Going with the Flow:

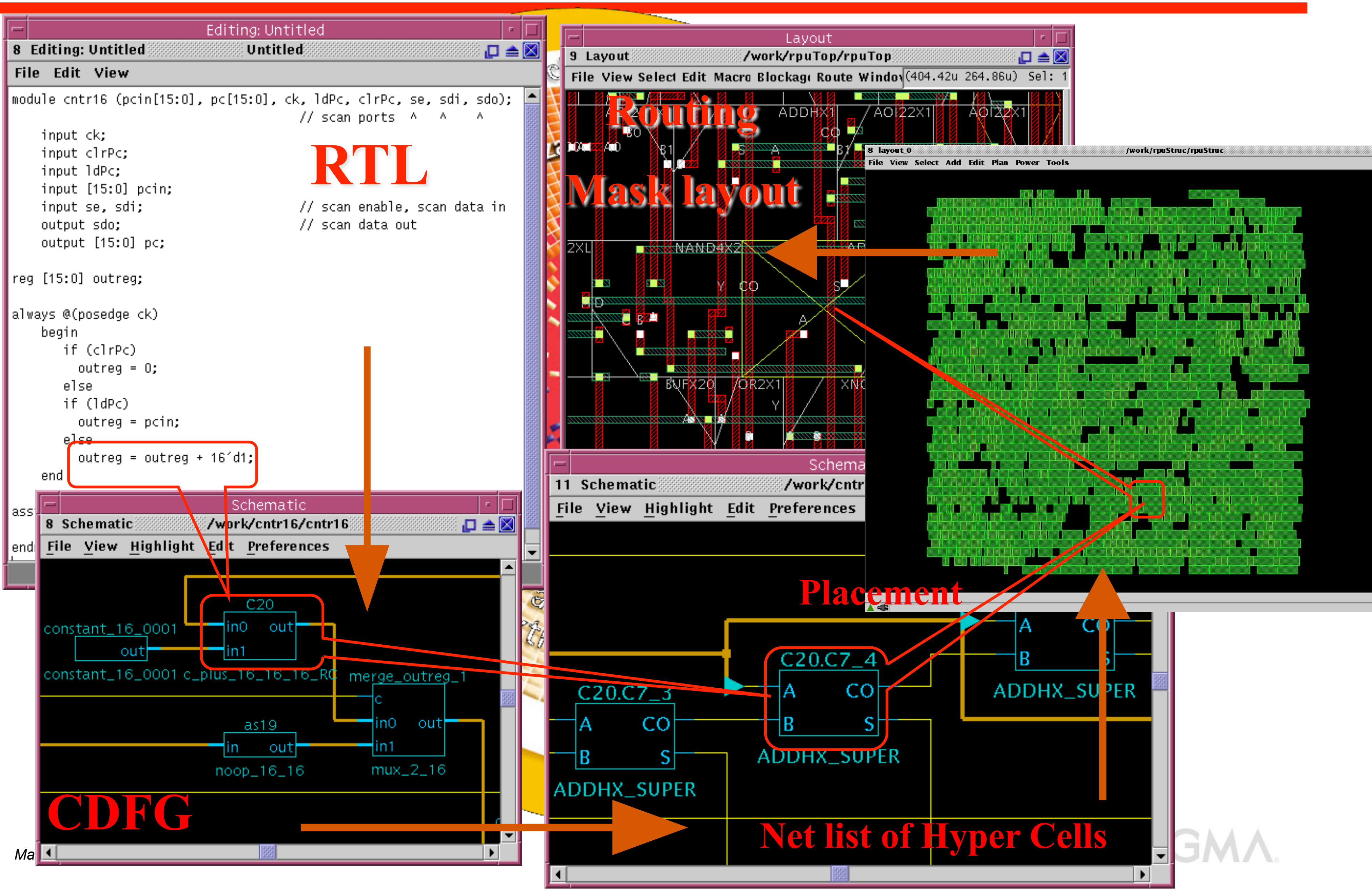
Bridging the Gap between Theory and Practice in Physical Design

Patrick Groeneveld,
Chief Technologist, Magma Design Automation
ISPD 2010 San Francisco

Overview: Physical Design Flows

- **The Nature of the PD problem**
  - Objectives
  - Algorithms
- **How to build a flow**
  - ABC
- **Tough problems:**
  - Crosstalk-induced delay
- **Proof of efficacy**

Physical Design Flow: from logic to physical

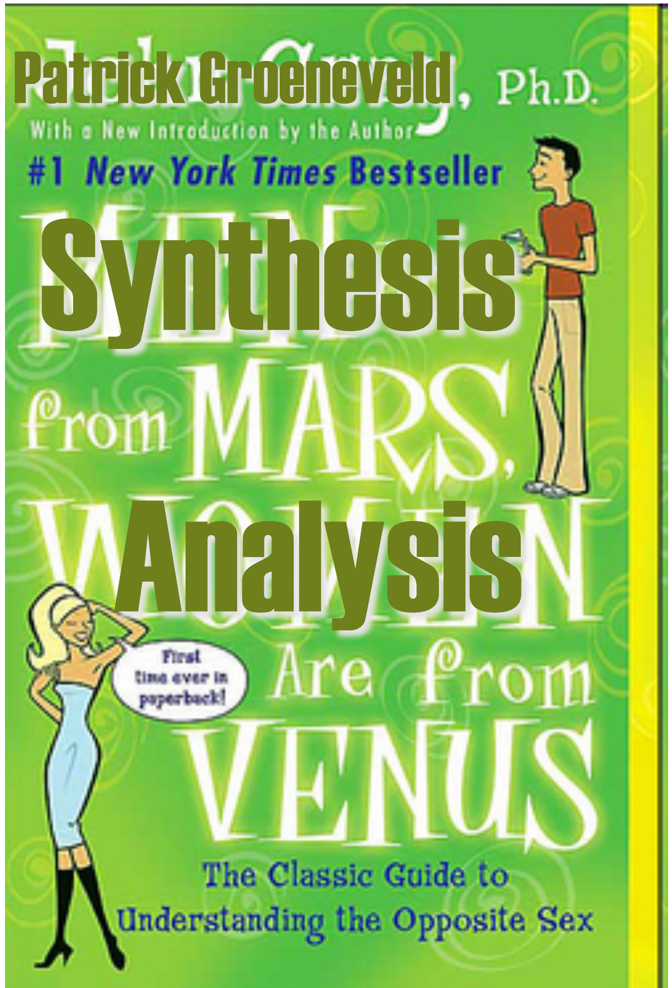


Synthesis is from Mars, Analysis is from Venus

- Sign-off tools:
- Verification, Extraction, STA, spice, DRC, LVS

- Highly accurate
- Big and slow
- Parallelizable

- Is the ‘whiner’



