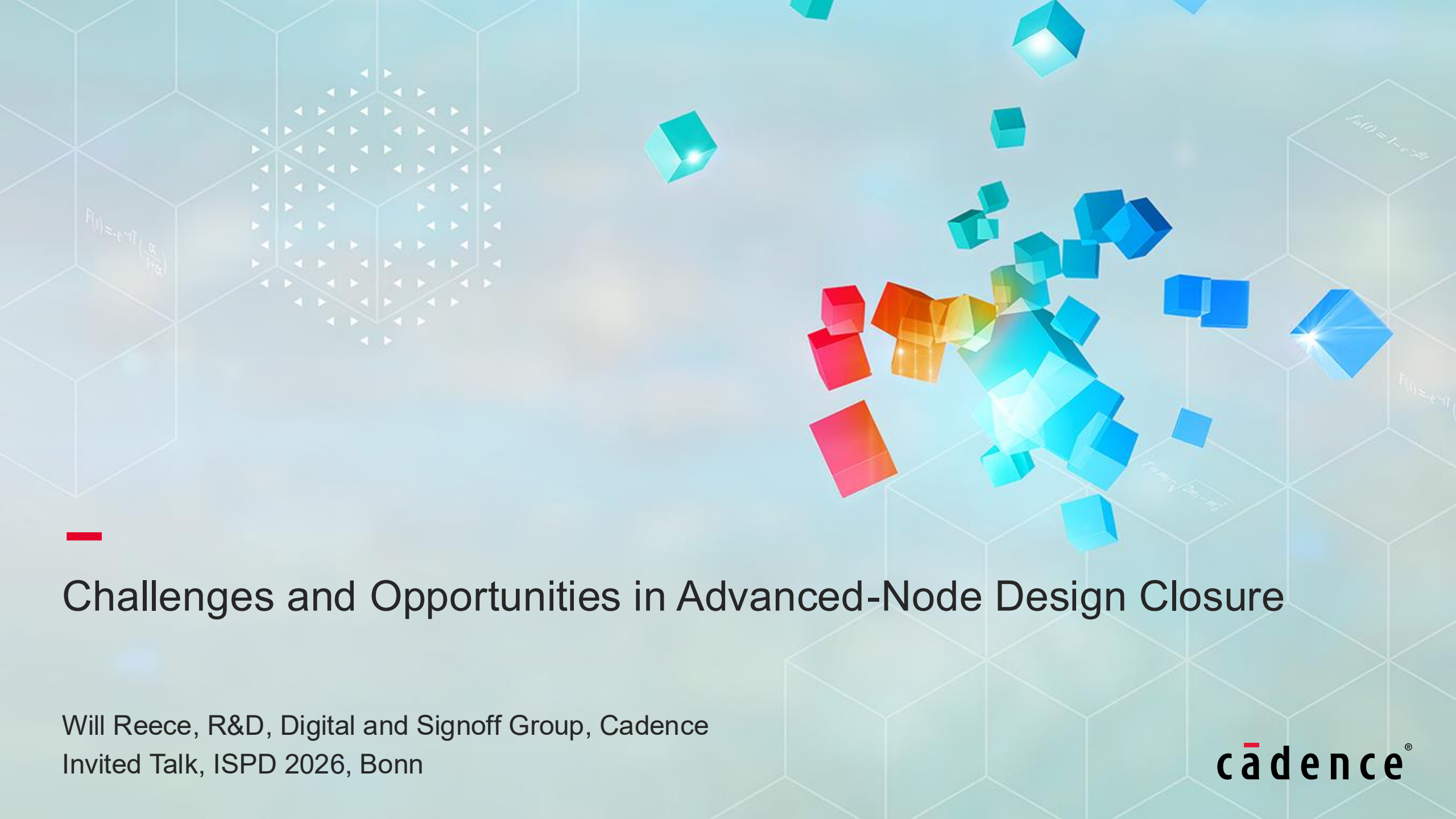


Challenges and Opportunities in Advanced-Node Design Closure

Will Reece, R&D, Digital and Signoff Group, Cadence

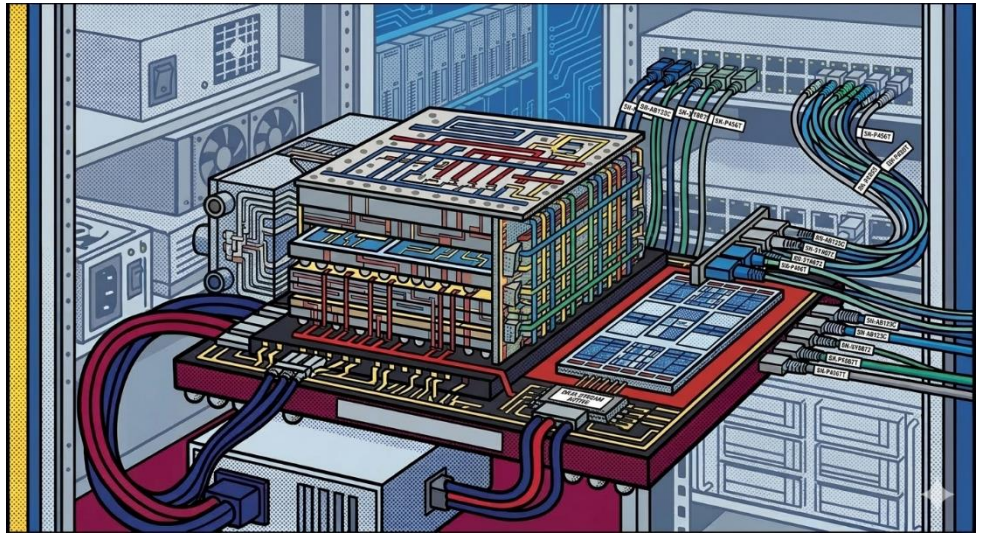
Invited Talk, ISPD 2026, Bonn

cadence®

Introduction

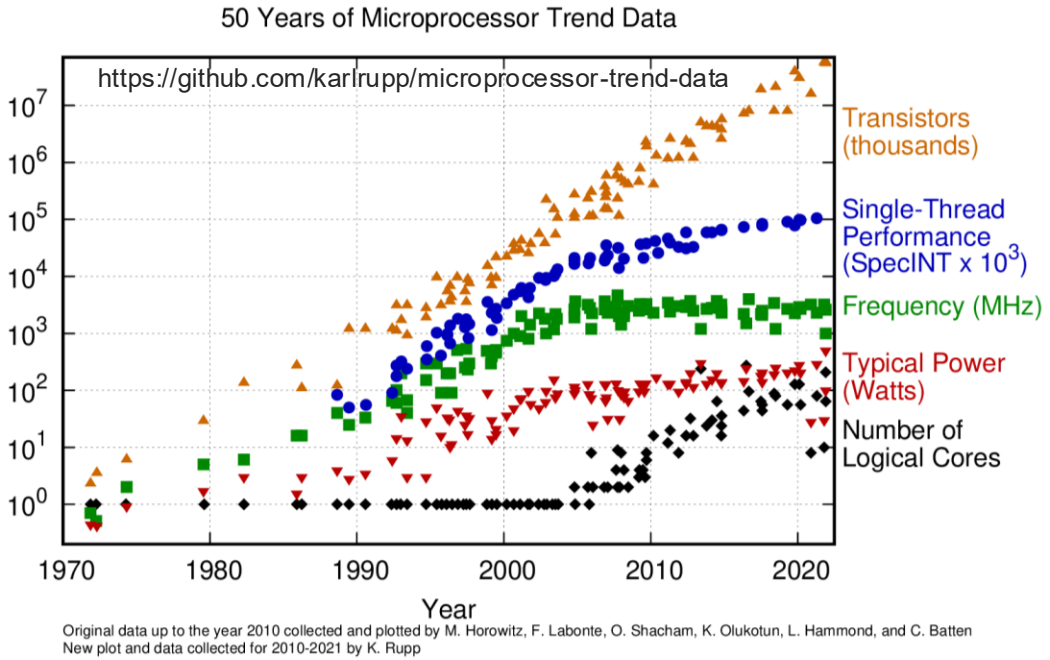
- EDA tools have revolutionized VLSI design
- Stunning variety of custom SoCs created each year
  - Consumer, mobile, GPU, AI acceleration, automotive
  - Often targeting advanced nodes
- The key challenge is managing ever-increasing complexity while closing timing
  - Increasing design scale
  - 3D-IC and Heterogeneous
  - Backside resources
  - Managing turnaround time
- Deploying AI can help, but further innovation is needed

Increasing Design Scale



- Massive increase in the size of designs
  - Hundreds of CPU, GPU, or AI cores → Huge instance counts
  - Designs often physically large, too
- Advanced nodes require expensive new checks
- Runtime, memory, algorithmic complexity
  - Require new strategies to manage
- 3D-IC allows for even larger designs with new constraints
  - Heterogeneous dies with different technologies and libraries are possible

Increasing Design Scale



- Massive increase in the size of designs
  - Hundreds of CPU, GPU, or AI cores → Huge instance counts
  - Designs often physically large, too
