SYNOPSYS®
Silicon to Software™

Interesting Problems in Physical Synthesis

Pei-Hsin Ho

Synopsys Fellow, Synopsys, Inc.

ISPD, Portland Oregon, 2017

Two Crises in Physical Synthesis

- Interconnect prediction and optimization
  - Resistance
  - Route DRC
- Runtime of physical design tools
  - CPUs
    - Clock frequency: plateaued out
  - Both ICC2 and Innovus have sped up through multi-threading
  - What is next?

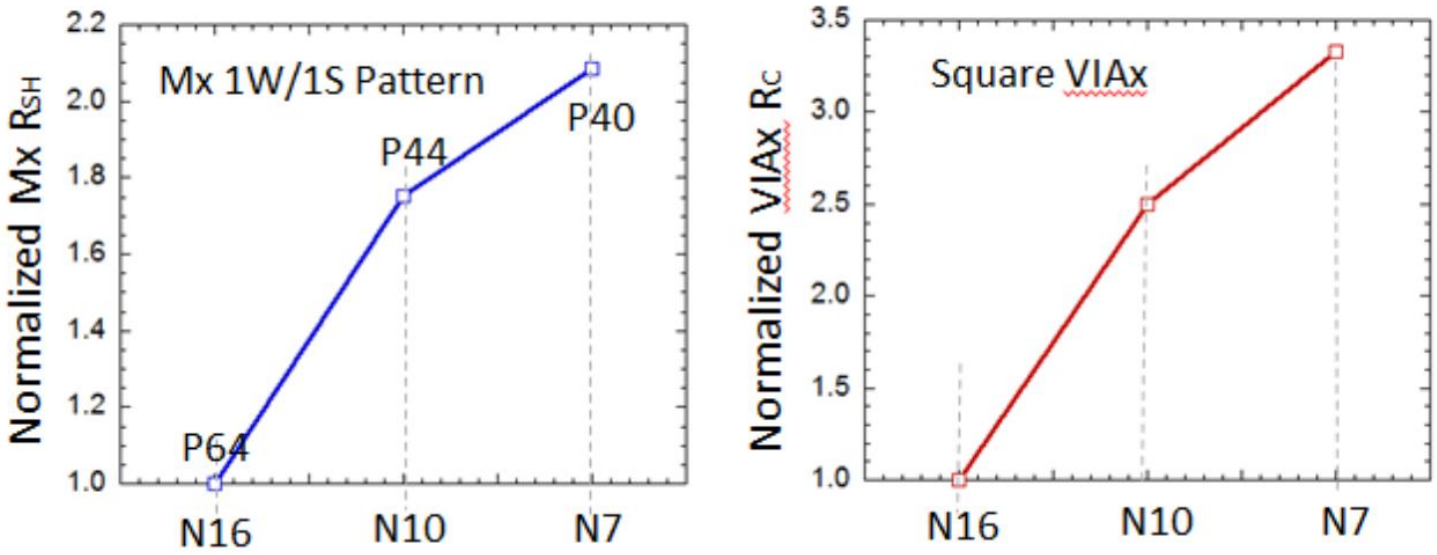
Two Crises in Physical Synthesis

- Interconnect prediction and optimization
  - Resistance
  - Route DRC

- Runtime of physical design tools
  - CPUs
    - Clock frequency: plateaued out
  - Both ICC2 and Innovus have sped up through multi-threading
  - What is next?