- Implementation tools:
- RTL synthesis, Placement, Routing, Optimization, Humans

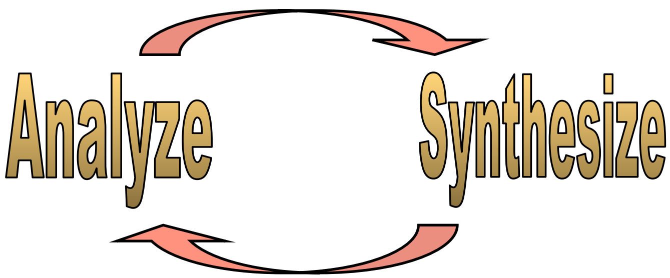
- Poor accuracy
- Lean, mean
- Tough

- Is the ‘hacker’

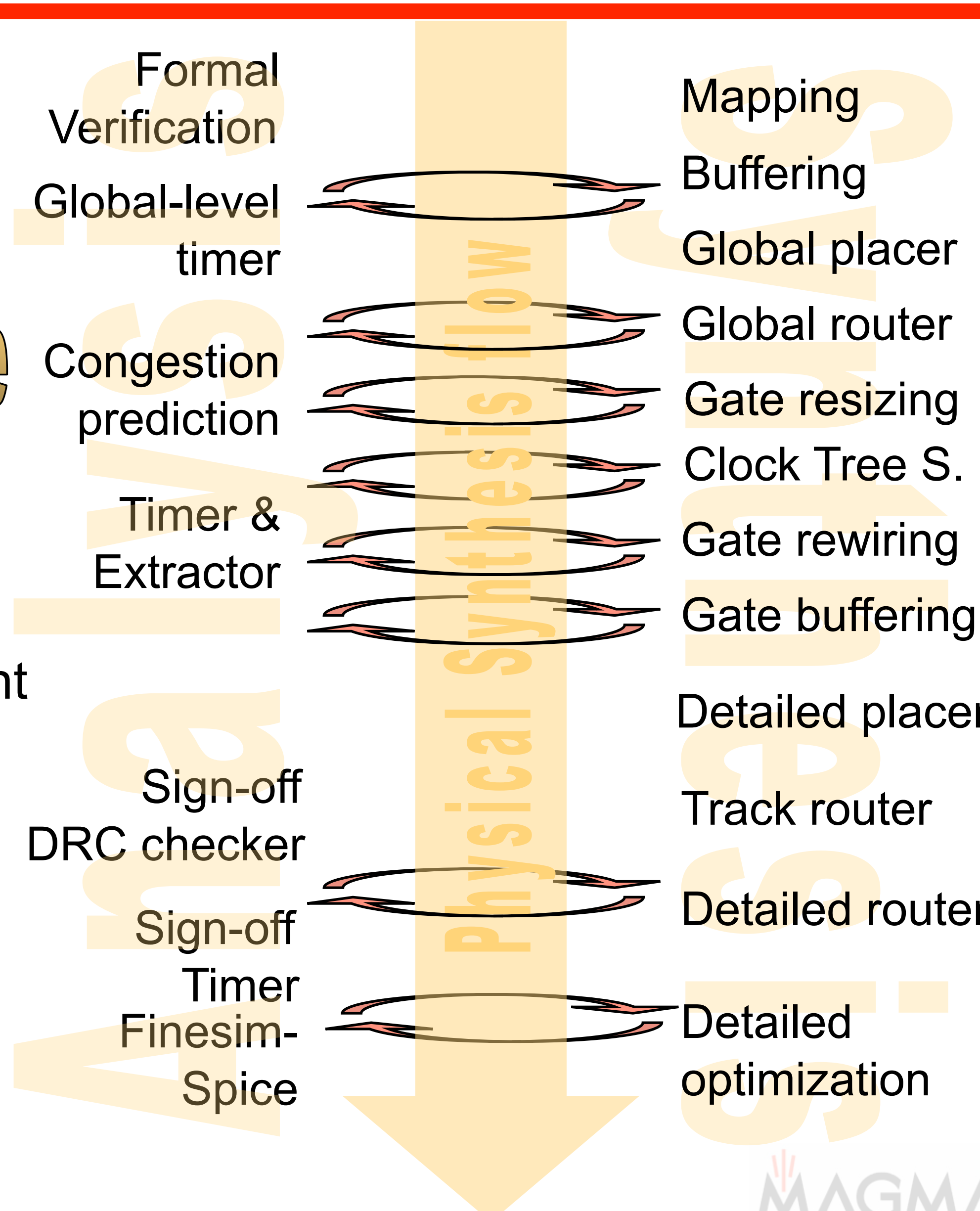
Need to make this work

Making this work in a Physical Synthesis Flow

Iterate:



- **Avoid loops:**
  - Gradual, Stepwise refinement
  - ABC flows
- **Speed up loops:**
  - Reducing analysis accuracy
  - Tricks: incremental analysis
  - Running tasks in parallel
  - Tight tool integration



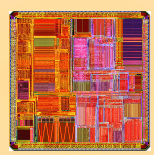
Physical Design:
Trade-offs between conflicting objectives

