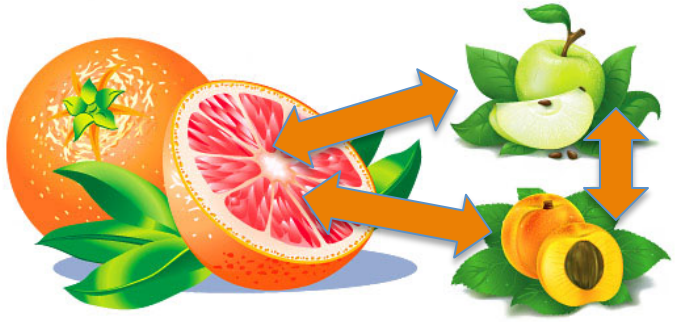
EDA,
or the art of trading off
apples, peaches and oranges

CANDE Workshop
November 10, 2011



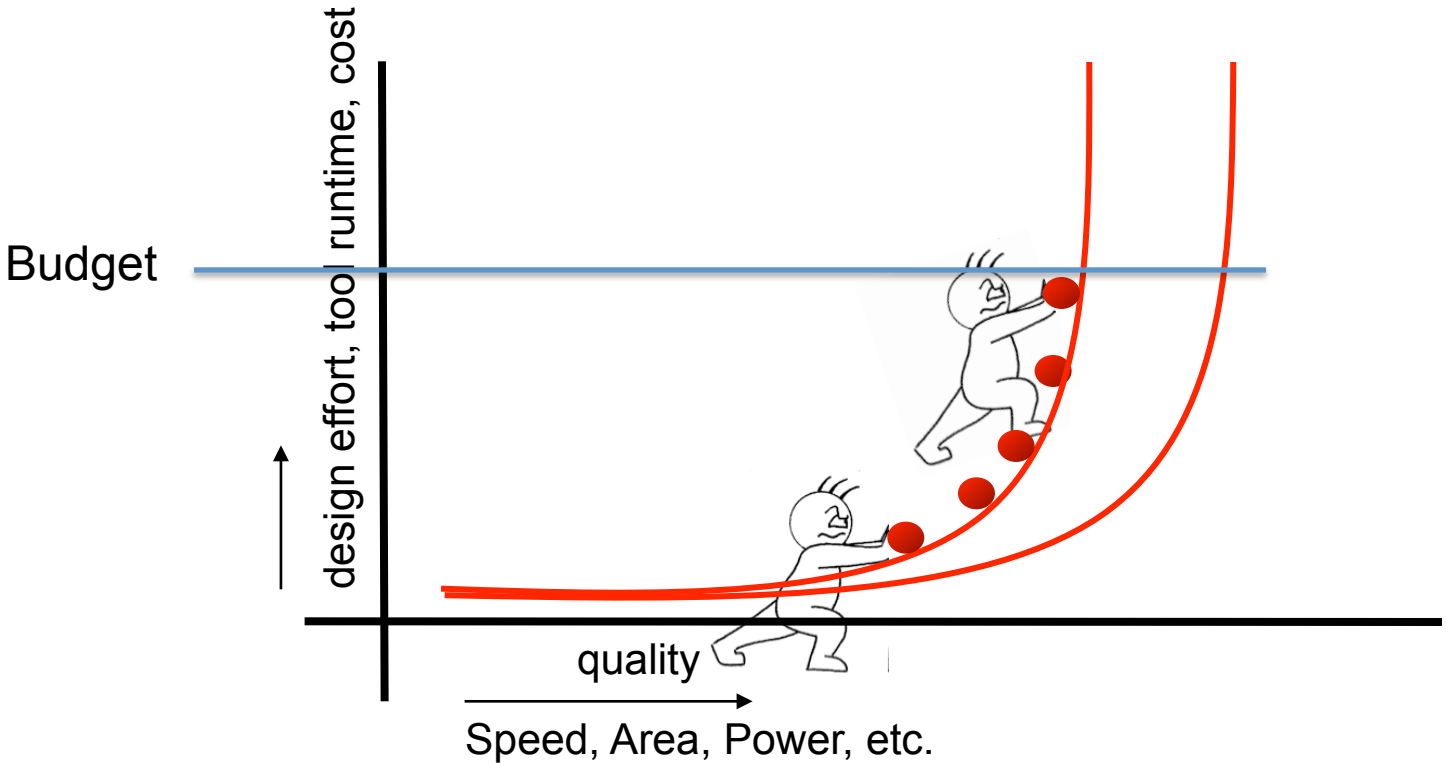
Patrick Groeneveld
Chief Technologist, Magma Design Automation

Silicon one

Kevin Trudeau, the king of Quacks

Trading off Apples, Peaches and Oranges…

Design *Effort* vs *Quality* tradeoff in EDA

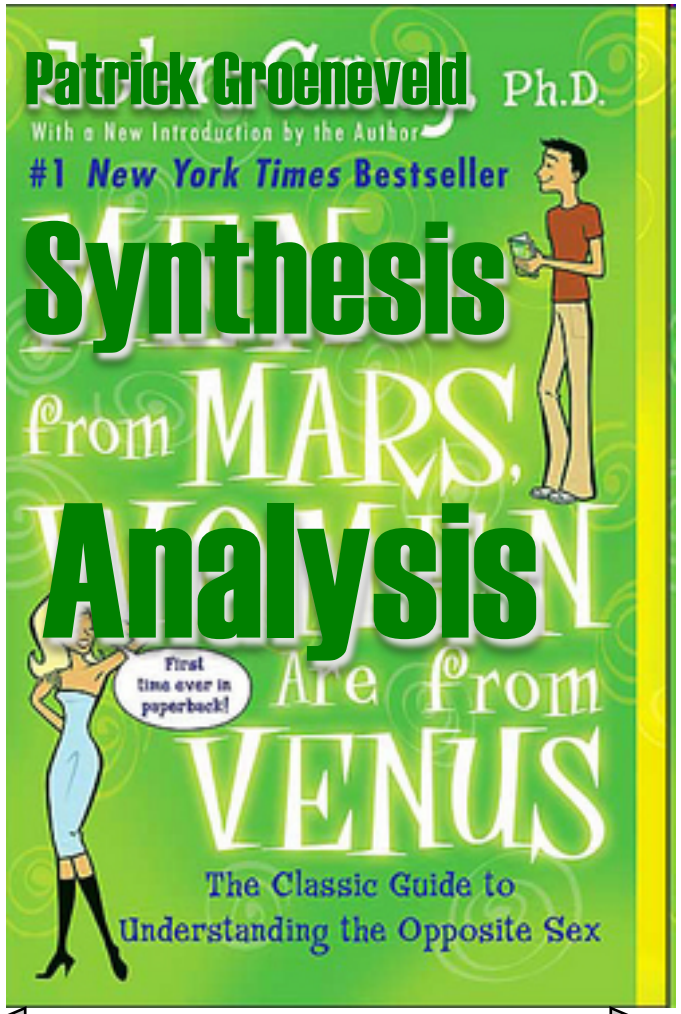


We'll settle for a common-sense point, given budget - back off, if there is not enough budget.

Synthesis is from Mars, Analysis is from Venus

Analysis

- Sign-off tools:
- Verification, Extraction, STA, spice, DRC, LVS
- Highly accurate
- Big and slow
- Parallelizable
- Is the ‘whiner’



Synthesis

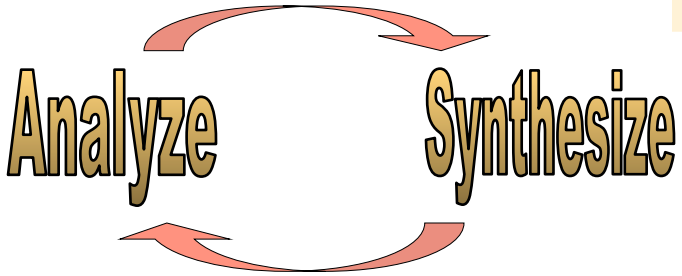
- Implementation tools:
- RTL synthesis, Placement, Routing, Optimization, Humans
- Poor accuracy
- Lean, mean
- Tough to parallelize
- Is the ‘hacker’

Need to make this ‘marriage’ work

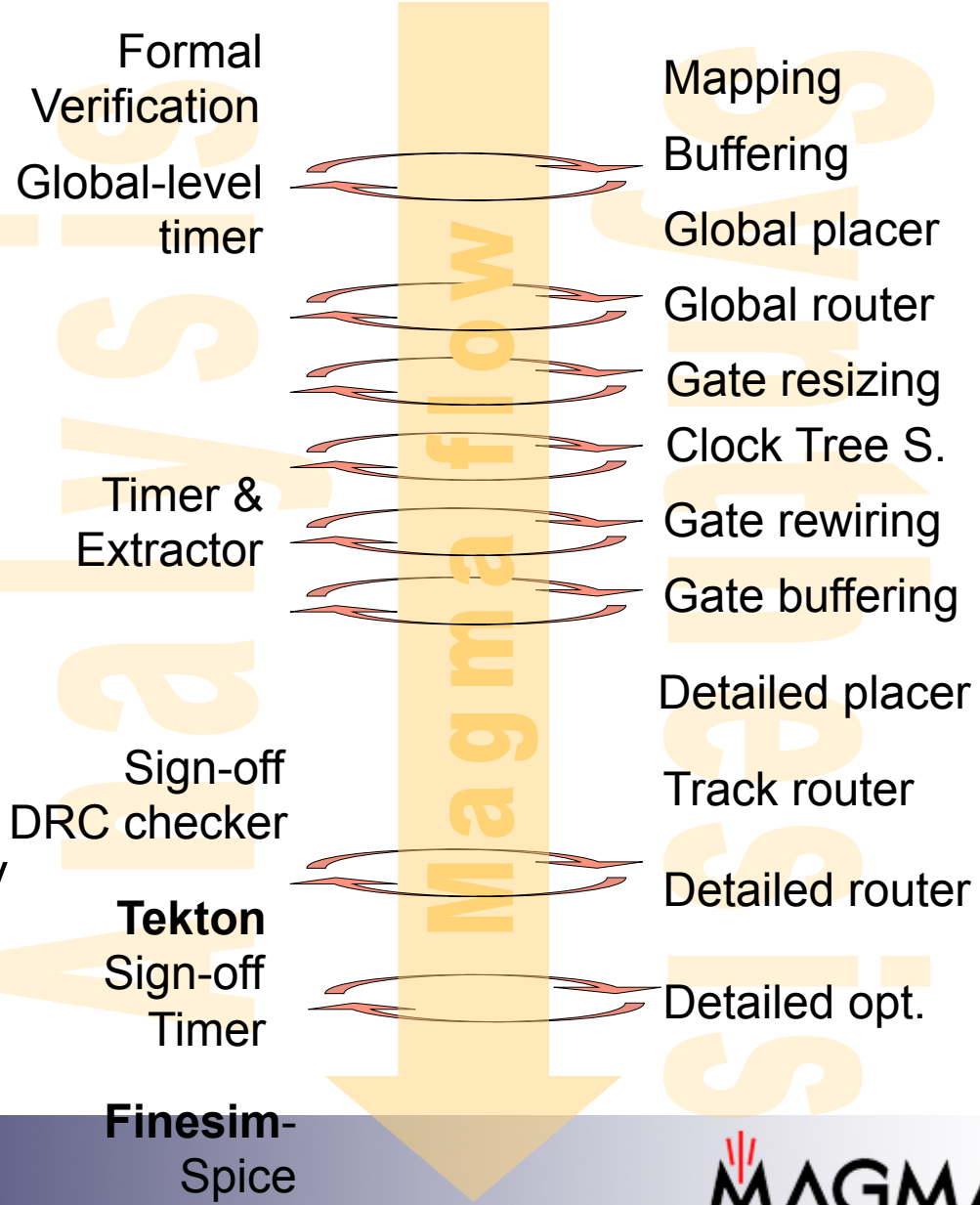
MAGMA

How design really works…

Iterate:



- Avoid loops:
  - Correct-by-construction methods
  - ABC flow
- Speed up loop by:
  - Reducing analysis accuracy
  - **Running tasks in parallel**
  - Take away walls between tools: Sign-off timer in the loop



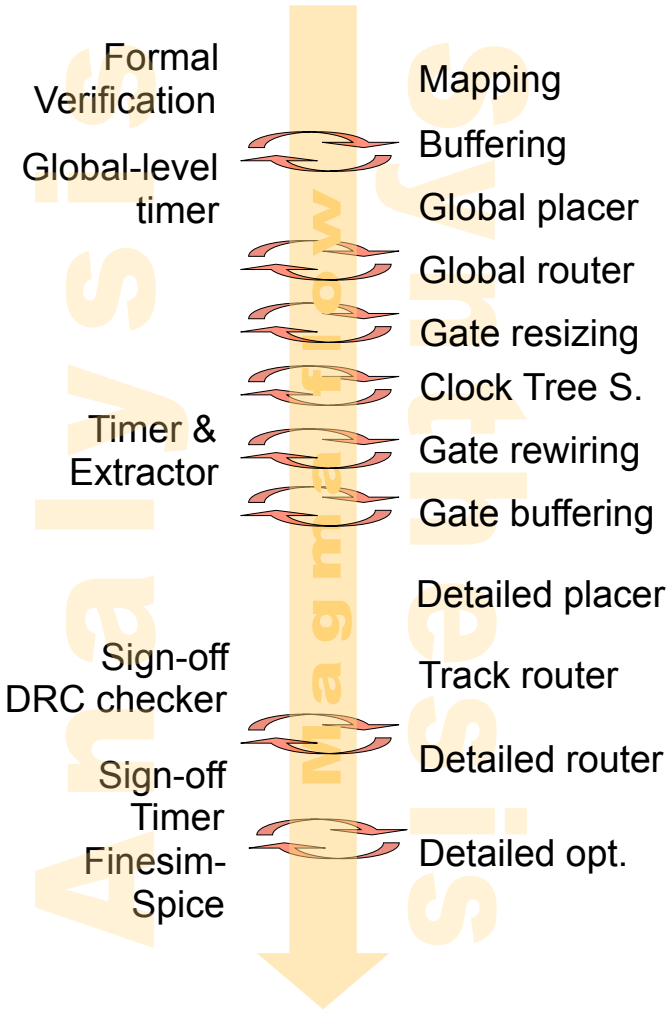
Building a Design Flow

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!

MAGMA

The ABC of a solid EDA Design Flow

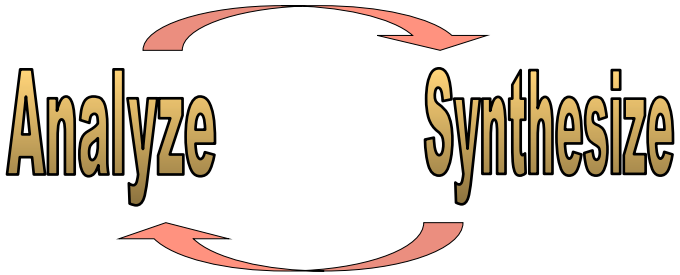
A: Avoid
Use pessimism to make problem unlikely, 'Correct by Construction'

More avoidance = worse results...

B: Build
Synthesize using an algorithm

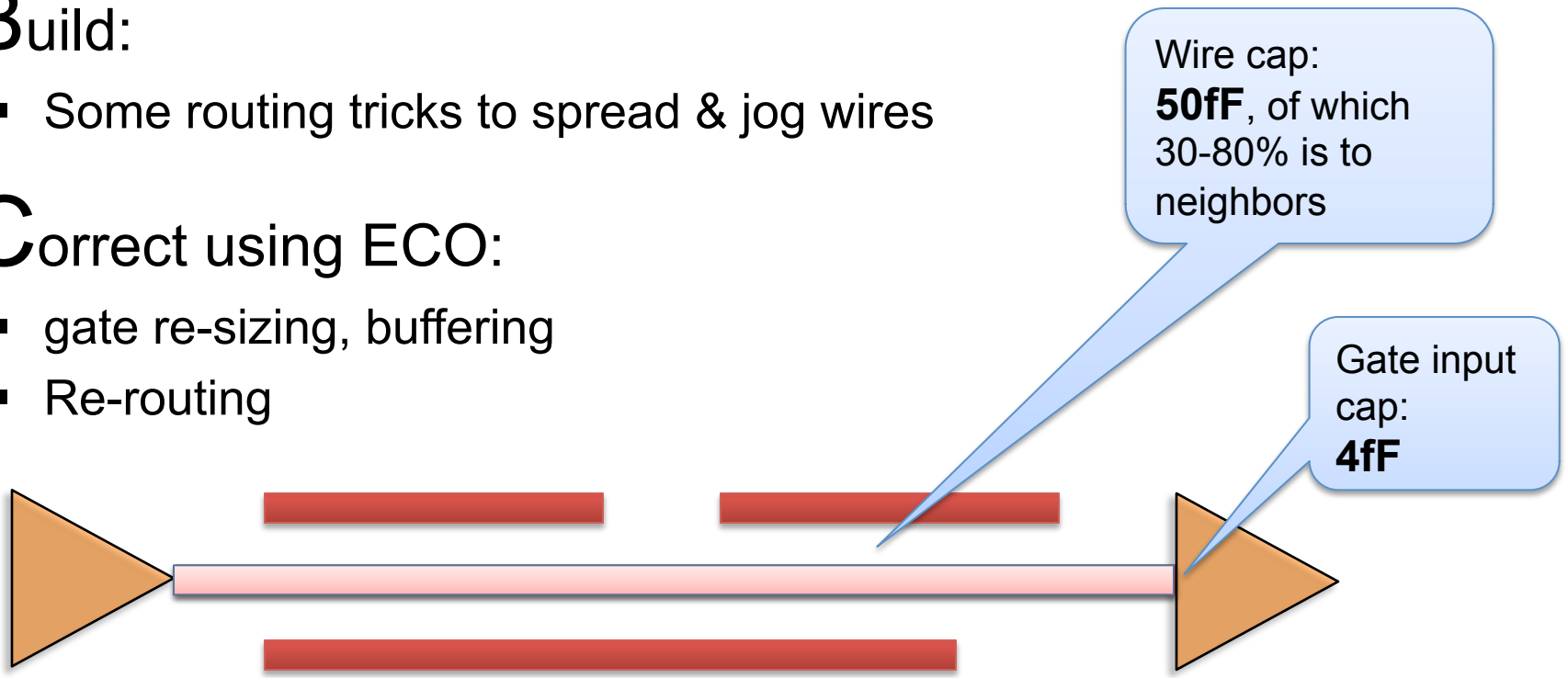
Synthesis is from Mars...