- Advanced nodes require expensive new checks
- Runtime, memory, algorithmic complexity
  - Require new strategies to manage
- 3D-IC allows for even larger designs with new constraints
  - Heterogeneous dies with different technologies and libraries are possible

Digital Implementation and Signoff – Angstrom Design

Complexity and Scale

16nm
- 1st g FinF
- Colo
- Dou patt
- On c varia
- Pron IR/E

12nm
- Improved pin access, rule compliance
- EM fixing
- Via alignment/insertion
- Boundary cell handling
- Routing halos

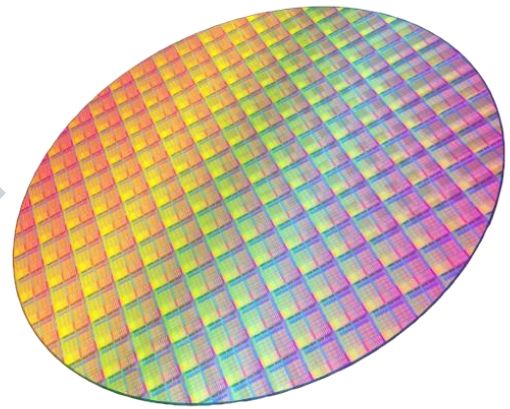
10nm
- Pin access placement
- Power integrity issues
- Wire resistance mitigation
- OCV handling
- IR/EM signoff

7nm/6nm
- Next-generation FinFETs
- Trim metal requirements
- Early power estimation
- In-design fixing for IR/EM
- No. of physical signoff rules
- Power convergence

5nm/4nm
- EUV layer support
- Via Pillar aware synthesis
- Via Pillar advanced support
- Metal handling
- In-design fixing for IREM
- Effective layer aware Optimization
- Long Range DRC checks

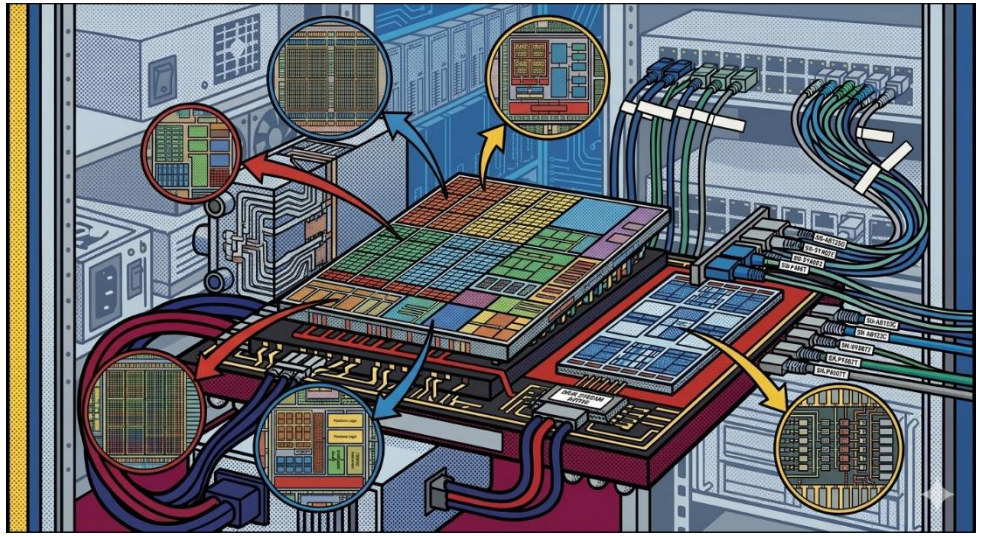
3nm/2nm
- **Complex FP rules**
- **Placement and DRC requirements**
- **MH cell, Color shifting rules**
- **EUV layer, Via Enc/Shift Rules**
- **LV SOCV Accuracy**
- **High Sigma Variation on cells**
- **Interleaving DPT layer stacks**

