From Hard-Coded Heuristics to ML-Driven Optimization: New Frontiers for EDA

Patrick Groeneveld
Cerebras Systems / Stanford

ML in the EDA Flow: Inside and Outside of the Loop

Logic Optimization

Placement-based Optimization (PBO, PlaceOpt)

Routing-based Optimization

C++

High Level Synthesis

RTL

Logic Synthesis

Mapped net list, minimu… …ed

Timing Opto

Conditioned net list

Global placement

Placed gates

Physical Opto 1

Placed Gates, global route

Global routing, Detailed placement

Placed gates, global ro…

Physical Opto 2

Detailed Placed gates, global+track route, scan wires

Detailed Routing

Routed design

Physical Opto 3

Mask pattern (GDS2)

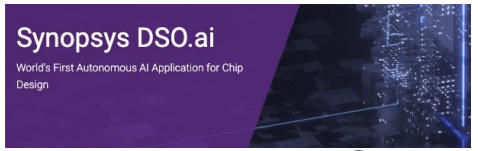
ML inside

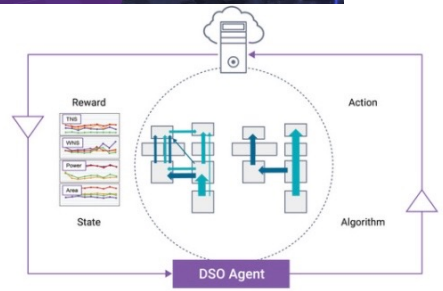
ML inside

ML inside

ML inside

ML outside





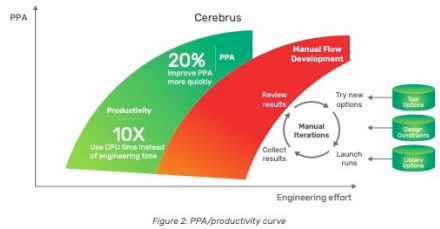
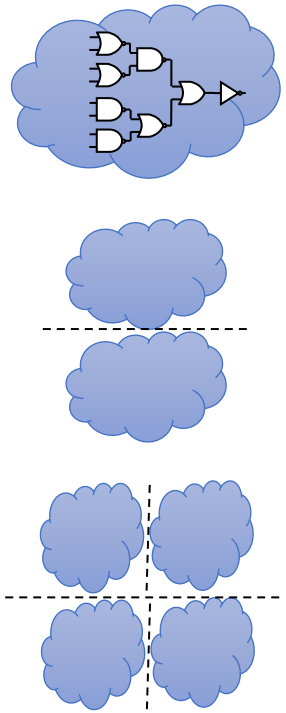


Figure 2: PPA/productivity curve

Placement Algorithms: overview

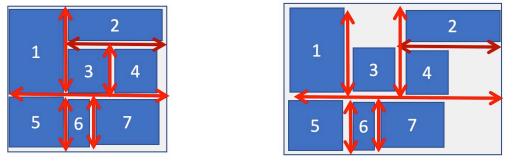
1970ies

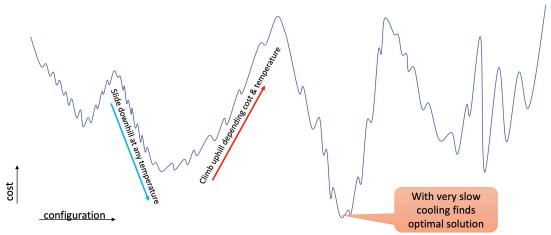
Partitioning-based
e.g., Recursive Min-Cut

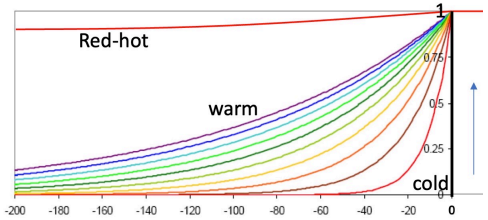


1980ies

Stochastic
Simulated Annealing







1990ies - today

Analytical

Minimize total quadratic wire length:

$$Q = \sum_{i=0}^{n} \sum_{j=i}^{n} w_{i,j} * \left((x_i - x_j)^2 + (y_i - y_j)^2 \right)$$

= Solving 2 (huge) matrices:

$$\begin{bmatrix} 3 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 100 \end{bmatrix}$$

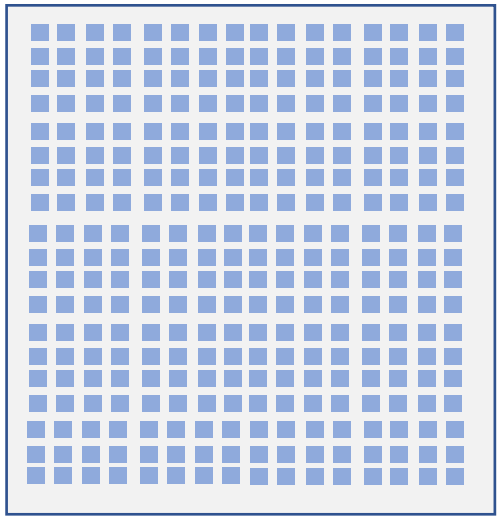
$$\begin{bmatrix} 3 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 3 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 50 \end{bmatrix}$$

2020ies onward?

Machine-**L**earning Driven

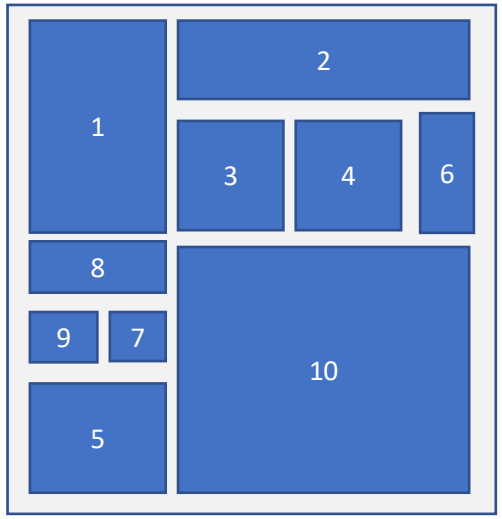
New!