Parasitic R under FinFET Technology

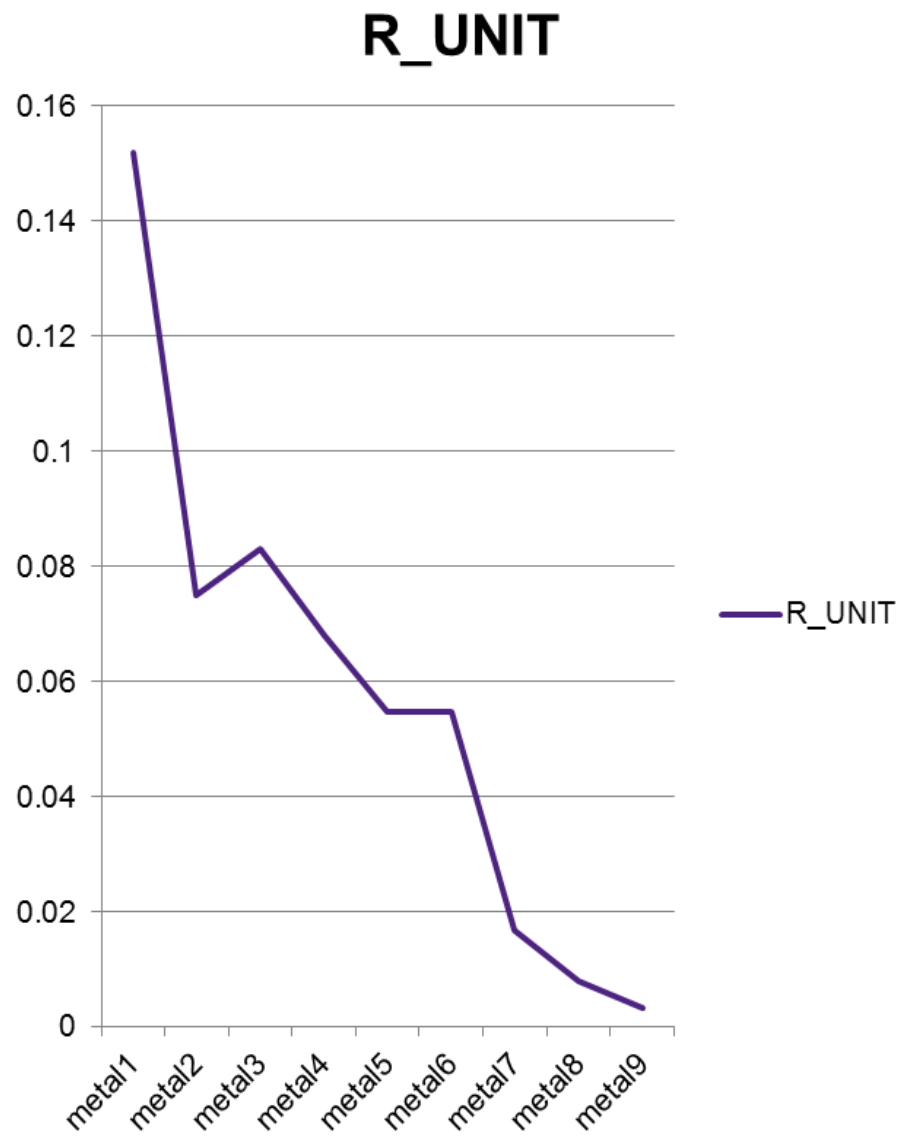
- FinFET
  - Cells: faster than planar
  - Wires and vias: increased resistance
    - Cliff Hou, ISSCC 2017 →



SYNOPSYS®

Large Spread in R across The Metal Stack

- Foundry' idea to speedup ICs with speedy cells and slow wires
  - Make a subset of the metal layers (the higher layers) faster
  - 50X difference in metal layer resistance is common
    - Bicycle vs. bullet train
- Vias also have very different resistances
  - Tradeoff between resistance and size
  - Double via
  - Via pillar



SYNOPSYS®

Mispredict R → Suboptimal Physical Synthesis

- Why is a large variation in R (or a large error in R estimation) a problem?
  - Interconnect must be buffered to maintain linear delay growth
  - Parasitic R determines how much wire a buffer can drive without incurring undue delay
    - But which layers will this net be routed on (bicycle or bullet train)?
  - Buffering change routing layers
    - Vicious circle: estimate higher R → more buffers → net routed on lower layers → even higher R
    - Buffers are expensive in power, delay and cell density
  - Example: 20 parallel 1mm wires

	WNS	TNS	Area	# buffers	Average gaps between buffers
Default prediction	-100	-1.28	78	400	50 um
Improved prediction	0	0	17	100	333 um