Speed

Run time

Human design effort

Leakage power



Area

Dynamic power

Routability

Yield, robustness

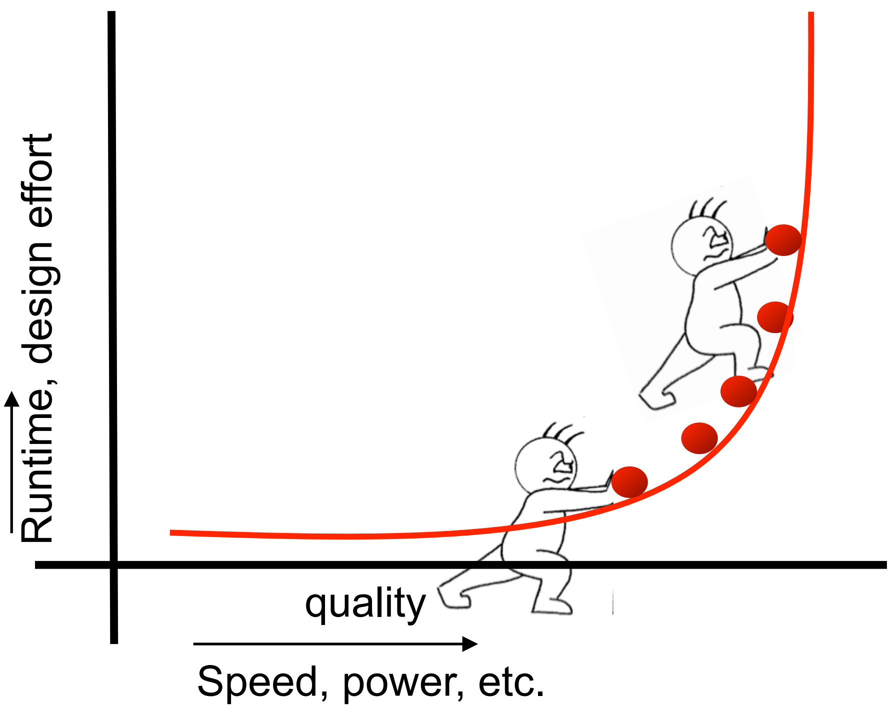
Corners

Manufacturability

The nature of the Physical Design 'beast'

Pushing *all* objectives simultaneously costs:

- **Human design effort,**
- **Run time**



MAGMA

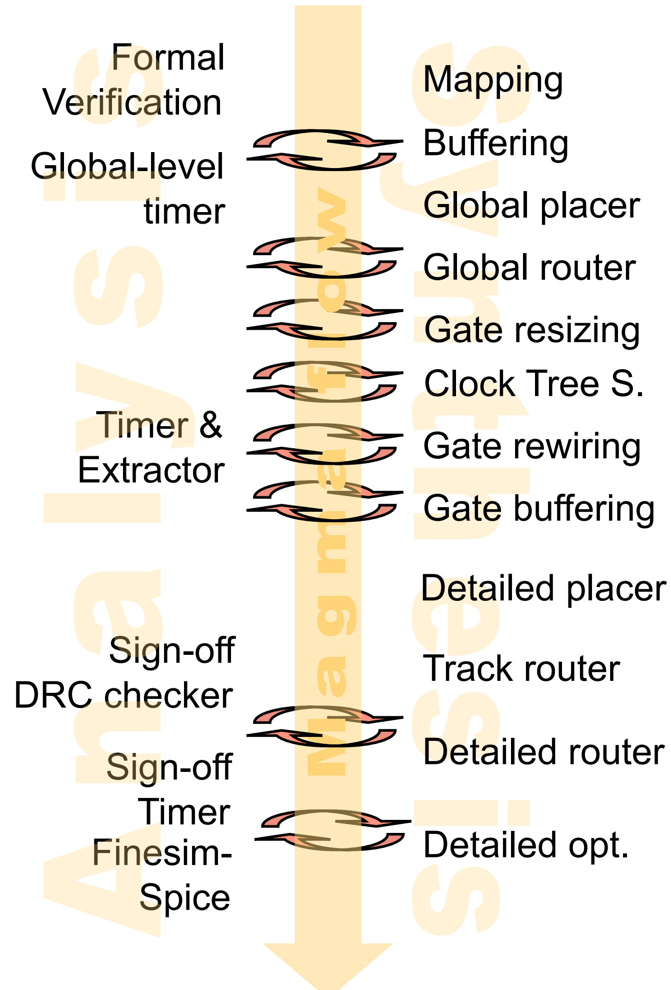
Building a Physical Design Flow

Observation 1:
Need gradual refinement flow using many algorithms

Observation 2:
Synthesis algorithms need highly simplified models of reality

Observation 3:
Synthesis algorithms cannot deliver good multi-objective trade-offs

Observation 4:
Optimizing a single objective often makes other objectives worse.



Optimal is not Optimal!

The ABC of a Physical Design Flow

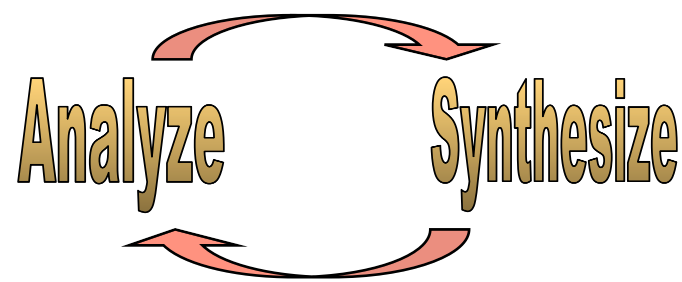
A: Avoid
Use pessimism to make problem unlikely, 'Correct by Construction'

More avoidance = worse results...

