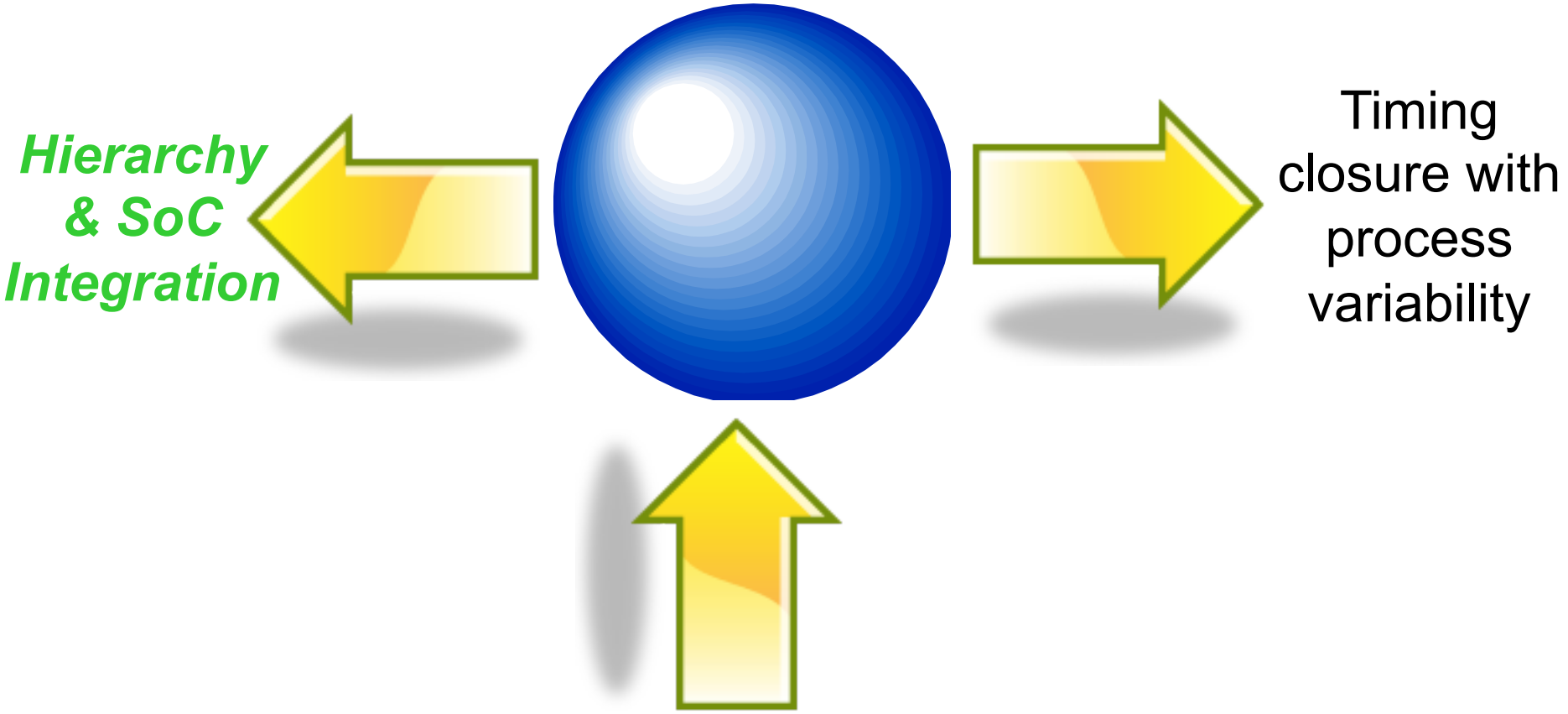
Electronic Design Automation

Static timing analysis: challenges and opportunities at 22nm and below

Kerim Kalafala, IBM

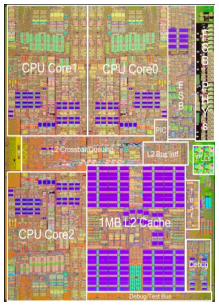
International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS) 2011

Grand challenges for static timing analysis

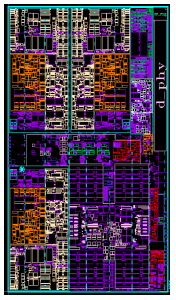


SoC and 3D Chip Stack Integration

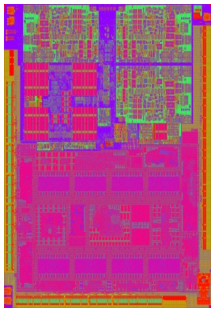
CPU



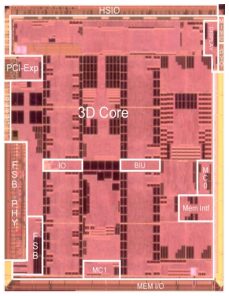
90nm, 2005



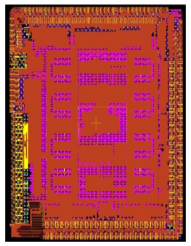
65nm, 2007



45nm, 2010
372M Transistors



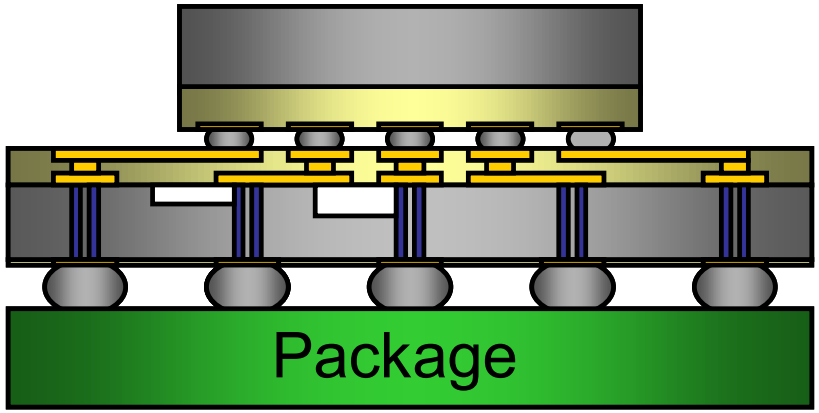
90nm, 2005



65nm, 2008

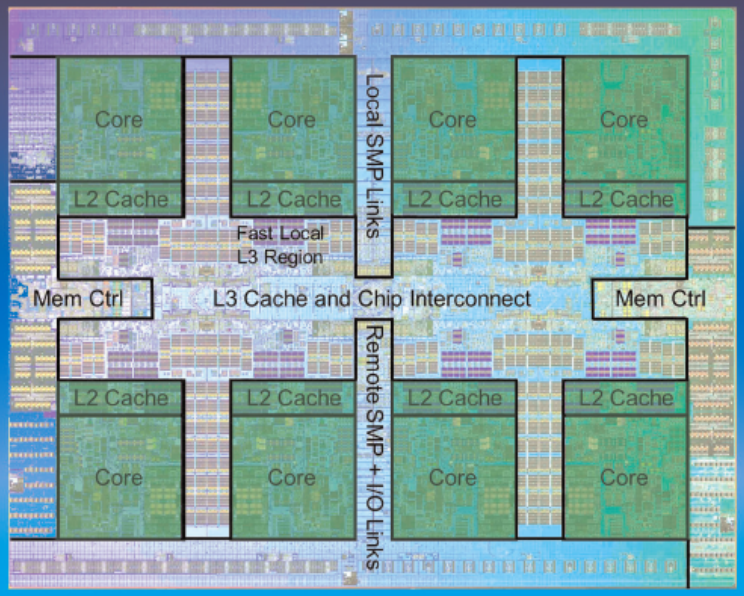
GPU

Courtesy Bob Drehmel, Deb Dyson (IBM)



IBM POWER7® Chip

VOLUME 55, NUMBER 3, MAY/JUNE 2011
IBM Journal of Research and Development
Including IBM Systems Journal



IBM POWER7 Technology and Systems

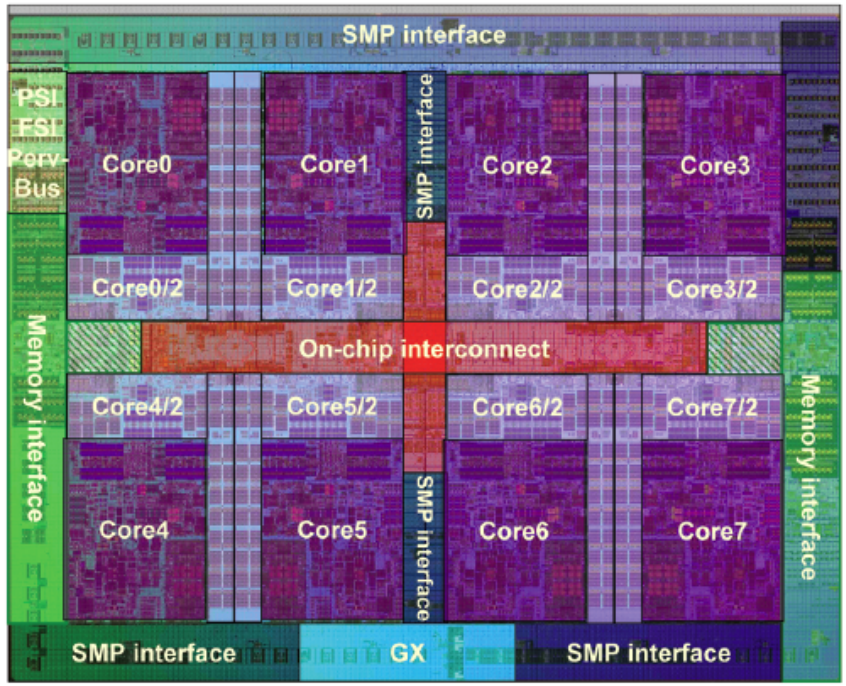
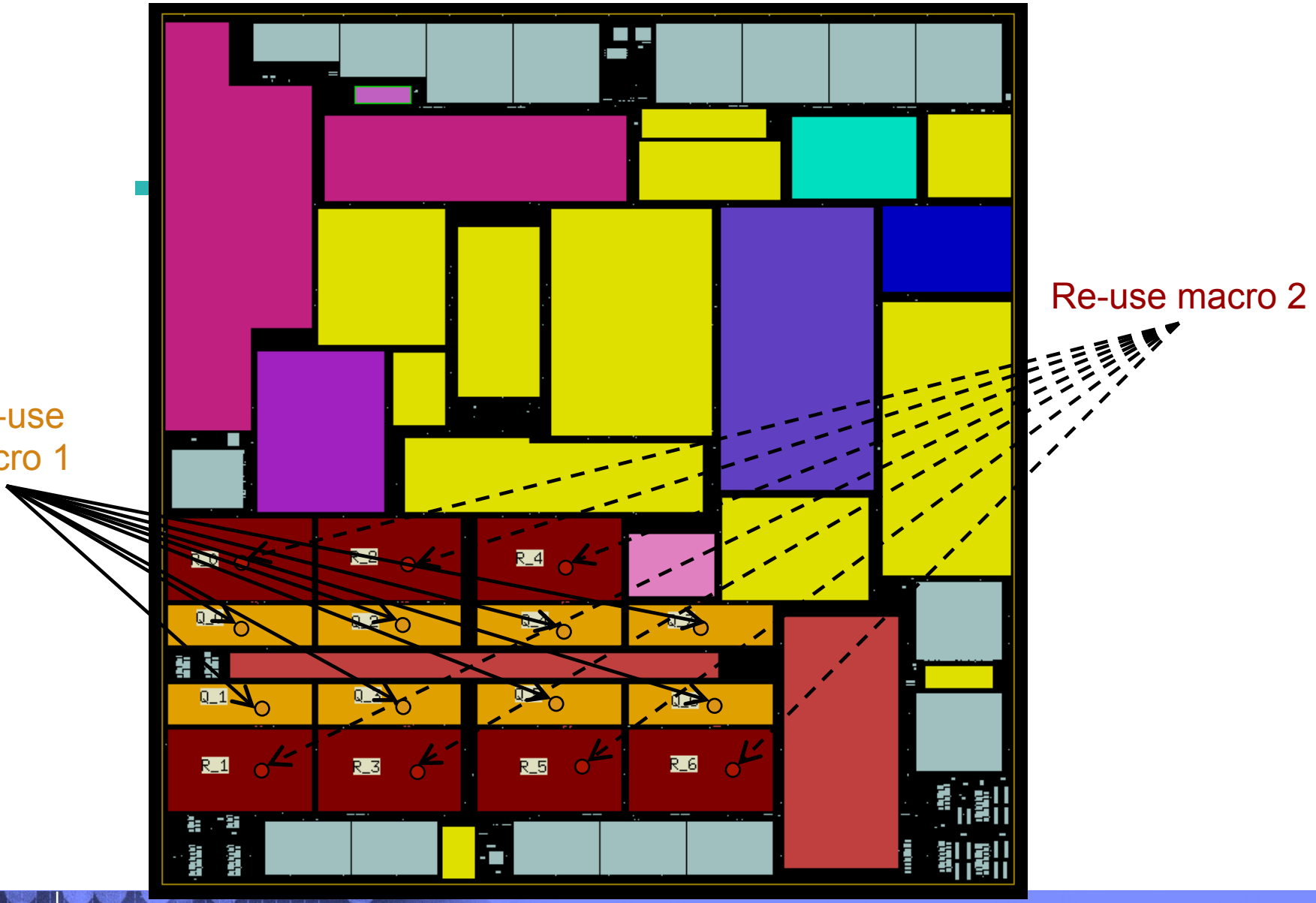