14 Angstrom
- **Foundry Rules Support**
- **Early Detailed Routing**
- **Back-Side Implementation Flow**
- **Thermal Optimization**



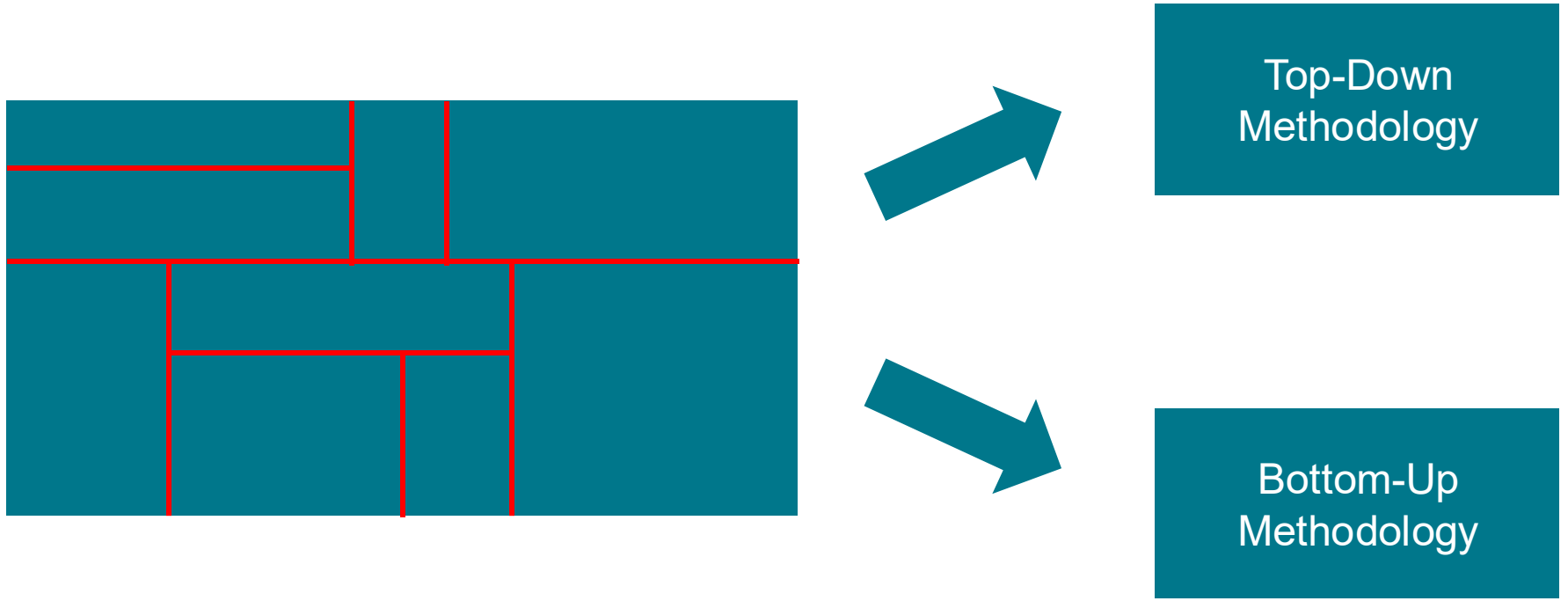
Success requires close collaboration between EDA, IP, and foundry

Increasing Design Scale



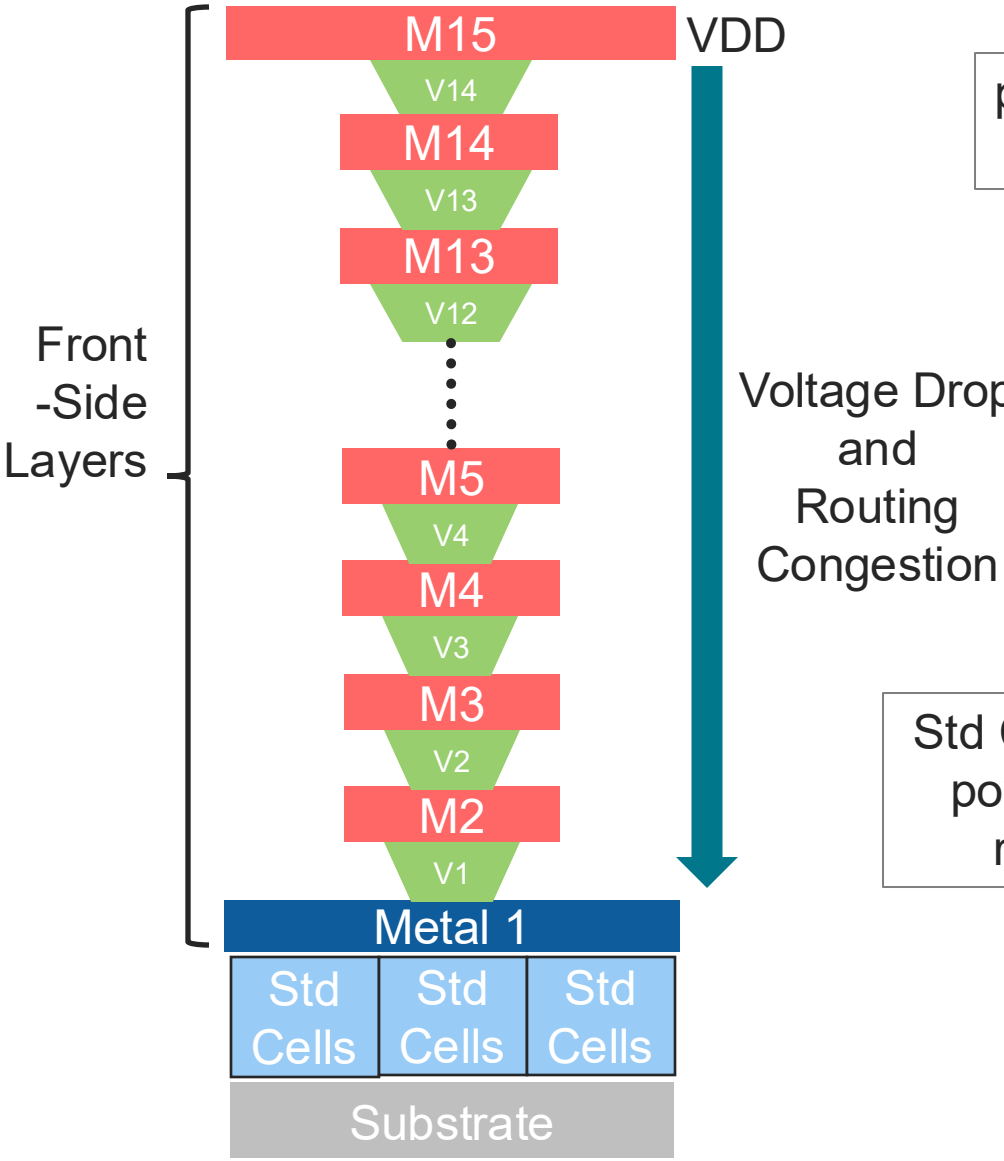
- The traditional solution is to partition the SoC into blocks by function
  - Work on these in parallel
  - Opportunities for component reuse → Significant runtime reductions
  - Combine hierarchically
- Challenges:
  - Maintaining engineering productivity when managing many blocks
  - Planning and budgeting – correlation matters
  - Timing closure at boundaries
  - Top-level complexity