B: Build
Synthesize using an algorithm

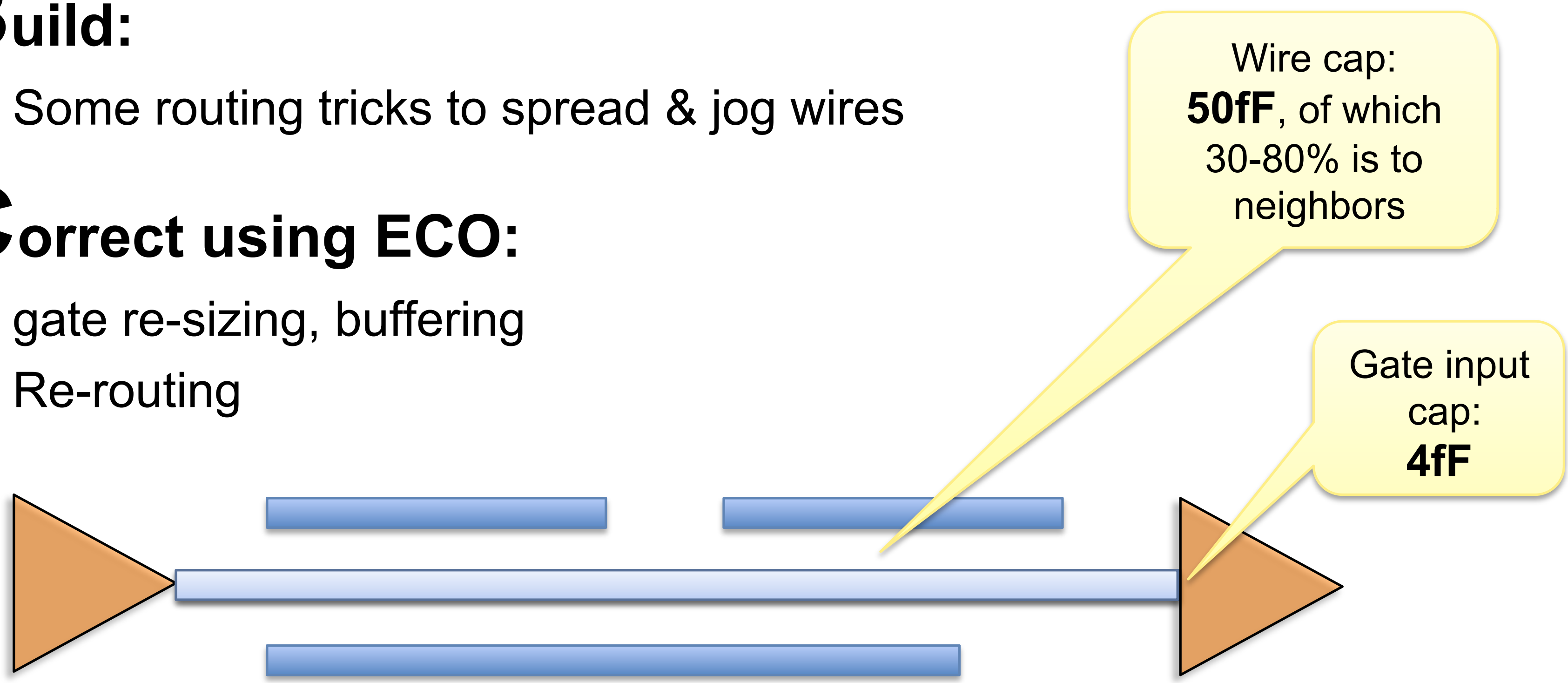
Synthesis is from Mars...

C: Correct
Fix each objective by incremental modifications (ECOs).



Example ABC: Combating crosstalk delay

- **Avoid: using 'pessimism':**
  - Size up all drivers: Costs cell area and power
  - Force double spacing NDR on many nets: Costs congestion = area
- **Build:**
  - Some routing tricks to spread & jog wires
- **Correct using ECO:**
  - gate re-sizing, buffering
  - Re-routing