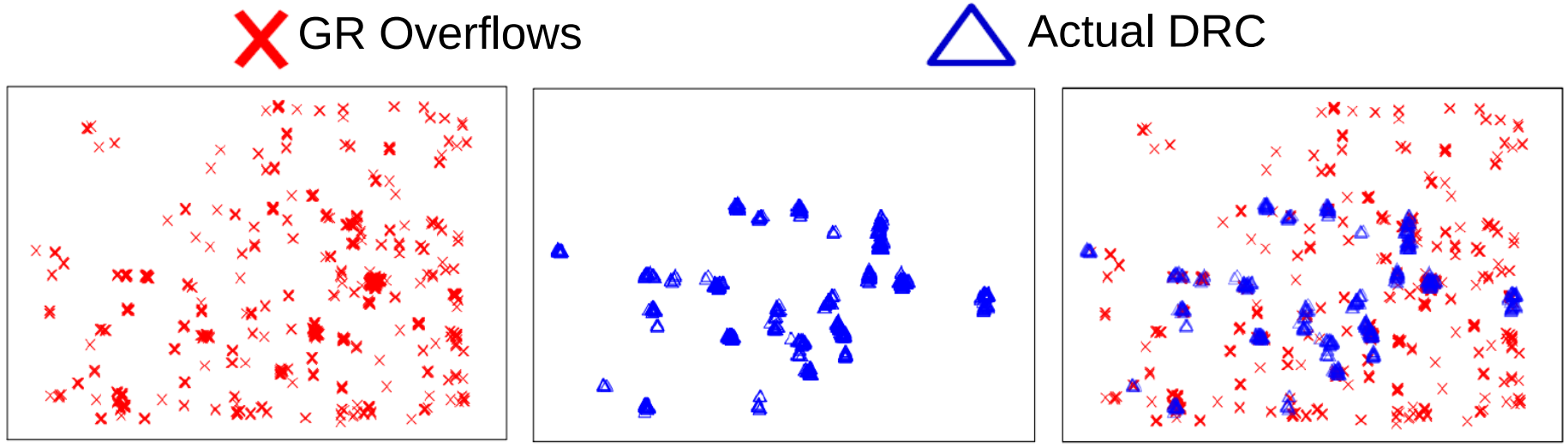
Interesting Problem 1: How To Better Predict R and Buffer The Net?

- How to better predict which layers this net will be routed on
  - Statistical analysis
  - Machine learning
    - Supervised
    - Unsupervised DNN

How About Routablity?

- Global route used to be effective in predicting routability (detailed route DRC)

- No longer the case:
  - W. Chen et al., ISPD 2017 →

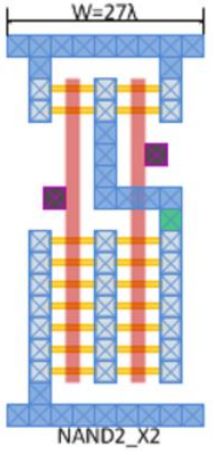


  - Why? Global route measures interconnect intensity, but not pin access or detailed-route design rules

SYNOPSYS®

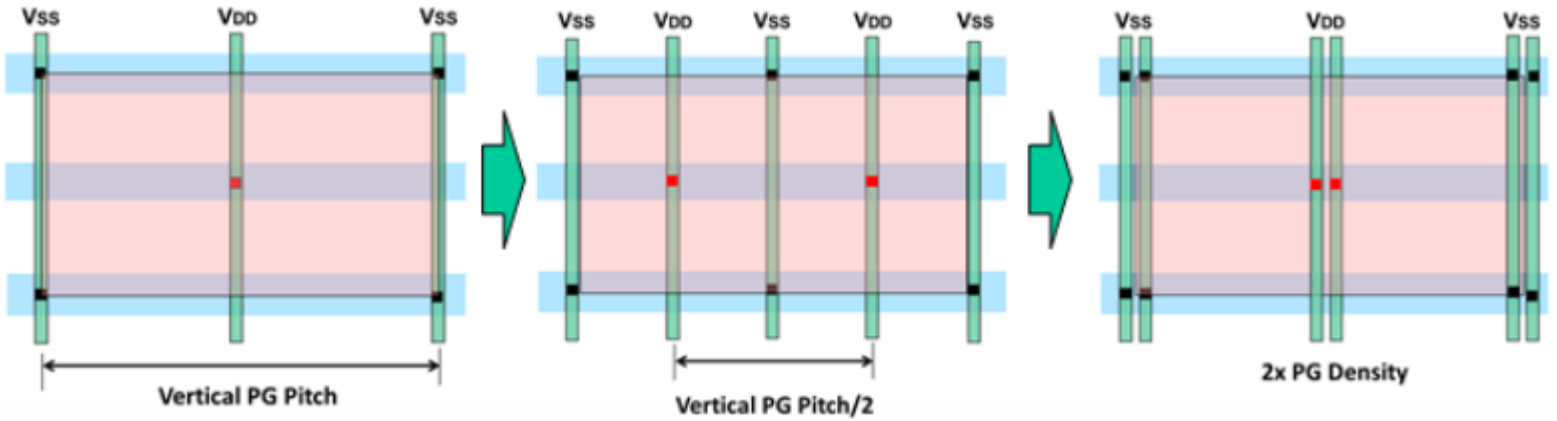
Pin Access under FinFET Technology

- Restricted pin placement in FinFET cells
  - Pin access point prevents fins
    - T. Cui et al., Great Lakes Symposium on VLSI 2015 →



