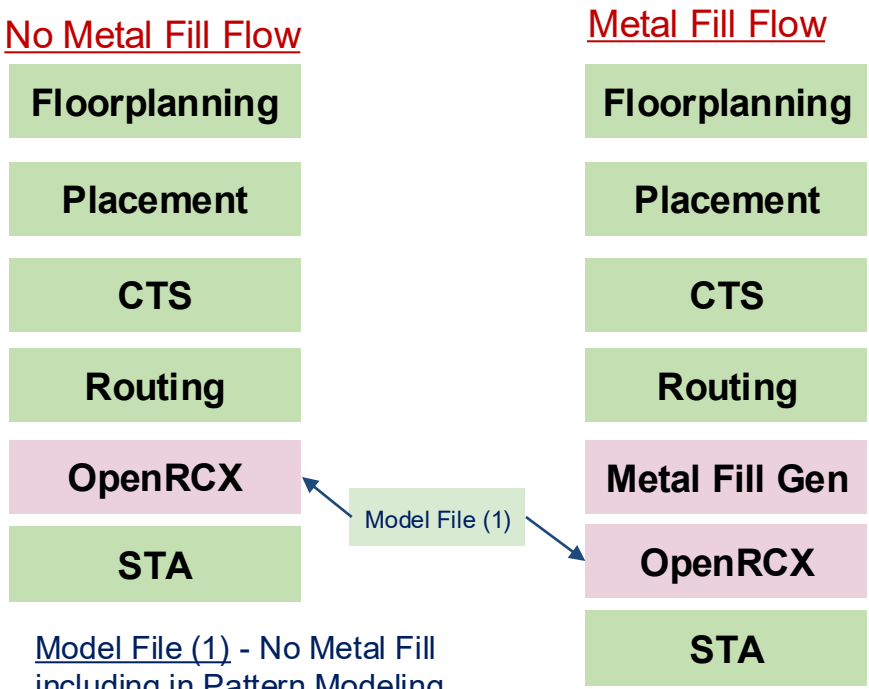
- w_m: Line width
- h_m: Metal height
- h_{ild}: Dielectric height
- l_f: Fill length
- w_f: Fill width
- h_f: Fill height
- s_x: Horizontal spacing between fill features
- s_y: Vertical spacing between fill features
- d_{ko}: Keep-out distance

In Figure 20 we sketch a practical approach to *intelligent* timing-driven fill.

Timing-Driven Fill
Loop:
0. Set an initial conservatism factor
1. Do (initial) RCX and STA
2. Identify timing-violating nets (TVNs) - i.e., timing-critical nets
3. Apply conservative net-protection (+keep-out distance and blocking $M+1/M-1$ layers) per TVN segment
4. Run (incremental) MC-Fill → target fill amount
5. PIL-FILL Synthesis:
5.1 Greedy insert fill in fill slack columns, targeting most-needy tiles and largest and largest-slack nets first
5.2 After K fill shapes have been inserted, re-run (incremental) STA based on ΔC's
5.3 Iterate until all required fill has been inserted (or, until no timing constraint looks safe)
6. Update Conservatism
6.1 Analyze windows that violate *min* constraints
6.2 Identify nets that belong to the windows that violate the constraints
6.3 Do (incremental) RCX and STA to change the conservatism factor of TVNs (return to Step 2)

Fig. 20. Timing-driven fill synthesis approach.

OpenRCX in the Design Flow



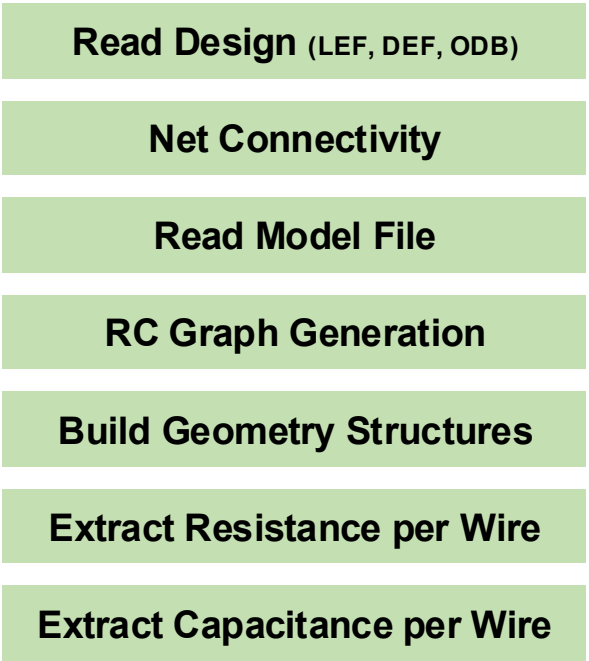
Model File (1) - No Metal Fill including in Pattern Modeling