Figure 2

POWER7 frequency domains.

OEM ASIC floorplan

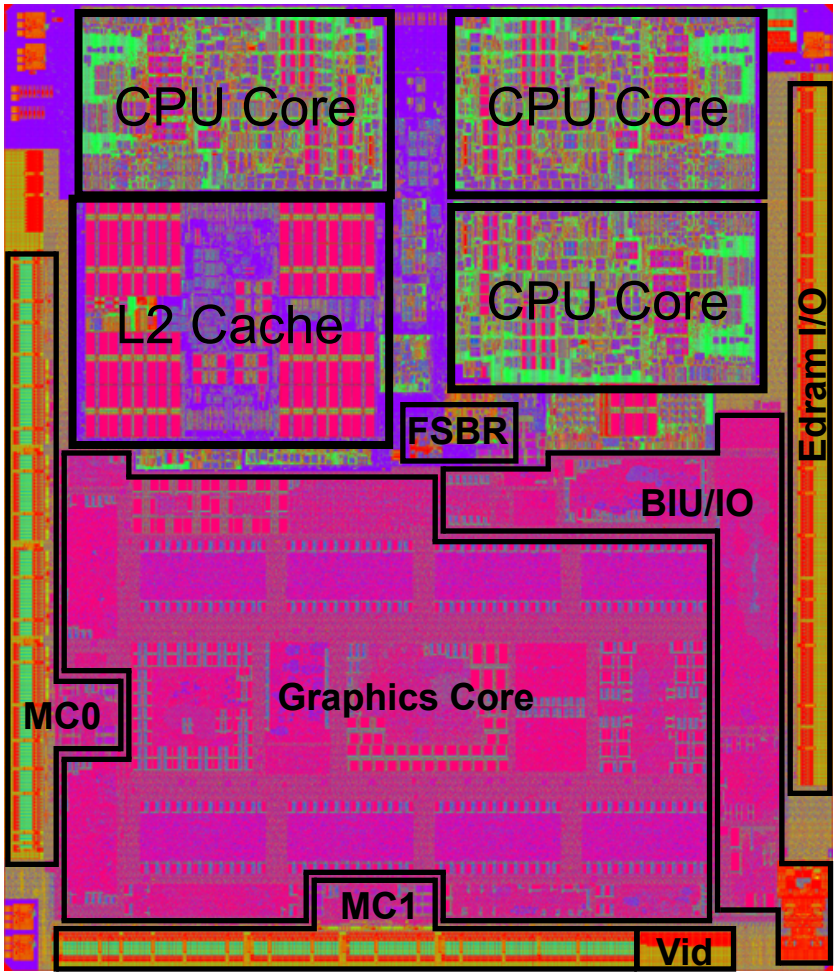
Courtesy Paul Zuchowski, Henry Burkhart (IBM)



Recent Gaming Design

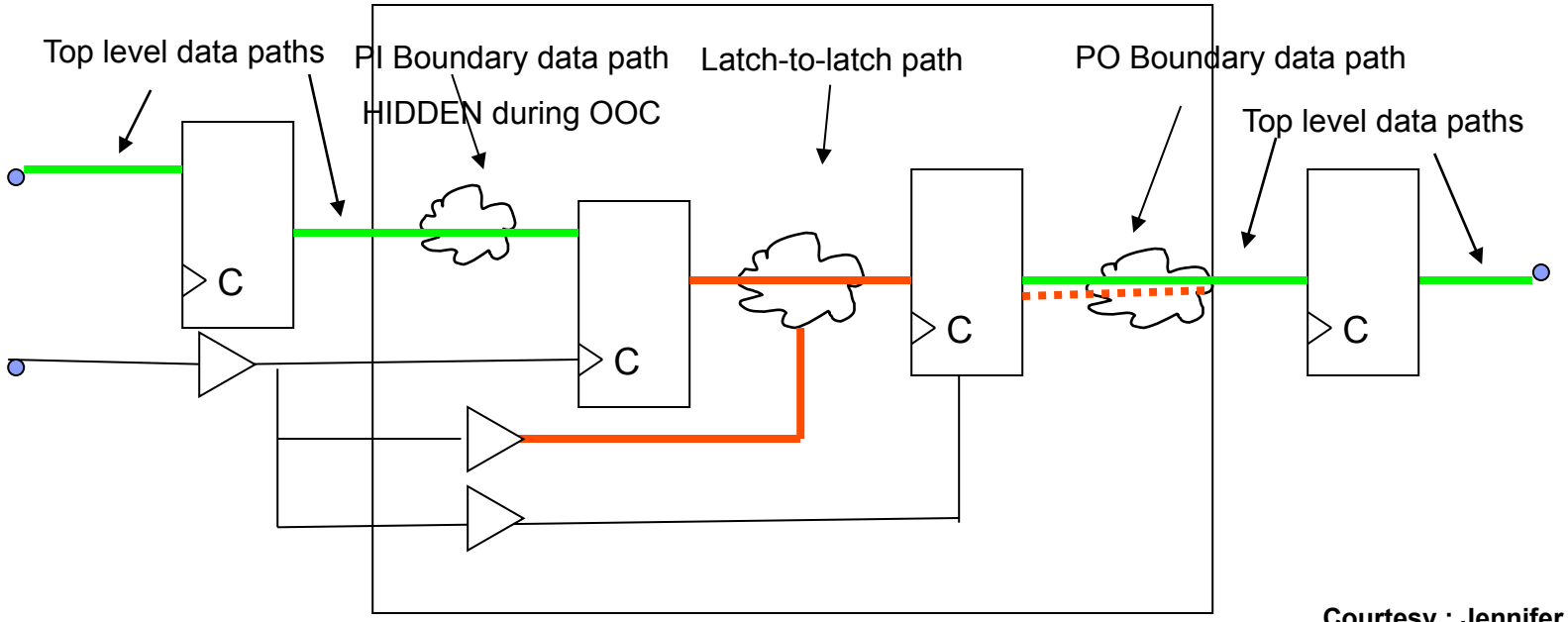
Courtesy Bob Drehmel, Deb Dyson (IBM)

- Chip Statistics
  - 372M transistors
  - 45nm SOI, Ultra-low k dielectric
  - 10 levels of metal
  - 153 array types, ~1000 instances
  - 1.8 million flip flops
  - 6 PLLs
  - 12 clock domains
  - Compared to 2005 CPU GPU
    - >60% Power Reduction
    - >50% Silicon Area Reduction
- Package Technology
  - 35mm FC-PBGA (3-2-3) build-up layers
  - Lidded Multi-Chip Module
  - High speed interface to on-module EDRAM
  - C4 Pitch: 151um minimum
- Power Delivery
  - Adaptive Power Supply (APS)
  - 8 Power Domains
- Manufactured by multiple foundries



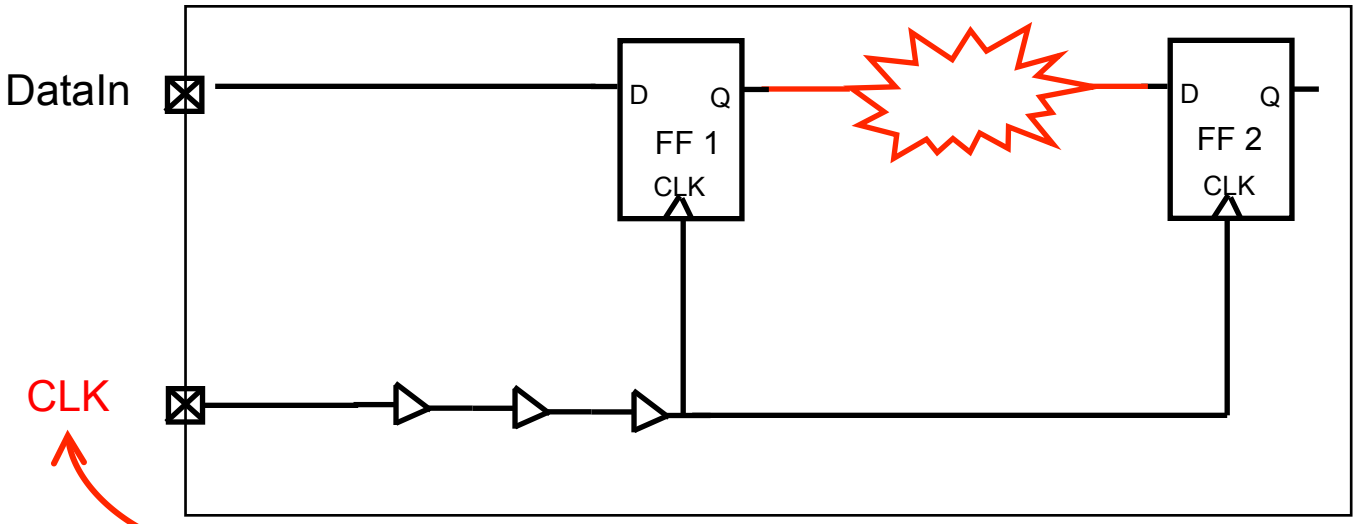
Divide and conquer via hierarchical abstraction

