Analysis is from Venus, Synthesis from Mars

Patrick Groeneveld
Chief Architect, Magma Design Automation

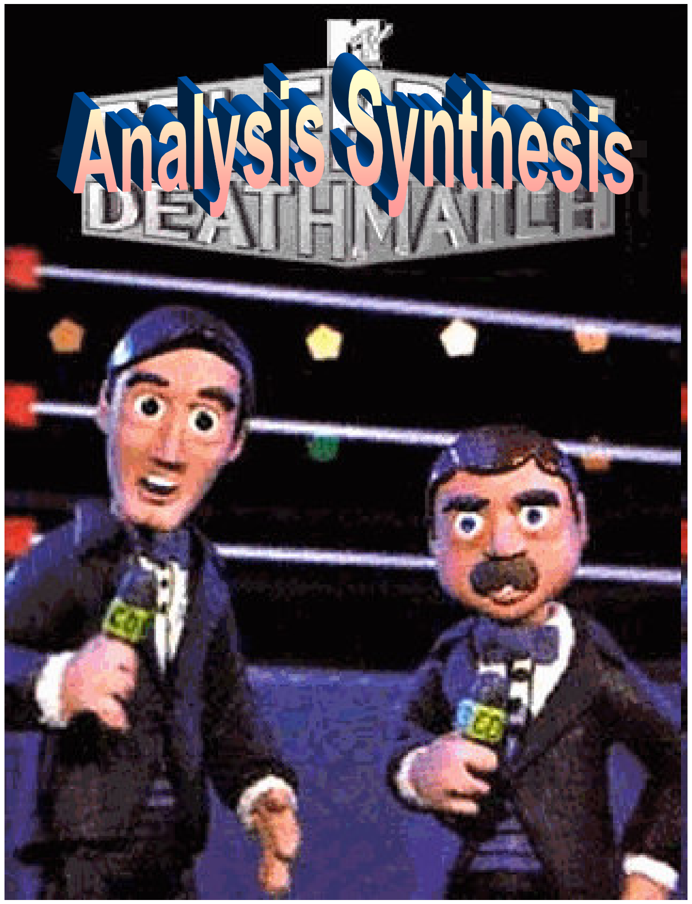
EDP 2007, Monterey

Summary

- **Focus on physical *synthesis* ASIC flow**
  - With given process technology, what can Physical Design Tools to handle DFM&Y?
- **Engineering principles**
- **Engineering a flow that improves Manufacturability and yield**
  - CMP
  - OPC
  - CAA
  - SSTA
- **Recommendations**

Analysis is from Venus, Synthesis from Mars

- Analyzes properties
- Many established models.
- Big and slow
- Highly accurate
- Each objective measurable
- Fully automatic
- Can’t fix anything

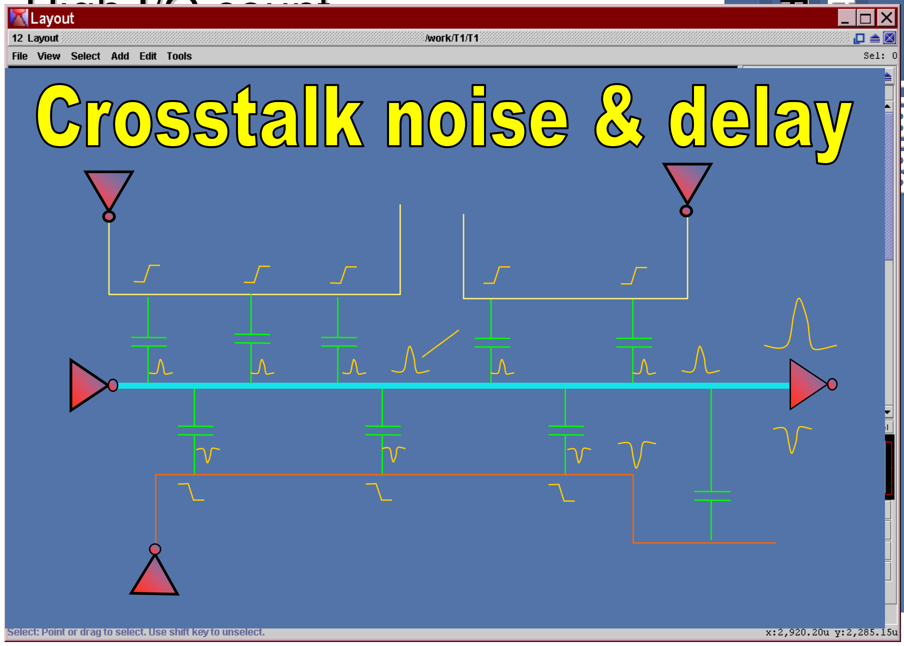


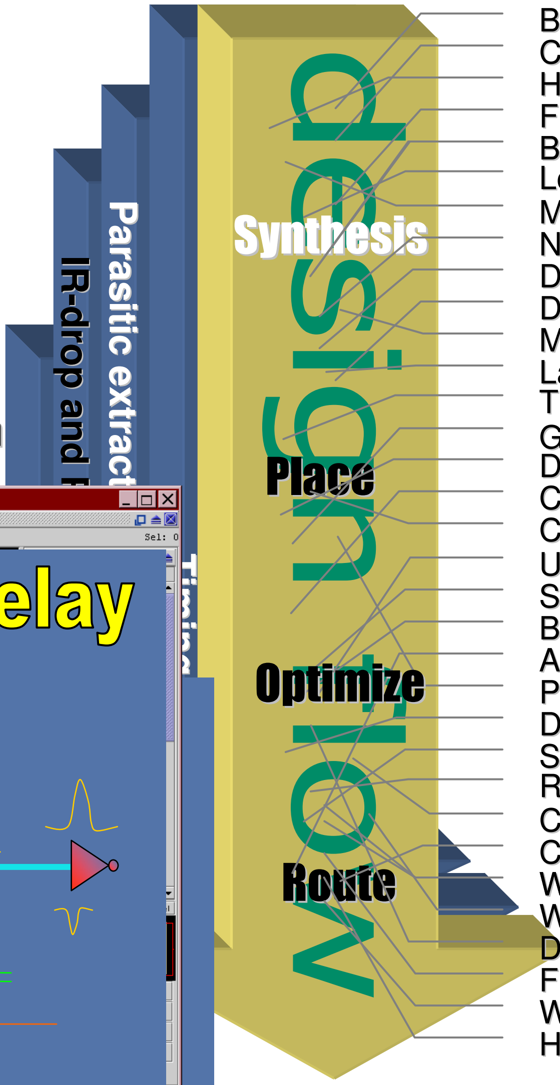
- Synthesizes things
- Very few models/methods used.
- Leaner, Faster
- Very poor accuracy
- Difficult handle multiple objectives.
- Needs manual help
- Is the builder

MAGMA

The anatomy of a Physical Synthesis flow

Timing closure (parasitic cap.)
Routing closure
Design scale, concurrent design
Testability
ECO capability
Clock skew
Low power requirements
IR voltage drop, Electromigration
High I/O count


Layout
12 Layout
/work/T1/T1
File View Select Add Edit Tools
Sel: 0
Crosstalk noise & delay
Select: Point or drag to select. Use shift key to unselect.
x:2,920.20u y:2,285.15u



BIST insertion
Clock gating
Hierarchy, Partitioning, design planning
Flip-chip packaging
Block/macro placement
Load buffering
Mapping for speed
Noise buffering
Diode insertion
Decoupling caps, package design
Multi-VDD regions
Large capacity and fast algorithms
Timing/sizing driven placement
Gate sizing
Delay buffering
Cloning, logic restructuring
Congestion control
Useful skew clock synthesis
Spare cell insertion
Balanced clock trees
Antenna-friendly routing, jumper insertion
Power infrastructure
Dual-hierarchy support
Scan chain reordering and routing
Rip-up and reroute
Correct-by-construction tools
Clock shielding
Wire spacing
Wire widening
Dual Vt support
Filling, slotting, router adaptations
Wire shielding
Hold time buffering

MAGMA

DFM and Yield are important, BUT…

- **Other objectives are relevant as well**
- **Area (cost)**
  - Bigger area avoids most DFM issues!
- **Timing performance**
  - X-talk
- **Correctness, Testability**
- **Design effort, complexity**
- **Power**

Must find reasonable trade-off

MAGMA

The truth about Physical Design Automation

Synthesis Algorithms do only *one* thing well

Cannot handle multiple objectives

System is easily over-constrained

Algorithms must use *inaccurate models* of the physical reality

Algorithmic steps do things that could cause problems at later steps

We often need to start over iterate to recover such errors

MAGMA

The ABC of a well-engineered IC design flow

A: Avoid

Detect specific problem patterns early, fix them

- Relies on prediction which
- does not have to be extremely accurate.