Note 1: if metal fill is present in the design it will be extracted as grounded wires.

Note 2: The challenge is that metal fill consideration is much larger area than pattern size; existence of fill must be either explicit or built in the extractor

Thanks: Dimitris Fotakis and David Overhauser

OpenRCX Internal Flow



Thanks: Dimitris Fotakis and David Overhauser

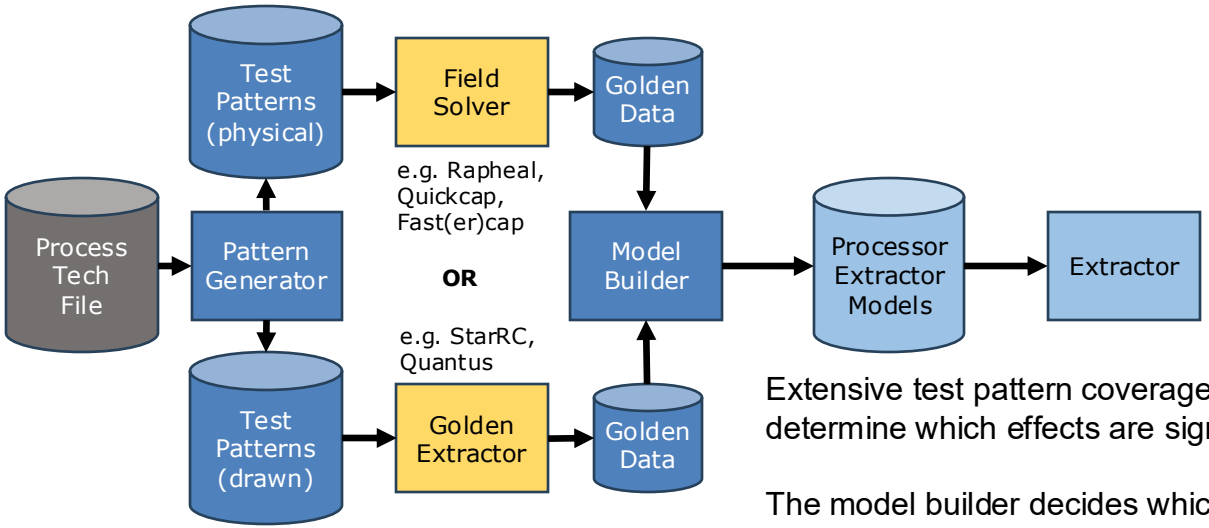
Extraction Model Creation

(2.5D) Extraction Model Creation

- Determine the characteristics of the process from the foundry technology information (e.g. itf or ict file)
  - Layer specs, wire shape, dielectrics stack, WEE tables, Thickness tables, Rho tables, …
- Create a large number of patterns of interconnect and transistor-level test structures
  - Interconnect-level test structures differ from transistor-level structures
- Run each pattern through a golden reference extractor
  - Field solver requires proper shape, enlargements, thickness and dielectrics
- Based on golden reference results, build extraction models
- Verify model results versus test results and design layouts