- **Macros are timed Out Of Context (OOC)**
  - latch-to-latch paths signed-off at the OOC level (red)
  - PI boundary nets are “hidden”, so they will not show up in the critical path report
- **Macro boundary paths are timed at the top level**
  - RLM boundary paths are signed-off at the top Level (green)

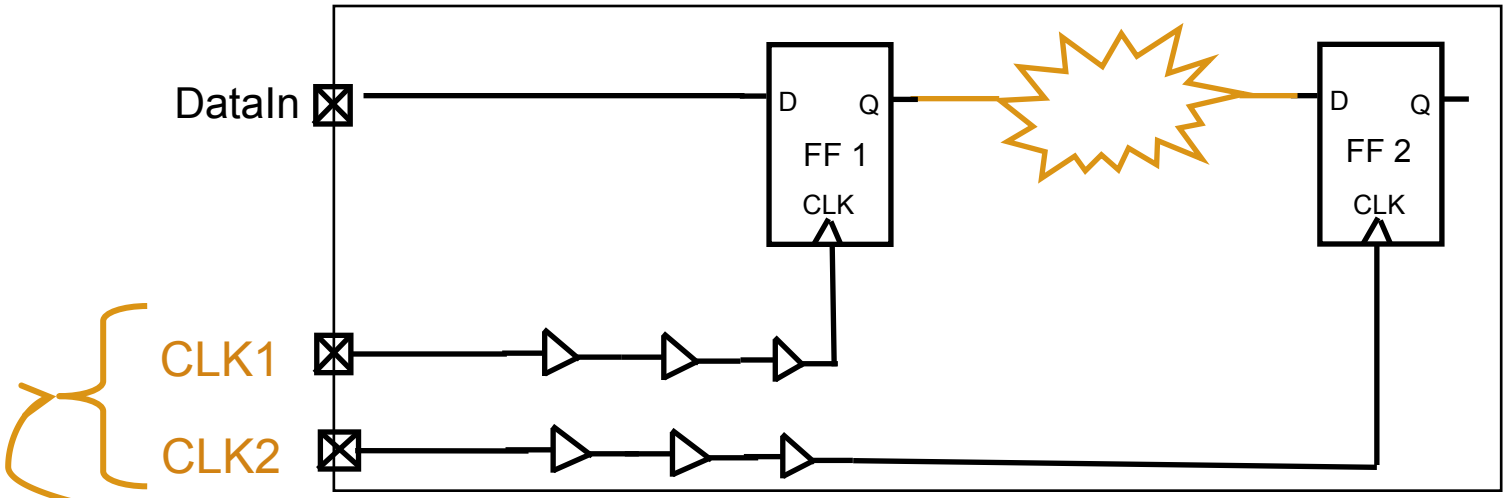


Courtesy : Jennifer Basile

Clocking considerations: duty cycle and skew



Need a duty cycle constraint for half-cycle paths

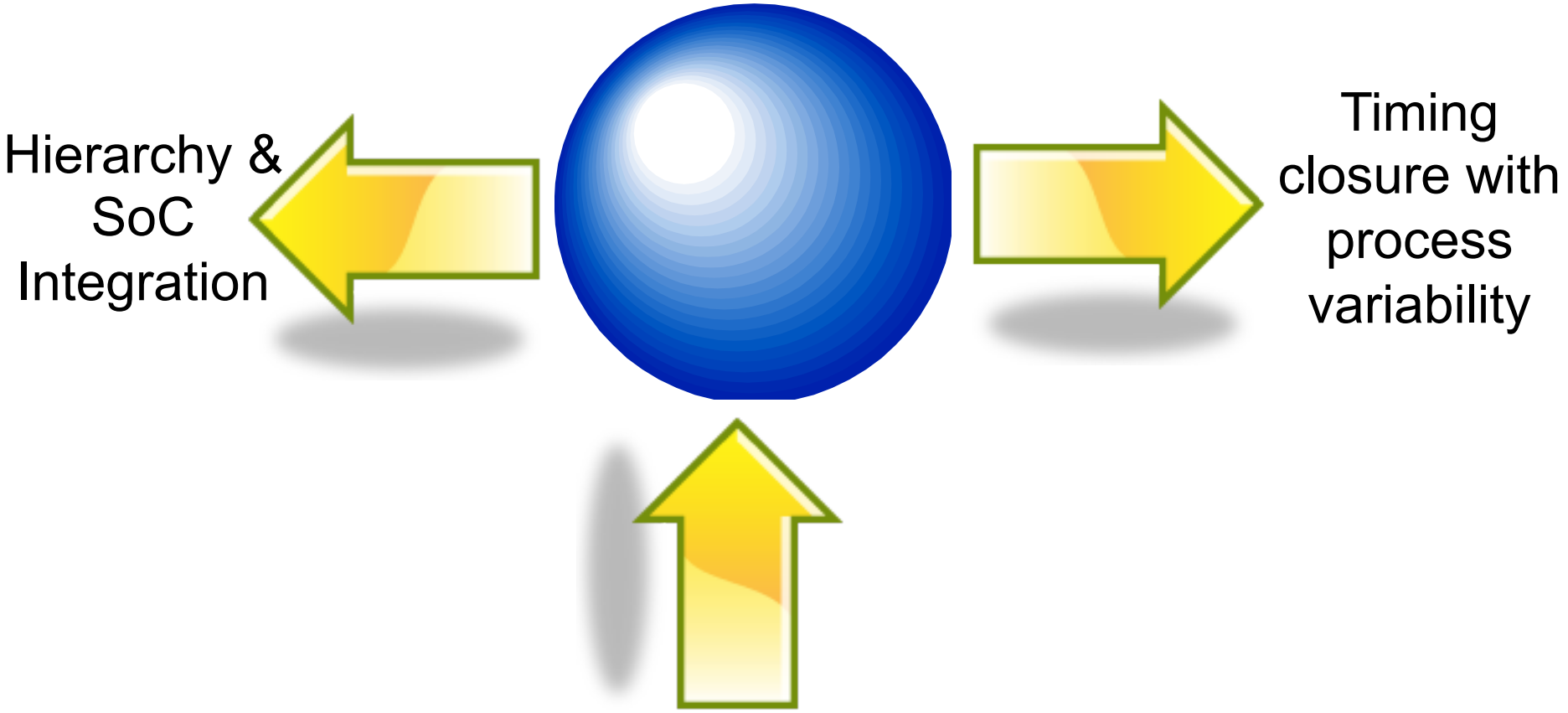


Need a skew constraint for inter-domain paths

General observations regarding hierarchy

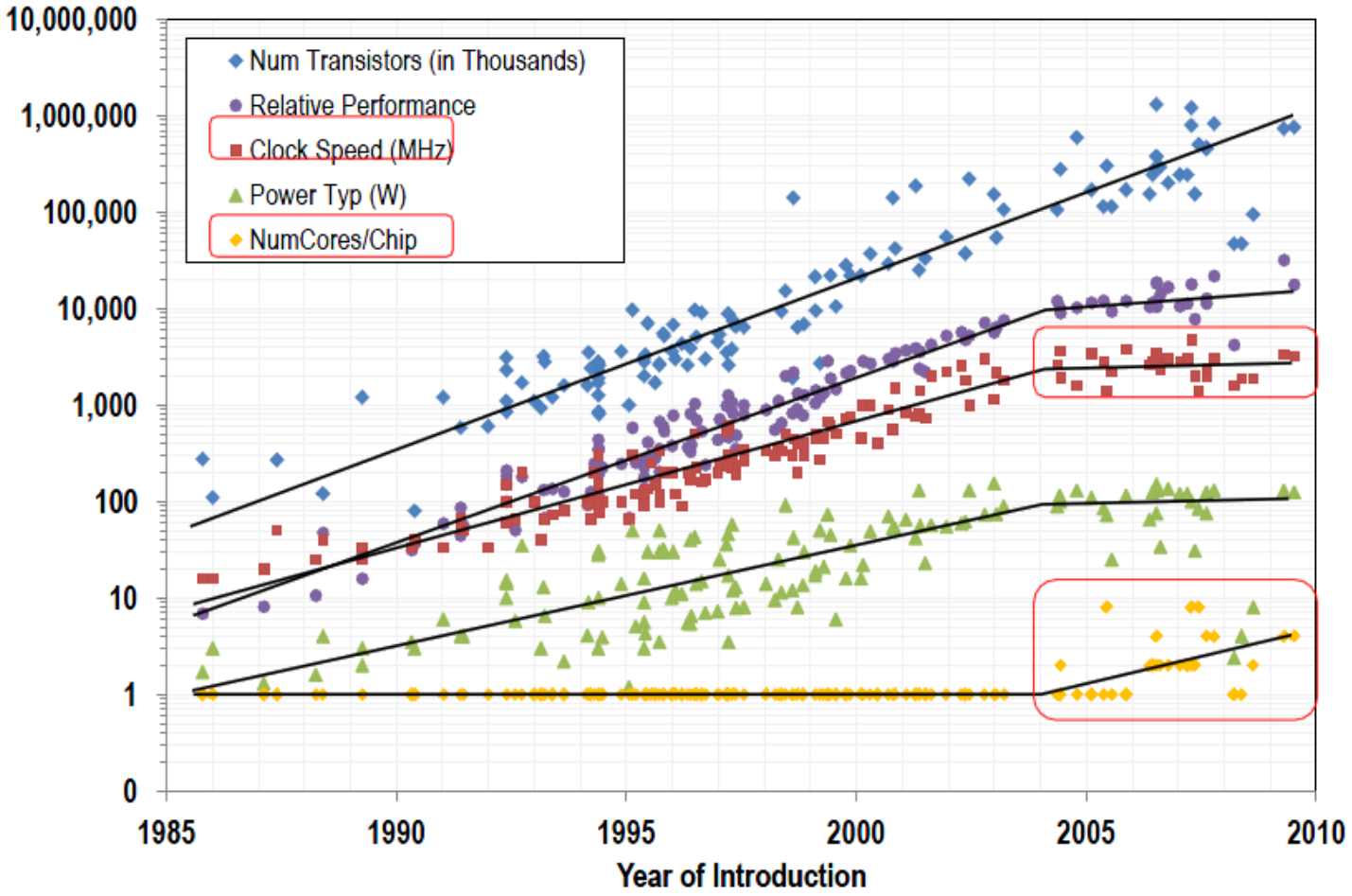
- **Suppose your design consists of:**
  - U unique macros
  - Which have an average graph size of S nodes
  - Which are re-used on average R times
  - And for which the ratio of abstract/detailed graph size is C
- **To analyze this design flat requires STA on overall graph size of: U * S * R**
- **By leveraging hierarchy we can reduce the total effective computation cost to:**
  - U*S (For out of context analysis) + U*(S*C)*R (For top-level analysis)
- **E.g.**
  - Let's pick some values: U = 50, S=100K nodes, R=5, C=0.2
  - Full flat analysis requires an STA graph of size: 25M nodes
  - By leveraging hierarchy, we reduce this to:
    - 5M (total size of all unique macros) + 5M (size of top-level with abstracted macros) = 10M node computations
  - 2.5X reduction in problem size
  - Closer to a 5X improvement in TAT if all unique macros can be analyzed in parallel across multiple systems
- **This is good, but we need more, otherwise:**
  - We will be fundamentally limited in macro size [S] (which makes design closure difficult)
  - And/or we will be limited in the # unique [U] IP components that can be co-analyzed in a single environment