Different Placement Problems



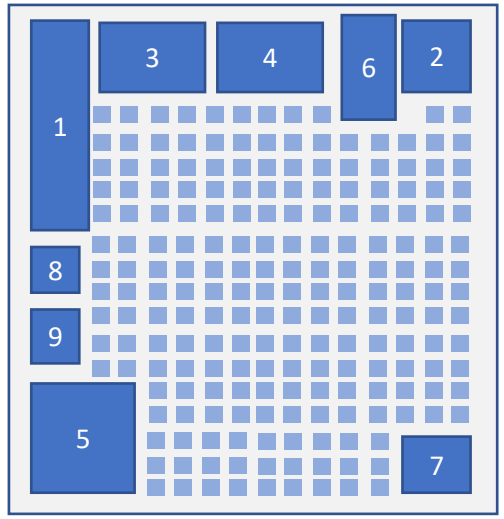
- **Cell Placement**
  - Millions of cells
  - (near-) uniform cell size
- FPGA and ASIC

Solved problem



- **Macro Placement**
  - 5-100 macros
  - Wildly different size and shape
  - Different orientation and flip

Not solved: inconsistent results
Manual expert often wins

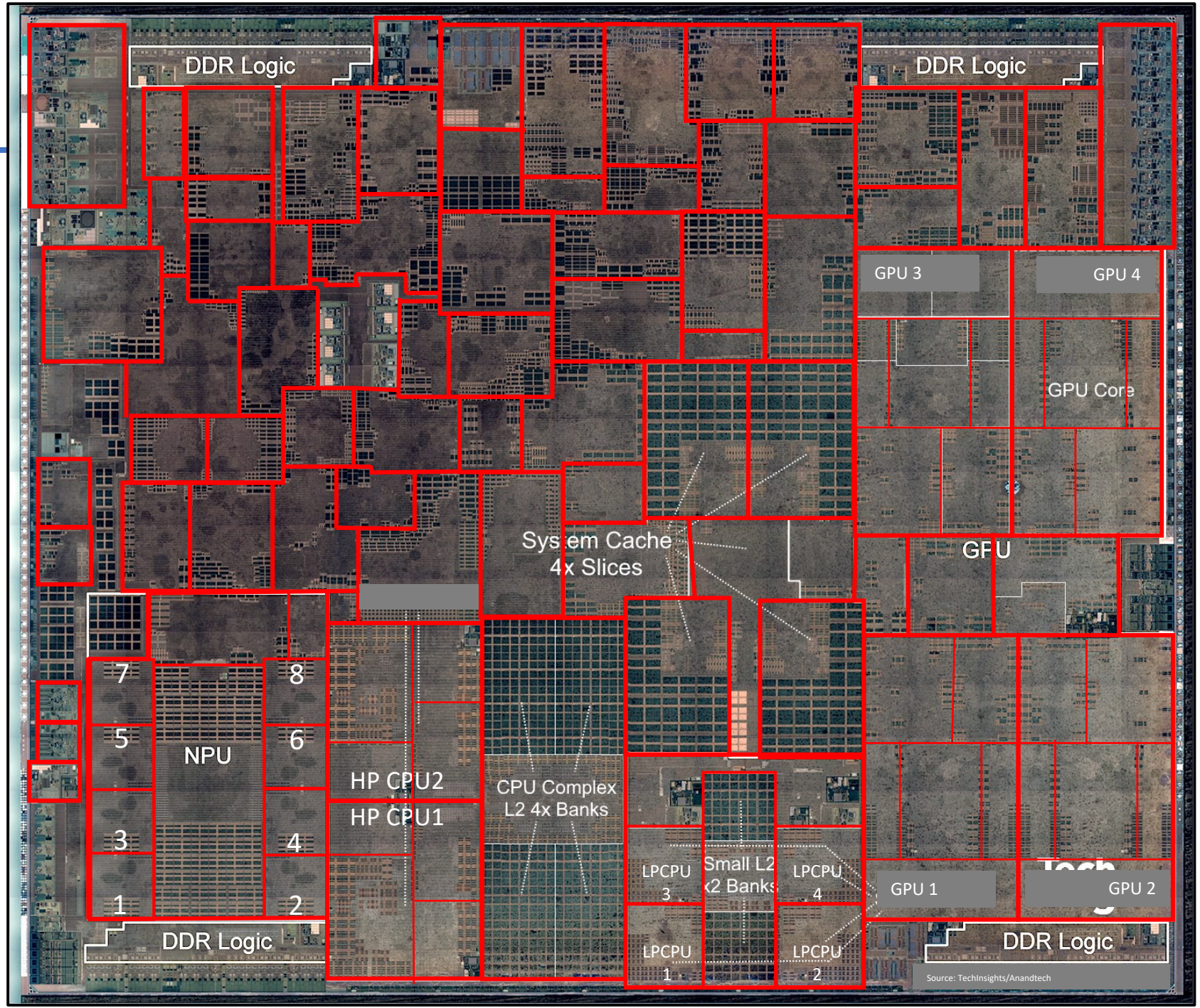


- **Block Placement**
- Combination of macro and cell placement.
- Essential for all ASIC placement