- Multiple patterning rules
    - L. Lucas et al., SPIE 2012 →

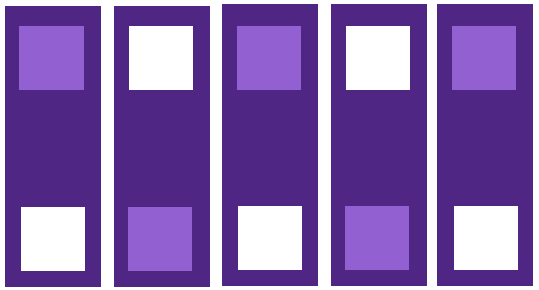
- Increased vertical PG density
  - Due to increased R
    - C. Hou, ISSCC17 →
  - Vertical PG blocks pin access

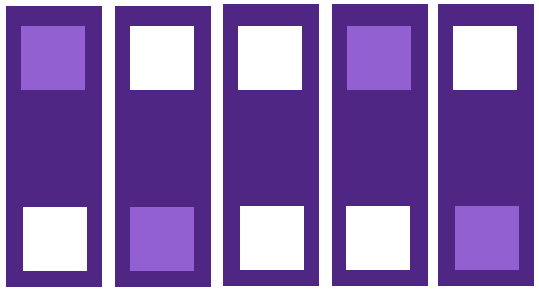


SYNOPSYS®

Reduce SAT Problem to Pin Access Problem

- Interaction between few access points and design rules:





Interesting Problem 2: How To Better Predict and Fix Detailed Route DRC?

- How to better predict detailed route DRC after global route?
  - Statistical analysis
  - Machine learning
    - Supervised
    - Unsupervised DNN
- Example:
  - W. Chen et al. ISPD 2017
    - Prediction: 74% true positive and 0.2% false positive
    - Optimization: Up to 76.8% DRC reduction

Two Crises in Physical Synthesis

- Interconnect prediction and optimization
  - Resistance
  - Route DRC

- Runtime of physical design tools
  - CPUs
    - Clock frequency: plateaued out
  - Both ICC2 and Innovus have sped up through multi-threading
  - What is next?

IC Complexity Has Been Growing

- ITRS Prediction of SoC Consumer Portable Design Complexity:

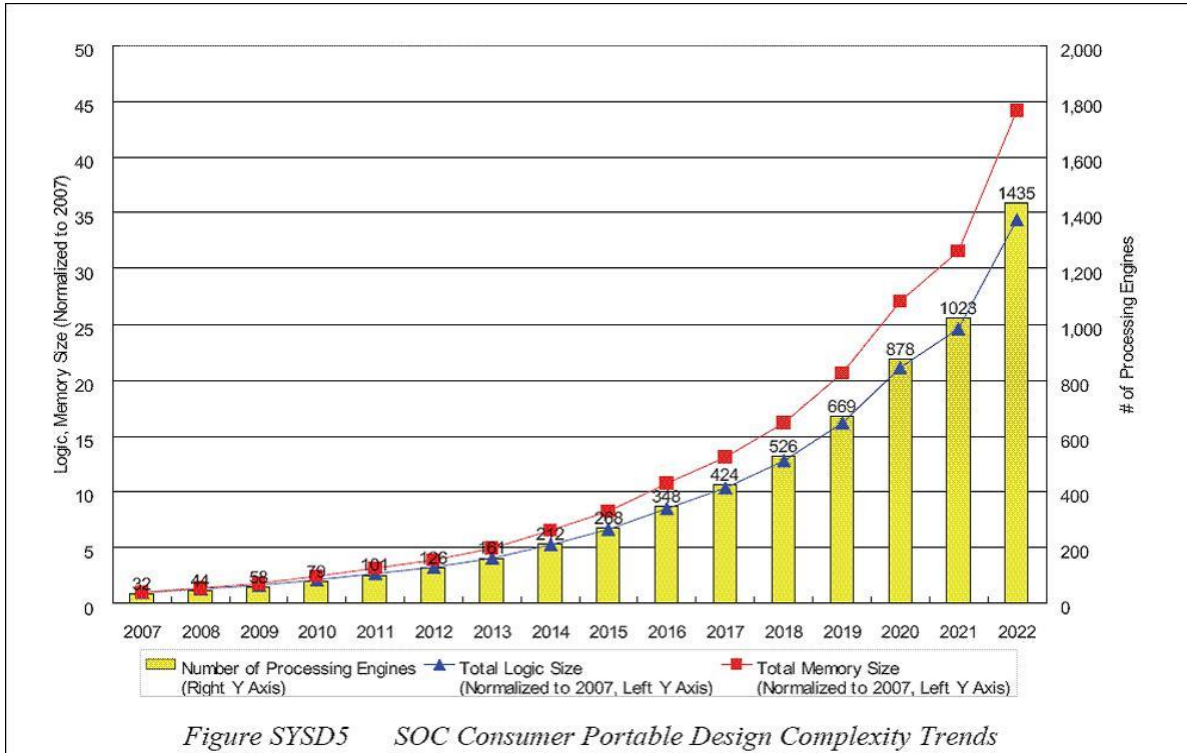


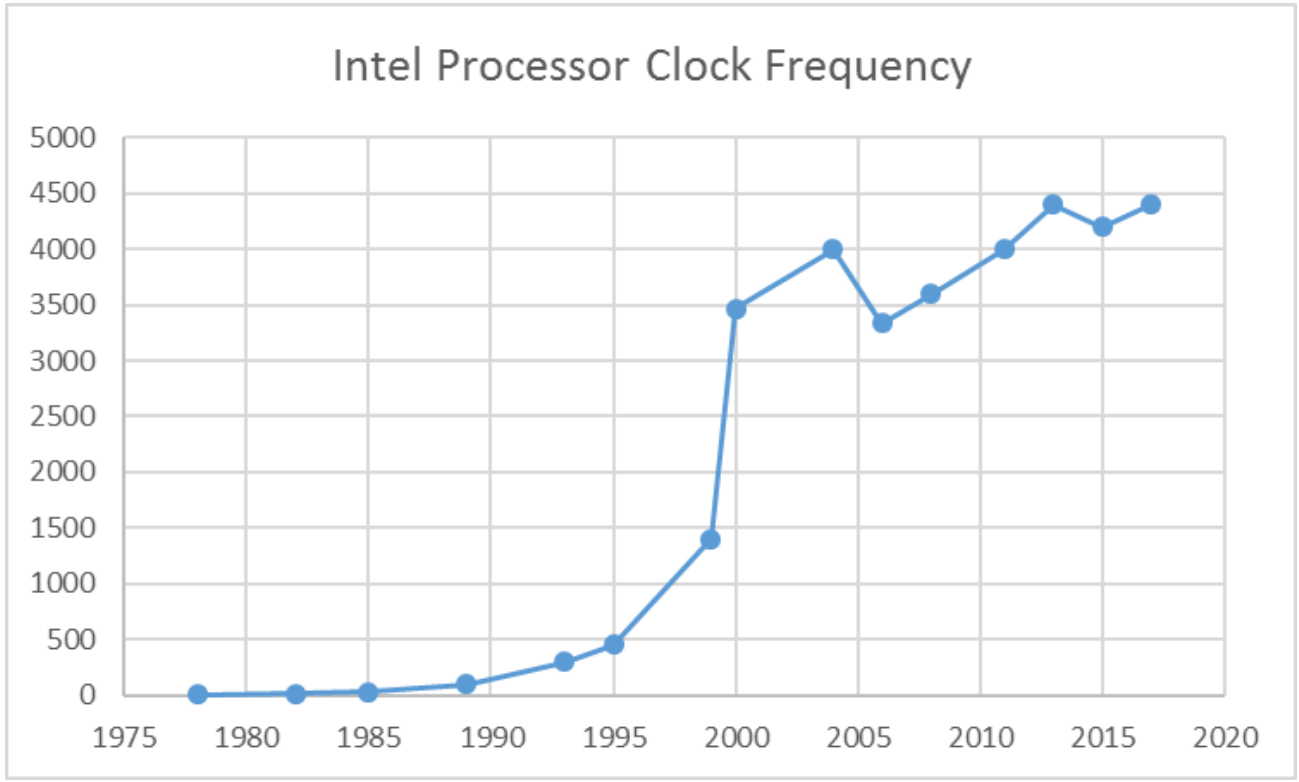
Figure SYSD5 SOC Consumer Portable Design Complexity Trends

- Performance of physical design tools like IC Compiler II and Innovus must grow, too!