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

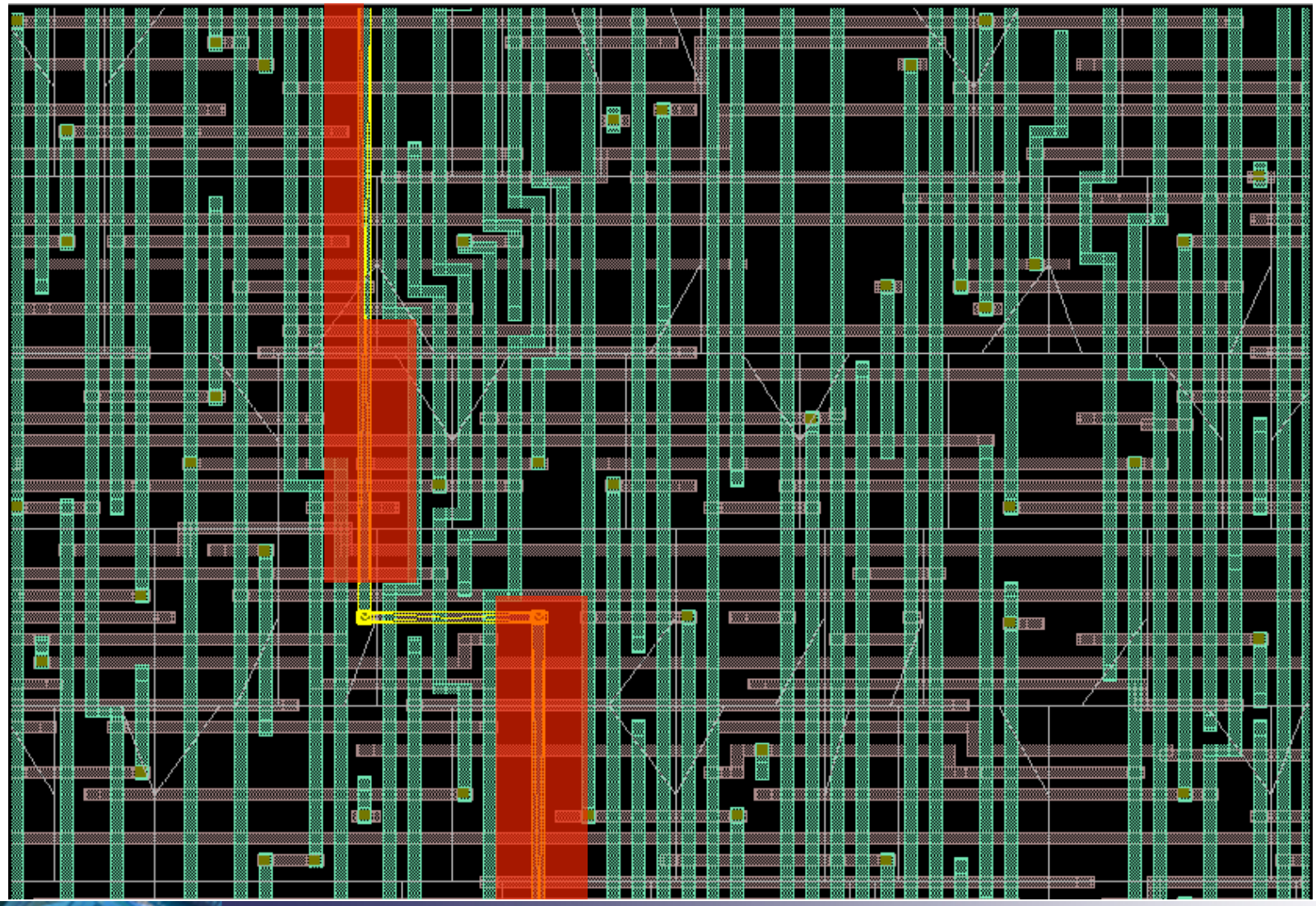


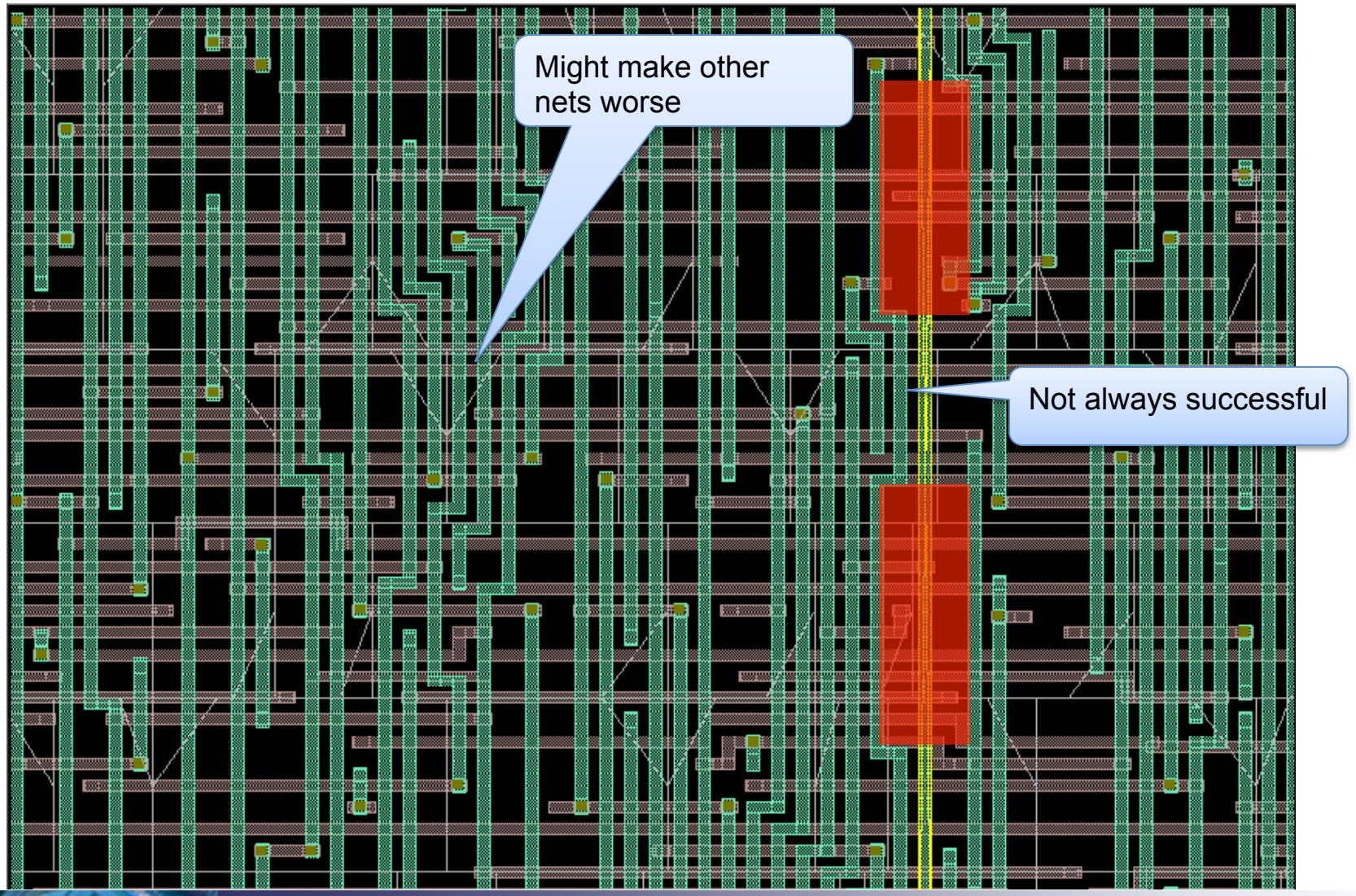
Example ABC: Combating crosstalk delay

- 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

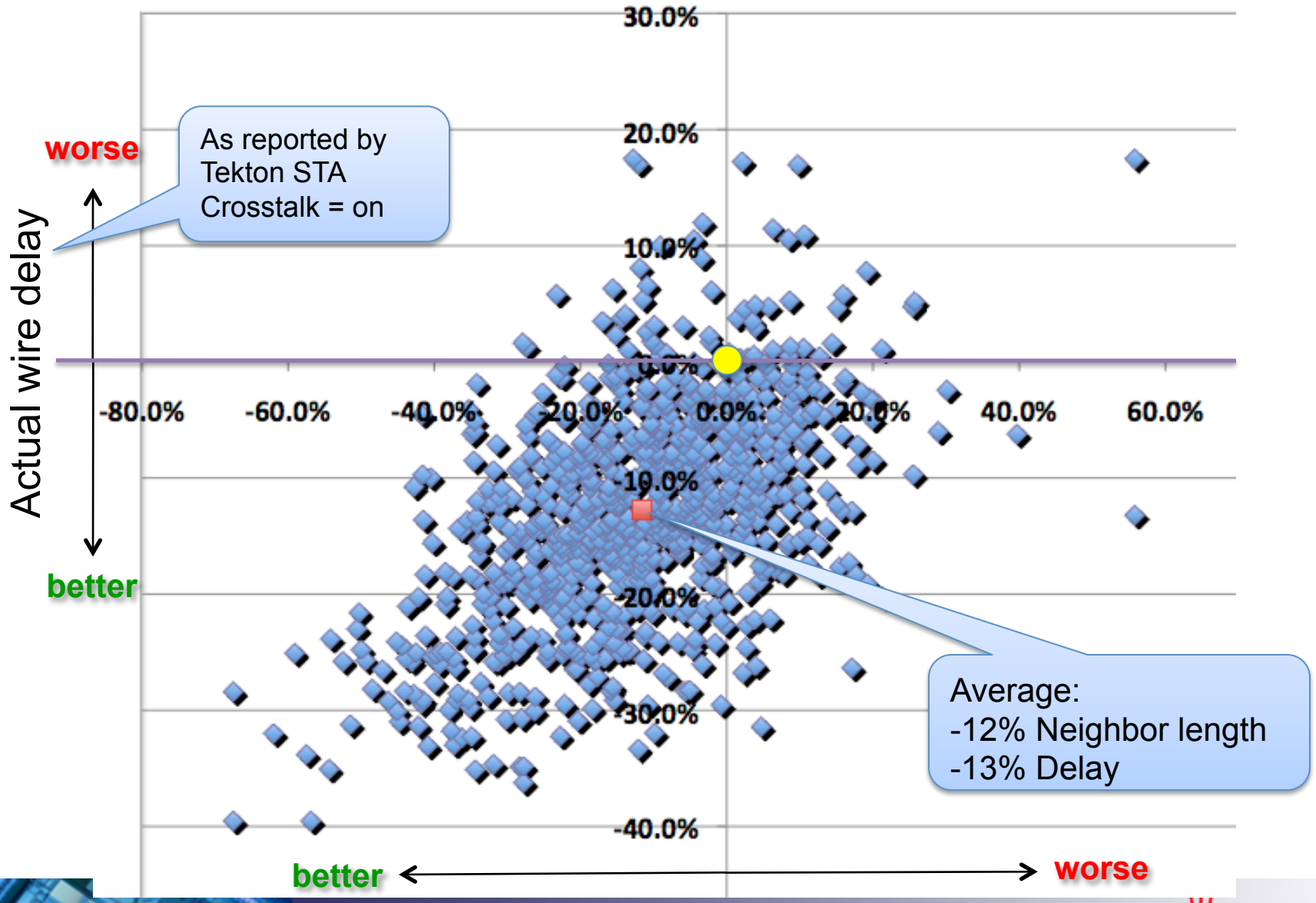


‘C’ routing improvement: pushing neighbors away

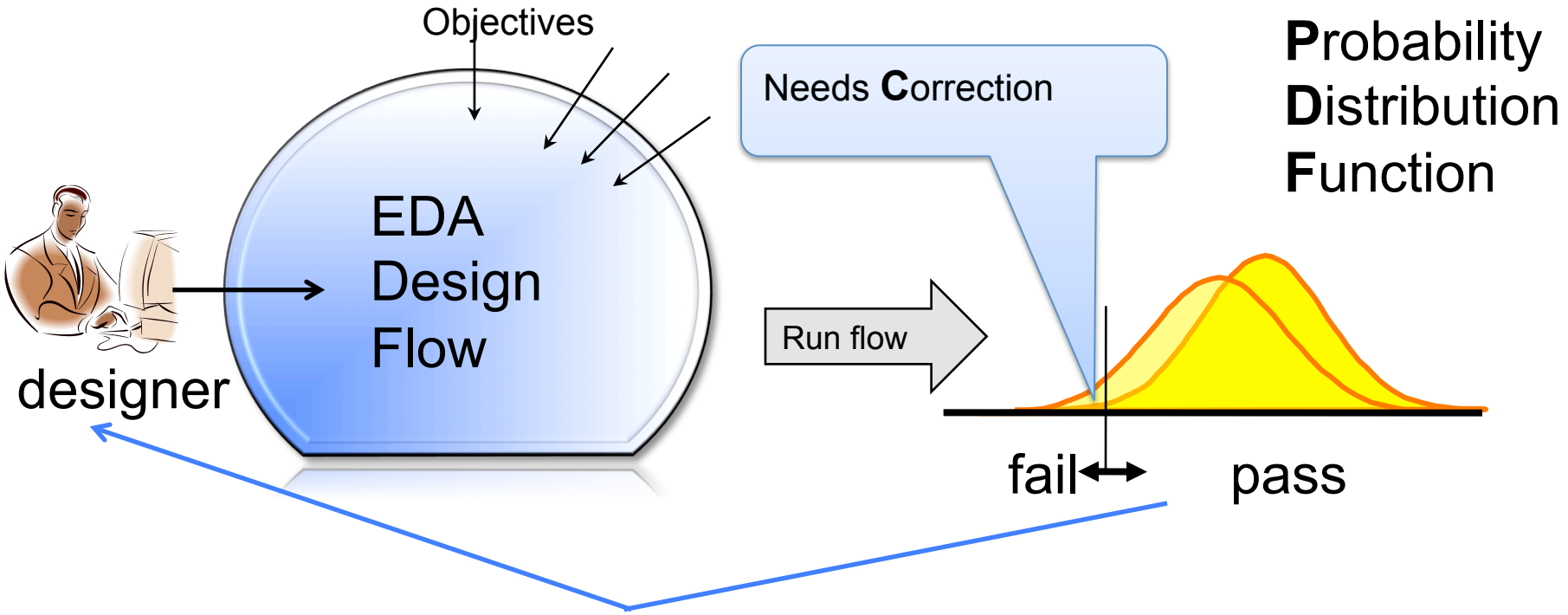



Might make other nets worse
Not always successful

Effect of this physical ECO on timing



Controlling the amount of Correction

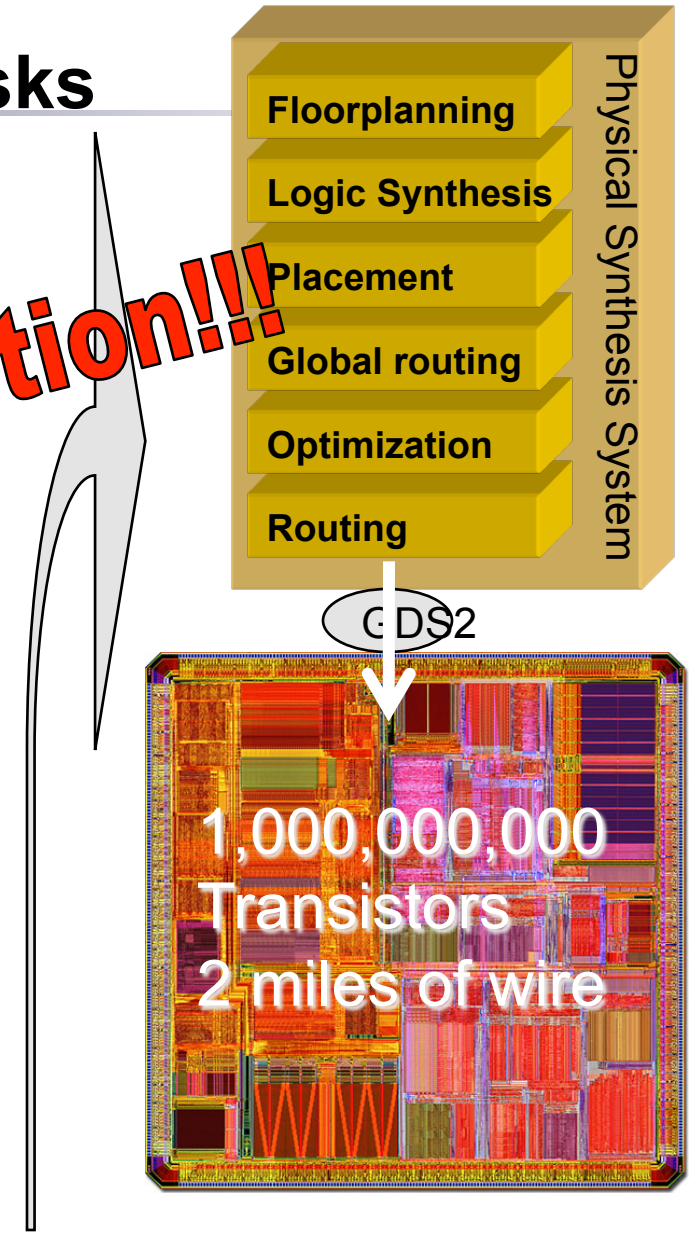


- Relax the objective
- More Avoidance (pessimism)
  - Which might deteriorate other objectives

Avoidance vs. Correction: masks

- Avoid:
  - DRC deck with hard rules
- Build:
  - Dijkstra grid 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!!!