B: Build

Synthesize using an algorithm on a simplified model.

- Capture 1st order effect of problem as objective.
- Shoot in the ball park, and hope for the best.

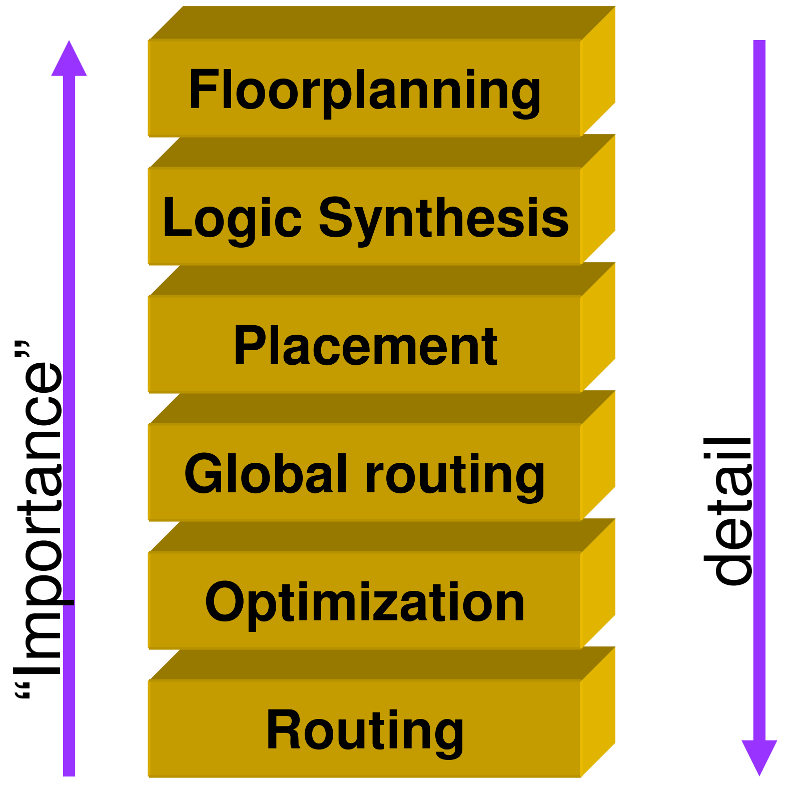
C: Correct

Perform accurate analysis, detect remaining problems and fix any problems by local modifications (ECO).

- This is typically slow and it
- might not work.
- If its real bad, iterate back to step A or B

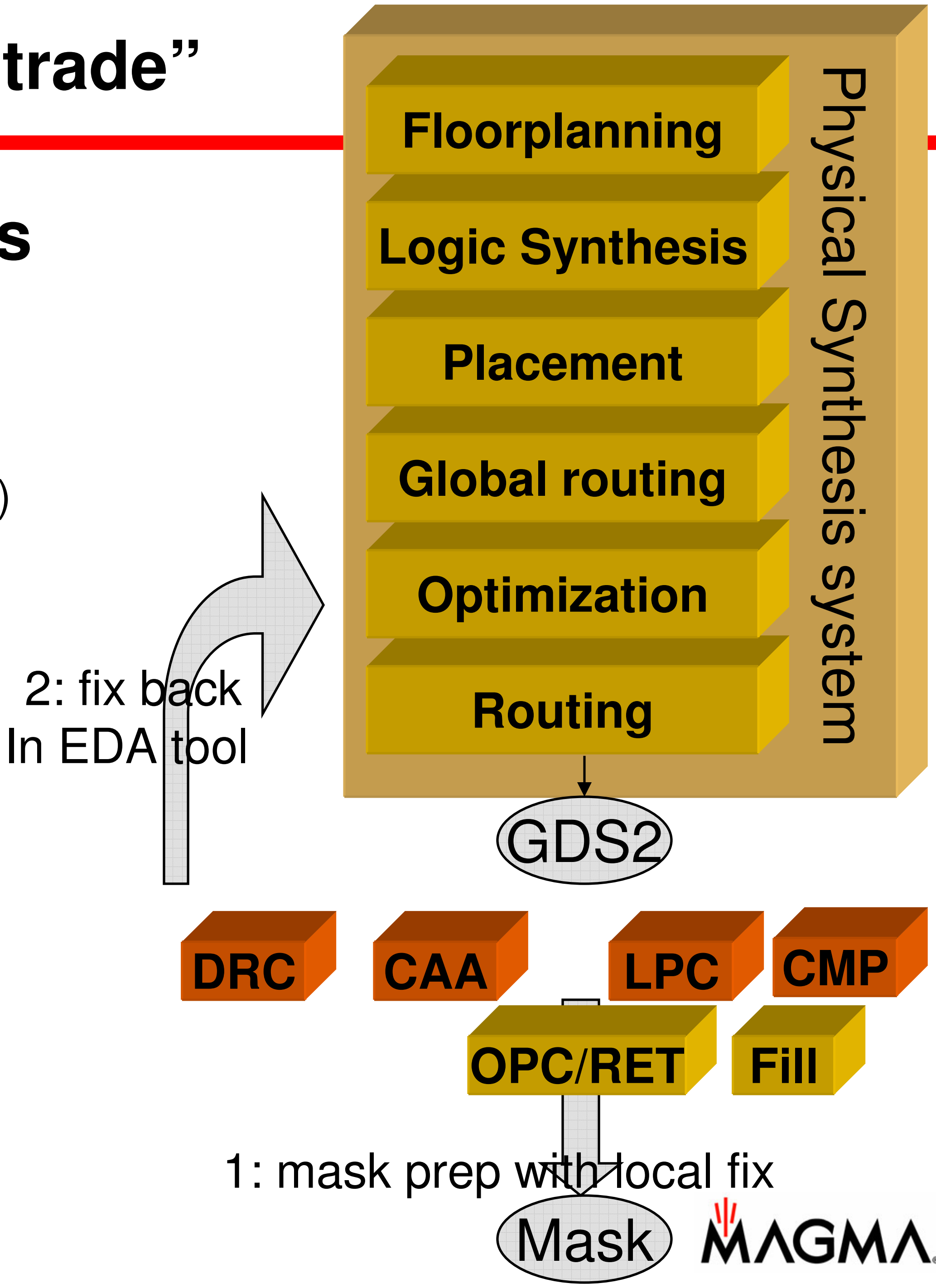
Guiding principles during Physical Synthesis

- **Stepwise refinement**
  - Use a number of build steps, each fixing an objective and adding detail
- **Avoid Correction iteration like the plague**
- **Use *in*accurate analysis**
  - Ballpark is enough, You're far off anyway
- **Keep sign-off levels alive**
  - Despite attacks



The tools of the "DFM trade"

- **Layout-level Analysis**
  - DRC (Design rule checker)
  - CAA (Critical Area Analysis)
  - LPC (Litho Shape Simulation)
  - CMP (Thickness Simulation)
  - SSTA (Statistical Timing)
- **Synthesis**
  - Pessimism and a
  - Bunch of hacks around existing tools…