SYNOPSYS®

Processor Clock Speed Has Barely Grow

- In the 13 years prior to 2004, Intel processor clock frequency improved by 20X
- In last 13 years since 2004, Intel processor clock frequency has improved by about 10%
  - No more automatic speedup of computation-intensive software

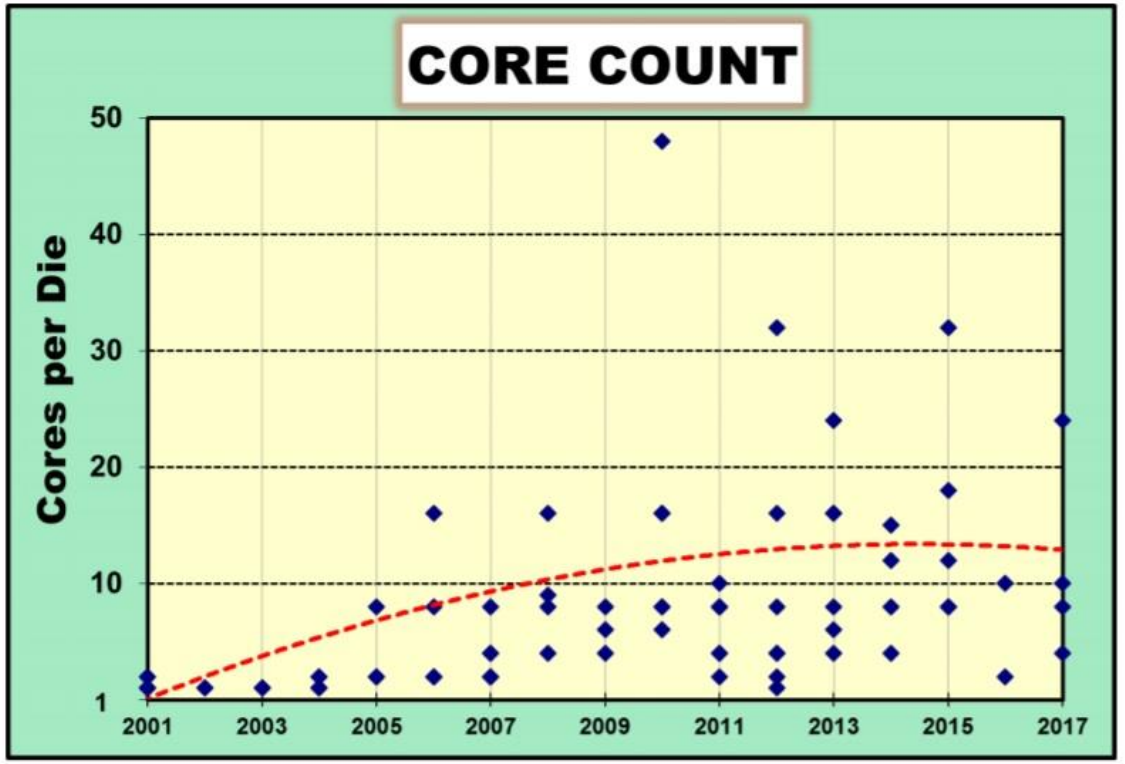


SYNOPSYS®

At Least We Have Multi-Core CPUs, Right?

- Both ICC2 and Innovus have been multi-threaded to use multi-core CPUs
  - But speedup usually plateaus out at around 16 cores

- Growth in core count has slowed
  - ISSCC 2017 general-purpose CPU core count →



SYNOPSYS®

Hardware Acceleration Becomes Necessary

- Cloud computing with GPUs
  - Amazon AWS
  - Microsoft Azure
  - Google cloud
  - Aliyun
  - Baidu
- Cloud computing with FPGAs
  - Amazon AWS
  - Microsoft internal
  - Baidu
  - Tencent
- Hardware acceleration of physical design tools!