Not solved: inconsistent results
Manual expert still wins

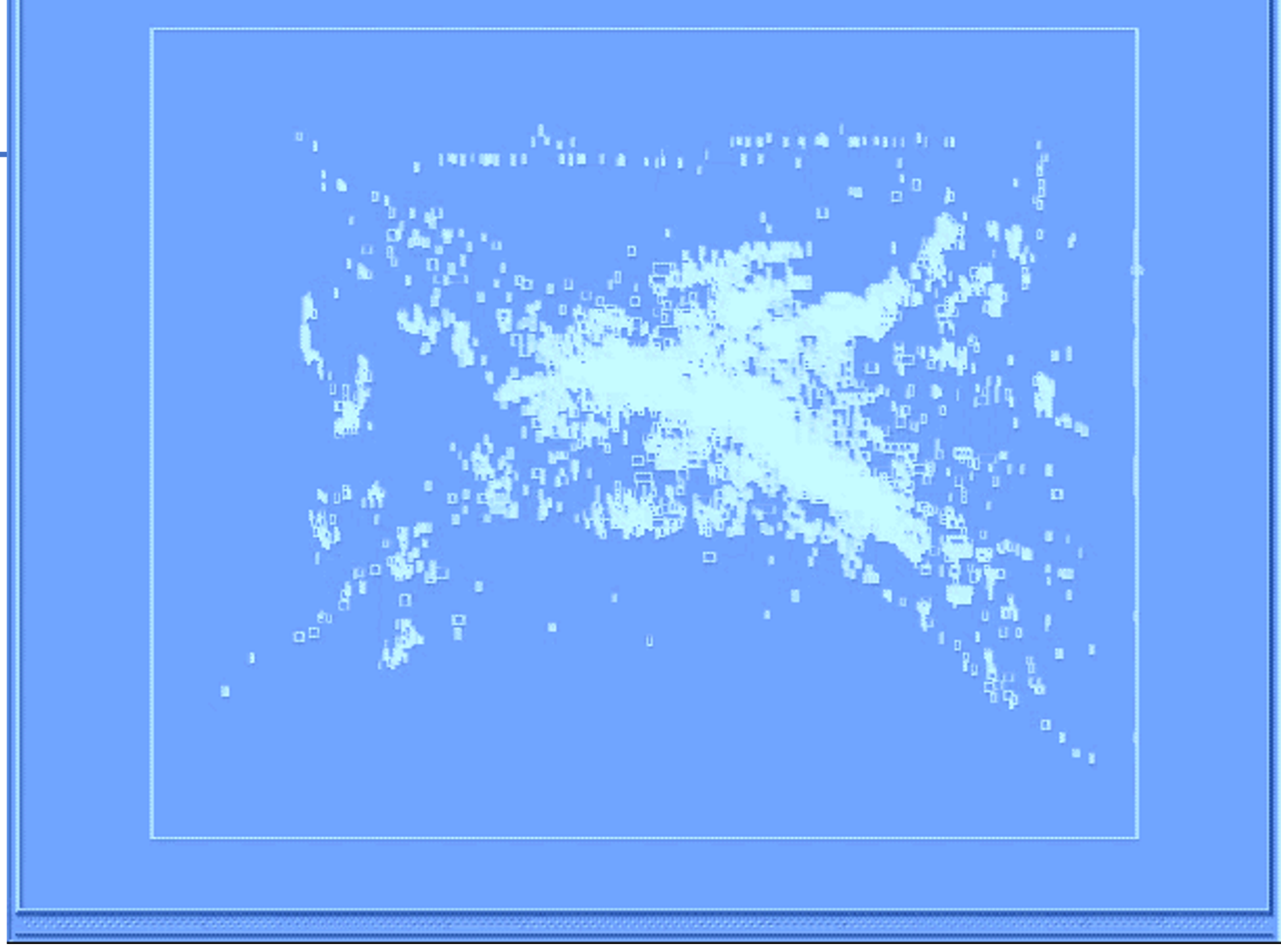
Mobile SoC

- Apple A12
- A macro placement of ~100 Blocks
- Each block: ~20% hard macros
- ~80% standard cells
- Sometimes blocks are carbon copies
  - ~75 unique blocks



Place cells:

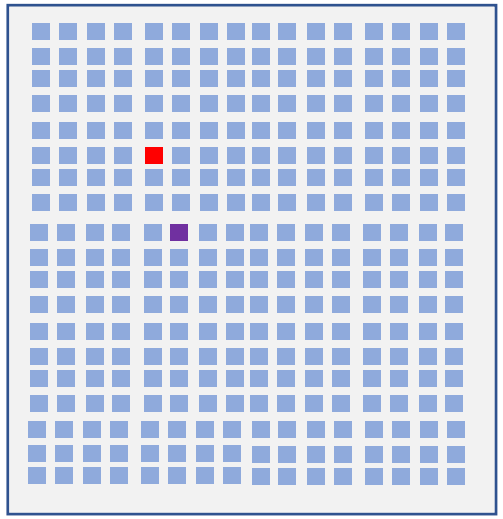
1. No overlap
2. Minimize total **wire length**
3. Routable: avoid **congestion**

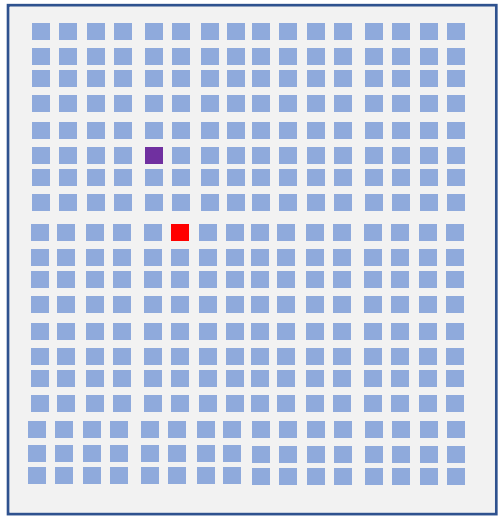


small ARM processor
20,000 cells

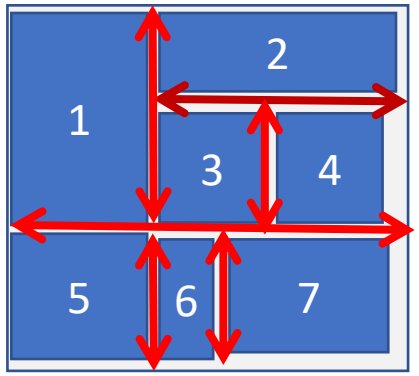
Cell Placement is Smooth, Macro is Rough