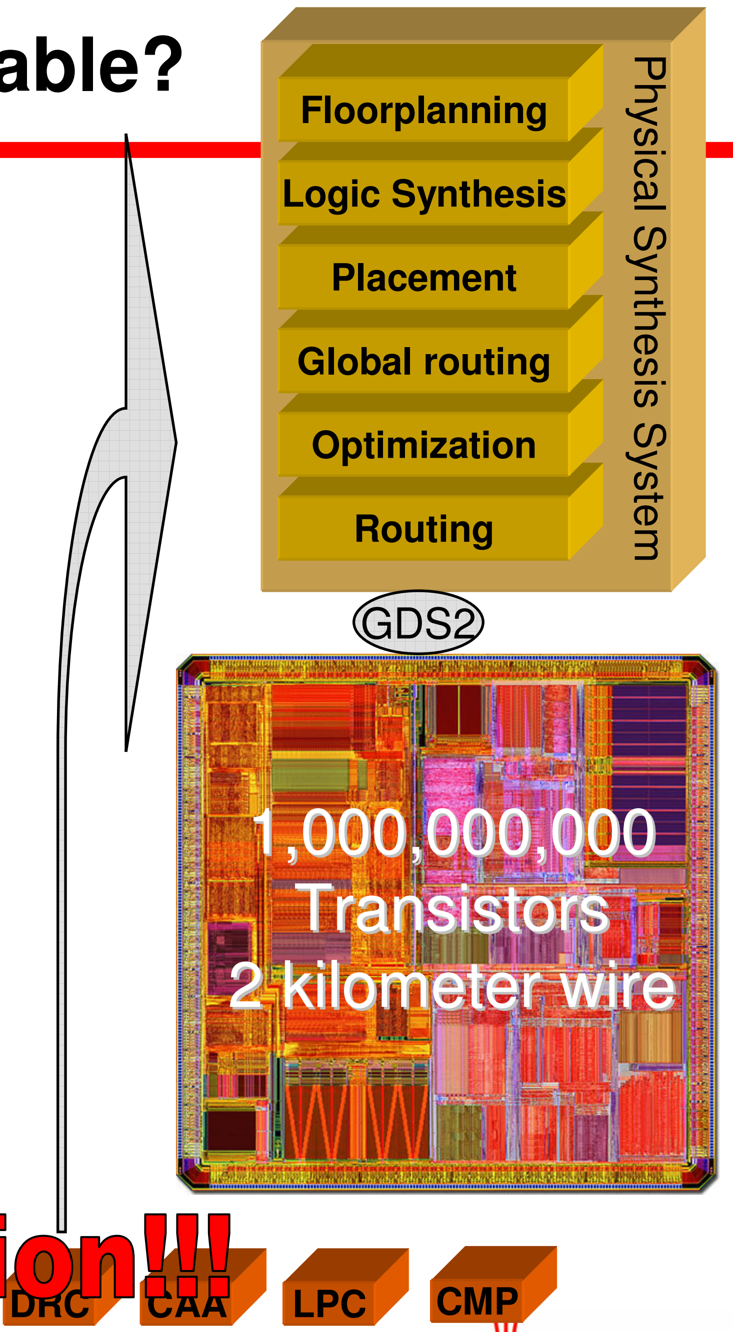
MAGMA

How much bad news is acceptable?

- … from your DFM (sign-off) analysis tools?
- < 100 violations
  - Manual fixes are feasible
- < 1000 violations
  - ECO-style fixes, rip-up and reroute
- > 10000 violations:
  - Re-run entire flow, and somehow do it better next time…

Better be 99.9999%

Correct by construction!!!



Avoidance (not patching) is key

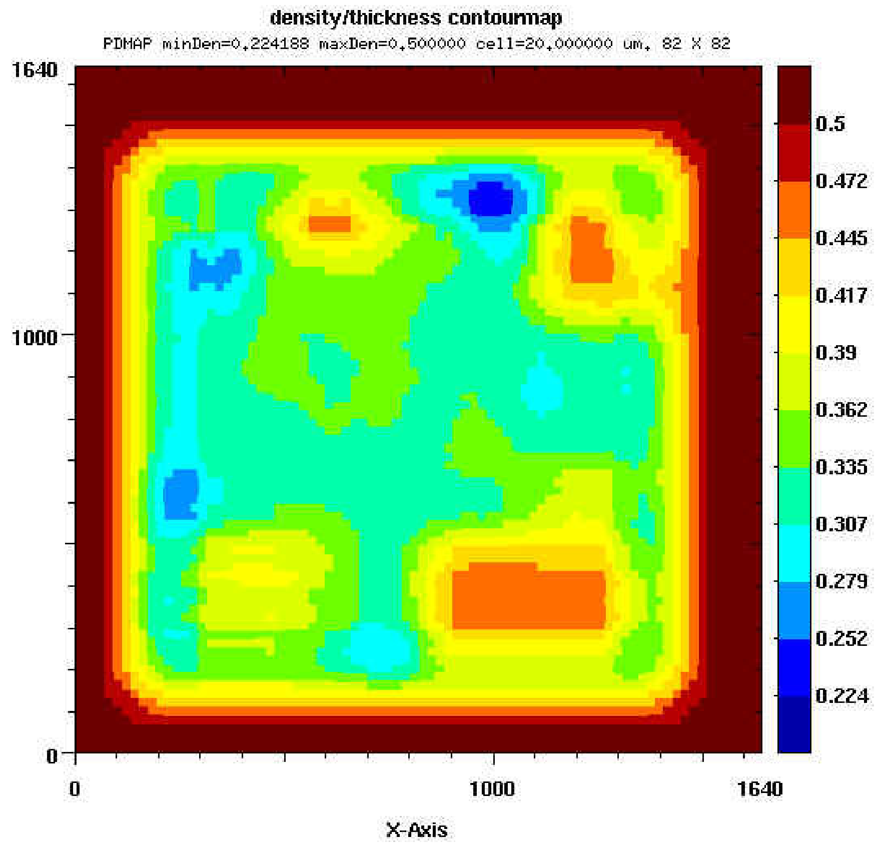
- **Sign-off abstraction levels are needed to avoid costly iteration:**
  - Logic: gates + nets
  - Physical: Standard cells + wires
  - GDS2: transistor pattern + wire pattern

- **Cells = place to hide mask and DFM issues**
  - Must remain rock-solid building blocks
  - This model was driver for Moore's law!

- **Wires = Interconnect mask**
  - Use grid abstraction for wire that ensure 99.9999%

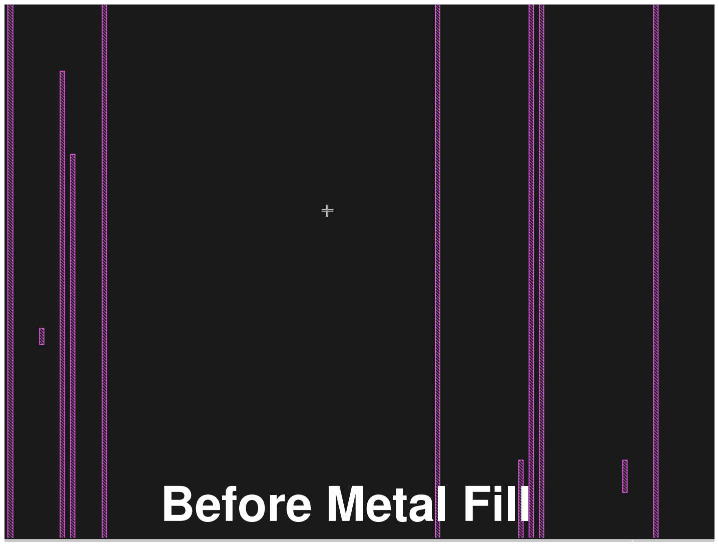
Problem 1: CMP variation (systematic yield loss)

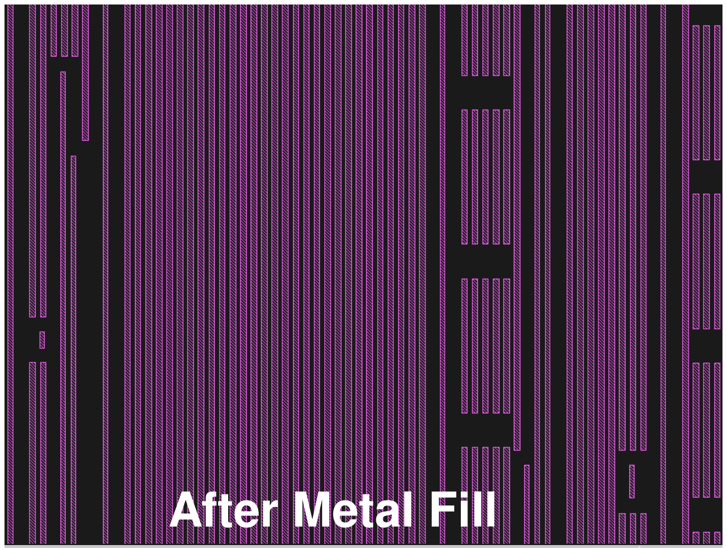
- **Problem:**
  - layout pattern density influences metal thickness, resulting in systematic wire resistance variations or failure.
- **Desired:**
  - keep (local) mask density variation within limits
  - Lower density = lower resistance
- **Simulator:**
  - CMP Thickness simulation tool



CMP variation: Plans of attack

- **1) Metal fill, encoded in layout “density rules”**
  - Reduce effect on delay/timing by ‘double spacing’ wires