Grand challenges for static timing analysis



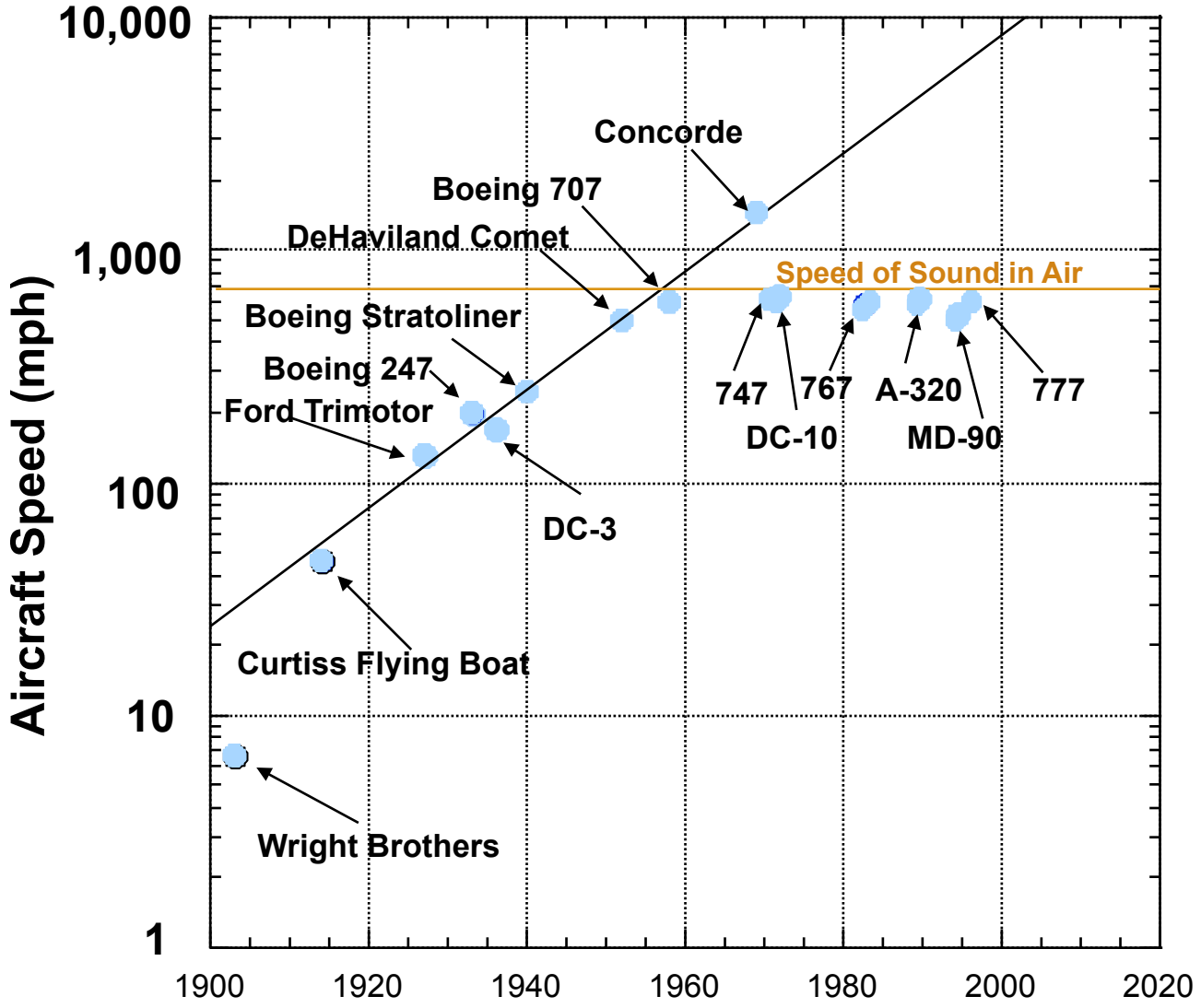
Post-exponential era in performance scaling

Microprocessor Performance Trends



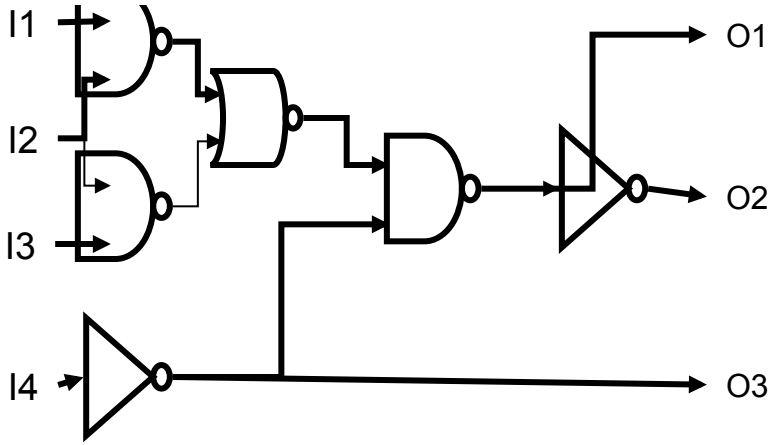
[The Future of Computer Performance: Game Over or Next Level?
Data by Mark Horowitz with input from Kunle Olukotun, Lance Hammond, Herb Sutter, Burton Smith, Chris Batten, and Krste Asanoviç]

Commercial Aviation

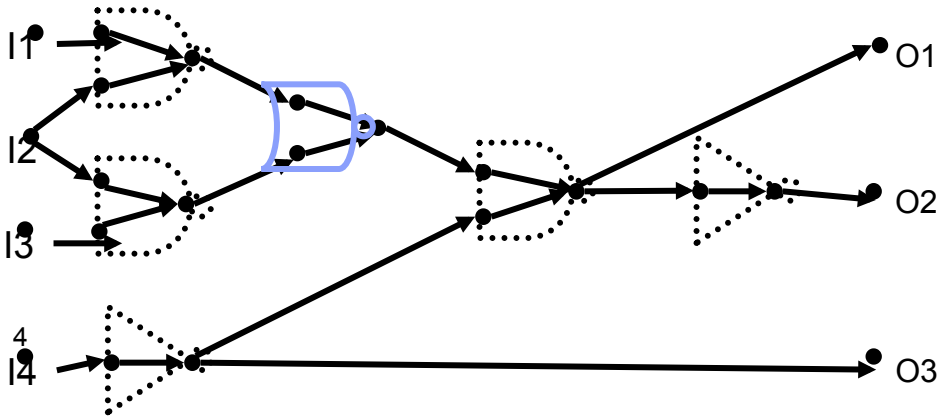


Source: F. M. Schellenberg, UCB Bodega Bay, 5-10-2007

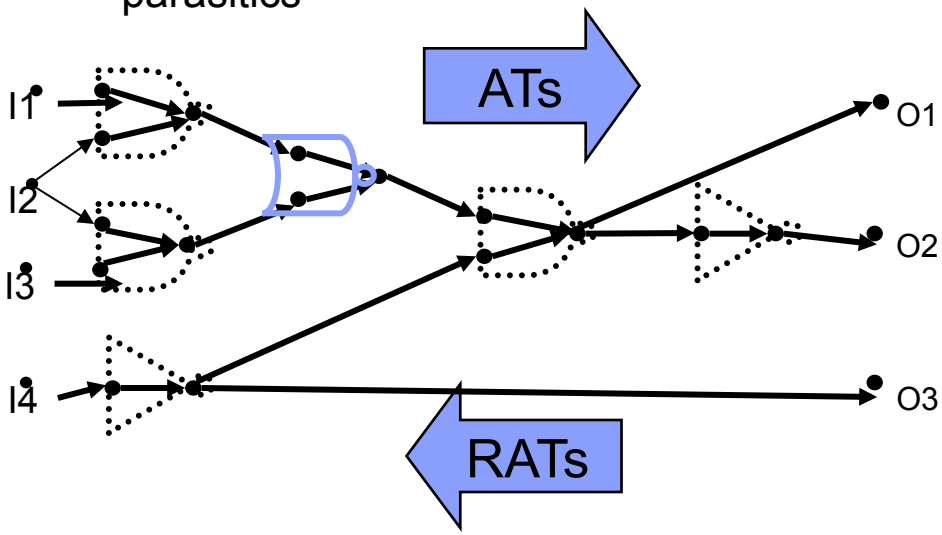
Block-based static timing analysis



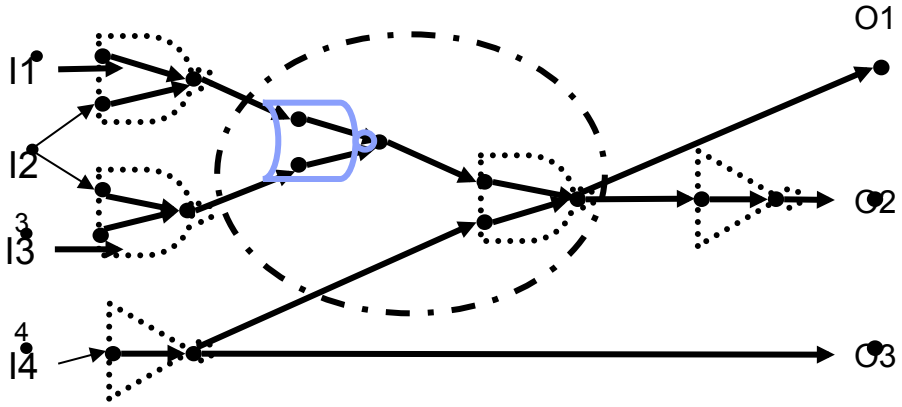
Step 1- Load netlist / assertions / parasitics



Step 2. – Create Timing graph (DAG) corresponding to the preceding circuit.