Thanks: Dimitris Fotakis and David Overhauser

Golden Reference Tool to Build Models



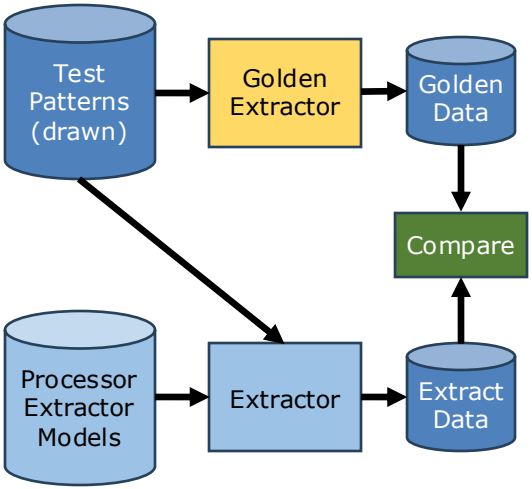
Extensive test pattern coverage helps determine which effects are significant

The model builder decides which effects need to be modeled in the extractor

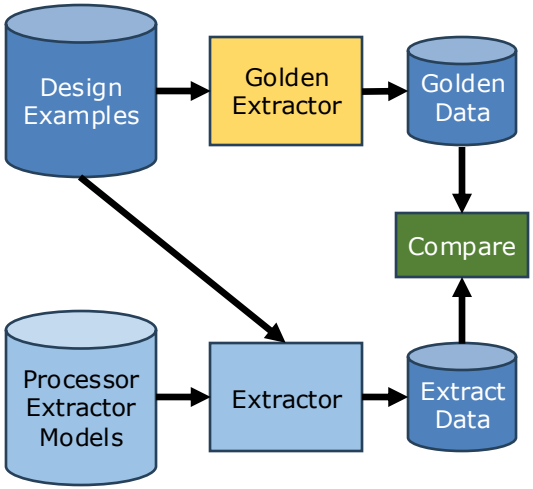
Note: OpenRCX does NOT include explicit Metal Fill in the modeling patterns

Thanks: Dimitris Fotakis and David Overhauser

Verifying Models



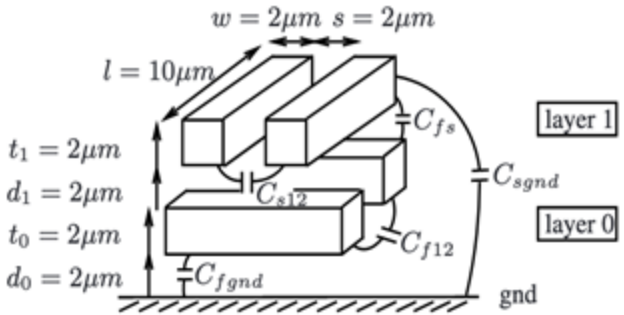
Verify using test patterns (once per process)



Verify using sample designs

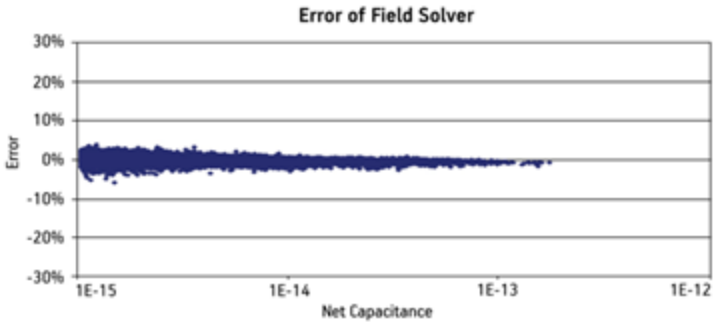
Thanks: Dimitris Fotakis and David Overhauser

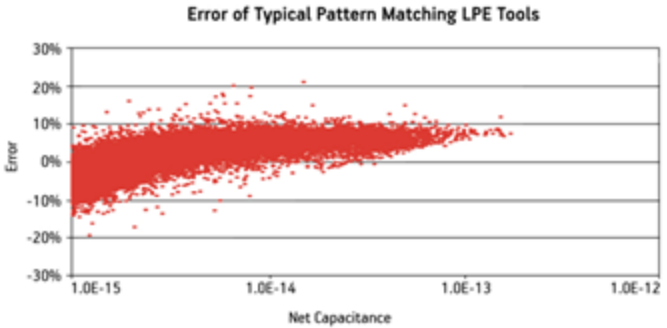
Extraction Accuracy Plots



Field solver error is generally versus a golden reference field solver (e.g. Quickcap vs. Raphael)