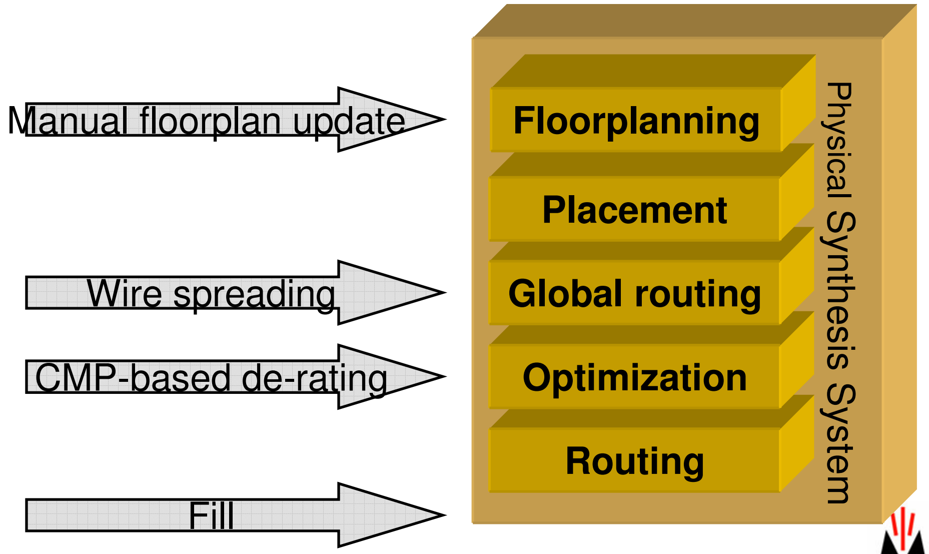




- **2) Manual floorplan updates**
- **3) Use fill-friendly power supply mesh pattern**
- **4) Force global router to spread wires even more**
  - This will start to cost wire length and contacts
- **5) Back-annotate density effect on delay into timer**
  - This could reduce pessimism in the delay calculator.

CMP variation synthesis: Score chart of methods

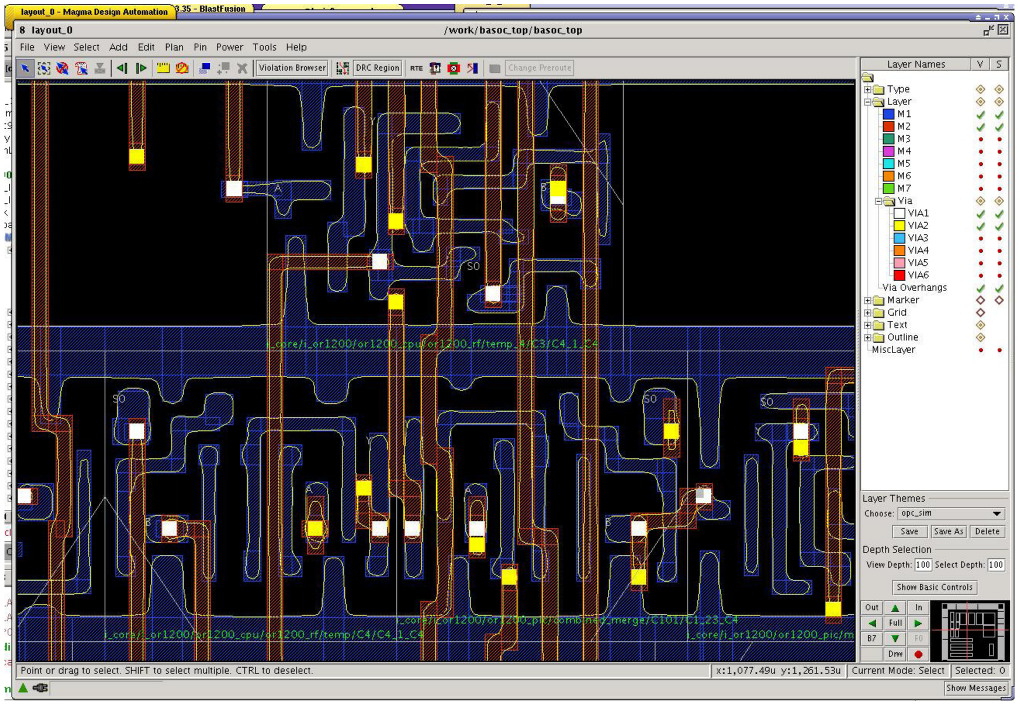
	How	The good	The bad
Fill insertion	**Pre/post GDS2 step**	**Effective**	**Adds wire cap.** **Huge files**
Global router wire spreading	**Costing global router**	**Helps other objectives**	**Not effective,** **Longer wires**
De-rate timer by CMP wire thickness.	**Optimization based on congestion data**	**Reduces pessimism, so smaller gates**	**Not effective** **Needs reliable process data.**

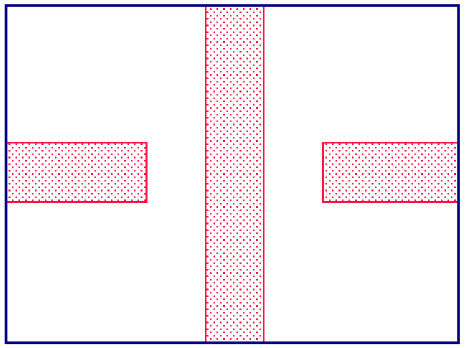


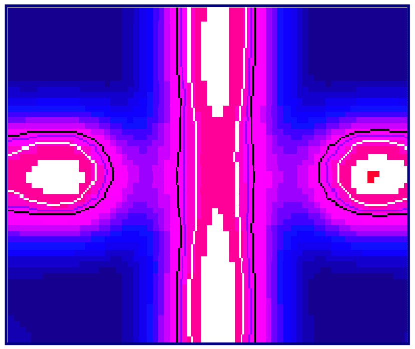
MAGMA

Problem2: Wire Layout printability (systematic)

- **The problem:**
  - What you see != what you get
  - Failures, resistance variations.
  - Depends on ***pattern*** in the local neighborhood
- **Goal:**
  - Achieve printability without impairing density and routability too much.
- **Simulator/sign off tools:**
  - Pattern matching DRC
  - LPC simulator (SLOW!)



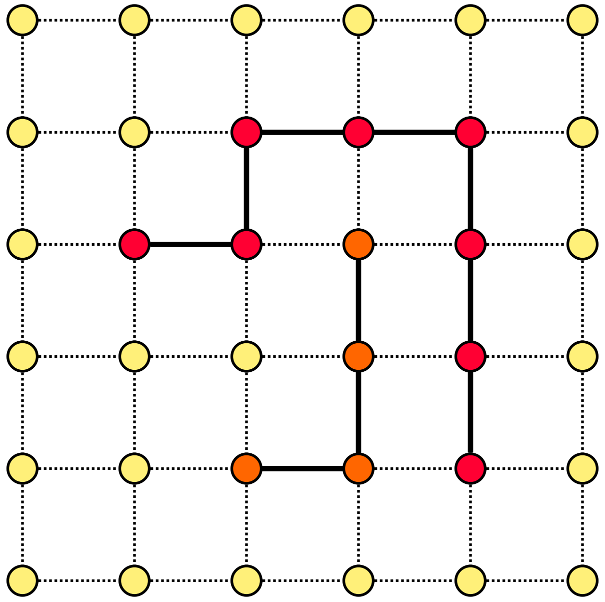


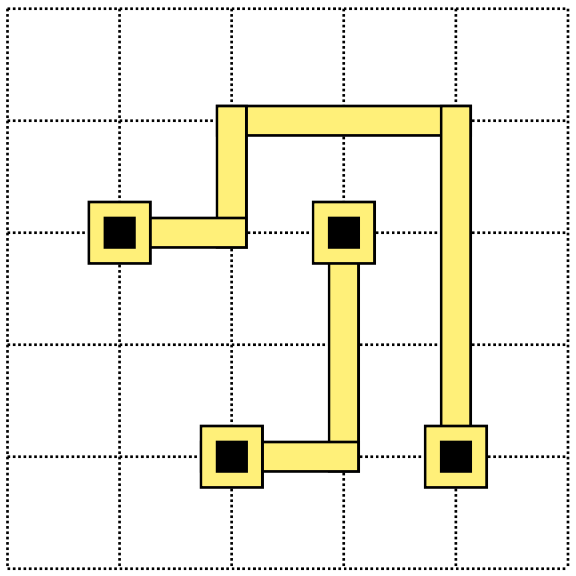


Wire layout printability: router foundation

- **Restrict the type of wire patterns that routers generate.**
  - Encode DRC rules using a regular 'grid graph'