- Cell placement change:
  - Easy because cell size are uniform
  - Small change = proportionally small effect

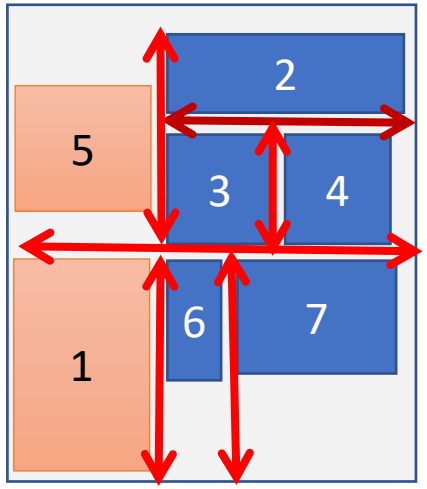




- Macro placement change:
  - Results might fit awkwardly due to non-uniform sizes
  - Most change have a big effect
    - No smooth gradient

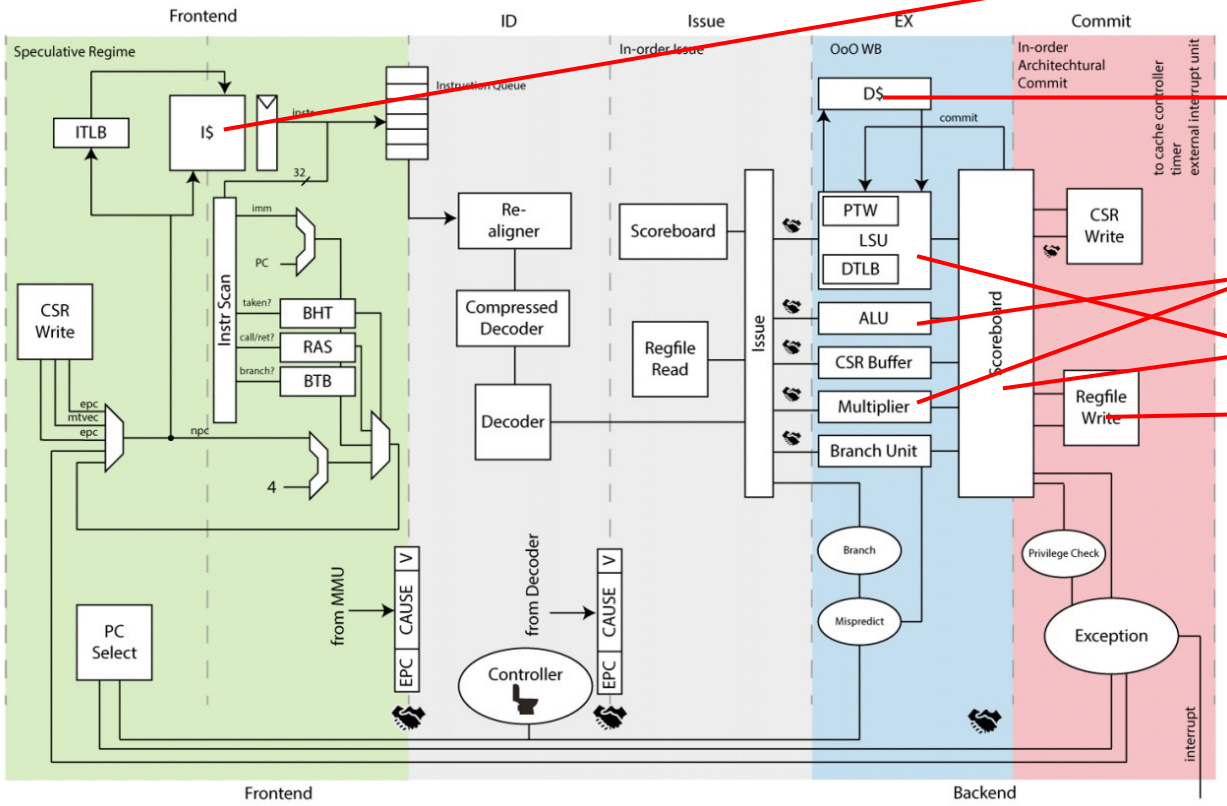


67V5V134V2HVH