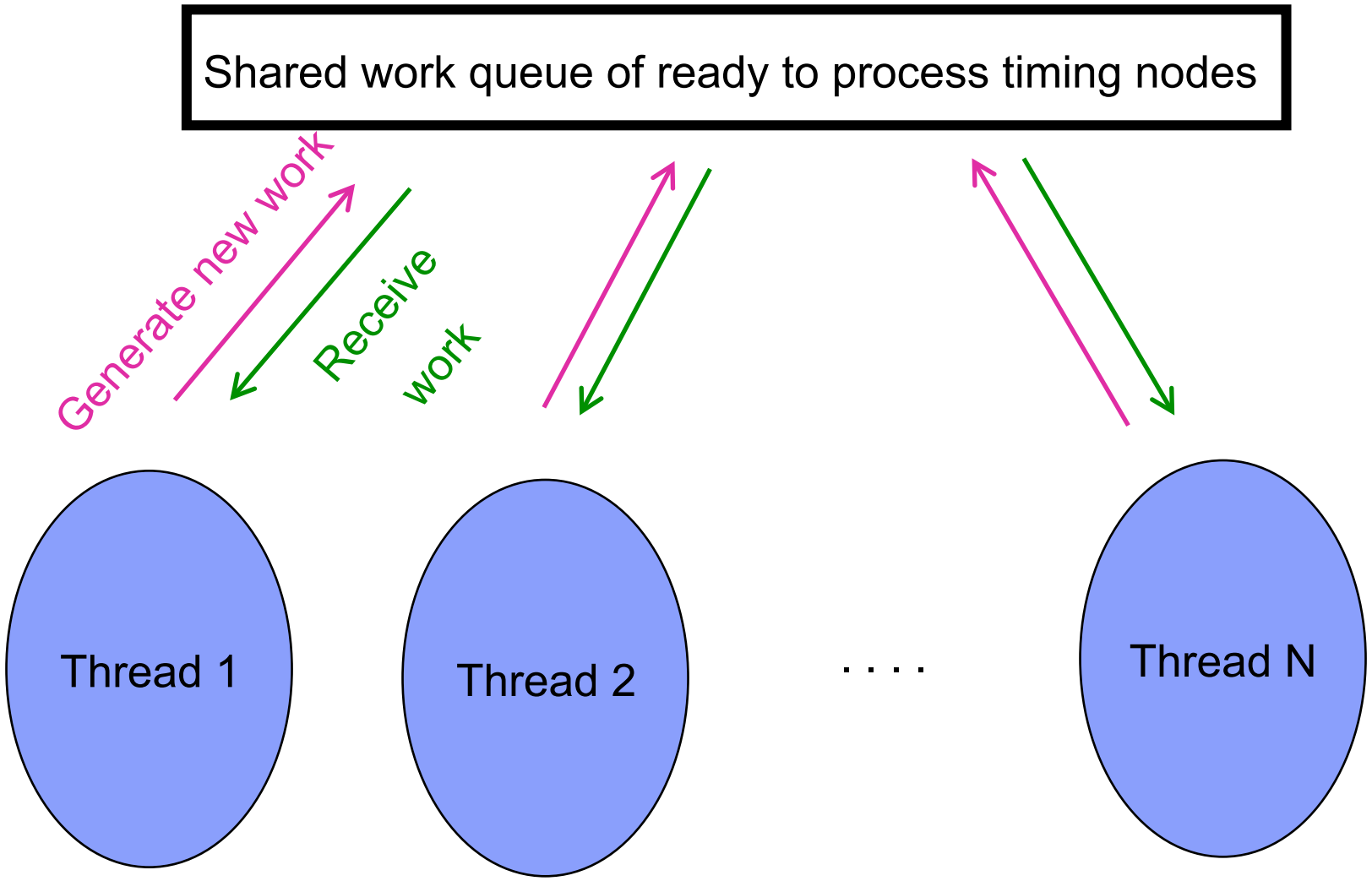


Step 3 – Propagate Arrival Times (ATs) Forward and Required Times (RATs) Backward

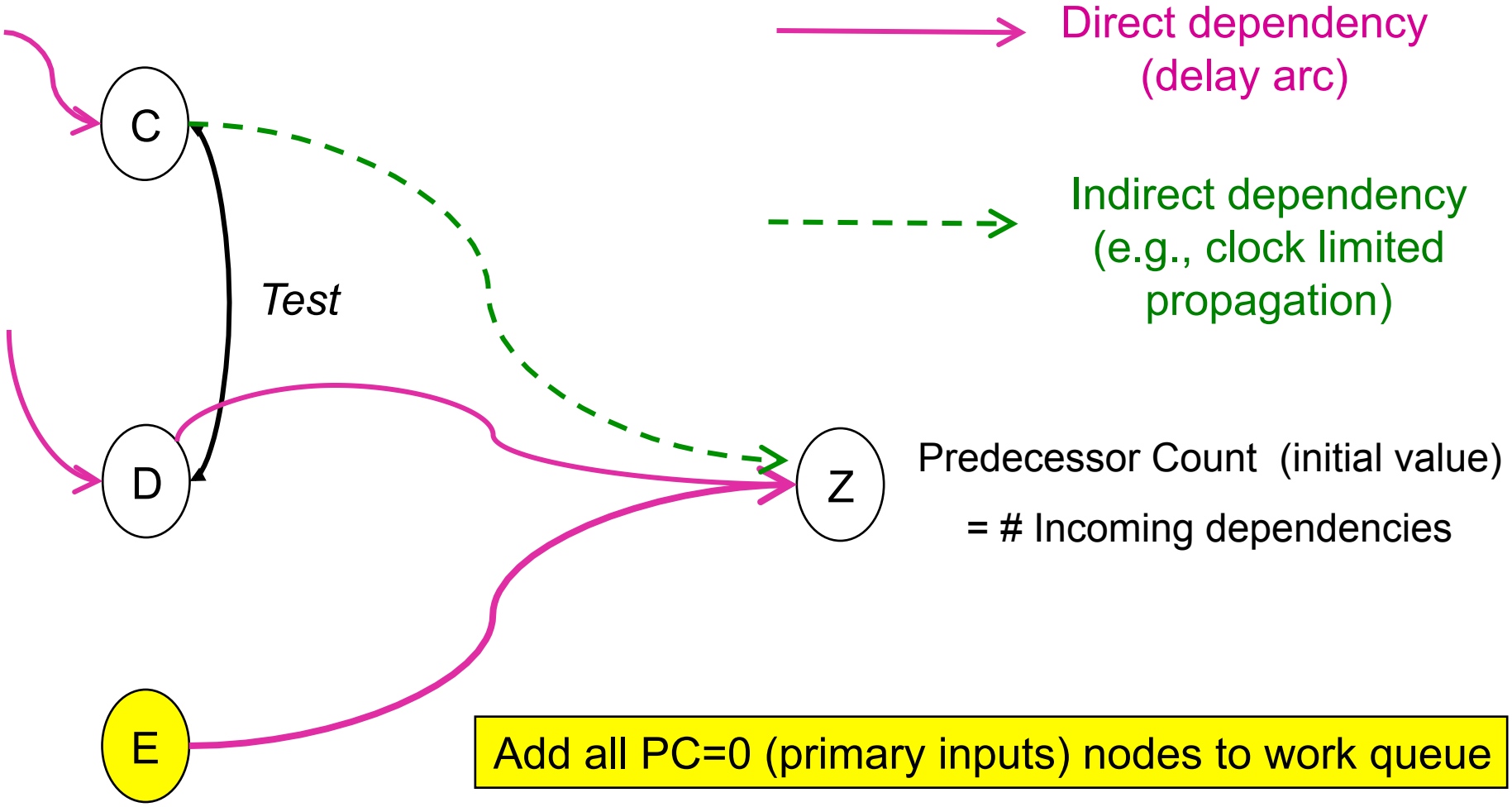


Step 4 – Fix-up and incremental re-analysis

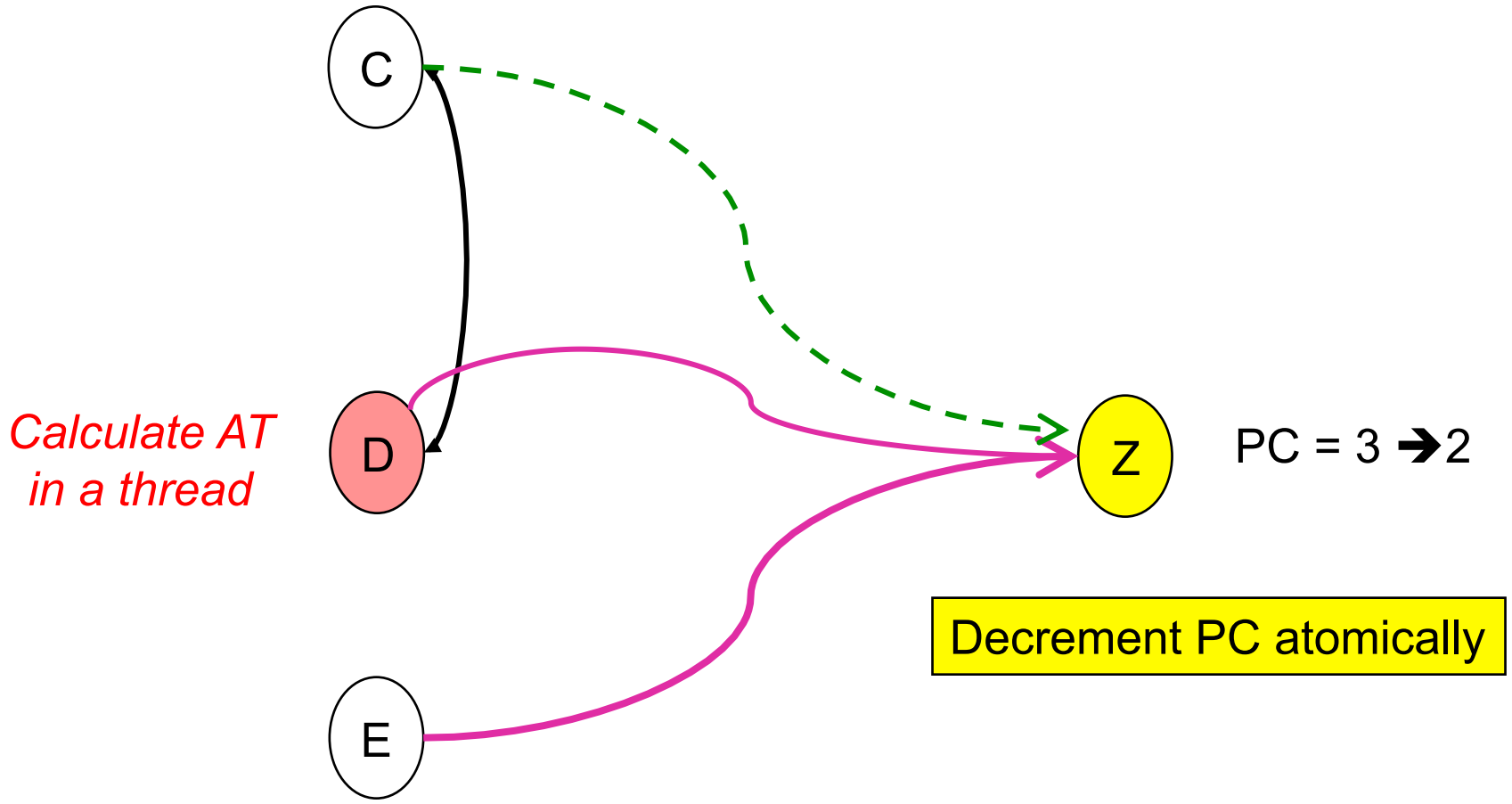
Key idea for parallelizing base AT/RAT propagation: dynamic work queue processing