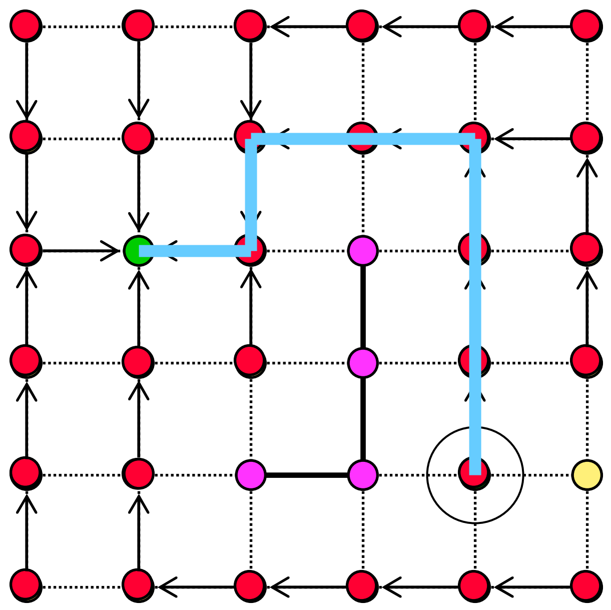
Smaller solution space!





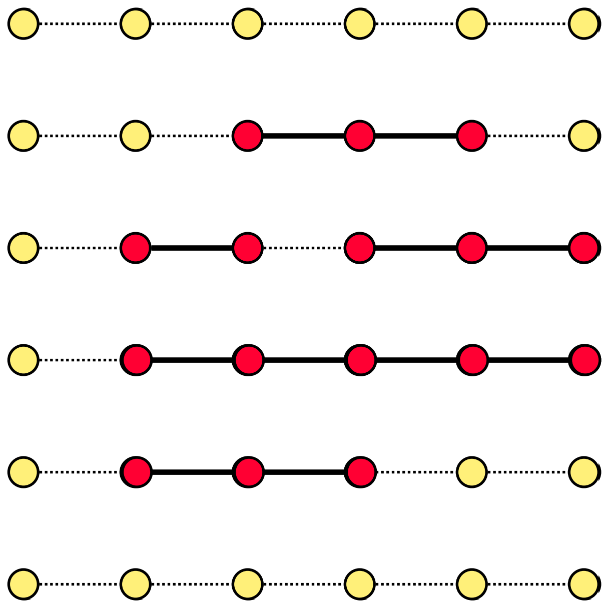
- **How to get printability rules in there?**
  - Local modifications of the grid graph can encode certain design rules, off grid elements

Workhorse: Dijkstra’s algorithm on a grid graph



- **The good: Guarantees to find shortest path, if it exists.**
- **The bad: Sequential: no guarantees for multiple nets**
- **Killer feature: Changing the edge weight can encode preferences.**

Wire Layout printability: plan of attack

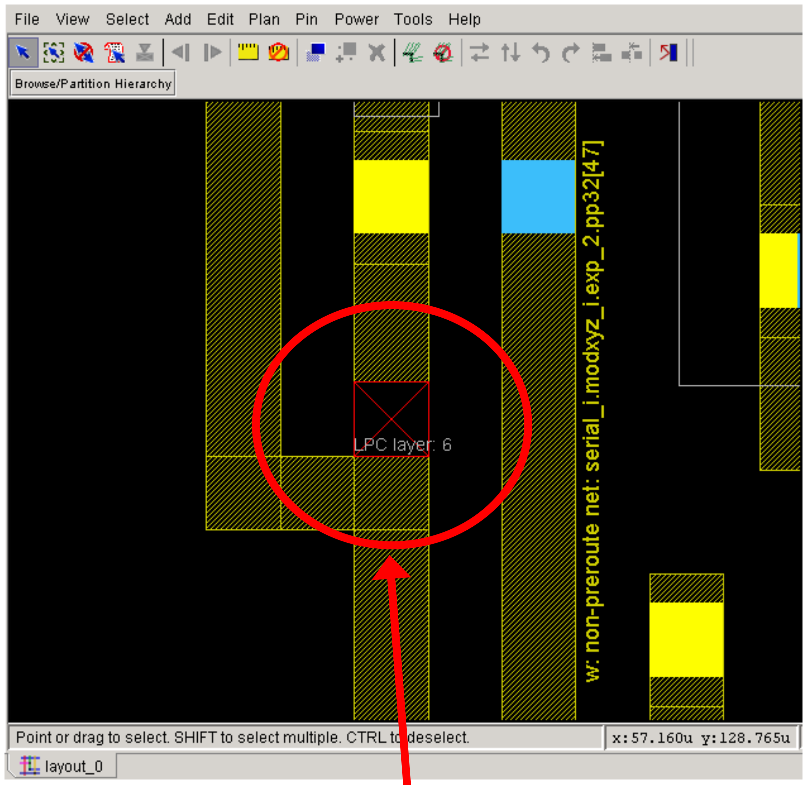


- **1: Pick proper grid spacing**
  - This is the starting point
- **2: Pre-condition grid landscape**
  - Block likely hot spots
  - Remove non-preferred direction edges
    - Creates lots of extra contacts
- **3: Cost the edges**
  - Make edges around likely hot-spots expensive
- **4: Patterns during path search**
  - Disabling stacked via avoids island problem
  - End-of-line rule compliance
- **5: Post-process routing patterns**
  - Rip-up-and-reroute, small modifications

MAGMA

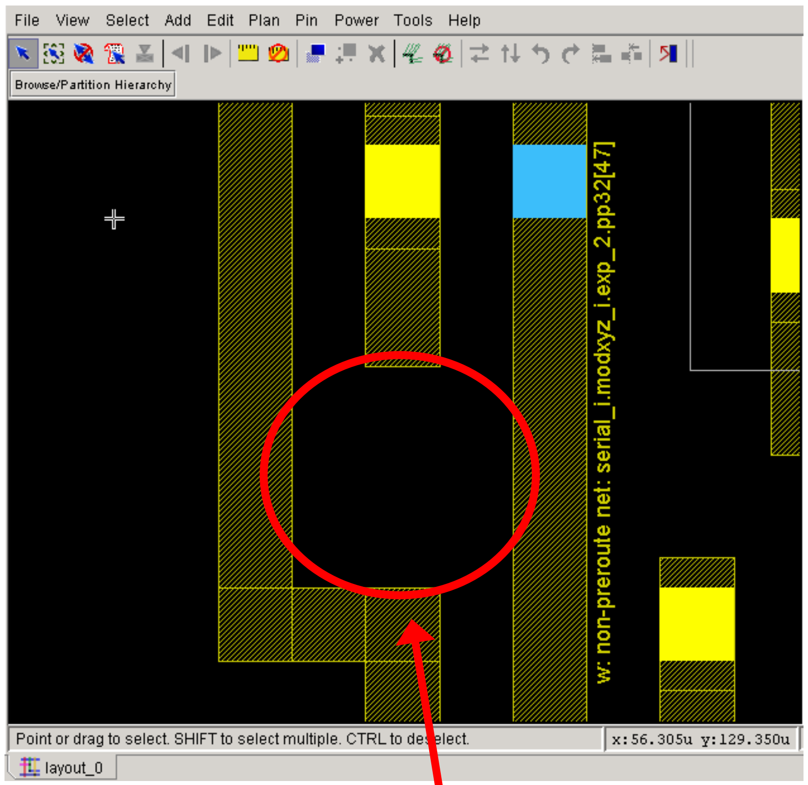
Litho Hot Spot Optimizations

- **Fix method: LPC hot spot resolved by re-routing**


File View Select Add Edit Plan Pin Power Tools Help
Browse/Partition Hierarchy
w: non-preroute net: serial_i.modxyz_i.exp_2.pp32[47]
LPC layer: 6
layout_0

Point or drag to select. SHIFT to select multiple. CTRL to deselect.

x:57.160u y:128.765u


File View Select Add Edit Plan Pin Power Tools Help
Browse/Partition Hierarchy
w: non-preroute net: serial_i.modxyz_i.exp_2.pp32[47]
layout_0

Point or drag to select. SHIFT to select multiple. CTRL to deselect.