GPU vs. FPGA

	GPU (Nvidia Tesla P100)	FPGA (Xilinx Ultrascale+ VU37P)
Forte	Parallel processing of floating-point operations	General purpose
Components	CUDA FP32 cores: 3584	LUTs: 1.3M; Flops: 2.6M; DSP: 12K
On-chip memory	3.5MB	UltraRAM: 34MB; BRAM: 8MB
HBM2	16GB	8GB
Interface	PCIe: 4 x 32GB/s	PCIe: 6 x 16GB/s; High speed transceivers: 128 x 4GB/s
Development cost	CUDA, OpenCL	OpenCL, RTL (synthesis, place and route are needed, 30 hr turnaround time)
Power	Higher	Lower
Cost	Lower	Higher

GPU vs. FPGA

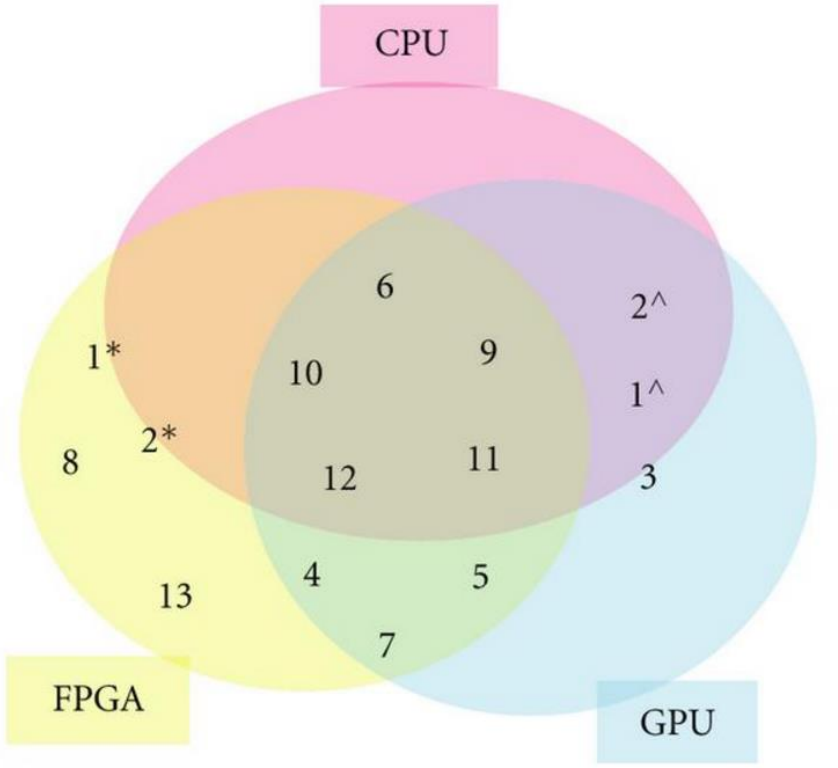
- Applications

GPU Strengths	FPGA Strengths
Fully parallelized: same control for all data (SIMD)	More complex control
Double, single or half precision floating-point data types	Custom data types
Floating-point operations	Fixed-point operations, bit-wise operations, etc.
Data larger than FPGA memory	Data fit in FPGA memory

GPU or FPGA

- Categories of HPC applications
  - UC Berkely's 13 dwarfs
    - R. Inta, et al., "The "Chimera": An Off-The-Shelf CPU/GPGPU/FPGA Hybrid Computing Platform,"International Journal of Reconfigurable Computing 2012

	Dwarf	Examples/Applications
1	Dense Matrix	Linear algebra (dense matrices)
2	Sparse Matrix	Linear algebra (sparse matrices)
3	Spectral	FFT-based methods
4	N-Body	Particle-particle interactions
5	Structured Grid	Fluid dynamics, meteorology
6	Unstructured Grid	Adaptive mesh FEM
7	MapReduce	Monte Carlo integration
8	Combinational Logic	Logic gates (e.g., Toffoli gates)
9	Graph traversal	Searching, selection
10	Dynamic Programming	Tower of Hanoi problem
11	Backtrack/ Branch-and-Bound	Global optimization
12	Graphical Models	Probabilistic networks
13	Finite State Machine	TTL counter