Layout Parameter Extractor (LPE) results are compared versus a field solver and LPE models are generally skewed positive so that net capacitance is never underestimated





Thanks: Dimitris Fotakis and David Overhauser

OpenRCX Calibration Patterns

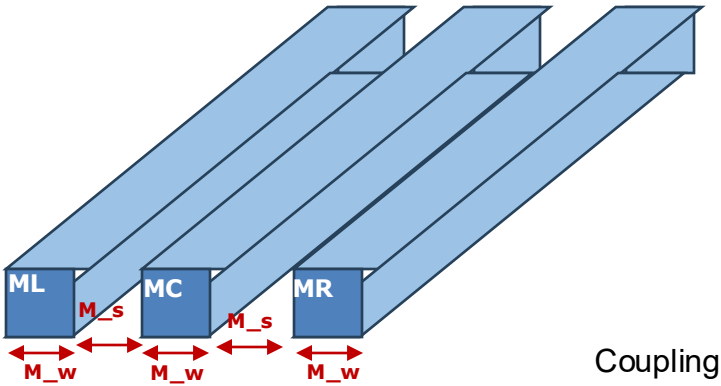
- Structures to calibrate OpenRCX are relatively simple
  - Designed for ~65nm node, in ~2004 !
- Coupling capacitances are computed only for same-layer neighbors and parallel diagonal wires
- Layers above or below are not necessarily adjacent layers (!)

- Min width-centric extraction rules are based on LEF
  - Non-min width only used on power nets, not signal nets
  - Non-min width infrastructure added in November 2024

- Resistance was also based on LEF, not calculated
  - Resistance as function of width and spacing added in November 2024

Thanks: Dimitris Fotakis and David Overhauser

OpenRCX Pattern A



Measure
CC_MC_ML
CC_MC_MR
CC_MC_gnd

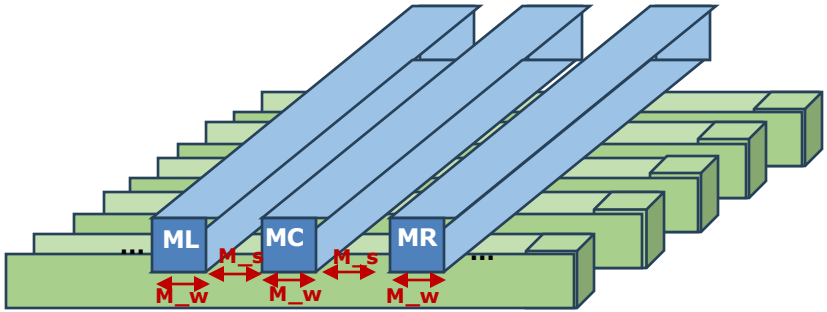
R_MC

Coupling Cap = f(M_w, M_s, M)
Grounded Cap = f(M_w, M_s, M)

R_MC

Thanks: Dimitris Fotakis and David Overhauser

OpenRCX Pattern B

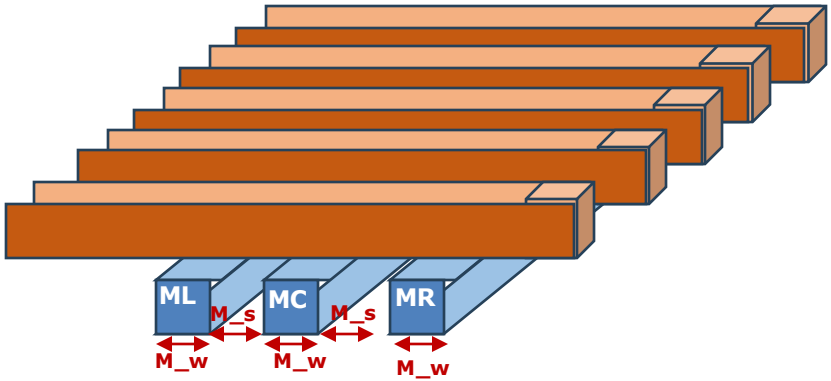


Measure
CC_MC_ML
CC_MC_MR
CC_MC_Munder
CC_MC_gnd

Coupling Cap = f(M_w, M_s, M, Munder)
Grounded Cap = f(M_w, M_s, M, Munder)