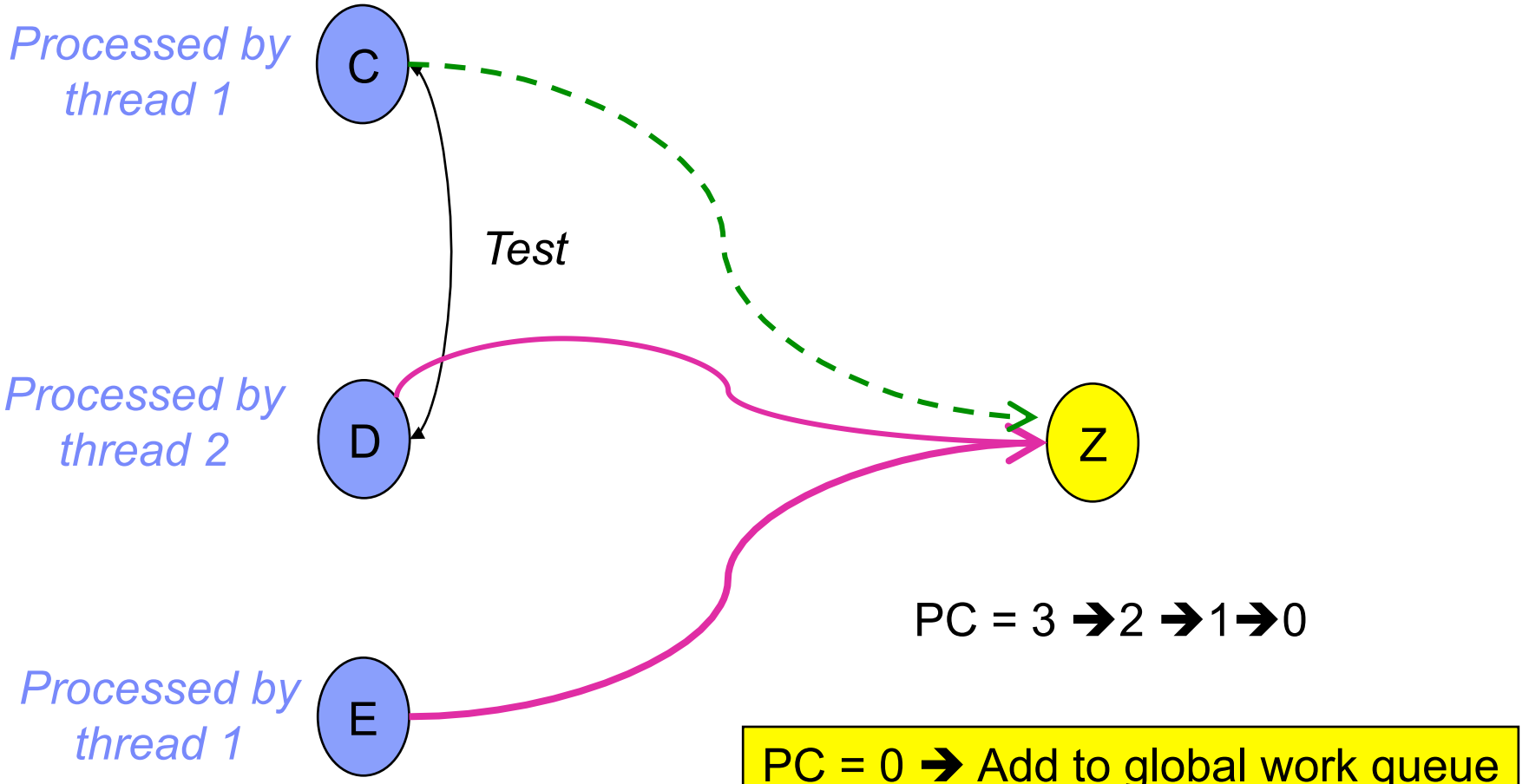
Calculation of initial predecessor counts