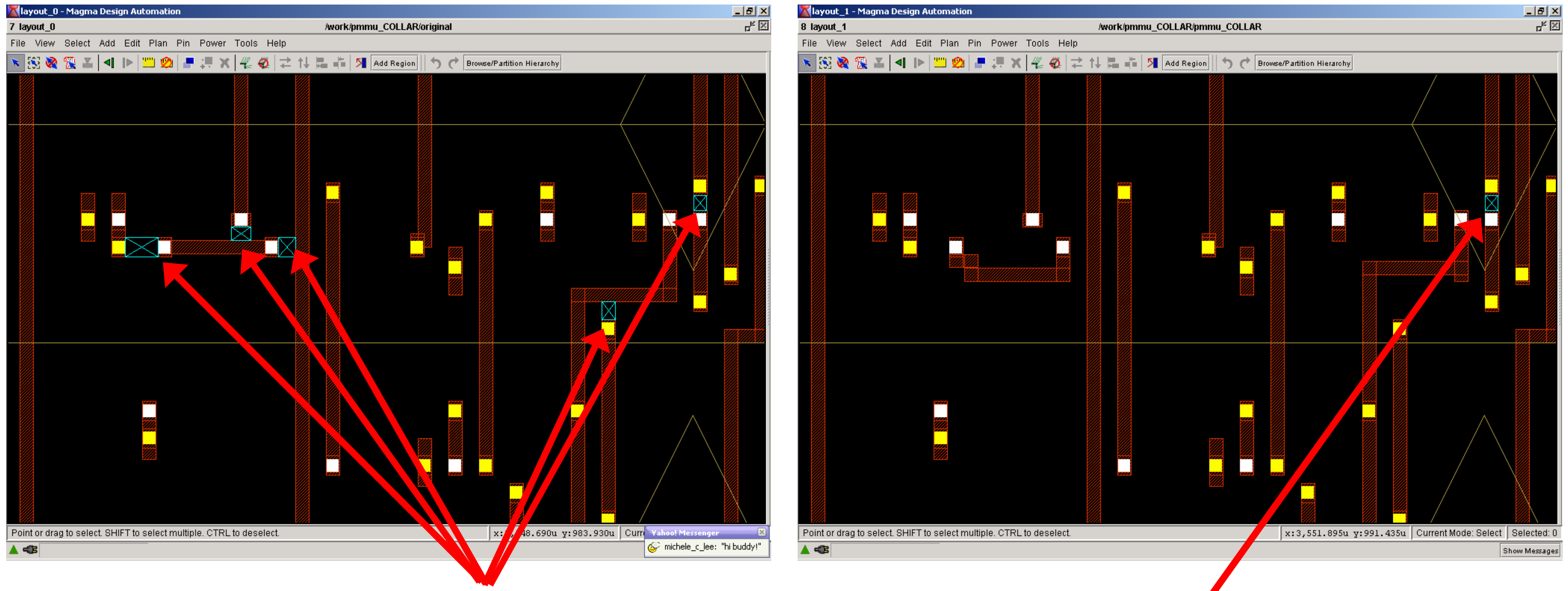
x:56.305u y:129.350u

LPC hot spot

Hot spot removed

LPC-based post processing

- Litho hot-spots detected using LPC
- Area blocked, and locally re-routed


layout_0 - Magma Design Automation
7 layout_0
/work/pmmu_COLLAR/original
File View Select Add Edit Plan Pin Power Tools Help
Add Region
Browse/Partition Hierarchy
Yahoo! Messenger
layout_1 - Magma Design Automation
8 layout_1
/work/pmmu_COLLAR/pmmu_COLLAR
File View Select Add Edit Plan Pin Power Tools Help
Add Region
Browse/Partition Hierarchy
Current Mode: Select
Selected: 0
Show Messages

Point or drag to select. SHIFT to select multiple. CTRL to deselect.

x: 48.690u y:983.930u

michele_c_lee: "hi buddy!"

Point or drag to select. SHIFT to select multiple. CTRL to deselect.

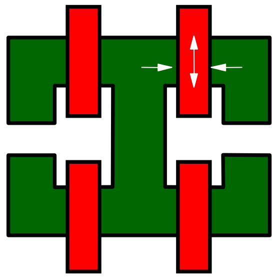
x:3,551.895u y:991.435u

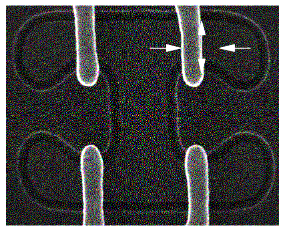
Five hot spots **4 fixed, 1 left**

Method is slow and has Limited strength

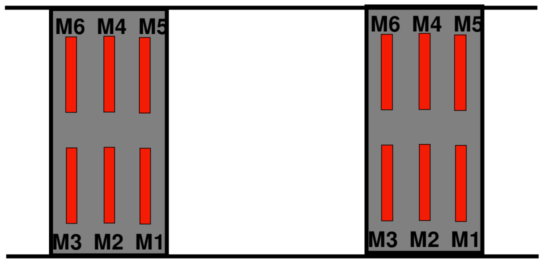
Printability of standard cells

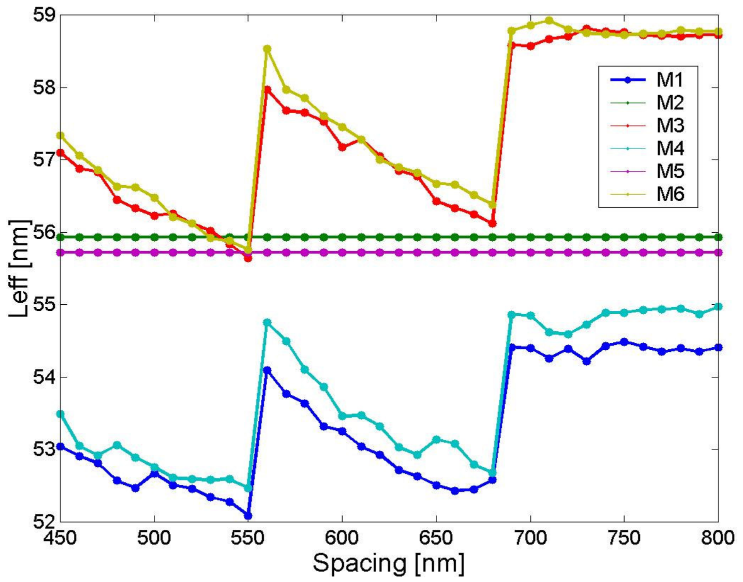
- **Cells can be designed manually, and checked extensively using LPC tools.**
  - Cell in must be printable!
  - Must be rock-solid building block





- **But Gate Length Depends on Cell neighbors**





- **Living with it: cell de-rating**
  - 5-10% impact on timing
  - 10-15% impact on leakage power

- **Avoiding is much better:**
  - Avoid by design **patterns** on standard cells
  - Certain detailed placement composability rules (conditional cell padding)

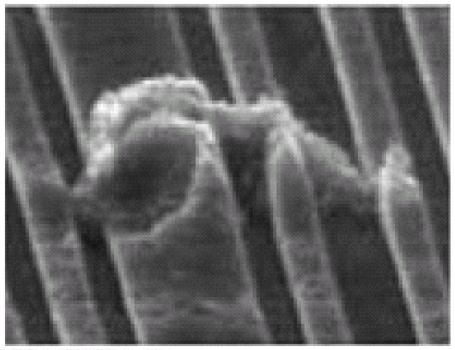
Layout synthesis for Printability: score card

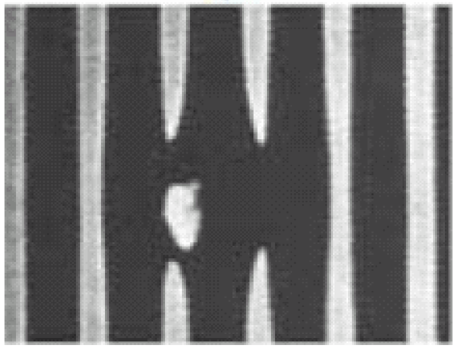
		The good	The bad
Grid pre-contioning	Hard Avoidance	Effective	Lower routability
No non-preferred	Hard Avoidance	Very effective fast	Routability, extra vias
Grid costing	Soft avoidance	Multiple objectives	Not very effective
Search patterns	Conditional avoidance	Effective	Slows down router
Post processing	Correction/ Patch-up		Slow, no guarantee
Standard cell design	Avoidance	Keeps abstraction	Pessimism, manual