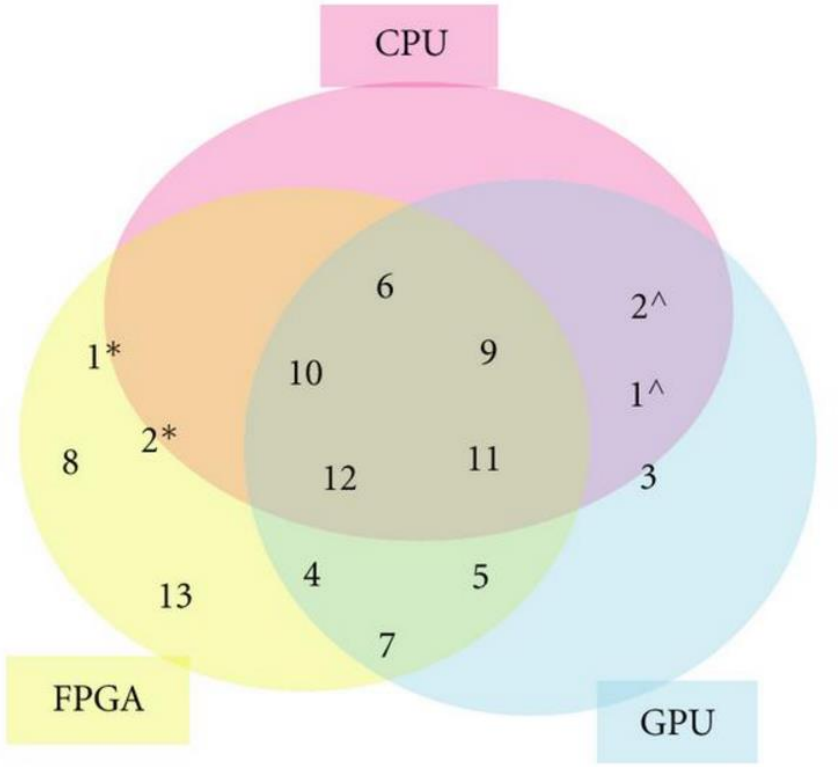


SYNOPSYS®

GPU or FPGA

- Examples: S. Che, et al. “Accelerating Compute-Intensive Applications with GPUs and FPGAs”
  - Gaussian elimination in double-precision floating-point matrix (Dwarf 1): GPU
  - Needleman-Wunsch protein sequencing (Dwarf 10): FPGA
  - DES data encryption (Dwarf 8): FPGA (GPU catches up with very large input size)

	Dwarf	Examples/Applications
1	Dense Matrix	Linear algebra (dense matrices)
2	Sparse Matrix	Linear algebra (sparse matrices)
3	Spectral	FFT-based methods
4	N-Body	Particle-particle interactions
5	Structured Grid	Fluid dynamics, meteorology
6	Unstructured Grid	Adaptive mesh FEM
7	MapReduce	Monte Carlo integration
8	Combinational Logic	Logic gates (e.g., Toffoli gates)
9	Graph traversal	Searching, selection
10	Dynamic Programming	Tower of Hanoi problem
11	Backtrack/ Branch-and-Bound	Global optimization
12	Graphical Models	Probabilistic networks
13	Finite State Machine	TTL counter



Interesting Problem 3: How to Accelerate Physical Design Using GPU, FPGA or Both?

- Example:
  - FPGA-based emulation speeds up RTL simulation by 0.5M times
  - GPU speeds up RTL simulation by 10 times
- Which dwarves does your tool use?
  - Machine learning
  - Analytical global placement
  - Analytical sizing
  - Incremental timing analysis
  - Detailed placement
  - Buffering
  - CTS
  - Routing

Interesting Problem 4: Design Efficient Physical Design Solution in C++ and RTL

- Interesting challenge:
  - Design an efficient solution in a combination of software and hardware
  - Turnaround time is much longer than compiling C++ code
    - FPGA place and route is very time consuming (30+ hours on more advanced FPGAs)
- Not very interesting challenge:
  - Servers accelerated by multiple FPGAs are hard to find
    - Observation: FPGA-based emulators are like FPGA supercomputers

Interesting Problem 5: How to Solve FPGA's Routability Problem

- FPGAs have limited routing resource

- Xilinx FPGAs have multiple dies: inter-die connections are fewer and slower than intra-die
  - Placer spreads the LUTs across two dies: may exceed inter-die connection capacity
  - Placer squeezes the LUTs within one die: pin density may become too high to be routable

Summary: Interesting Physical Synthesis Problems

- Problem 1: How to better predict R and buffer the net
- Problem 2: How to better predict and optimize route DRC
- Problem 3: How to accelerate physical design using GPU, FPGA or both
- Problem 4: Design efficient physical design solution in C++ and RTL
- Problem 5: How to solve FPGA’s routability problem