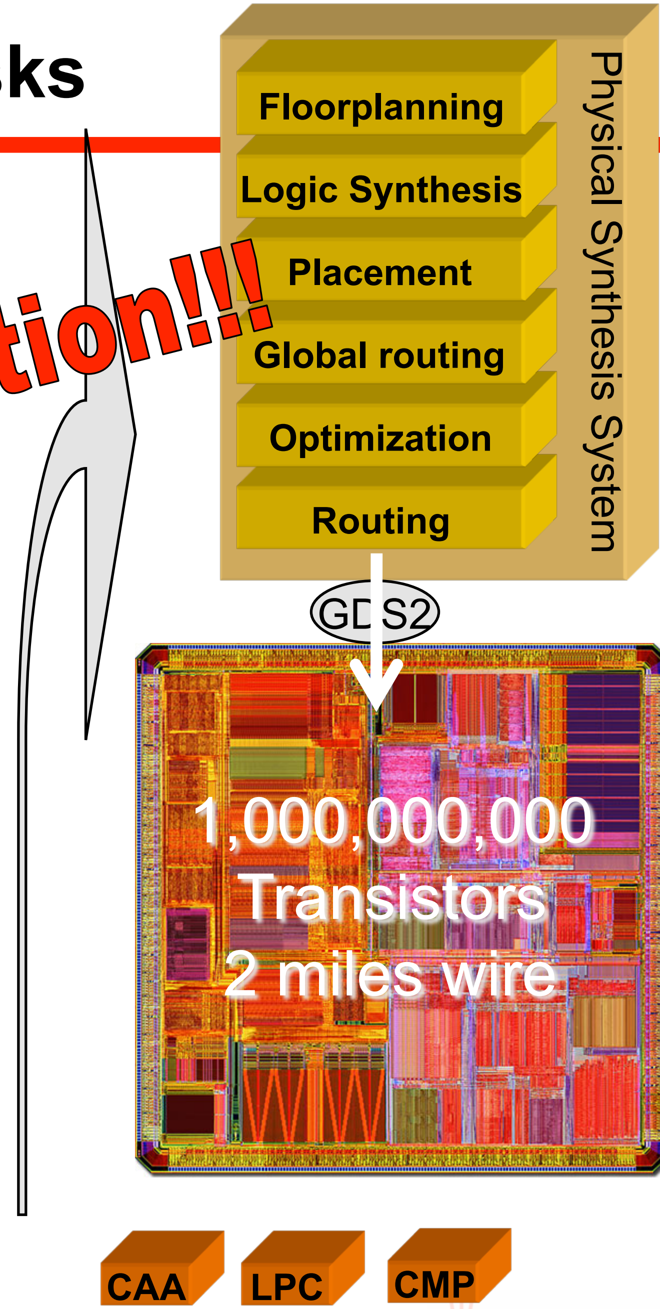


MAGMA

Avoidance vs. Correction: masks

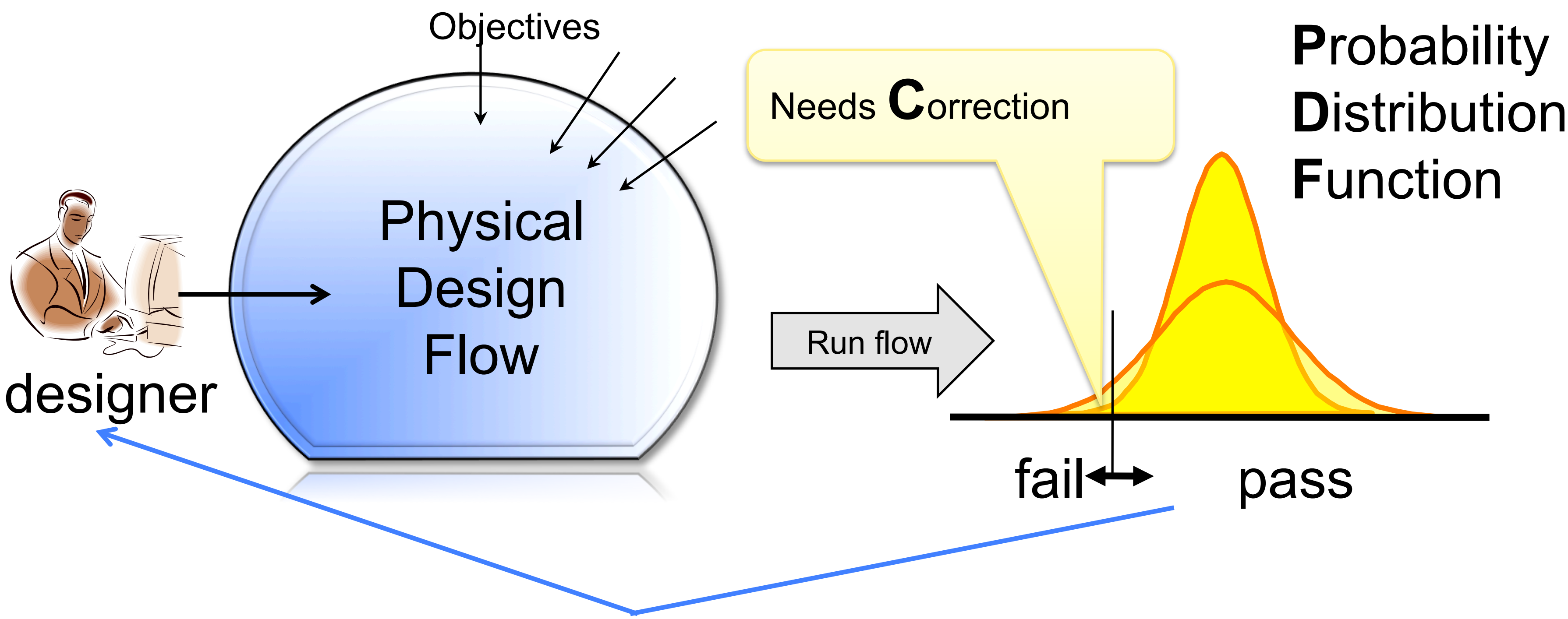
- **Avoid:**
  - DRC deck with 'hard' rules
- **Build:**
  - Dijkstra-based expansion + hacks
- **Correct:**
  - Analyze using DRC, CAA, LPC
  - Fix incrementally using R&R
- **How many failures are acceptable?**
  - < 100 violations: Manual fixes are feasible
  - 1000-10000 violations: Automatic ECO-style fixes, rip-up and reroute
  - > 10,000 violations ???????

Better be 99.999% Correct by Construction!!!