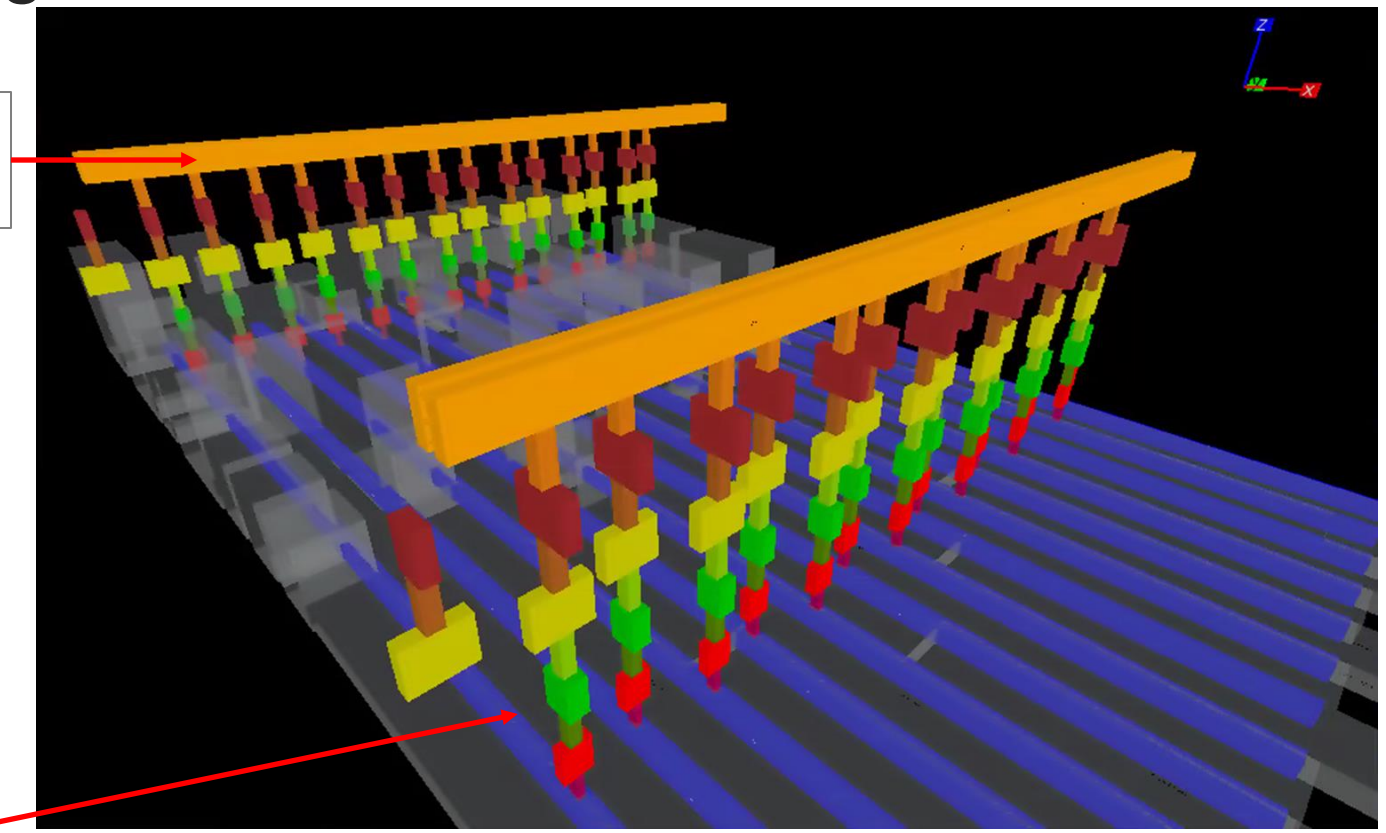
Hierarchical Planning



- Non-trivial budgeting of resources and timing
- Automation required to handle increasing design scales
  - Opportunities for AI-driven improvements