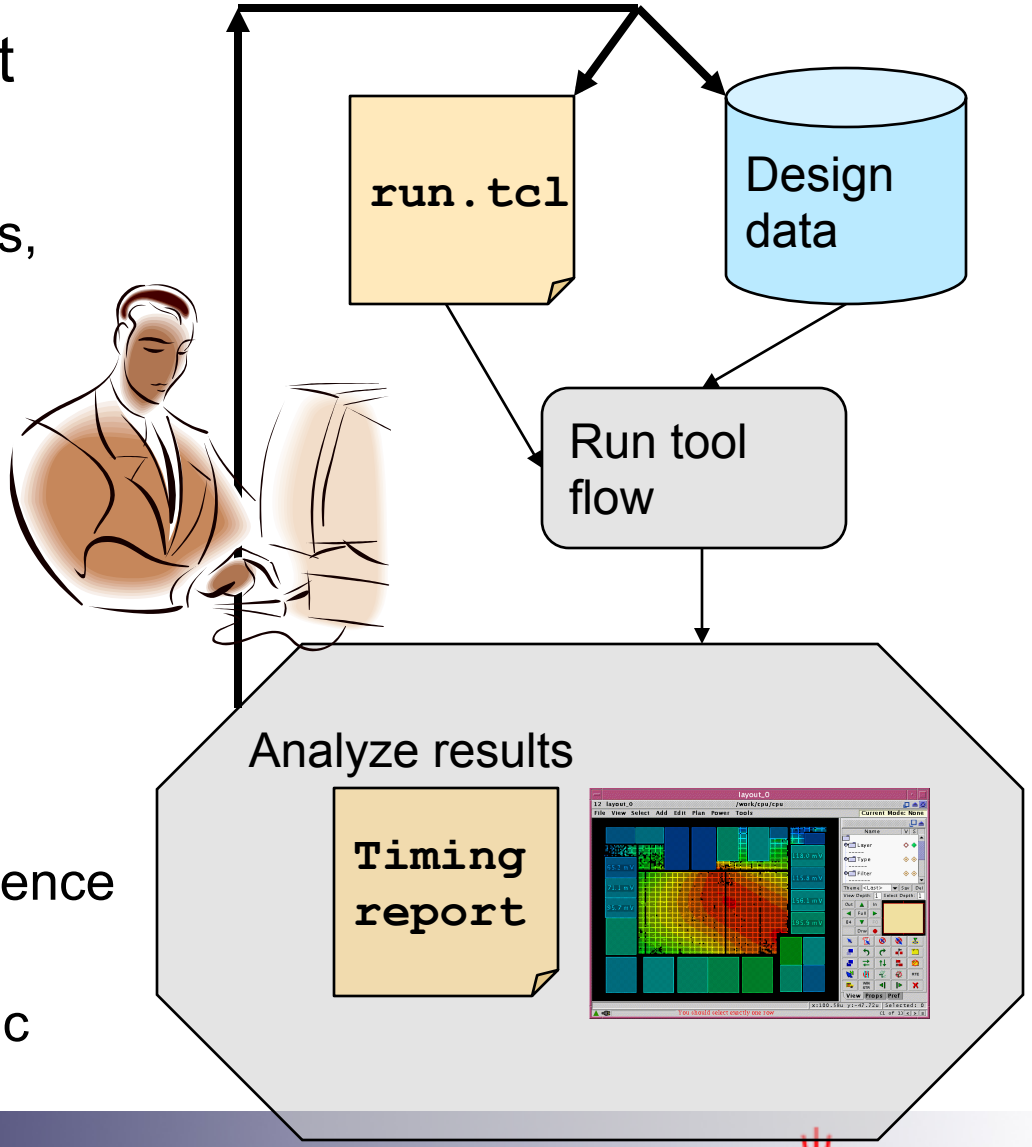


MAGMA

How to tune the EDA flow?

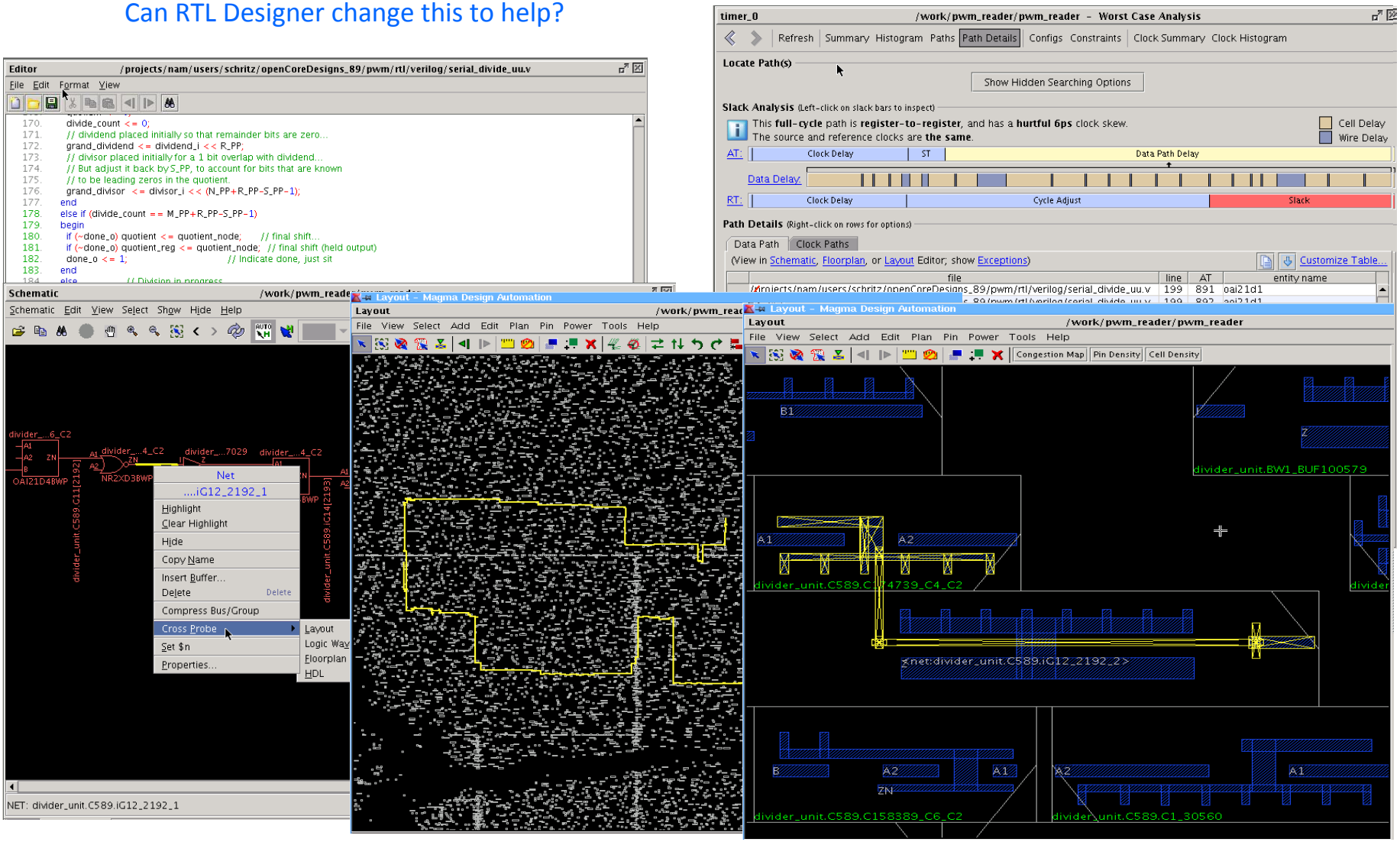
- Tuning of the TCL script
- First time:
  - Poor local optimum, bugs, mistakes
- Tune flow+data
  - Better local optimum.

- But:
  - Loop is slow
  - Tool talks gibberish
  - Result depend on experience of engineer.
  - Hacks are design-specific