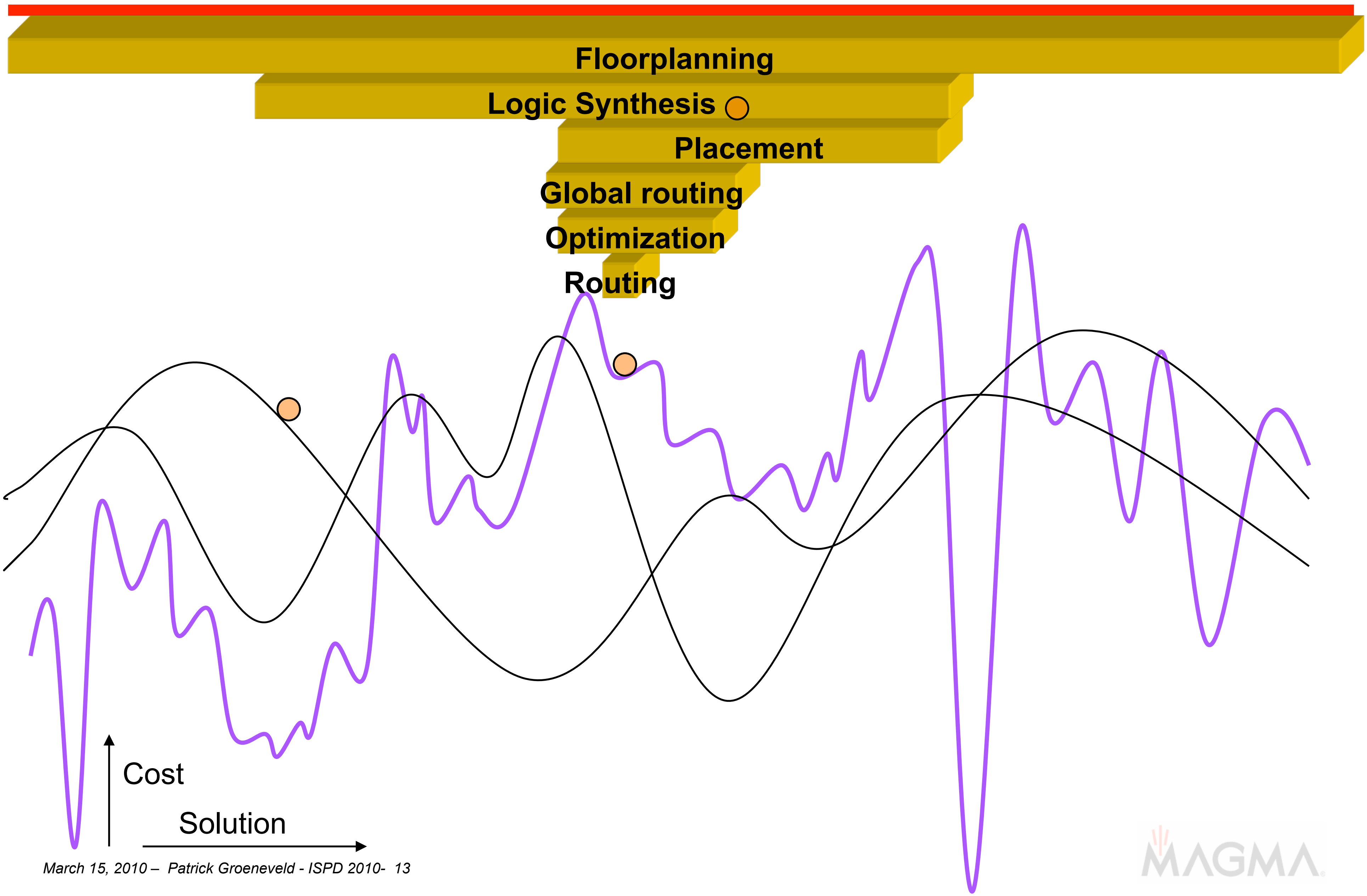
MAGMA

Controlling the amount of Correction



- **Relax the objective**
- **More Avoidance (pessimism)**
  - Which might deteriorate other objectives

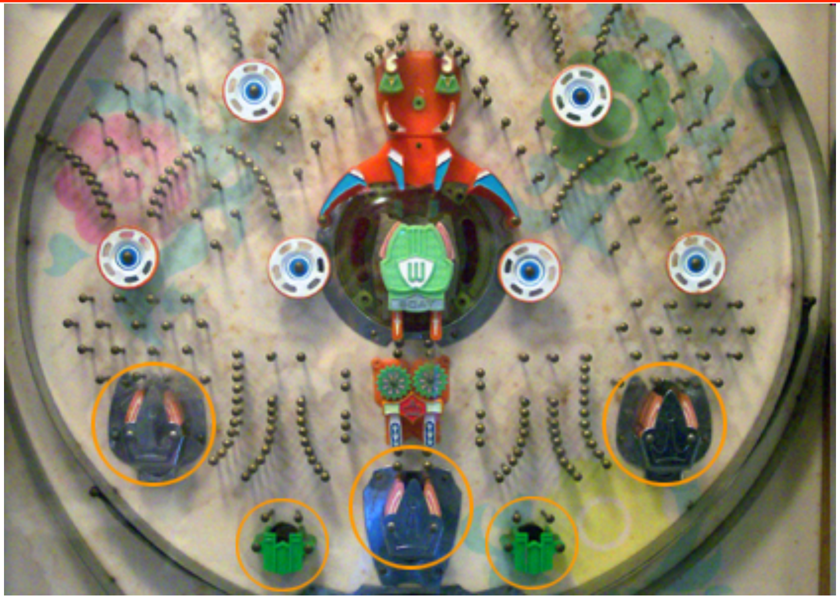
Local Optima in a Physical Design Flow



The Physical Design Flow as a Pachinko Machine

- **Run flow:**
  - End up an one of the local optima.
- **Re-run:**
  - typically get same results
    - (Multi-processing alert!!)
- **Re-run with small change**
  - Could be significant difference
- **Changes:**
  - Irrelevant order changes
  - Additional steps/algorithms
  - Changing constraints, tuning, etc.

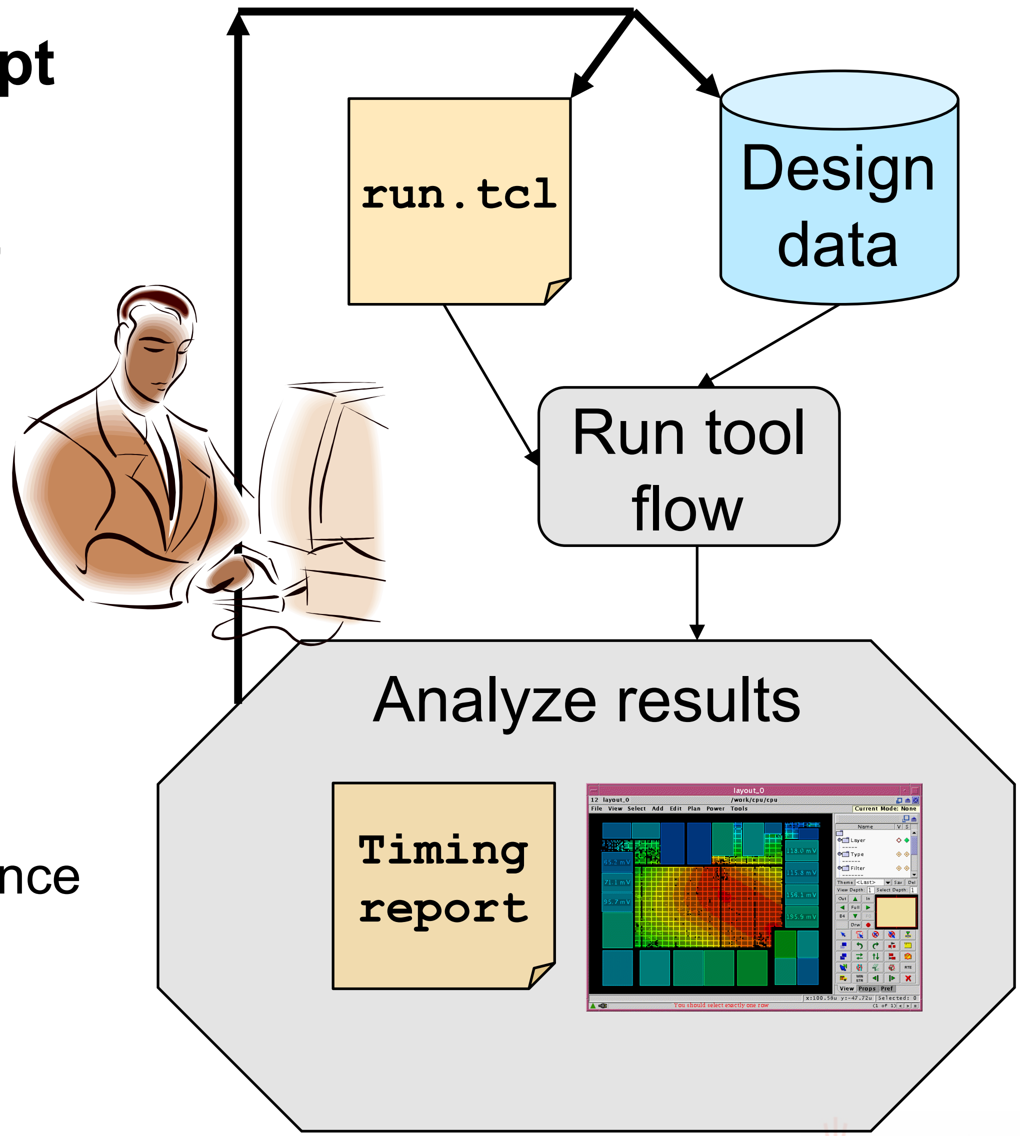
- **Good/bad results depend on:**
  - 'ease' of the design
  - Flow set-up/tuning
  - Design structure (e.g. data paths)
  - **Coincidence**