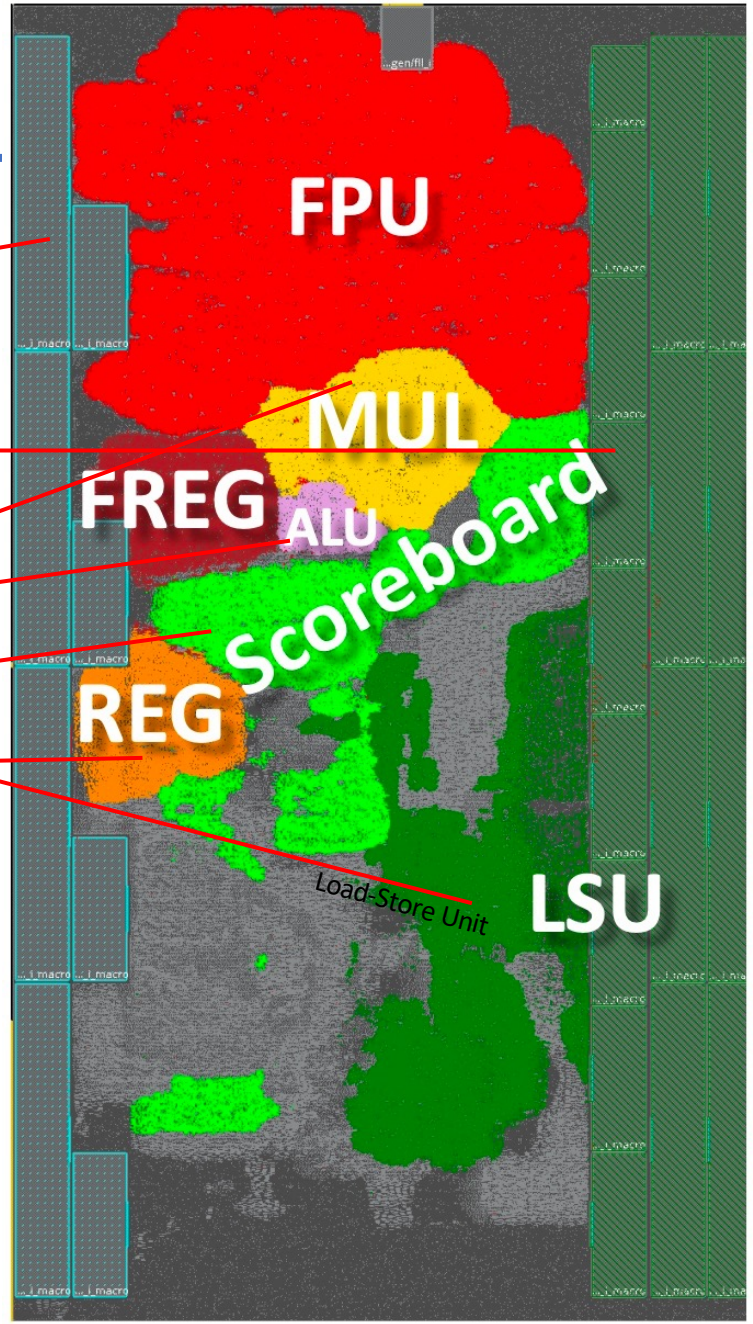


67V1V534V2HVH

Ariane RISC-V Processor: EDA Dislikes hard marcos



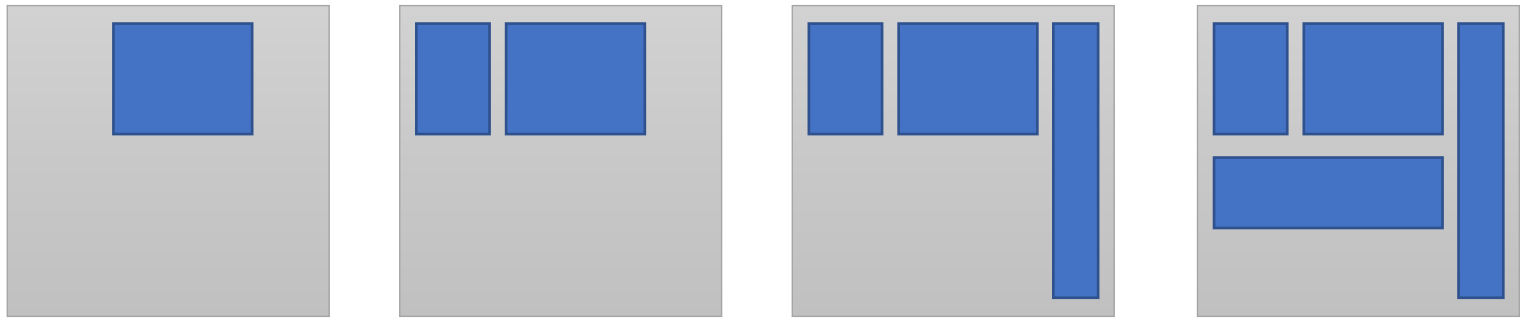
Block Diagram



ASIC Layout

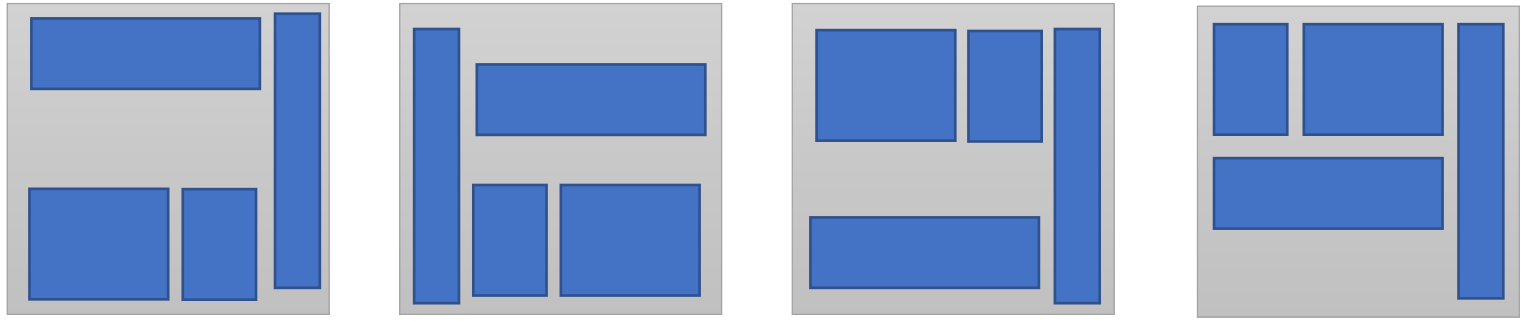
Source: Florian Zaruba, ETH Zürich
https://riscv.org/wp-content/uploads/2018/05/14.15-14.40-FlorianZaruba_riscv_workshop-1.pdf