Front-Side Routing Challenges

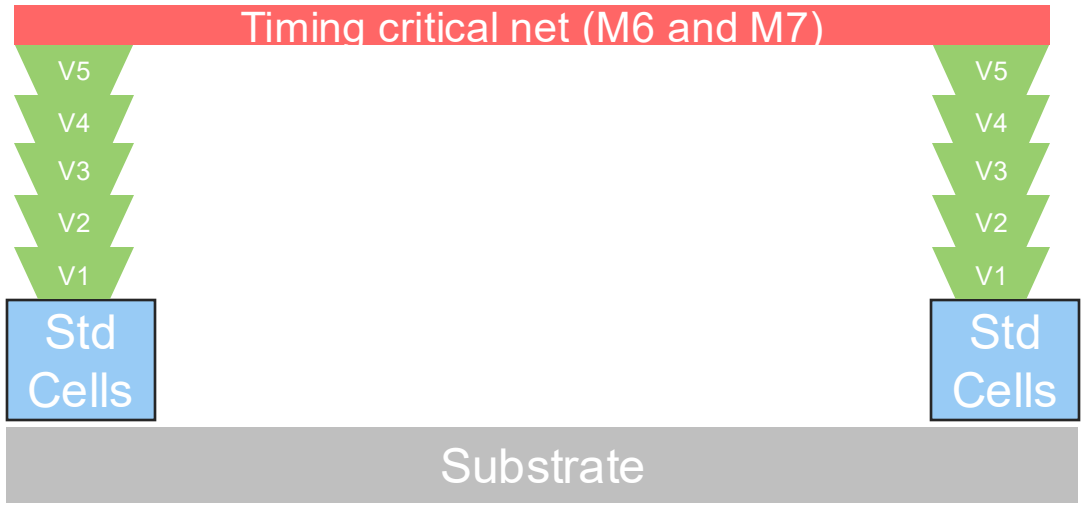




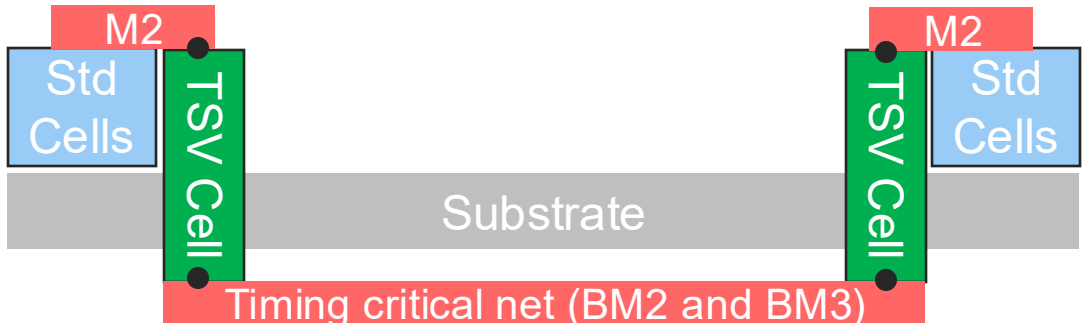
- Dense power via pillars across the floorplan can trigger signal routing congestion
- May have to increase floorplan area to manage congestion

Back-Side Signal Routing

Front-Side Signal Routing



Back-Side Signal Routing

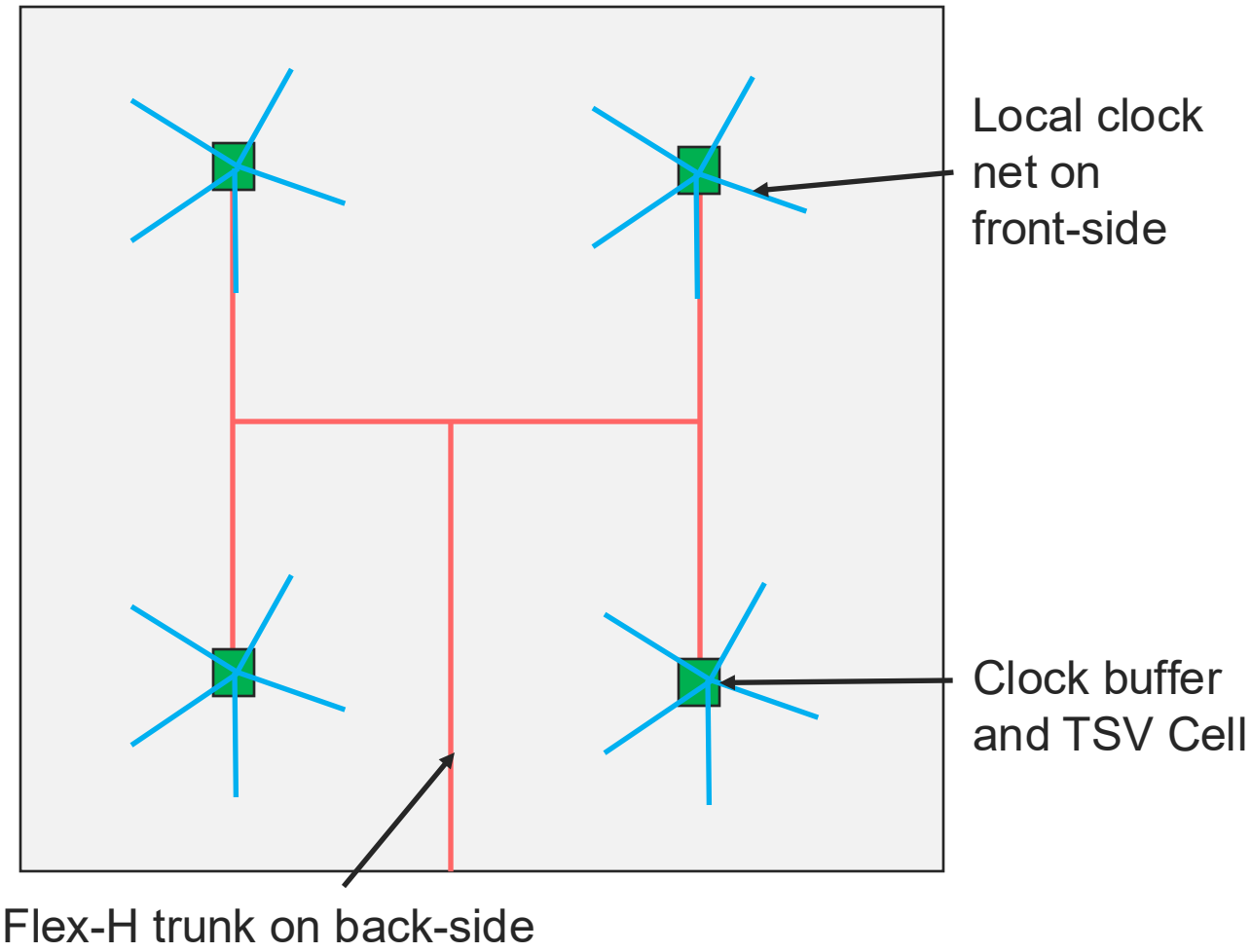


- For long timing-critical nets, upper routing layers are used
- Via pillars go from standard cell pin to upper layers
- Can cause routing congestion in local area

- Innovus™ GigaOpt™ assigning long timing critical nets to back-side layers
  - Net parasitics are reduced, improving timing
  - No via pillars on the front side, reducing local congestion