MAGMA

How to tune the flow?

- **Tuning of the TCL script**
- **First time:**
  - Poor local optimum, bugs, mistakes
- **Tune flow+data**
  - Better local optimum.
- **But:**
  - Loop is slow
  - Tool talks gibberish
  - Result depend on experience of engineer.
  - Hacks are design-specific



PD Flow tuning for best out-of-the-box results

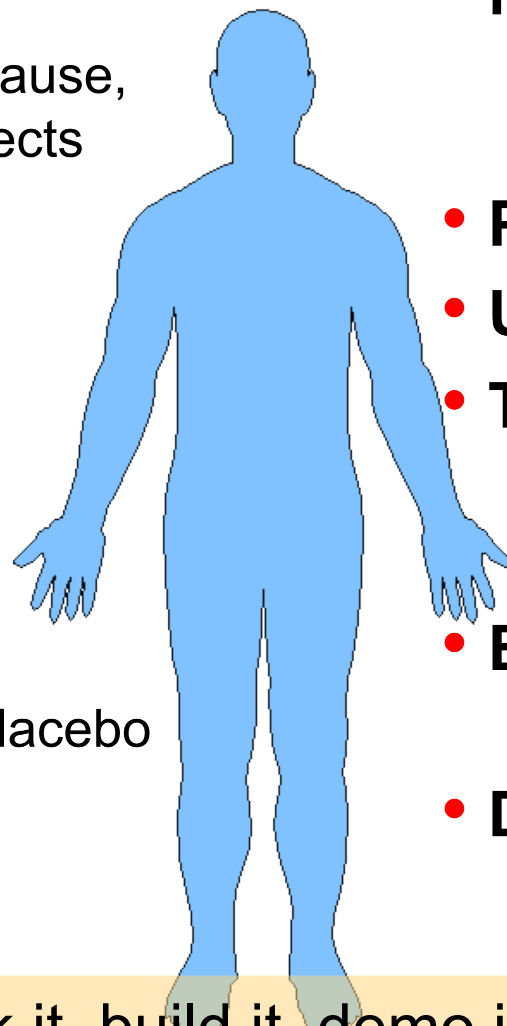
- **Goal:**
  - Improving the chance of ending up in a good local optimum. (that is: move the mean for better QOR)

- **That requires:**
  - Good understanding of cause, actions, side-effects
  - Statistical ***evidence of efficacy***
- **Issue:**
  - Effects and side-effects are hard to predict
  - How to distinguish design-specific noise from real improvements?

Not easy!

Medical tools vs. Physical design tools

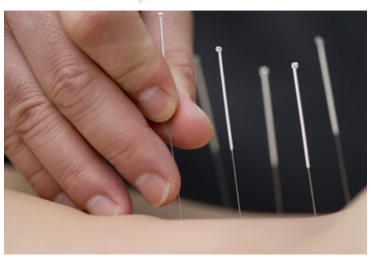
- **New drug**
  - Biological model of cause, actions and side-effects
- **Develop it**
- **Test tube test**
- **Test on animals**
  - Efficacy,
  - side effects
- **Clinical trials**
  - Large double-blind placebo -controlled tests
- **FDA-approval**

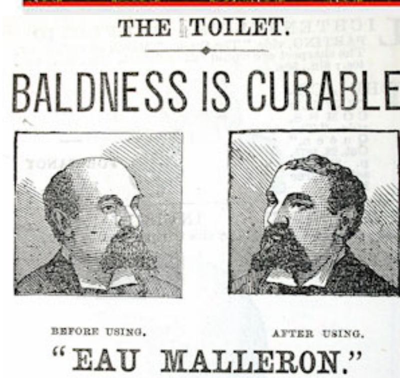


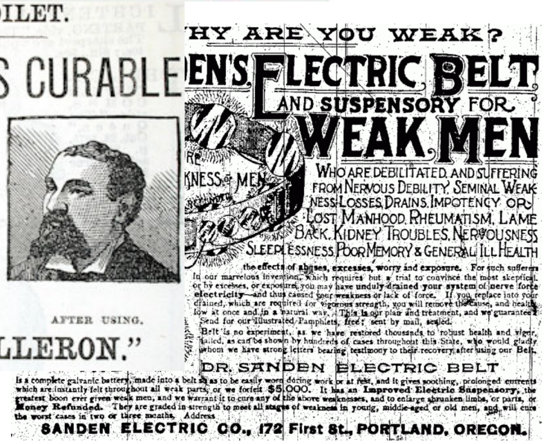
- **New flow component**
  - Based on electrical/ physical plausability
- **Program it (C++/TCL)**
- **Unit test**
- **Test on small testcases**
  - Debug program
  - Efficacy, side effects
- **Beta test**
  - Hope that customers use it
- **Deployment**
  - Go for it!