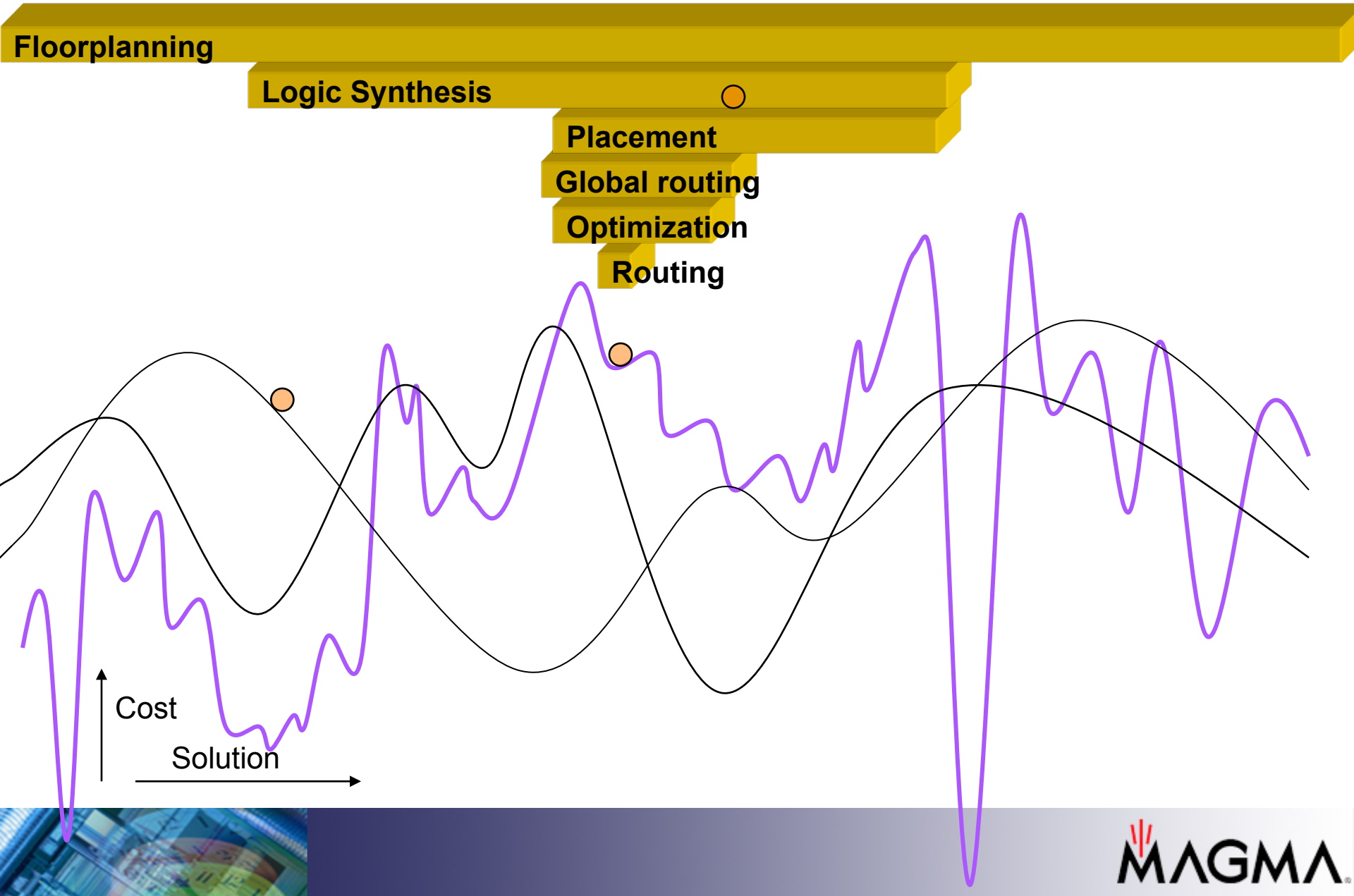
Debugging: finding what's wrong

1 line of RTL caused 16 gates in critical path
Can RTL Designer change this to help?



MAGMA

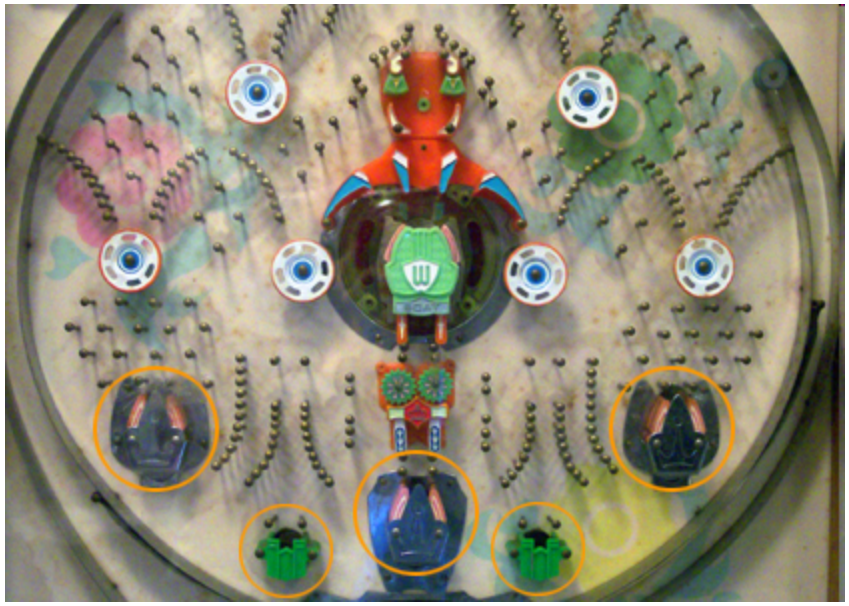
Local Optima the Design Flow



The EDA 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**

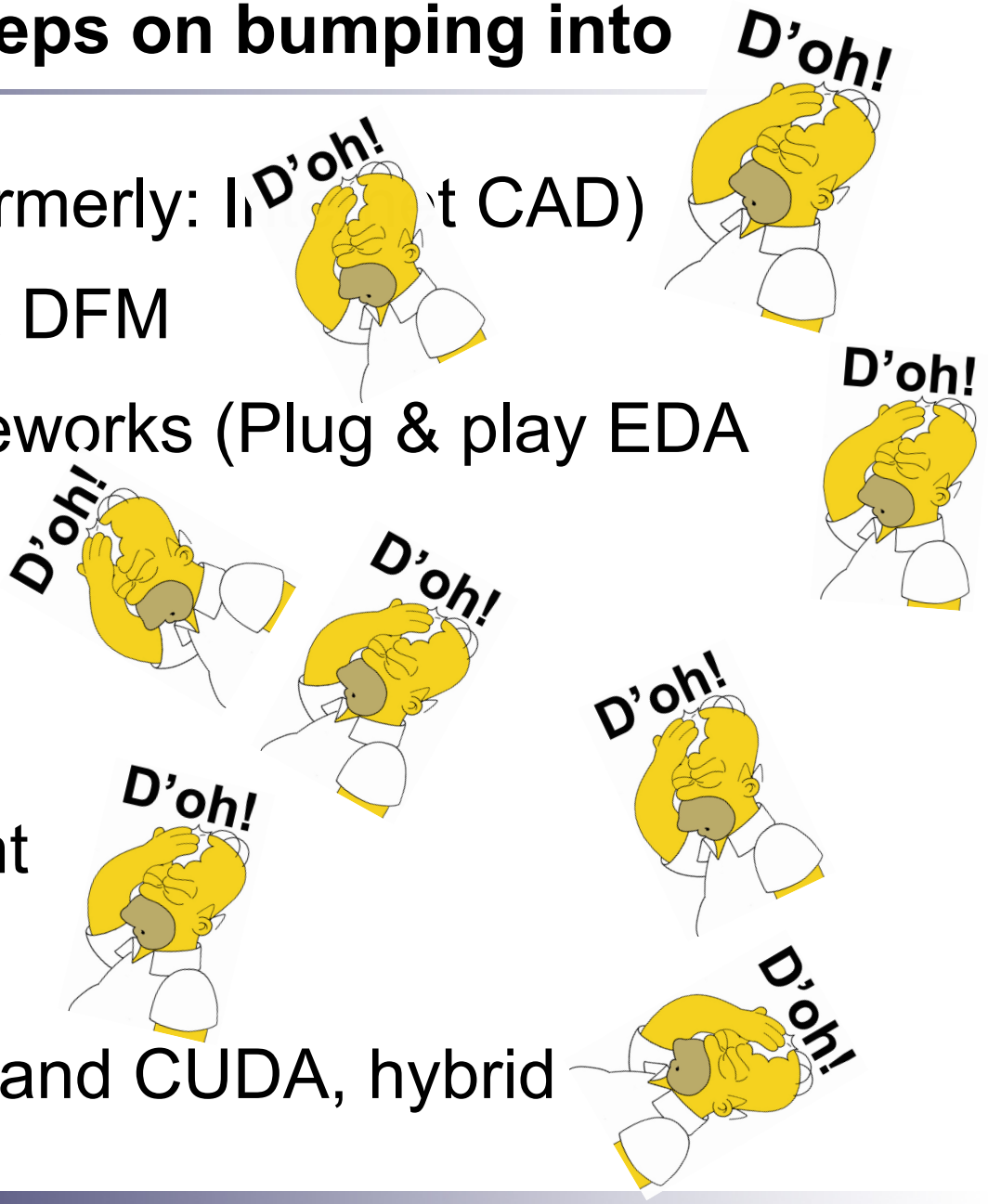


A donkey doesn't bump into the same stone twice

MAGMA

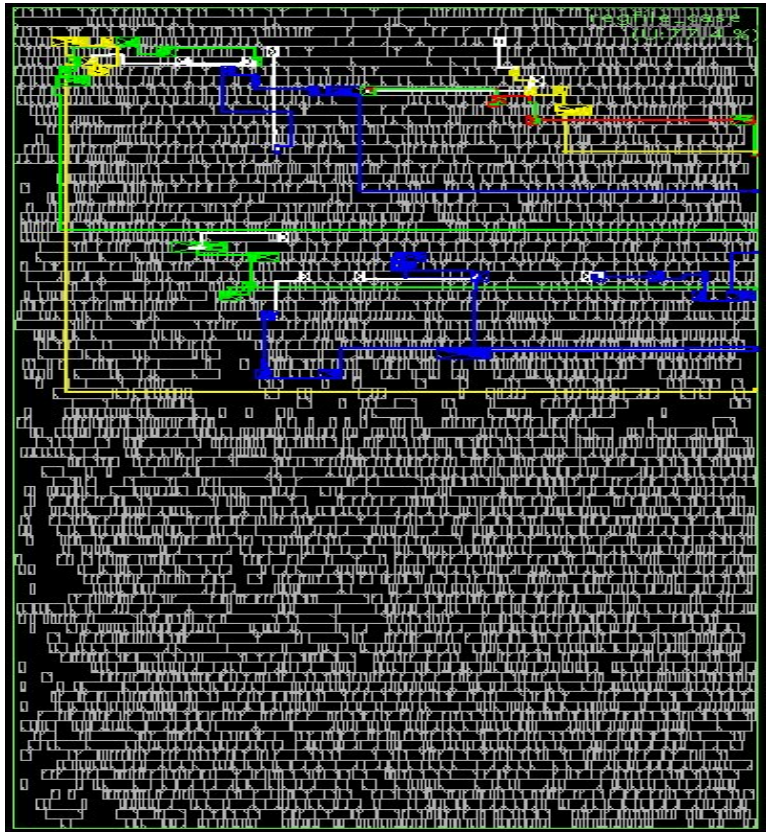
Bad ideas that EDA keeps on bumping into

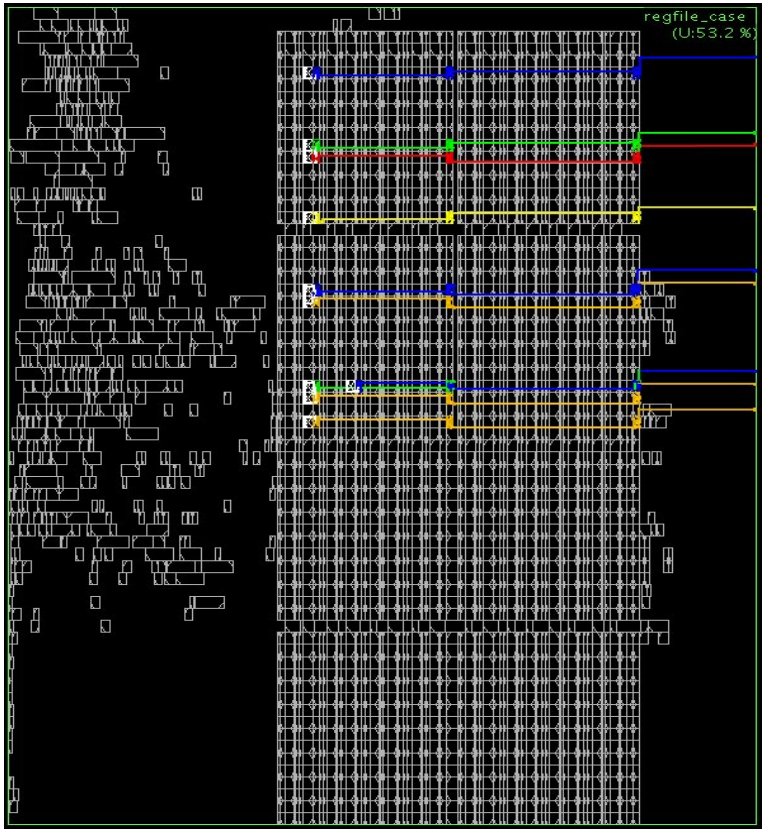
- Cloud computing (formerly: I[illegible]t CAD)
- Model based DRC & DFM
- Common CAD frameworks (Plug & play EDA tools)
- Thermal placement
- X-architecture
- Structured placement
- Multi-core EDA
- GPU's and OpenCL and CUDA, hybrid