TSV Cell = Thru-Substrate Cell

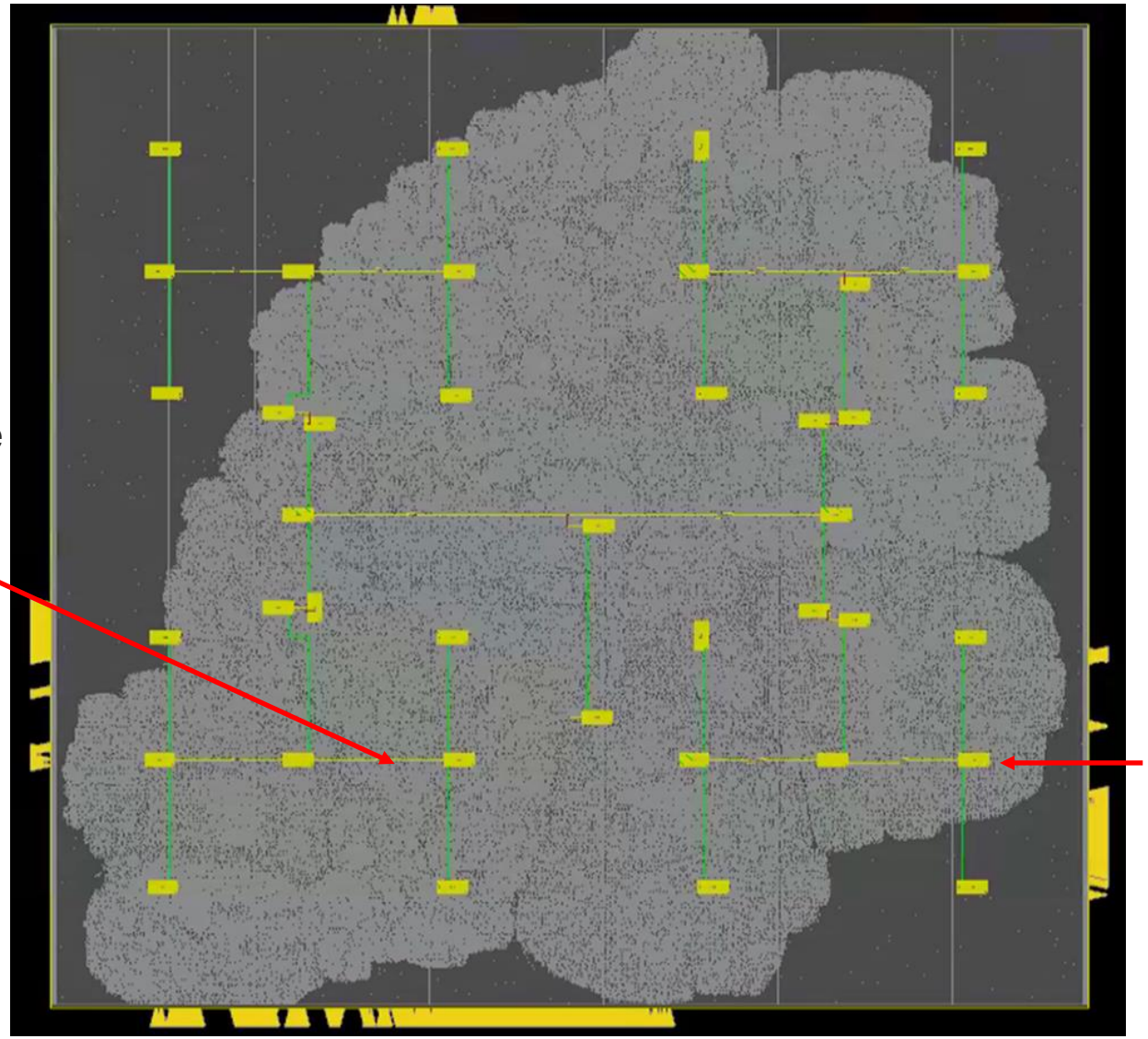
Back-Side Flexible H Clock Tree



- Flex-H trunk routing generally on upper front-side layers
  - Requires via pillars
  - Reduced signal routing resources
- Innovus™ Implementation can route Flex-H trunk on back-side layers
  - Reduced front-side routing congestion
  - Improved timing of clock tree