“Engineers: think it, build it, demo it, declare victory”

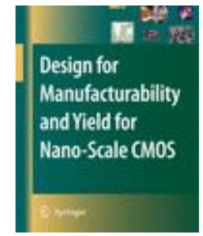
Lack of evidence = quackery







Physical Design is not exempt`:

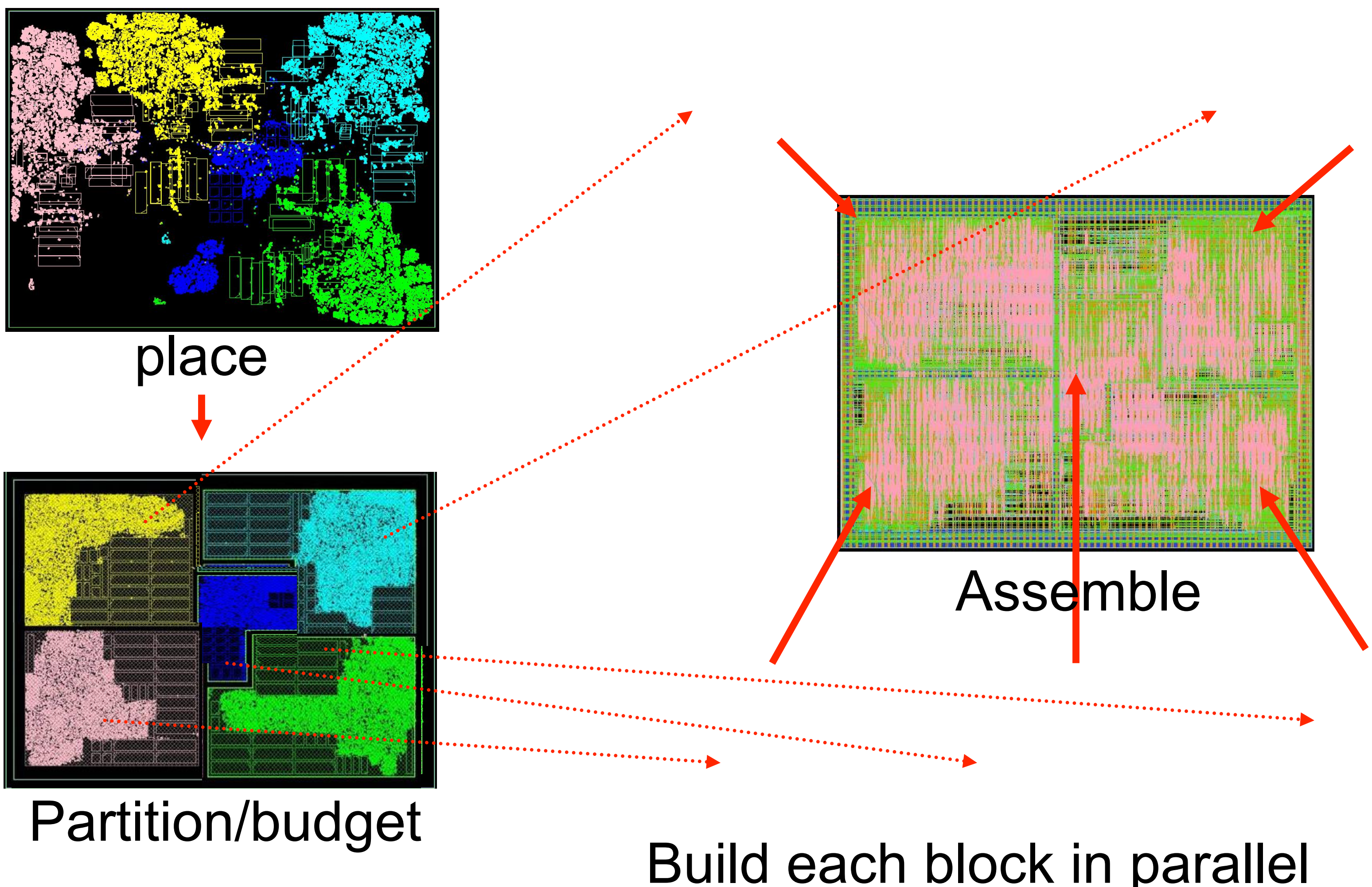


- Structured placement
- Thermal-driven placement
- Plug 'n play tool interoperability
- Running PD tools in parallel on a GPU.
- Gridless routing
- X-Architecture

Apply skeptical wisdom

- **“Humans are amazingly good at self-deception”**
  - This looks soooo good, therefore this *must* work
- **“If it has no side effects, it probably has no effects either”**
  - Example: improving temperature gradients will cost timing you! Are you really willing to pay based on the evidence?
- **“Do not confuse association with causation”**
  - “I took this airborne pill, and I did not get sick”
  - “I used this DFM optimizer, and the chip yields!
- **“The plural of ‘anecdote’ is ‘anecdotes’, *not* data”**
  - Result could be a random effect, or another side effect
  - No substitute for unbiased placebo-controlled tests
  - Only large data sets are statistically relevant

Coarse-grain partitioning to speed up



MAGMA

Summary: it's the flow, not the algorithm!

- **Need to deal with conflicting objectives**
- **Careful tuning of:**
  - Clever **A**voidance (as little as surgical as needed)
  - Incremental **C**orrection.
- **Need to focus on the dominant issues:**
  - Timing: very poor delay predictability
  - Design scale: keeping up with Moore's law
- **Be skeptical and honest!**
  - Negative results are as important and positive!