Updating predecessor count using atomic operations

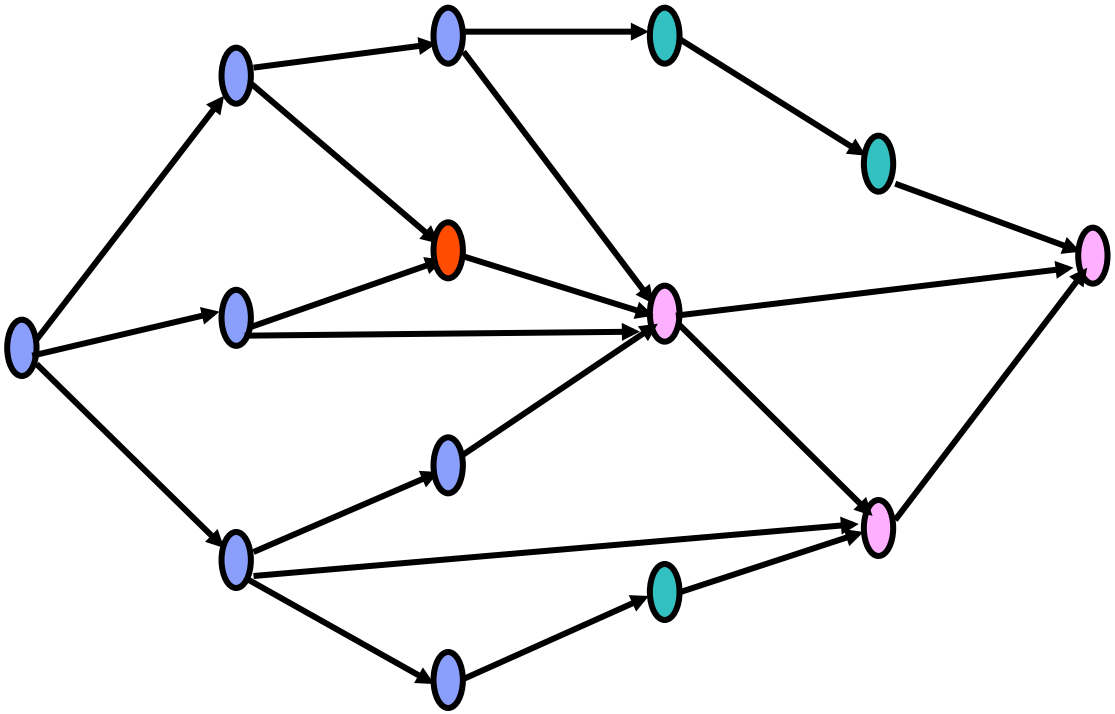


Generating new work based on updated predecessor counts

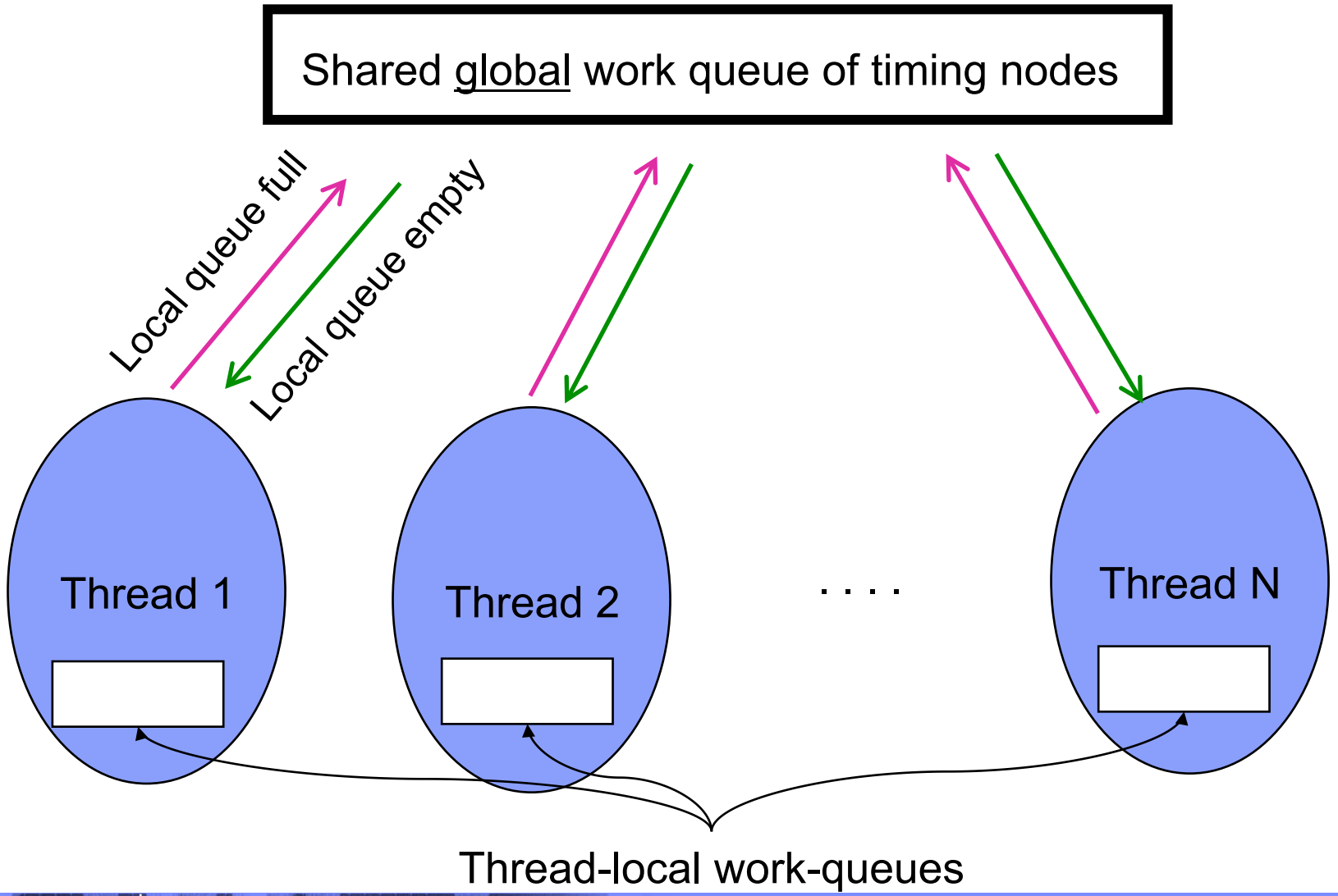


Benefit of dynamic processing

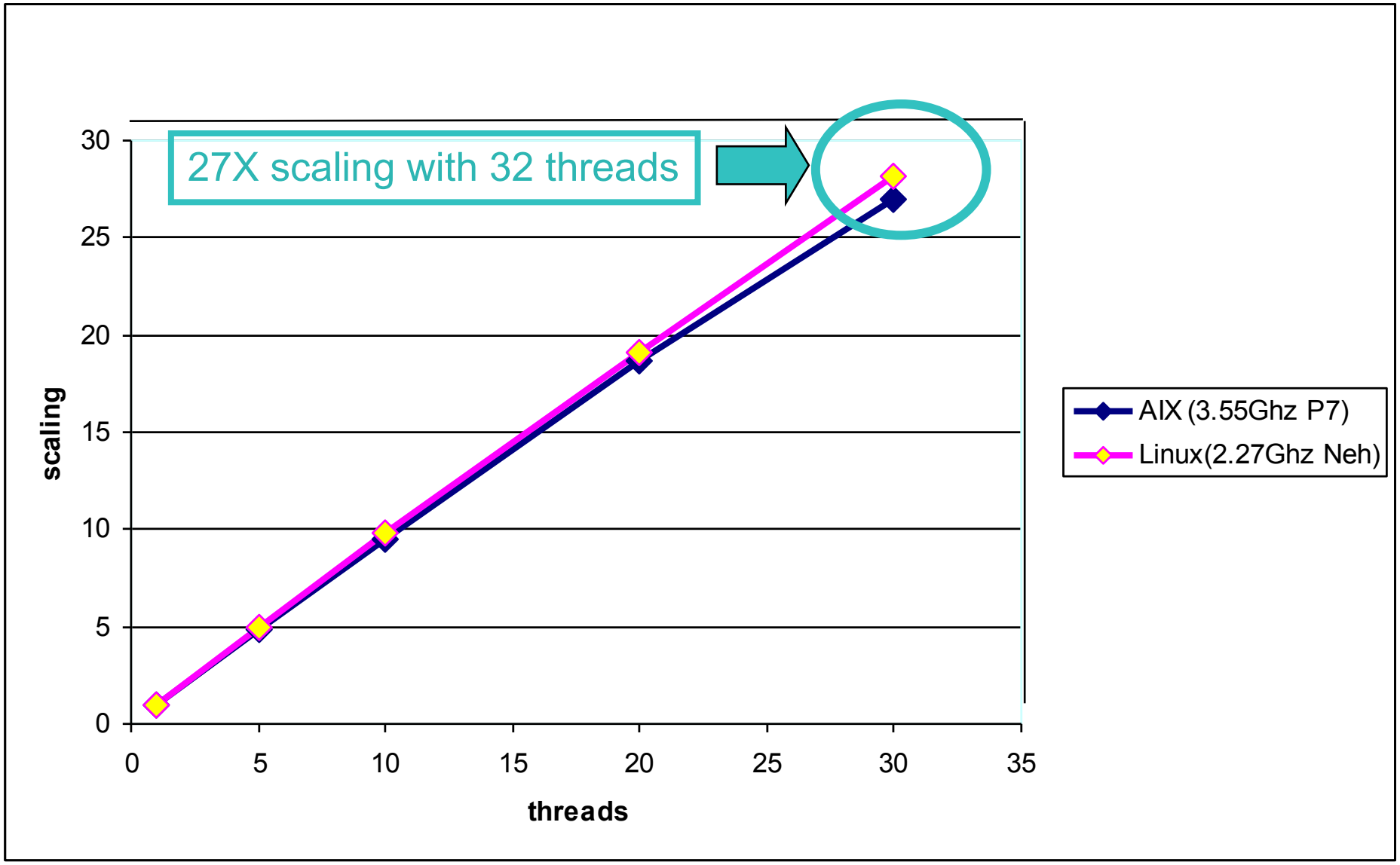
- Bottleneck (e.g., complex delay calculation)
- Nodes where work can proceed independent of bottleneck



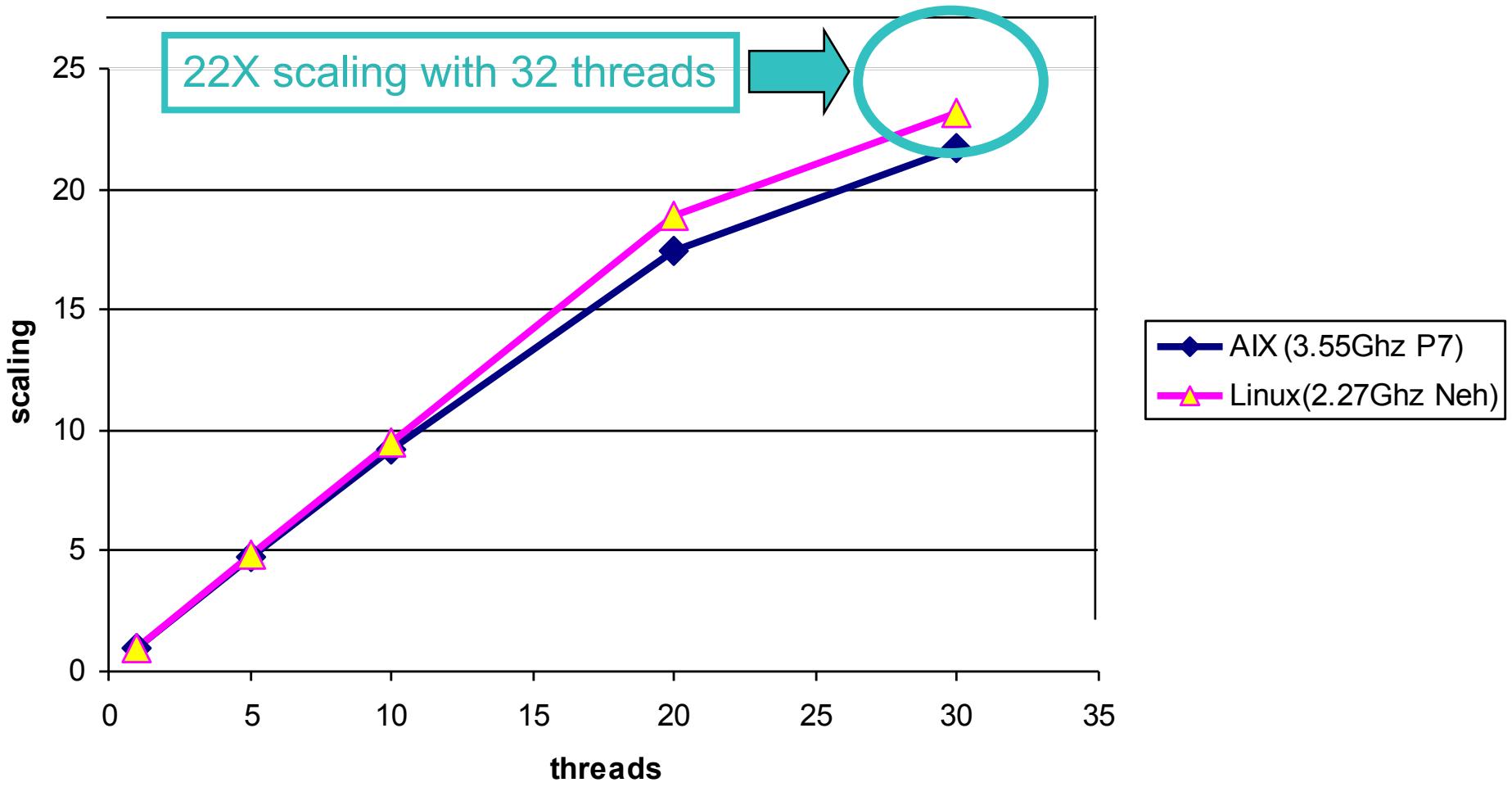
Use of local queues to minimize global queue locking



Multi threaded dynamic arrival time scaling



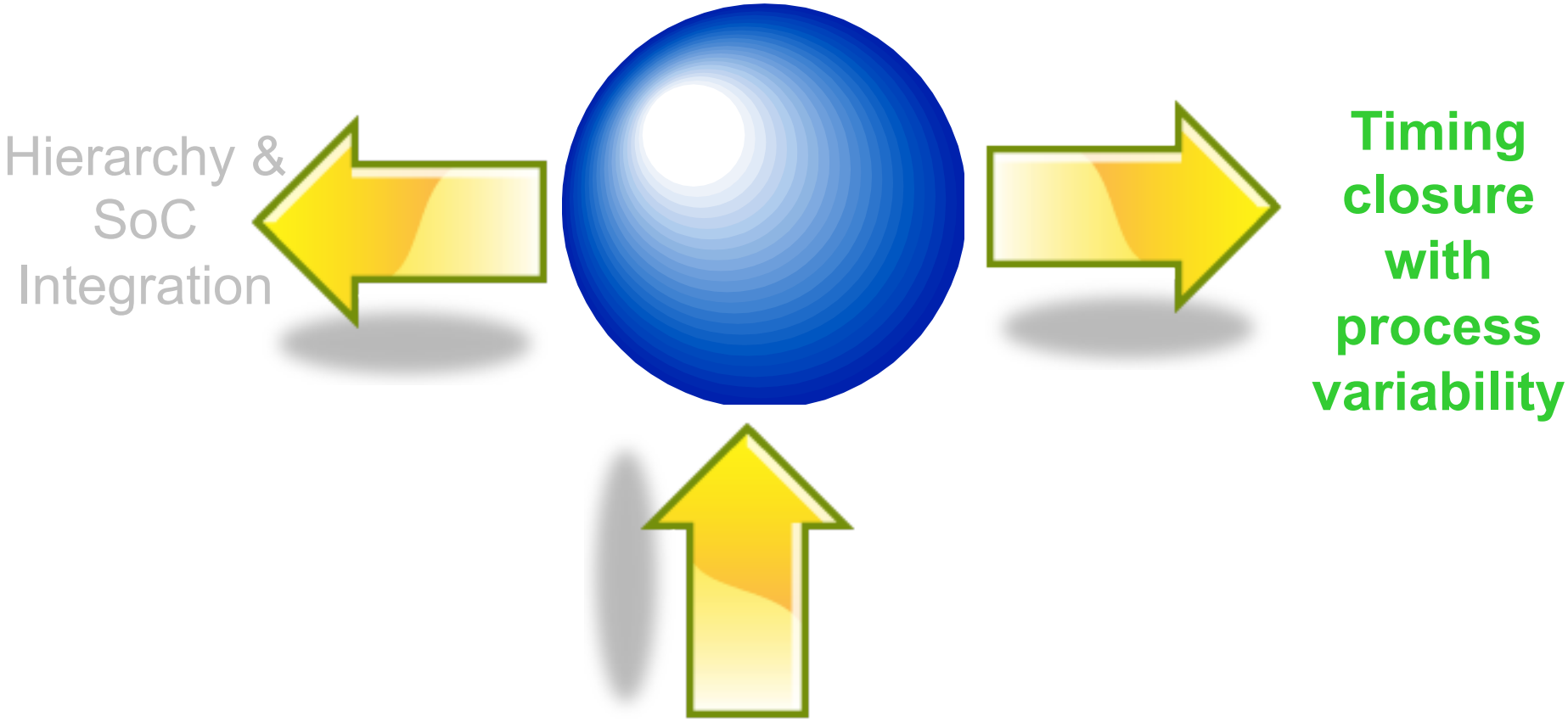
Multi threaded dynamic required time scaling