MAGMA

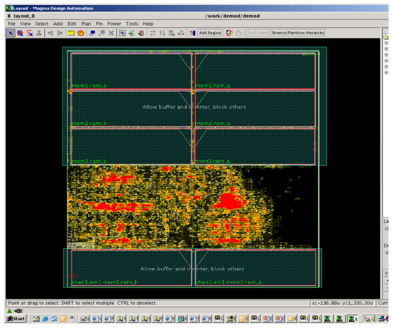
Problem 3: Random layout yield loss

- **The problem is simple:**
  - Random particles cause opens and shorts, resulting in yield loss
- **Analysis tools:**
  - Predict (relative) yield using CAA that measures 'critical area'
- **Synthesis mantra:**
  - Its OK to slip a few of these!
- **Issues:**
  - Trade-off between various factors nebulous

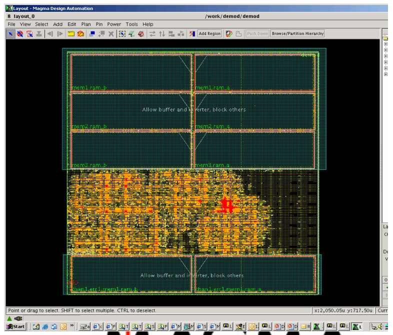




CAA before

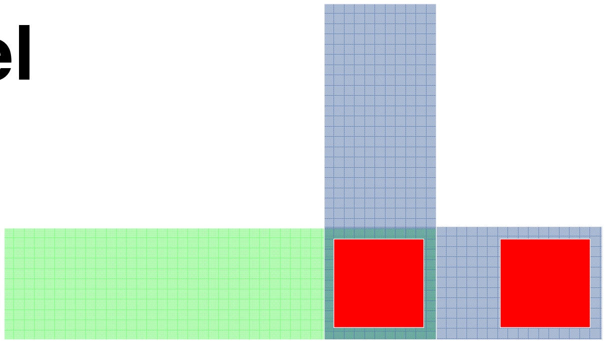


CAA after



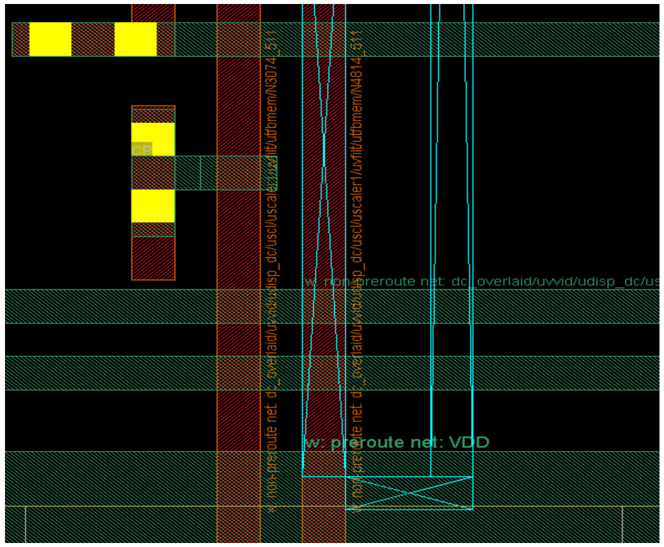
Combating random yield loss

- **Redundancy at system level**
- **Minimize contacts**
- **Make contacts redundant**
  - >80% without drawbacks
  - Most standard cell libraries are NOT redundant-via capable!!
- **Spread wires**
  - Global routing
  - During Detailed routing
  - Using postprocessor
- **Widen wires**
  - Using postprocessor

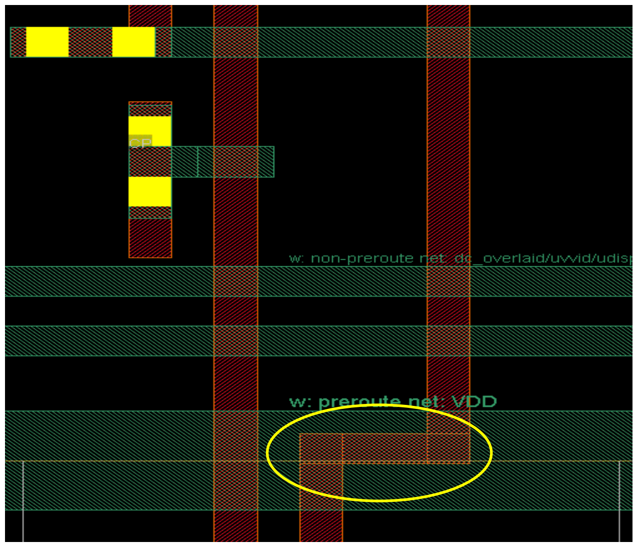


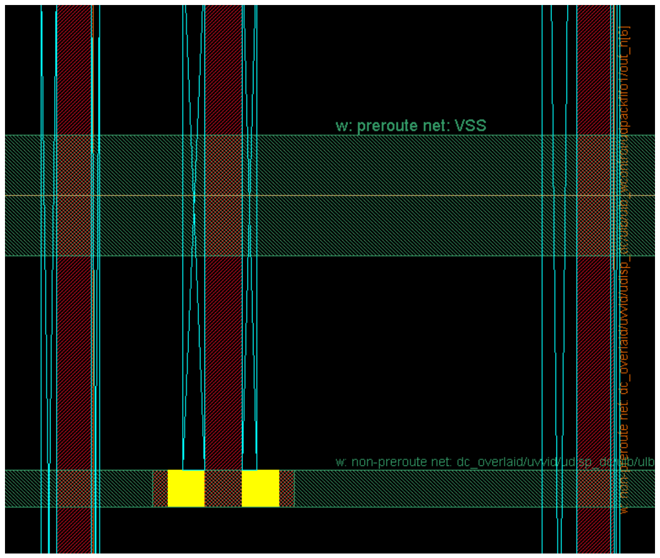
Assumes small defects dominate

Wire Spreading/Widening postprocessing

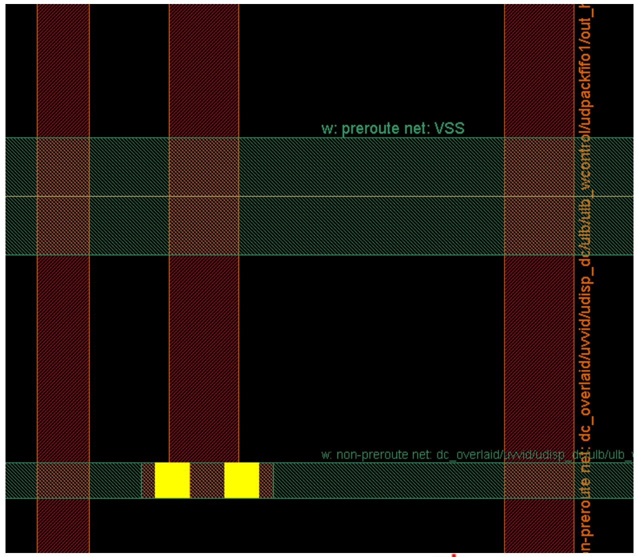


Wire Spreading





Wire Widening


w: preroute net: VSS
w: non-preroute net: dc_overlaid/uvvid/udisp_dc/ulb/ulb
n-preroute net: dc_overlaid/uvvid/udisp_dc/ulb/ulb_wcontrol/udpackfifo1/out_