Thanks: Dimitris Fotakis and David Overhauser

OpenRCX Pattern C

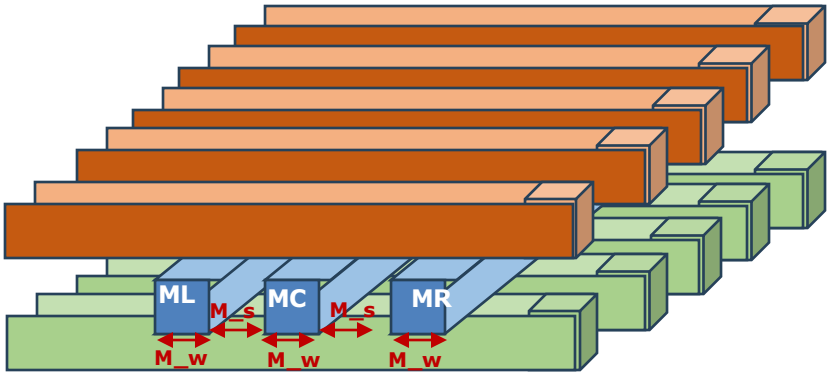


Measure
CC_MC_ML
CC_MC_MR
CC_MC_Mover
CC_MC_gnd

Coupling Cap = f(M_w, M_s, M, Mover)
Grounded Cap = f(M_w, M_s, M, Mover)

Thanks: Dimitris Fotakis and David Overhauser

OpenRCX Pattern D

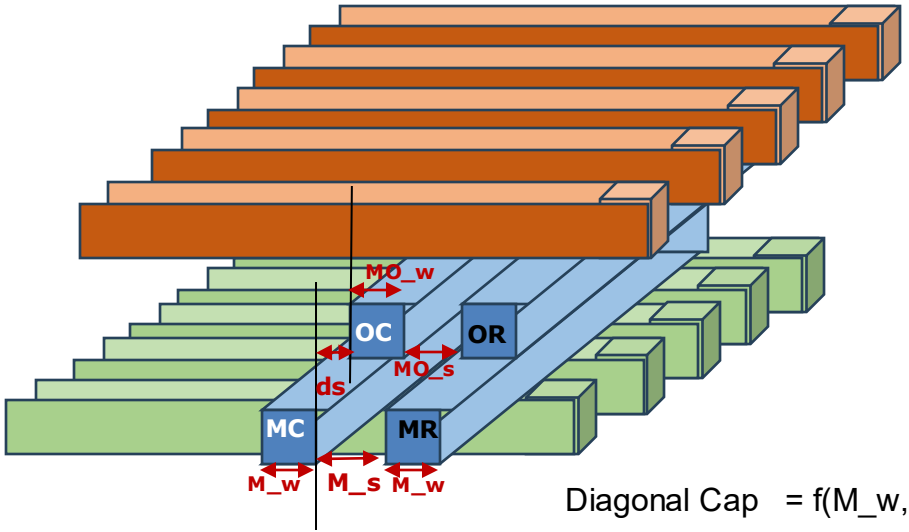


Measure
CC_MC_ML
CC_MC_MR
CC_MC_Mover
CC_MC_Munder
CC_MC_gnd

Coupling Cap = f(M_w, M_s, M, Mover, Munder)
Grounded Cap = f(M_w, M_s, M, Mover, Munder)

Thanks: Dimitris Fotakis and David Overhauser

OpenRCX Pattern E



Measure
CC_MC_OC
CC_MC_gnd

Diagonal Cap = f(M_w, M_s, ds, M, Mover)