Basic Macro/Block Placement Methodologies



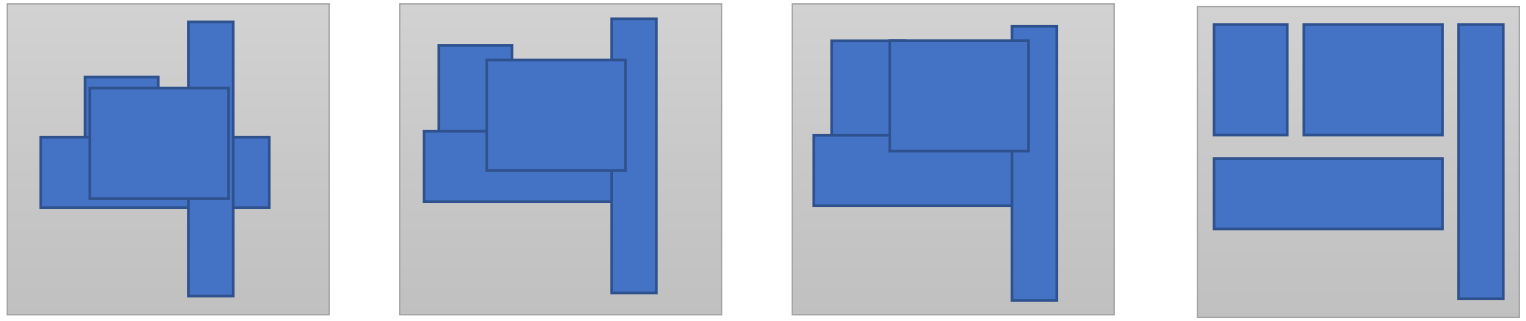
Sequential

Blind



Iterative restricted to legal intermediate states

Bumpy

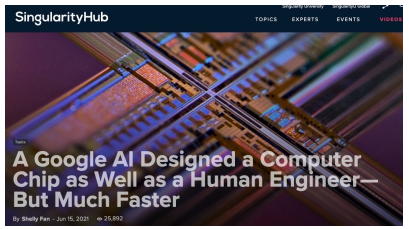


Iterative with Illegal intermediate states

Barely legal

“A Graph Placement Methodology for Fast Chip Design”

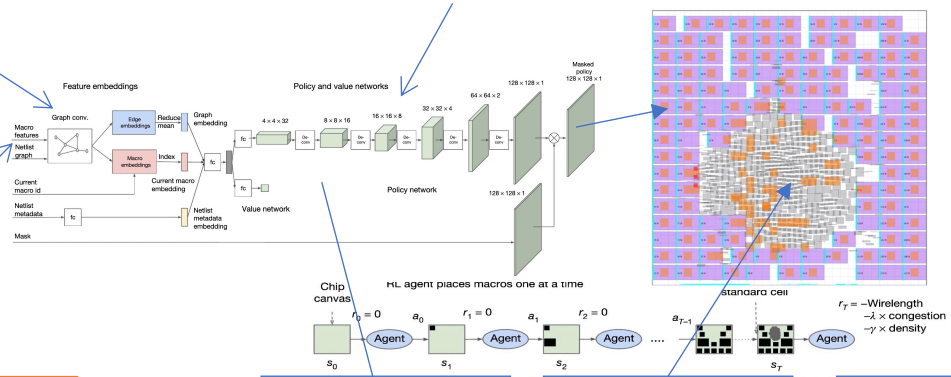
Mirhoseini et. al. Nature, 9 June 2021



Pre-trained on 10K Labeled placements

Pre-trained on 20 similar TPU blocks

Netlist connectivity global chip properties



Simulated Annealing Post-processor

Block Netlist of n Macros and standard cells

Cluster standard cells

RL Agent Places Macros one-by-one

Force-directed placement of clustered standard cells

Calculate reward = Wirelength + Congestion + Density

Mirror and Rotate

Un-cluster standard cells

Run Cadence or Synopsys Industrial Place and Route tool

Actual Timing, wire length and congestion

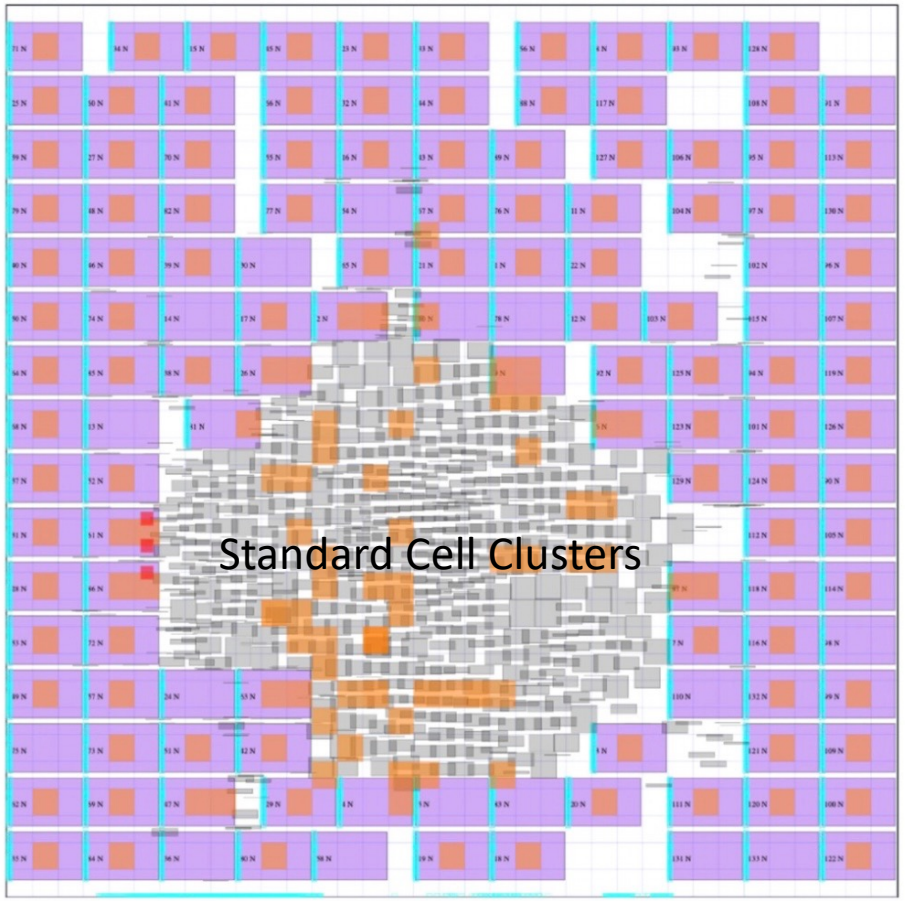
~1 second/iteration

6 hours of refinement(= ~25K iterations?)
= 2000 hours CPU time (with Volta GPU)