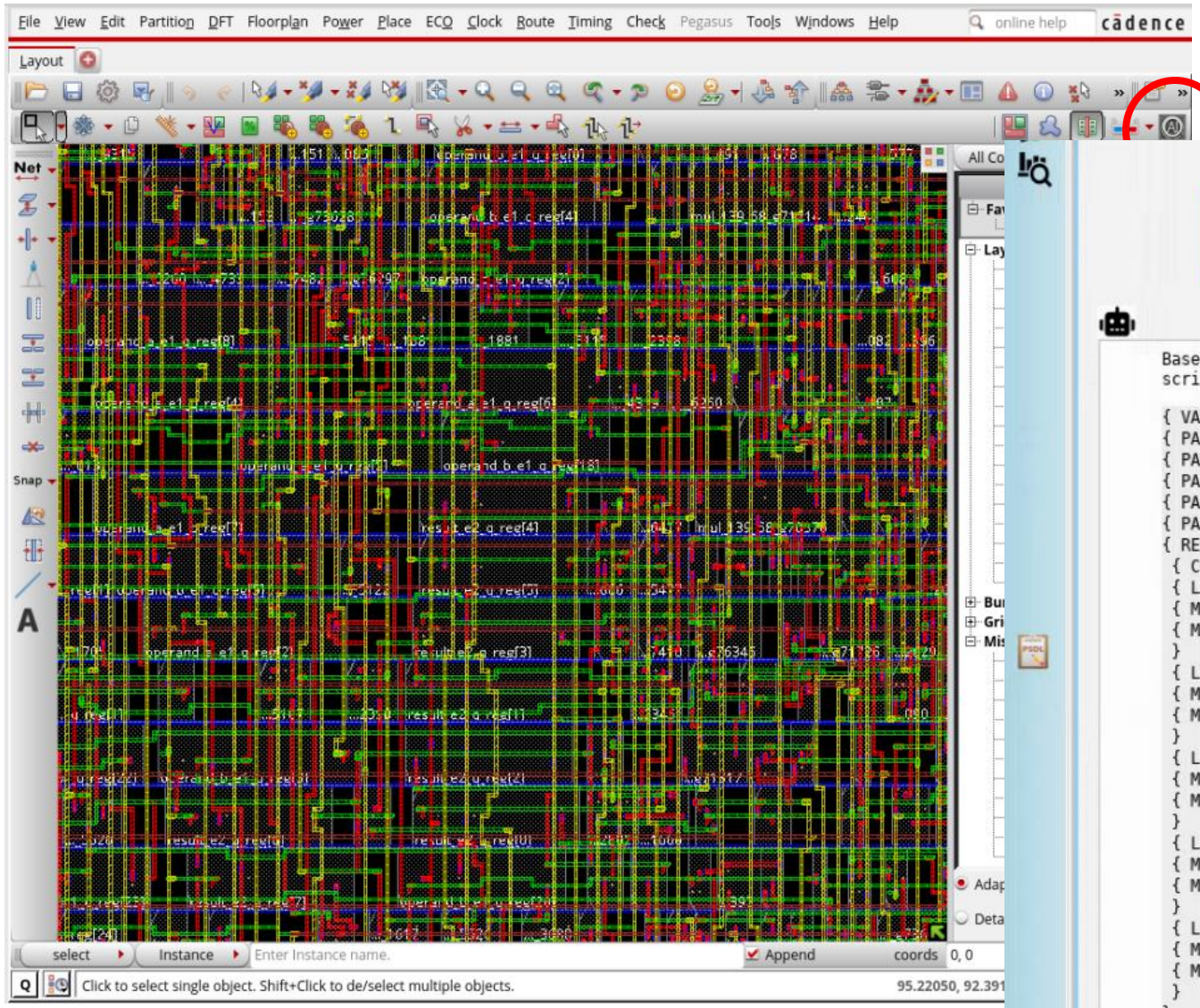
TSV Cell = Thru-Substrate Cell

Back-Side Flex-H Clock Tree – Example



TSV Cell = Thru-Substrate Cell

AI for Designer Productivity



Help me generate a PSDL script for my design. Design info: It has 5 layers, layer names are Metal1,Metal2,Metal3,Metal4,Metal5. Default width is 1,2,3,4,5. Metal1 PG is followpin, Metal2,Metal3,Metal4 PG is staple, Metal5 is stripe. Metal2,4 is vertical , Metal3,5 is horizontal. Please generate PG in core area and the variables can be defined by yourself.

Based on the provided design information, I will generate a PSDL script for your design. Here is the script:

```
{ VARIABLE { m1_width 1 } { m2_width 2 } { m3_width 3 } { m4_width 4 } { m5_width 5 } }
{ PATTERN m1_followpin { TYPE FOLLOWPIN } }
{ PATTERN m2_staple { TYPE STAPLE } { DIRECTION VERTICAL } { WIDTH (m2_width) } { LENGTH 1.0 1.0 } }
{ PATTERN m3_staple { TYPE STAPLE } { DIRECTION HORIZONTAL } { WIDTH (m3_width) } { LENGTH 1.0 1.0 } }
{ PATTERN m4_staple { TYPE STAPLE } { DIRECTION VERTICAL } { WIDTH (m4_width) } { LENGTH 1.0 1.0 } }
{ PATTERN m5_stripe { TYPE STRIPE } { DIRECTION HORIZONTAL } { WIDTH (m5_width) } }
{ REGION
 { COREAREA }
 { LAYER Metal1
 { METAL m1_fpin m1_followpin { NET VDD } }
 { METAL m1_fpin_vss m1_followpin { NET VSS } }
 }
 { LAYER Metal2
 { METAL m2_vdd m2_staple { NET VDD } { OFFSET 0.0 0.0 } { STEPDISTANCE 1.0 1.0 } }
 { METAL m2_vss m2_staple { NET VSS } { OFFSET 0.0 0.0 } { STEPDISTANCE 1.0 1.0 } }
 }
 { LAYER Metal3
 { METAL m3_vdd m3_staple { NET VDD } { OFFSET 0.0 0.0 } { STEPDISTANCE 1.0 1.0 } }
 { METAL m3_vss m3_staple { NET VSS } { OFFSET 0.0 0.0 } { STEPDISTANCE 1.0 1.0 } }
 }
 { LAYER Metal4
 { METAL m4_vdd m4_staple { NET VDD } { OFFSET 0.0 0.0 } { STEPDISTANCE 1.0 1.0 } }
 { METAL m4_vss m4_staple { NET VSS } { OFFSET 0.0 0.0 } { STEPDISTANCE 1.0 1.0 } }
 }
 { LAYER Metal5
 { METAL m5_vdd m5_stripe { NET VDD } { OFFSET 0.0 0.0 } { STEPDISTANCE 1.0 1.0 } }
 { METAL m5_vss m5_stripe { NET VSS } { OFFSET 0.0 0.0 } { STEPDISTANCE 1.0 1.0 } }
 }
}
```

Please note that I've defined the variables for the metal widths, and used them in the PATTERN and METAL statements. I've also assumed that the PG structure should be generated in the core area, and used the COREAREA keyword to define the region.