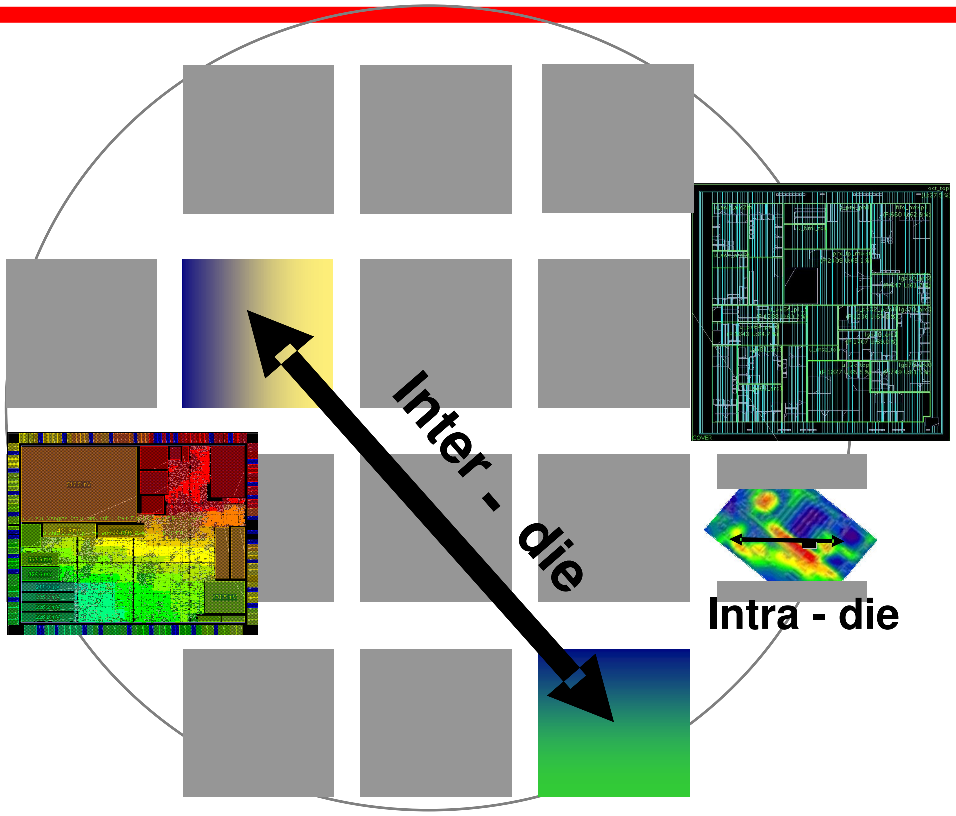
MAGMA

Avoiding Random yield loss: scorecard

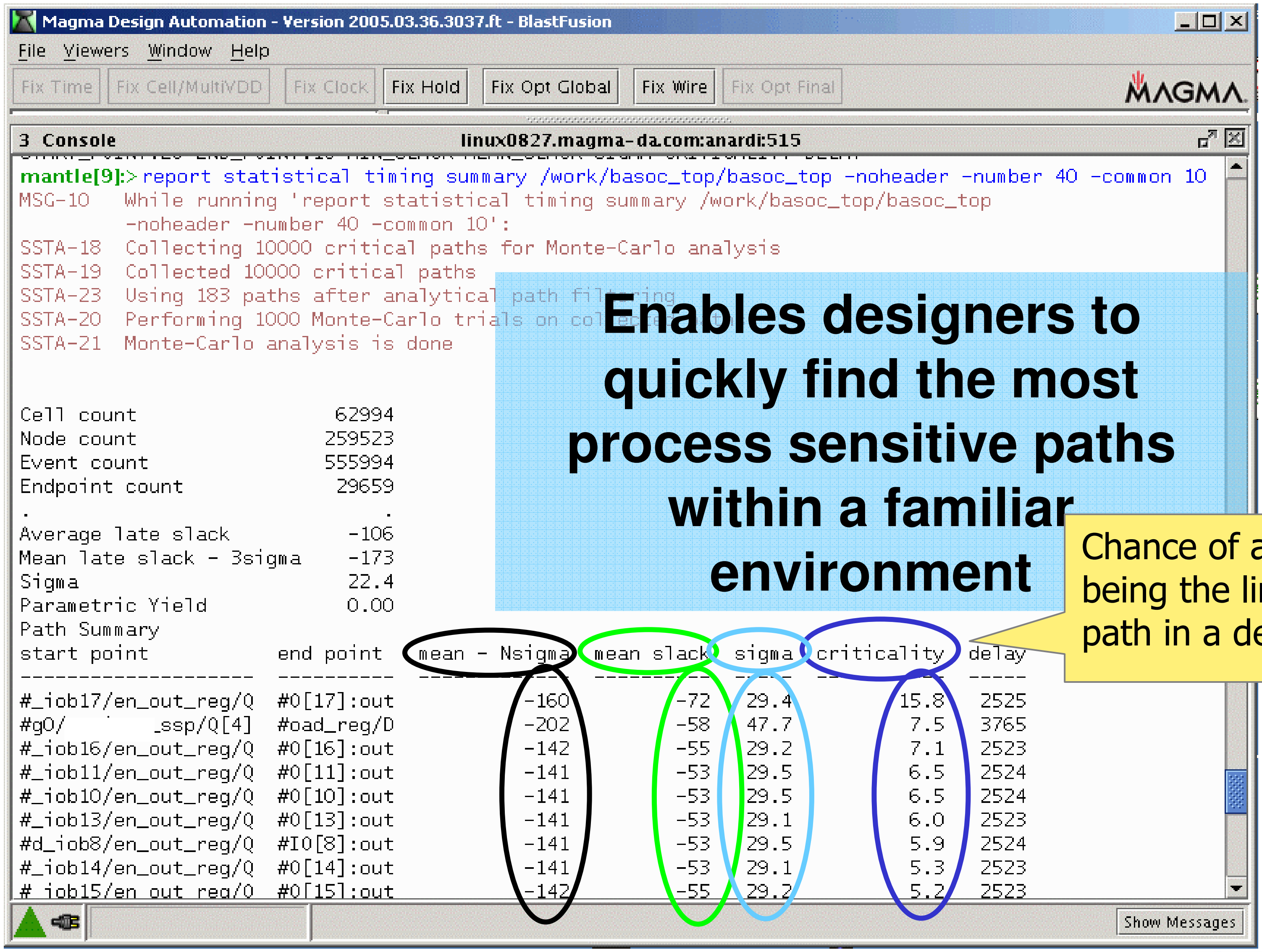
	Method	The good	The bad
Via reduction	Router costing	OK	Implies less spreading
Redundant vias	Post-process	Effective and Cheap	Last 10% are hard
Spreading	Global router costing		Longer wires, more vias
Spreading	Detailed router cost		Routability
Spreading	Post processing	Cheap	Longer wires
Widening	Post processing	Cheap	Forbidden pitch

Other yield issues: parameter variation

- **Predictable:**
  - Temperature,
  - Voltage drop
- **Random:**
  - OCV
- **Attack plan:**
  - Reduce pessimism by back-annotating
  - Fast Multi-corner-multi-mode optimization
  - SSTA



Quartz SSTA Analysis report



```
mantle[9]:> report statistical timing summary /work/basoc_top/basoc_top -noheader -number 40 -common 10
MSG-10   While running 'report statistical timing summary /work/basoc_top/basoc_top
         -noheader -number 40 -common 10':
SSTA-18  Collecting 10000 critical paths for Monte-Carlo analysis
SSTA-19  Collected 10000 critical paths
SSTA-23  Using 183 paths after analytical path filtering
SSTA-20  Performing 1000 Monte-Carlo trials on collected paths
SSTA-21  Monte-Carlo analysis is done


Cell count                        62994
Node count                       259523
Event count                      555994
Endpoint count                    29659
.                                     .
Average late slack                 -106
Mean late slack - 3sigma           -173
Sigma                              22.4
Parametric Yield                   0.00
Path Summary
start point             end point    mean - Nsigma  mean slack  sigma  criticality  delay
---------------------   ----------   -------------  ----------  -----  -----------  -----
#_iob17/en_out_reg/Q    #0[17]:out            -160         -72   29.4         15.8   2525
#g0/          _ssp/Q[4] #oad_reg/D            -202         -58   47.7          7.5   3765
#_iob16/en_out_reg/Q    #0[16]:out            -142         -55   29.2          7.1   2523
#_iob11/en_out_reg/Q    #0[11]:out            -141         -53   29.5          6.5   2524
#_iob10/en_out_reg/Q    #0[10]:out            -141         -53   29.5          6.5   2524
#_iob13/en_out_reg/Q    #0[13]:out            -141         -53   29.1          6.0   2523
#d_iob8/en_out_reg/Q    #I0[8]:out            -141         -53   29.5          5.9   2524
#_iob14/en_out_reg/Q    #0[14]:out            -141         -53   29.1          5.3   2523
#_iob15/en_out_reg/Q    #0[15]:out            -142         -55   29.2          5.2   2523
```

Enables designers to quickly find the most process sensitive paths within a familiar environment

Chance of a path being the limiting path in a design.

Summary

- **DFM&Y is a design problem!**
  - No single point solution, but a combination of methods
  - Delicate trade-off throughout flow:
    - Between avoidance and fix steps, and between other objectives
- **Keep GDS2 sign-off levels alive in 45nm!**
  - Cannot afford loops that involve slow analysis
  - Standard cells must remain rock-solid building blocks
  - Wires layout patterns must be restricted
- **0-order DFM wisdom for synthesis tools:**
  - Bigger cells = better
  - Lower density = better
  - Keep it regular and uniform

DFM without analysis!

- **Must be 99.99% correct-by-construction**
  - … because iterative fixing is insecure and slow