Thanks: Dimitris Fotakis and David Overhauser

Potential Roadmap …

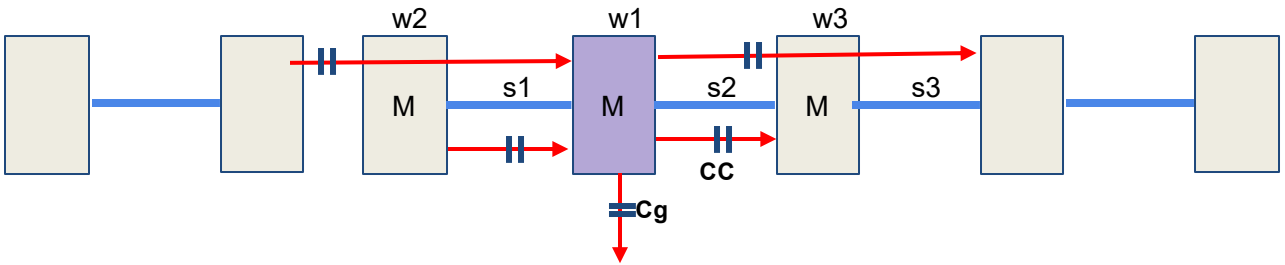
(Deeper dive in May 29 guest lecture: Dimitris Fotakis)

- Multi-width and variable spacing
  - OpenRCX is written to support this
  - Extraction rules need to be expanded
- Coupling capacitance to secondary neighbors
  - OpenRCX and rules still need to support
- Resistance dependent on neighbor
  - Merged in November 2024
- Lithography effects
  - Handled at model creation (drawn to physical) for field solver

Thanks: Dimitris Fotakis and David Overhauser

Non-Symmetric Coupling to 1st, 2nd Neighbors

Not currently implemented, but infrastructure exists as of November 2024
(see extFlow_v2, extmain_v2 etc.)

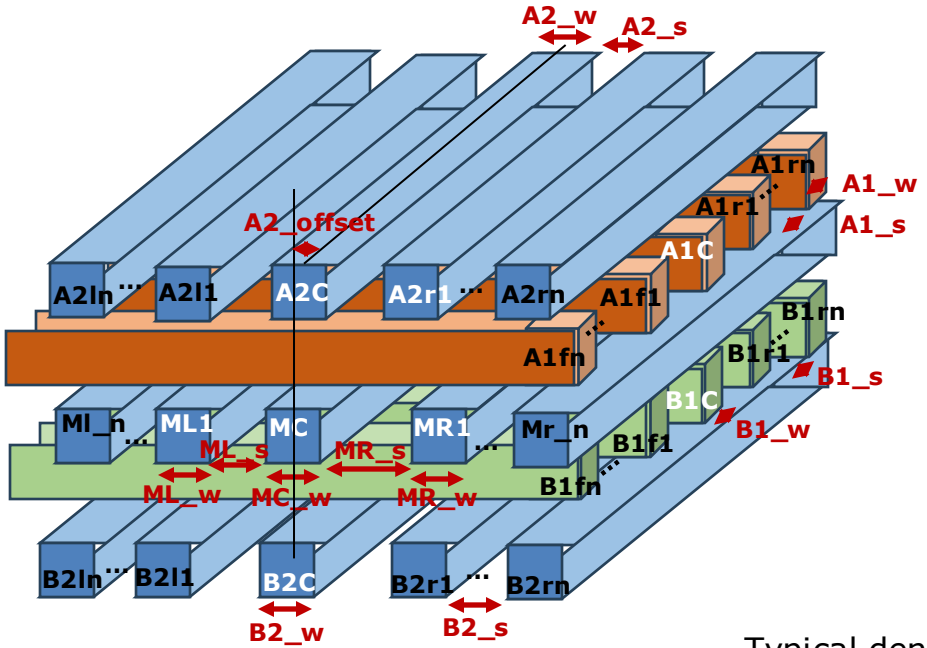


Coupling Cap= function(w, s, M), height is set

Fringe (Ground) Cap= function(w, s, M)

Thanks: Dimitris Fotakis and David Overhauser

More? E.g., HVH Routing



Measure

CC_MC_ML1
CC_MC_MR1
CC_MC_A1C
CC_MC_A2C
CC_MC_B1C
CC_MC_B2C
CC_A1C_B1C
CC_MC_A2r1

R_MC

Typical density area $(50um)^2$

Thanks: Dimitris Fotakis and David Overhauser