EDA is Dumber than a Donkey, example #1

- Structured Datapath Placement



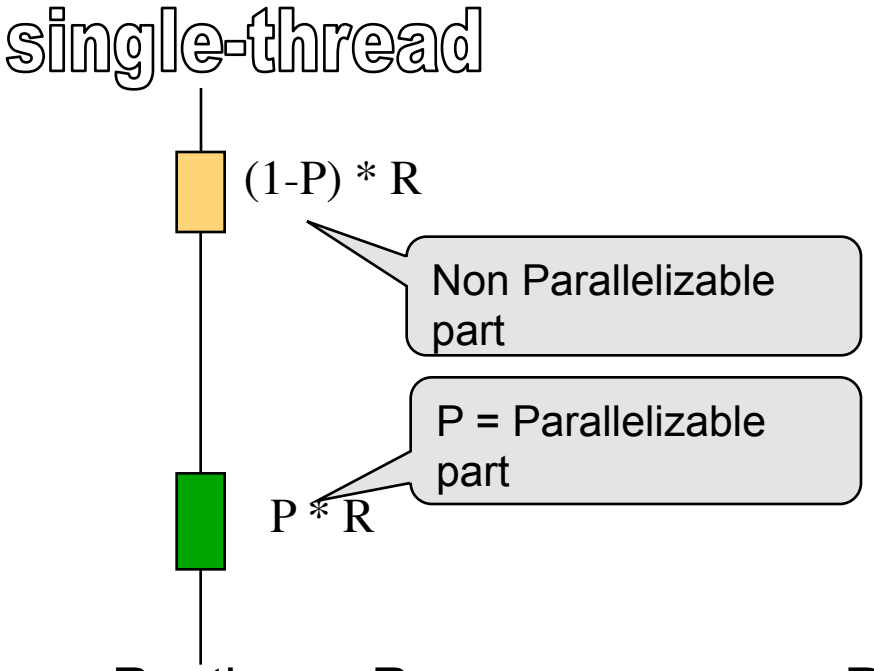


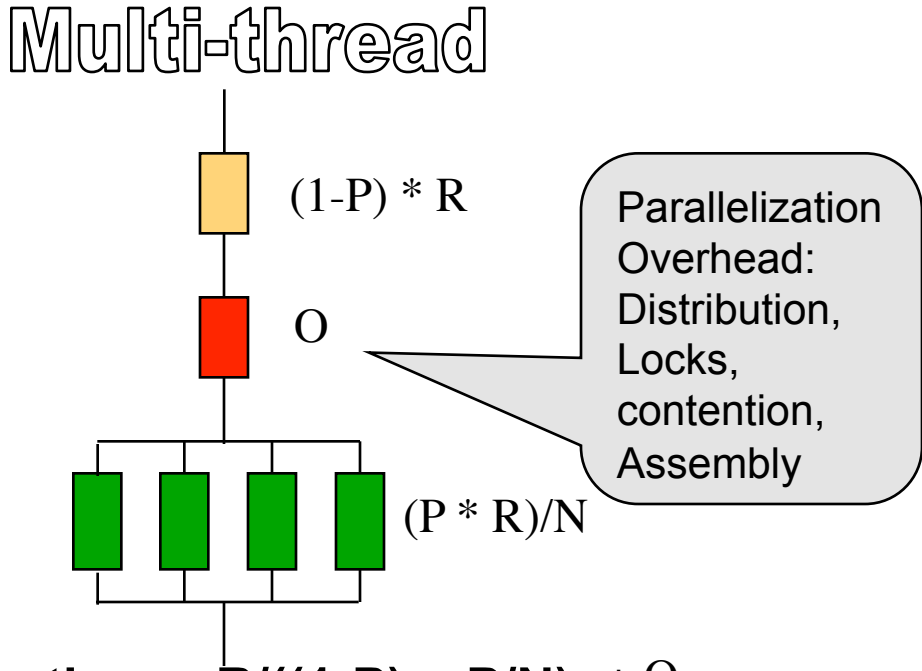
MAGMA

#2 Donkey moment example: Multi-core

"You need to rethink your algorithms"

Amdahl's law: Why parallelization gain tapers off





- Runtime = R
- **Run time = R/((1-P) + P/N)** + O

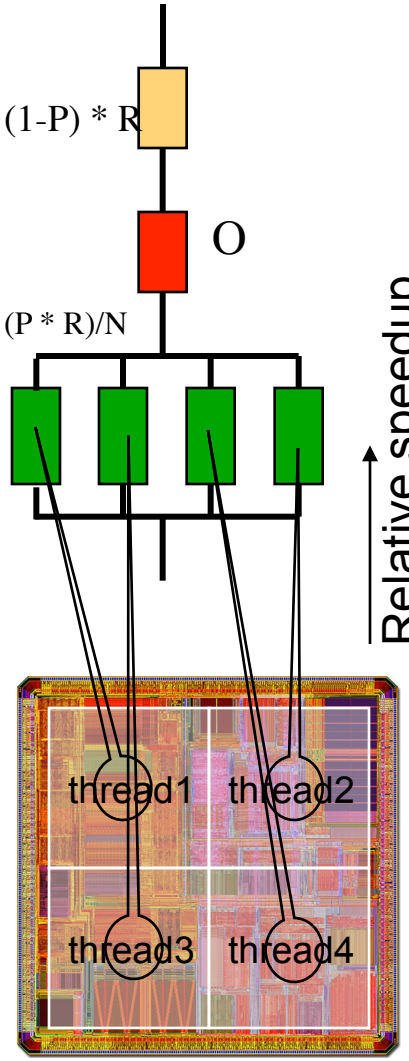
P	Maximum speedup	Reality
50%	2x	0.8x
80%	5x	2.0x
90%	10x	2.5x
95%	20x	2.8x