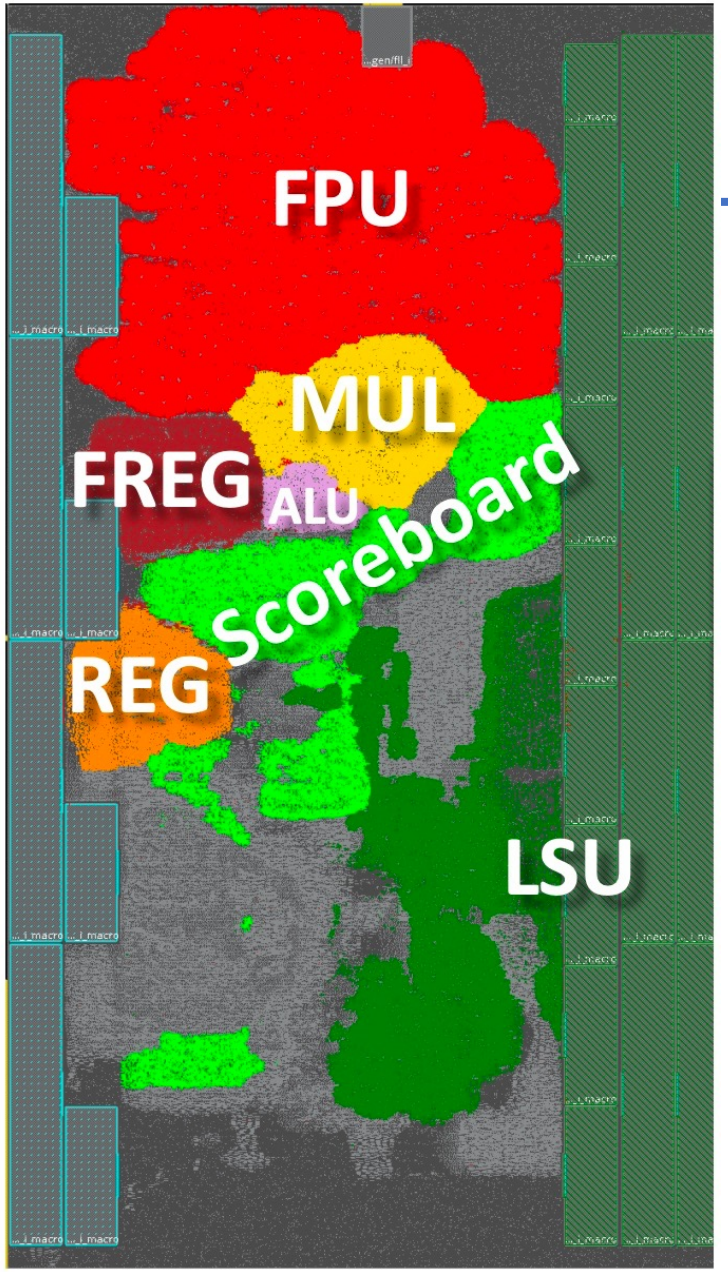
4-8 hours

Manual Macro Placement
(claim: 6 weeks ?!?)