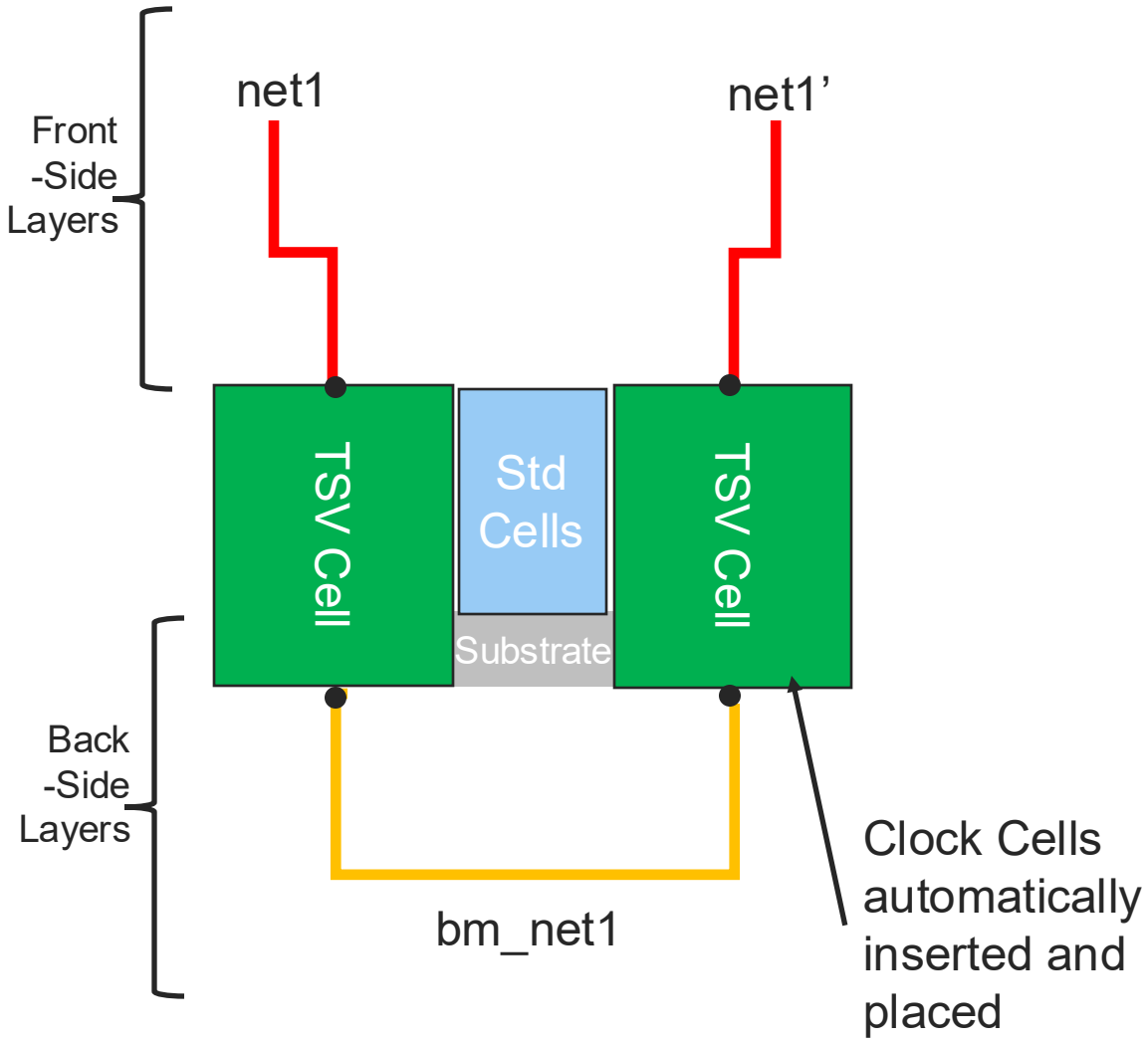
AI for Design Optimization

- ML and AI techniques are widely deployed in chip design
- Reinforcement learning and GNNs are attractive from first principles
  - Representative training data can be hard to find at the required scale
- Zero-shot learning with LLMs interesting new area for research
  - Non-determinism is a challenge

Summary

- EDA tools have enabled chip design at scale
  - A huge impact on the modern world
- SoC design complexity and scale continue to increase
  - Continued tool evolution is required
- AI can help with PPA and designer productivity
  - Progress made, but potential for further gains

Cadence Angstrom Back-Side Implementation Flow



- Cadence supports a complete back-side implementation flow
  - Back-side layers defined in Tech LEF
  - GigaOpt™ assigns nets to back-side to improve timing
  - GigaPlace™ places TSV cells
  - Flex-H CTS routes clock trunk nets on back-side
  - Quantus™ Extraction engine back-side routing aware
  - NanoRoute automatically routes both front- and back-side clock/signal nets
  - Voltus™ IR analysis includes back-side PG
- Custom TSV Cells used to connect front-side metal to back-side metal
  - During design export and timing analysis, front-side and back-side nets merged

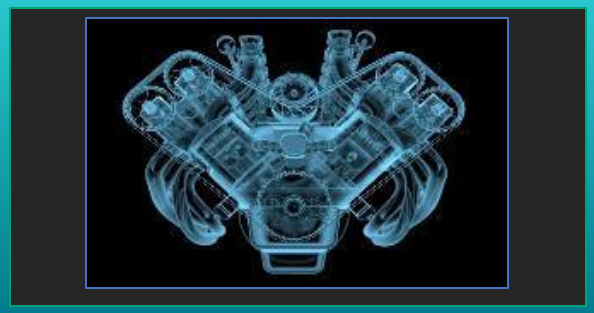
TSV Cell = Thru-Substrate Cell

Cadence Design Excellence and First Silicon Success

Productivity

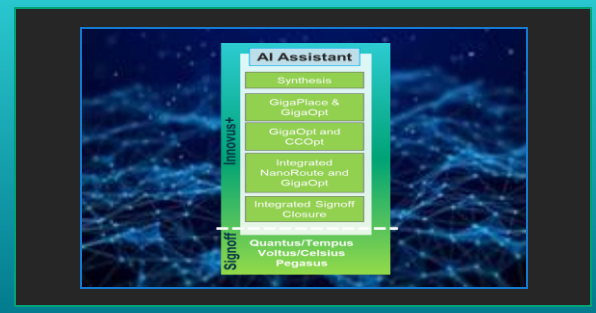
Design	Implementation	Electrical Signoff	Physical Signoff
Stratus™, Modus, Conformal®	Genus™, Innovus+™, Joules™	Quantus™, Tempus™ Voltus™, Celsius™	Pegasus™, DFM

Digital Full Flow
RTL – GDSII



Core Engine Excellence

Massively Parallel



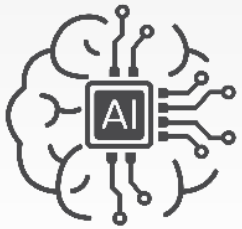
Full Flow Integration

Large-Scale AI-Driven Optimization with Cadence Cerebrus AI Studio

Cadence Cerebrus® AI Studio

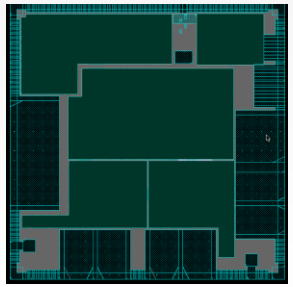
User Assistant Agent

- Doc Query
- Python Script Generation
- Design Debug
- Data Cleanup

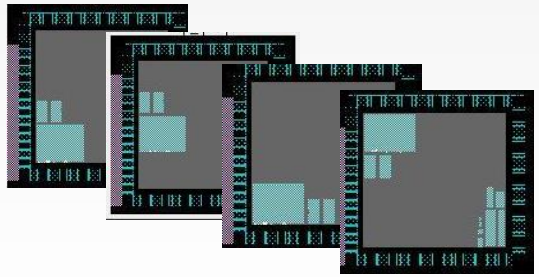


Design Planning Agent

- Partition Shaping and Sizing
- Pin Assignment
- Feedthroughs
- Pipelining

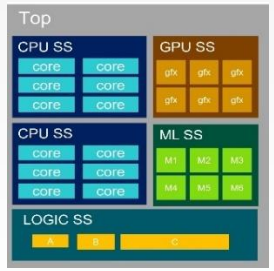


Block Implementation Agent



One engineer, multiple blocks

Design Closure Agent



- Distributed Signoff Optimization
- No Design Assembly
- Full Flat STA Accuracy

Cadence® JedAI Data Platform

AI Assistant

Multi-Design/Multi-Run Studio Manager

Advance Design Analytics

Accumulated Learning

“Faster design time” *“Engineering productivity”* *“Differentiated PPA”*