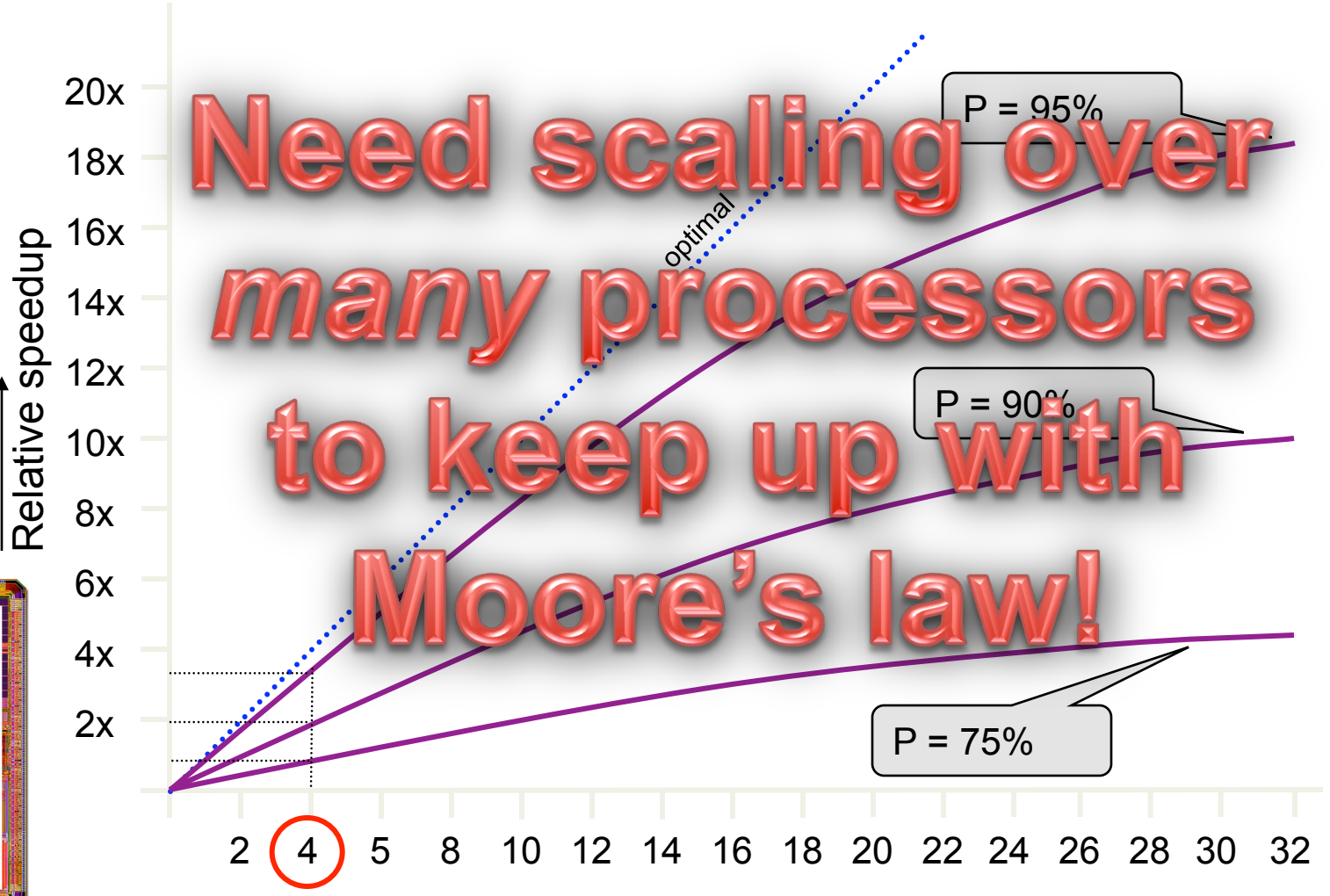
Must push P to 100%

Must minimize O

MAGMA

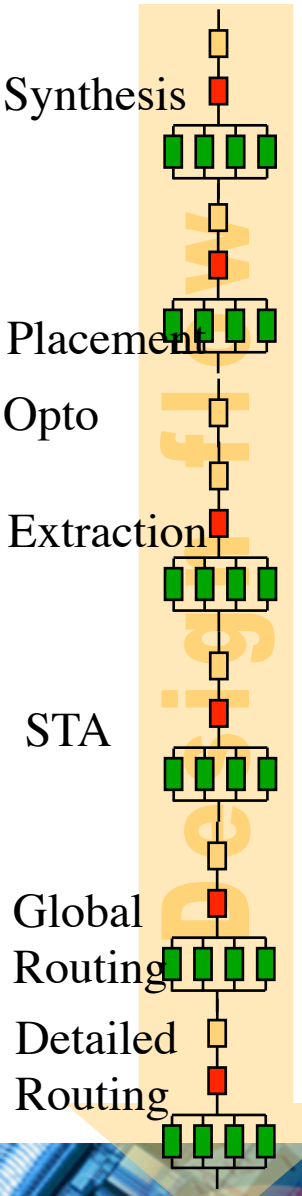
Parallelizing a single step in the flow

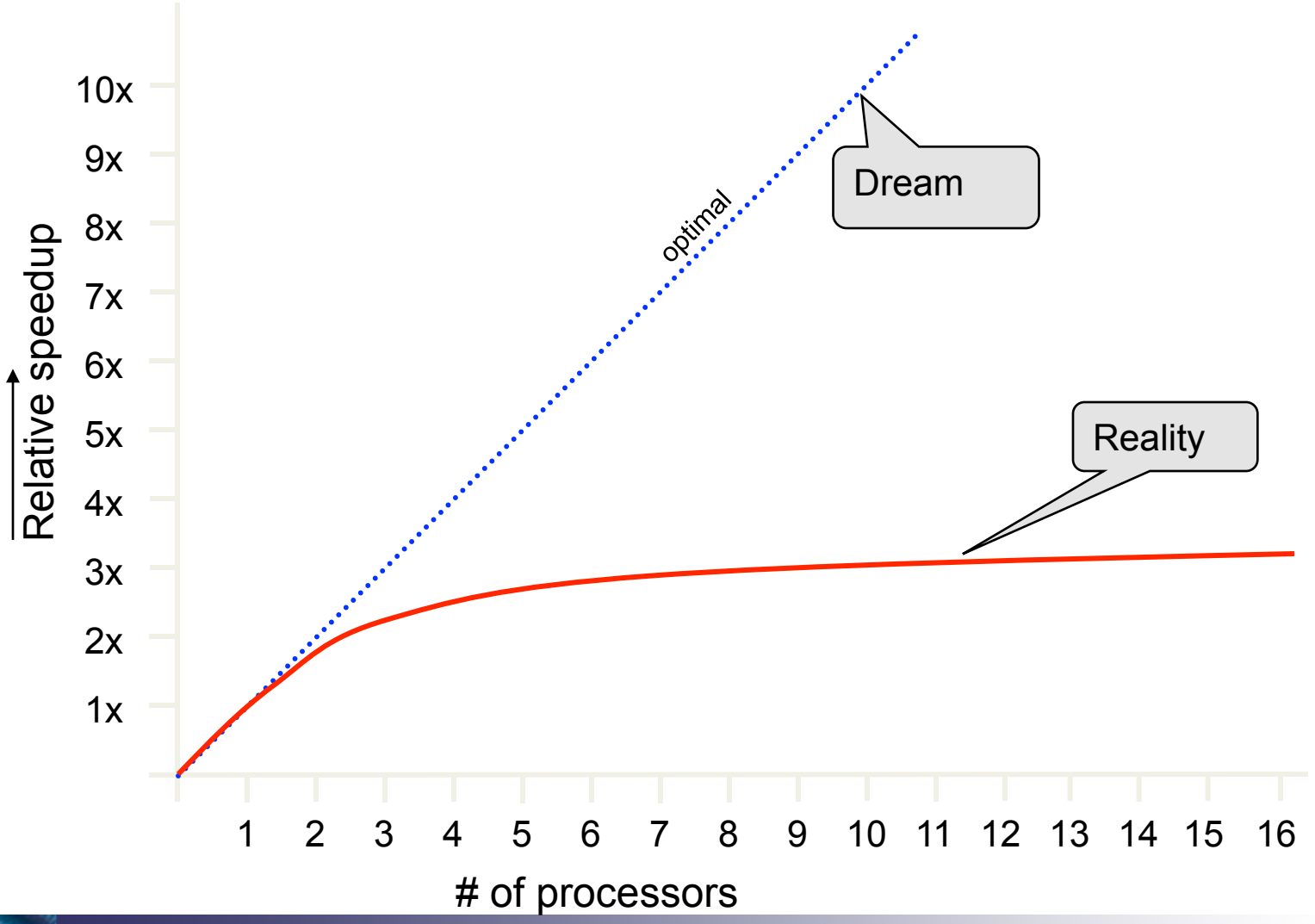




Need scaling over many processors to keep up with Moore's law!

Parallelizing the flow: Can we break the barrier?

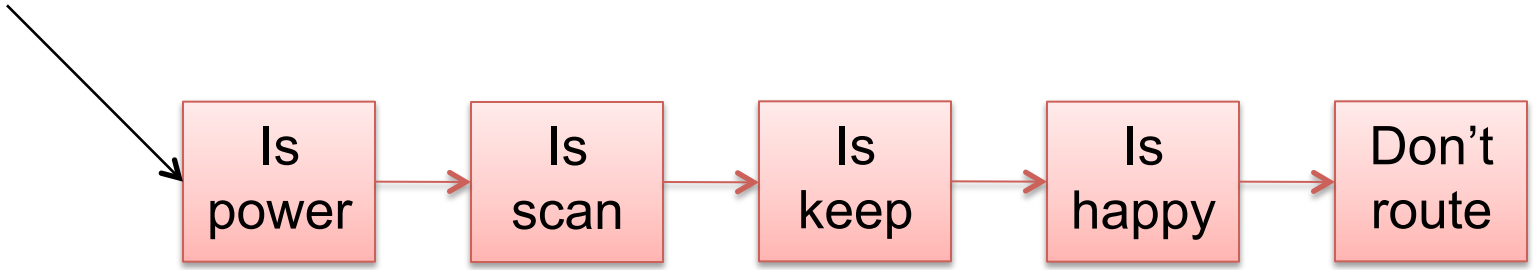




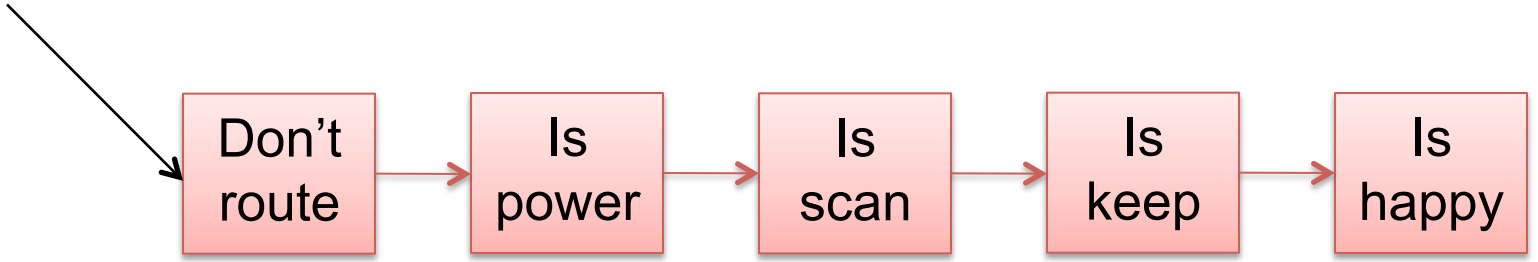
MAGMA

Parallel locking

Net properties



Clever idea: reorder after read: popular objects get in front



Since read messes with the list, I need a lock on EVERY read

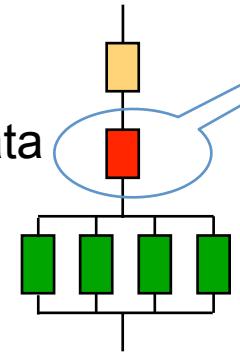
Unlocking parallel potential

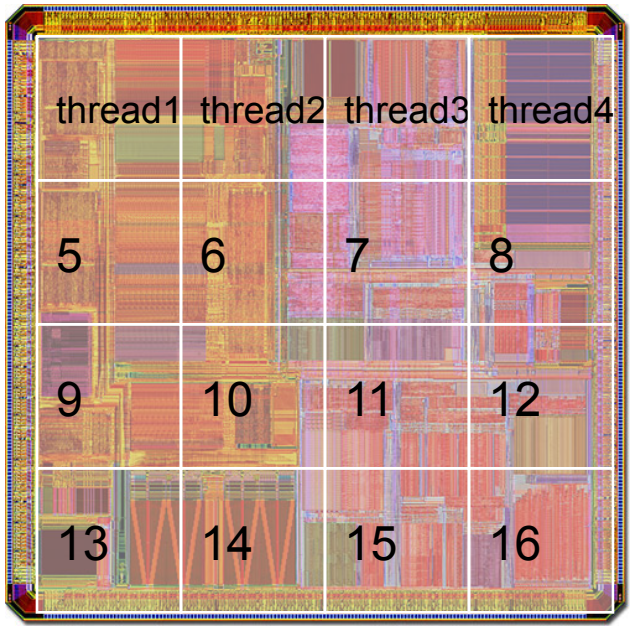
- Locks can easily kill potential multithread gains.
- Avoid locks: Duplicate contended data
  - Sledge hammer: duplicate all data (OS support for that)
  - Costs time and memory
  - Complicates code
- Avoid locks: by construction
  - Work on non-overlapping data

Best: have zero interaction between threads

Parallelization requires extremely *low* overhead

- Resource bottlenecks
  - Bandwidth to memory or disk
  - Many EDA problems have poor data locality due to design size



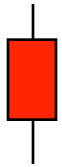


- Design partitioning and re-assembly
  - Non-trivial for EDA problems

- Interactions between threads
  - Data dependencies between threads kill speedup
  - No locks!!

Need 100% independent partitions

Partitioning is *Evil* for synthesis



- Why is it evil?
  - Overall quality suffers
    - Cannot optimize across boundaries
  - Partitioning problem is proven tough
  - Good partitions take (non-parallelizable) effort!
    - Algorithmic
    - Need to duplicate data

Partitioning:
A necessary evil
for the sake
of parallelism?

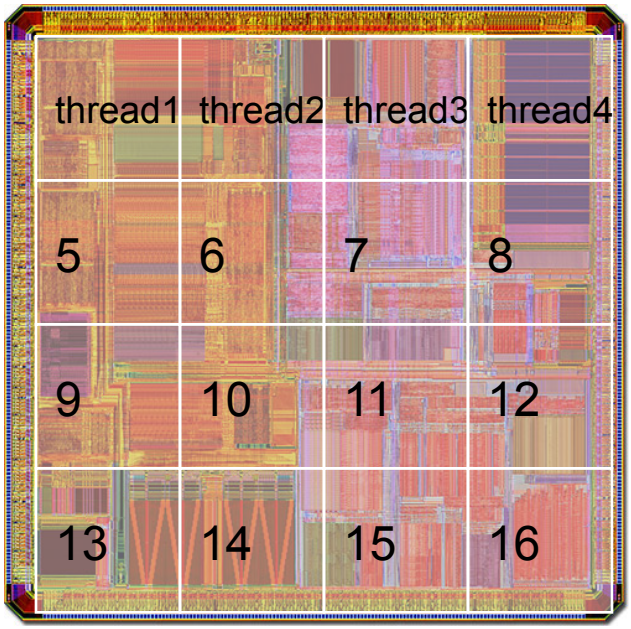
How to partition a problem for parallelism?

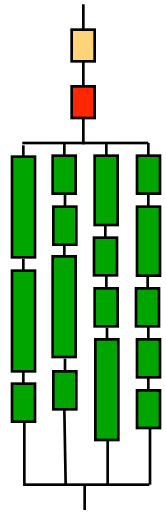
- Observation 1:
  - ***Analysis*** tools are much easier to parallelize
  - They do not change design state

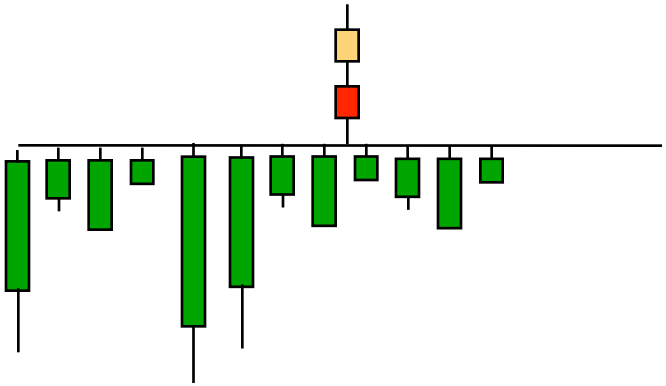
- Observation 2:
  - ***Synthesis*** tools change design state
  - Design changes while its being worked on.