Ariane RISC-V Processor Benchmark

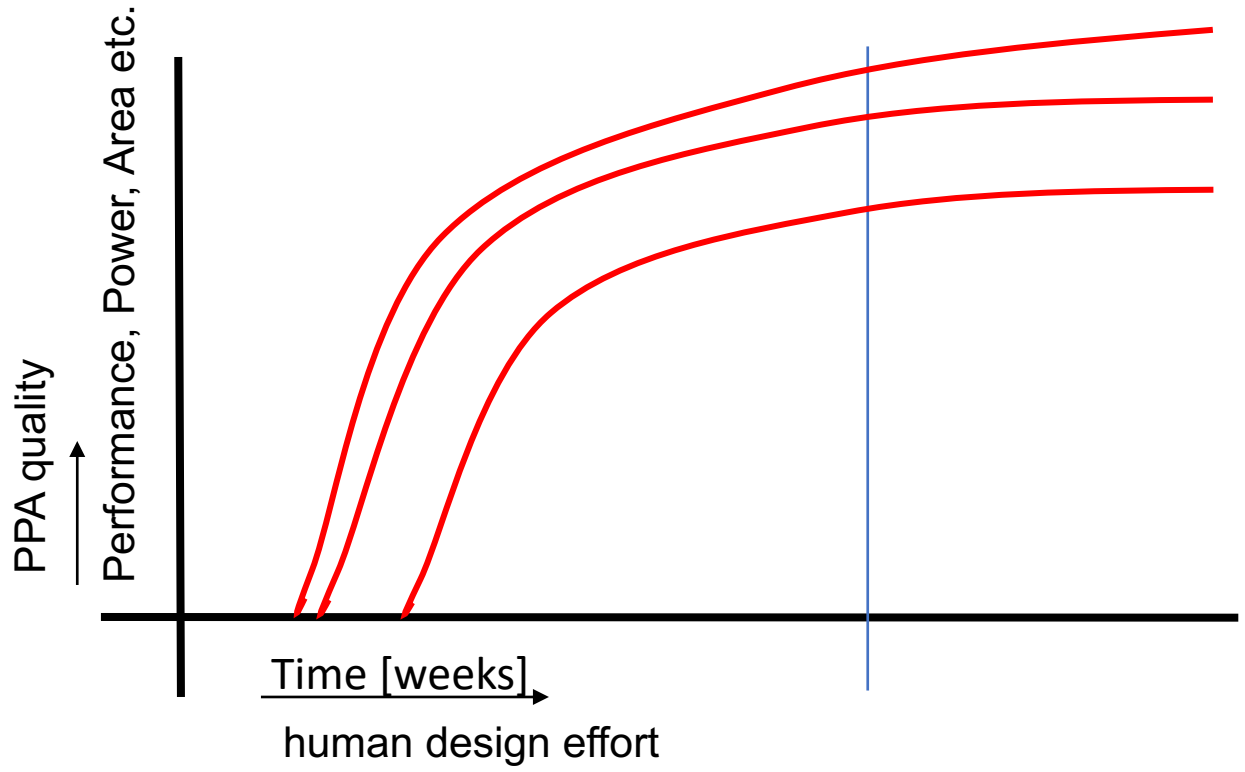


Ariane from:
“A Graph Placement Methodology for Fast Chip Design”

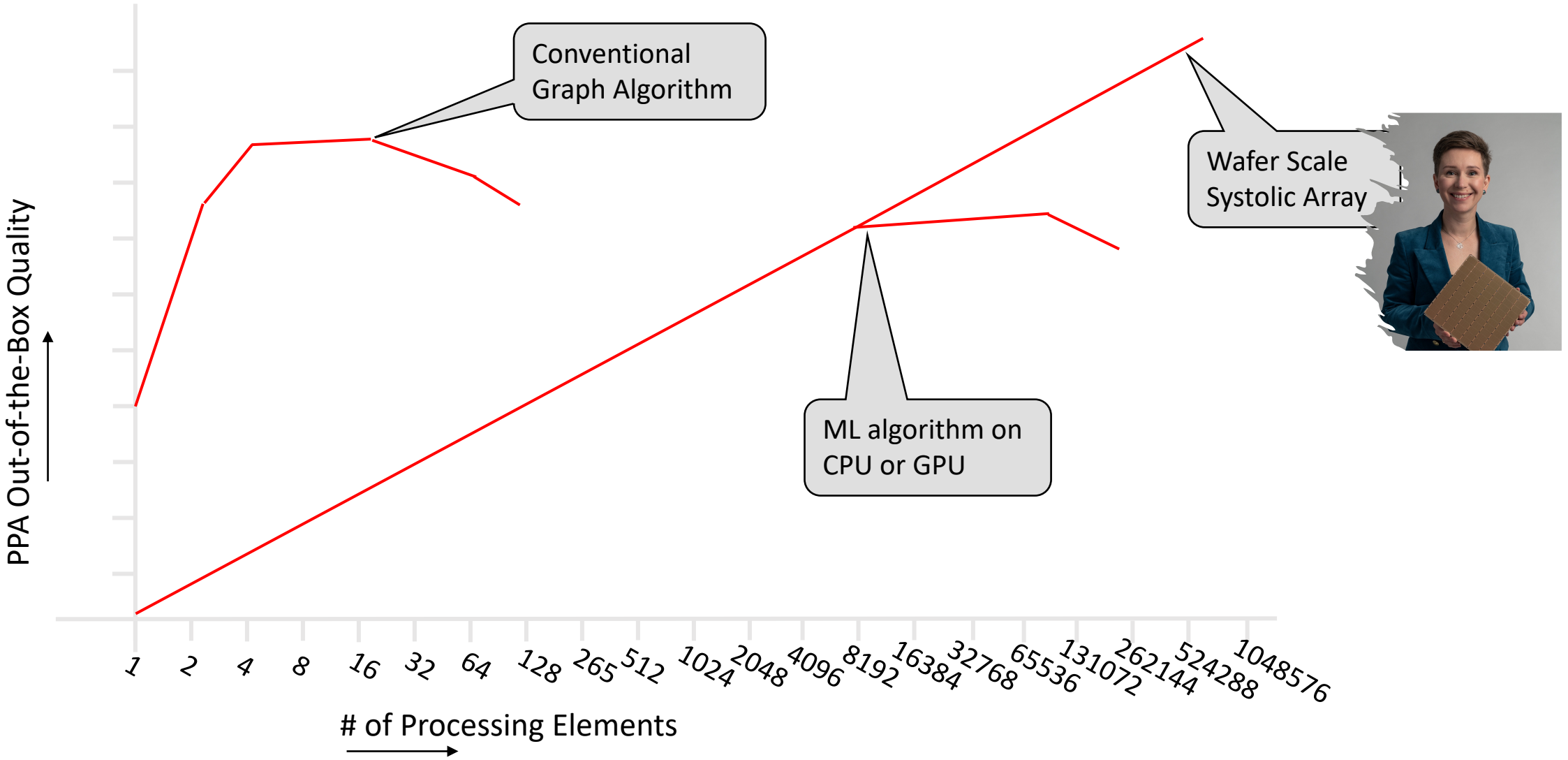


Ariane as shown in:
https://riscv.org/wp-content/uploads/2018/05/14.15-14.40-FlorianZaruba_riscv_workshop-1.pdf

AI outside of the loop: Tweaking Parameters



ML Inside: Quality vs CPU



Conclusion: Machine Leaning in EDA

- A new wave of ML-powered EDA tools is coming
- ML applications likely hit a few sweet spots:
  - Finding patterns
  - Complex interactions across tools, tuning algorithmic parameters
  - Mechanism for brute force trial-and-error
  - New ideas!
- Challenges:
  - Finding the proper embedding (modeling) of the circuit
  - Needs expensive hardware
  - Beating ‘conventional’ algorithms at their own game

Thank You!