Issue: Load distribution

- Load is not predictable





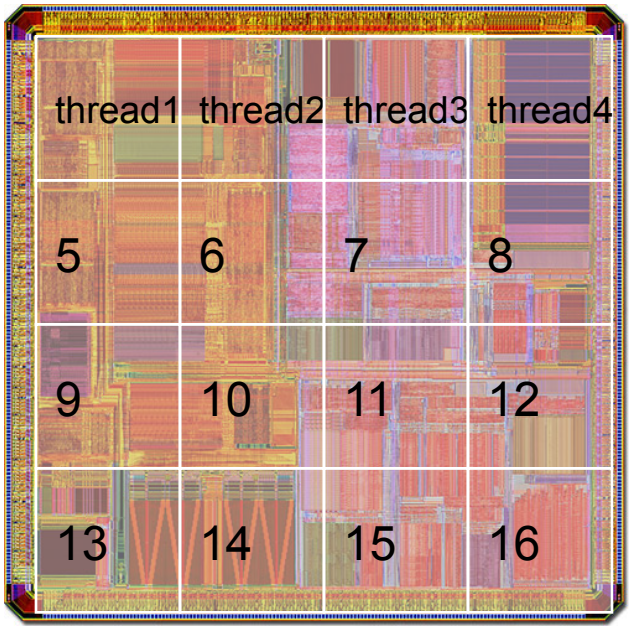


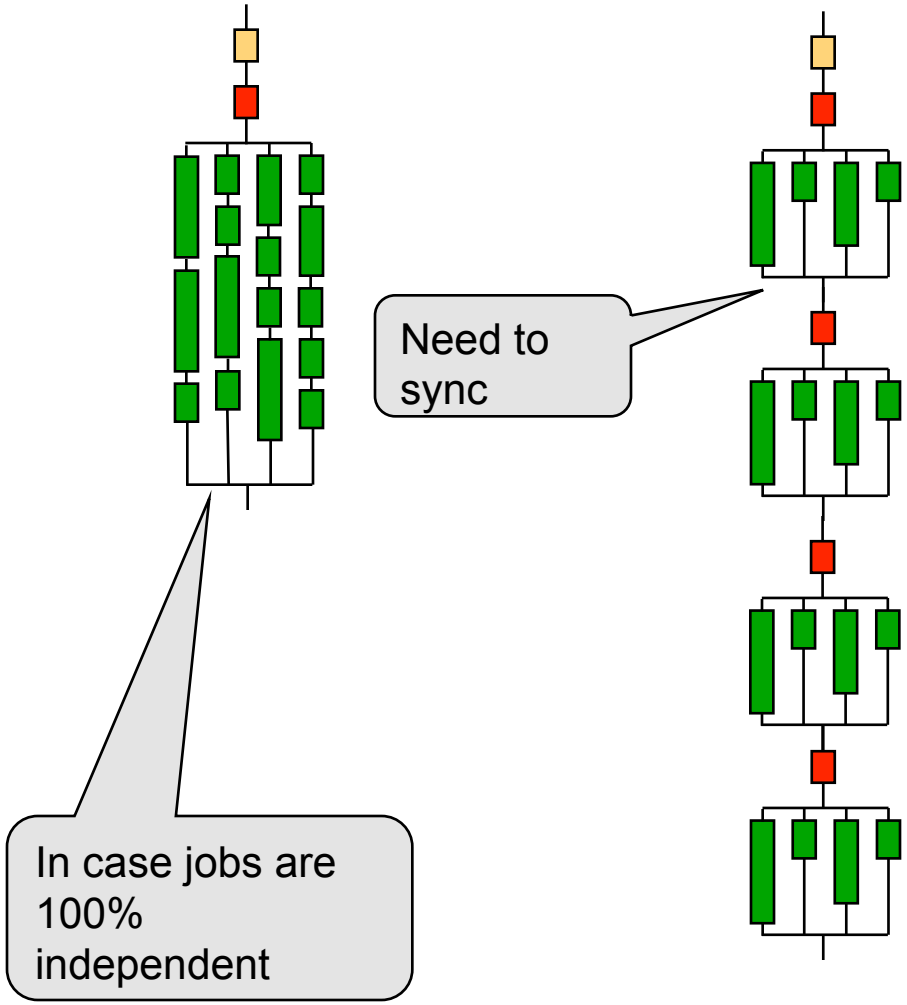
16-core: Effective utilization:10%

4-core: Effective utilization:
95%

Issue: Repeatablity: parallelism's silent killer

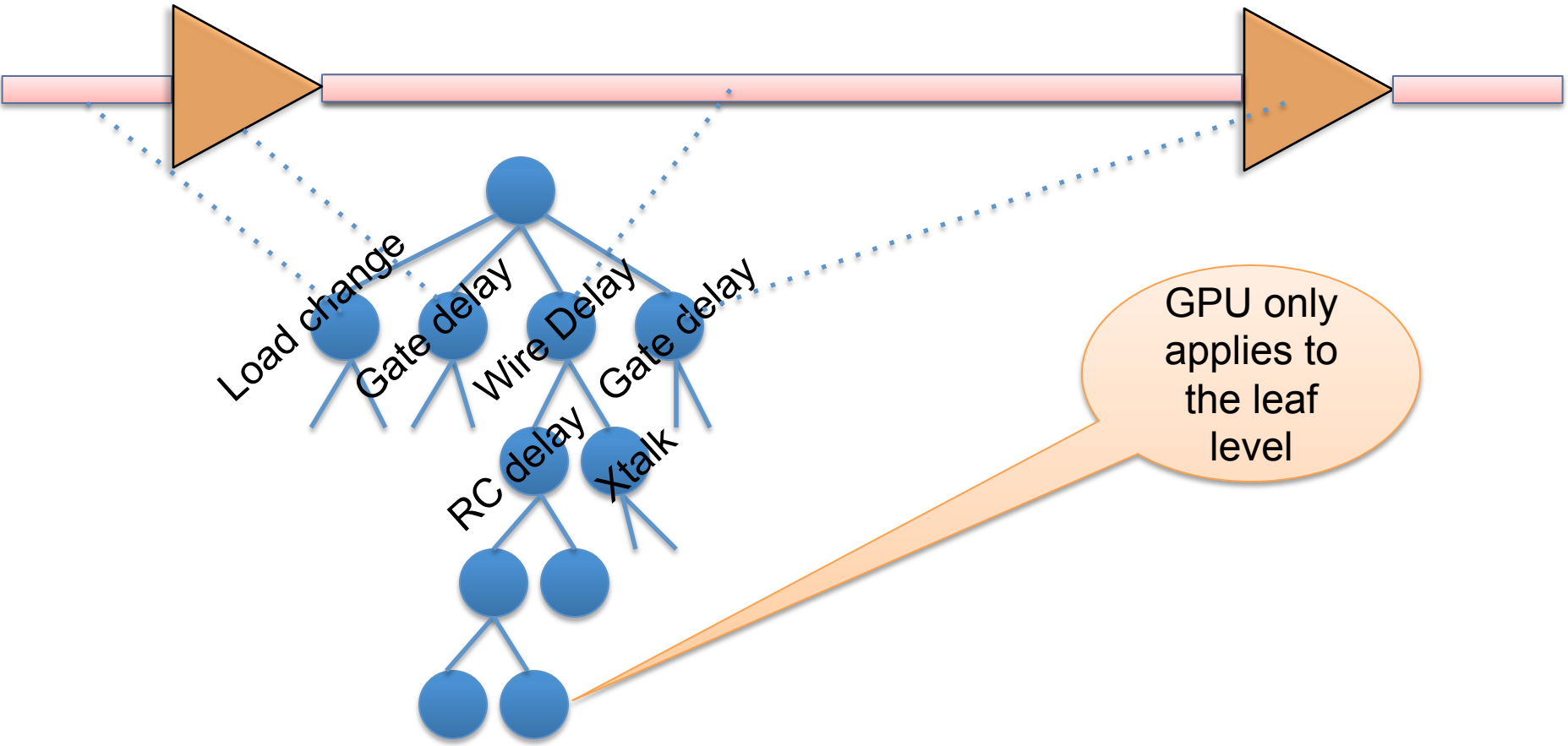
- 4 processors, 16 jobs to do.





CUDA & EDA: What's wrong with this picture??

Why Friends don't let Friends program OpenCL/CUDA



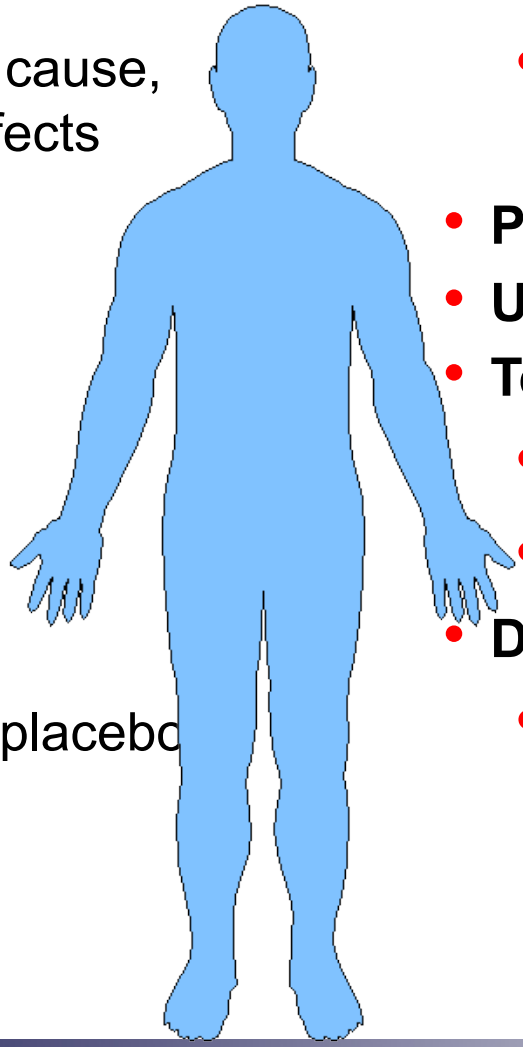
There is little glory in writing low-level hacks

Run!

- Hybrid solutions are bad ideas

Medical tools vs. EDA tools

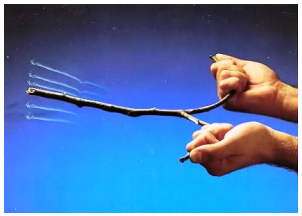
- New drug
  - Biological model of cause, actions and side-effects
- Develop it
- Test tube test
- Test on animals
  - Efficacy,
  - side effects
- Clinical trials
  - Large double-blind placebo controlled tests
- FDA-approval
- Deployment

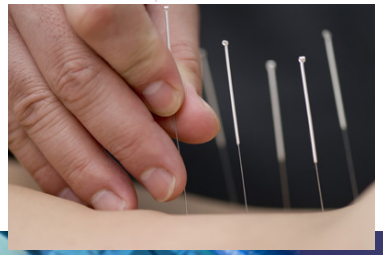


- **New flow component**
  - Based on electrical/ physical plausibility
- **Program it (C++/TCL)**
- **Unit test**
- **Test on small testcases**
  - Debug program
  - Efficacy, side effects
- **Deployment**
  - Go for it!

"Engineers: think it, build it, demo it, declare victory"

Lack of Evidence = Quackery





EDA
is not exempt:

OpenCL

- Structured placement
- Thermal-driven placement
- DFM-driven design
- Plug ‘n play tool interoperability
- Hybrid GPU/CPU EDA tools.
- Gridless routing
- X-Architecture

MAGMA

Skeptical wisdom for EDA

- "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