Let's go back to our earlier example

- **U = 50, S=100K nodes, R=5, C=0.2**
- **By leveraging hierarchy, we obtained:**
  - 2.5X reduction in problem size
  - Closer to a 5X improvement in TAT if all unique macros can be analyzed in parallel across multiple systems
- **Saw we have 16 core SMP machines available and we can achieve an 8X overall improvement from fine-grain parallelism**
- **Now for STA:**
  - 20X TAT improvement vs. full flat analysis (running on a single machine)
  - 40X TAT improvement vs. full flat analysis (if we distribute macro analysis across multiple machines)
  - All without relying on improvements in single threaded h/w performance
- **Multiple avenues to drive scaling**
  - As cores/machine (& memory bandwidth) increases ➔ Deploy multiple threads per job
  - As system memory (& bandwidth) increases ➔ Deploy multiple in parallel jobs per machine

Grand challenges for static timing analysis



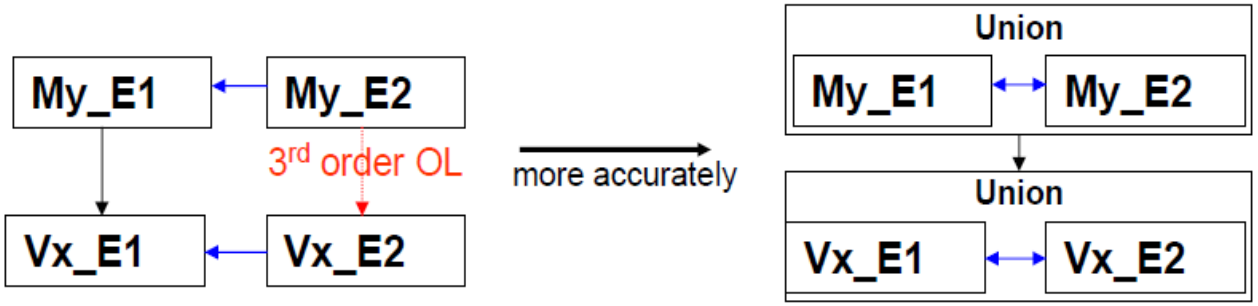
Delay impact due to variability

Parameter	Delay Impact
BEOL metal (Metal mistrack, thin/thick wires)	±15 %
Environmental (Voltage islands, IR drop, temperature)	±15 %
Device fatigue (NBTI, hot electron effects)	±10%
V_t and T_{ox} device family tracking (Can have multiple V_t and T_{ox} device families)	± 5%
Model to hardware uncertainty (Per cell type)	± 5%
N/P mistrack (Fast rise/slow fall, fast fall/slow rise)	±10%
PLL (Jitter, duty cycle, phase error)	±10%

Accept risk of failing hardware, or …
… conduct many timing runs, or …
… build in large and painful guard-bands

Double Patterning Technology (DPT): DPT-enhanced Extraction

Double Patterning introduces new 'intra-level' overlay errors

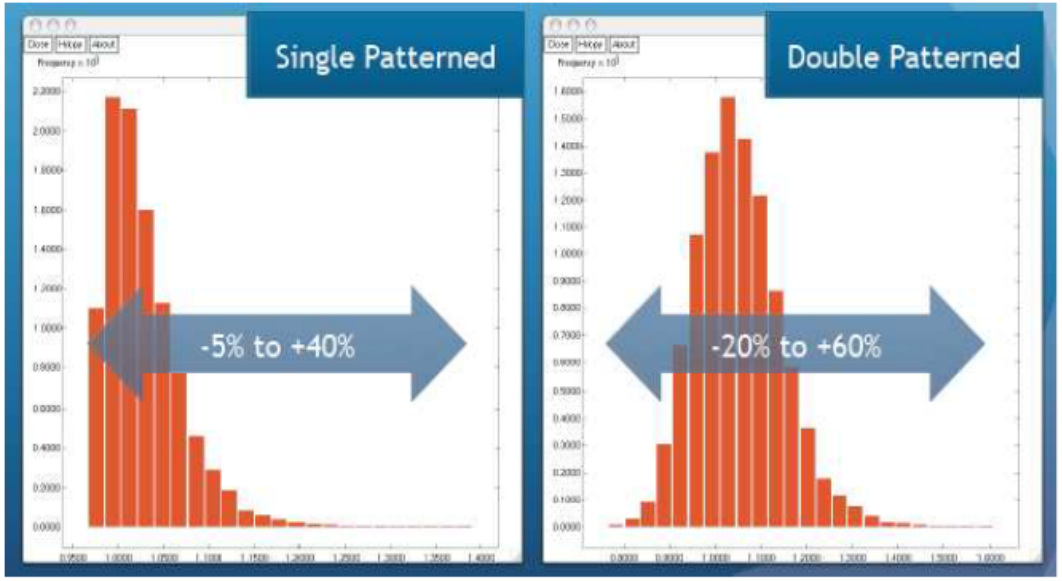


New intra-level overlay error affects electrical performance:

- accurate extraction will need to be 'color aware'
- designers will want to control color of critical circuits

Exploring RC variation of last wire in 5 wire group.

10% variation in each as applicable.

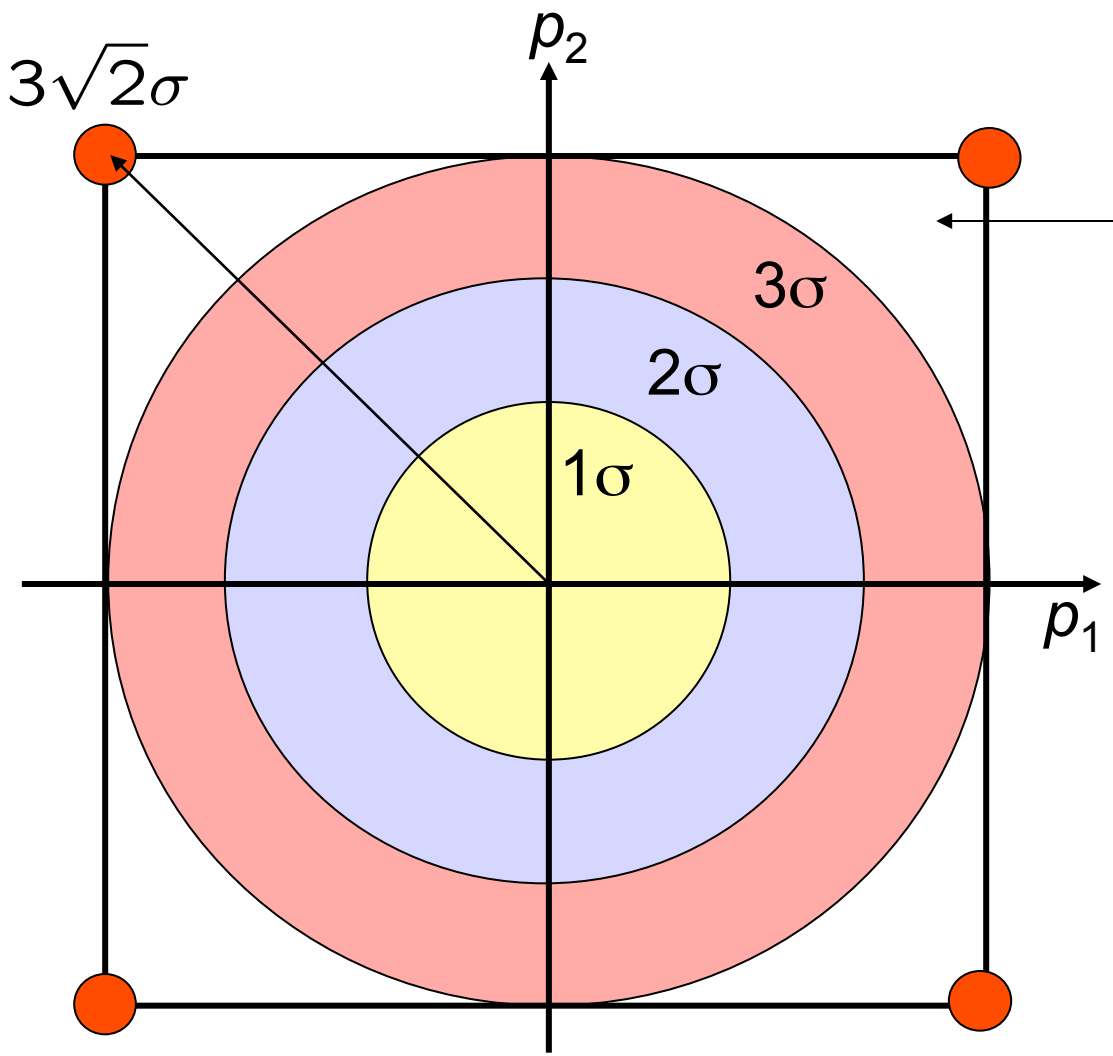


A Designer's Guide to Sub-Resolution Lithography: Enabling the Impossible to get to the 15nm Node.

Lars Liebmann (IBM), Andres Torres (Mentor Graphics). Presented at DAC 2011.

(Also based work on timing impact of double patterning by Sani Nasif, IBM Austin)

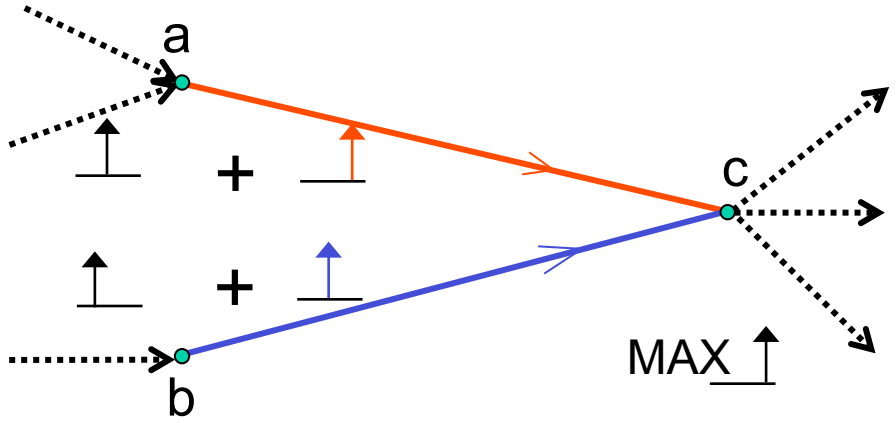
The (hyper-) sphere vs. the (hyper-) cube



- **Worst of all red corners is "exhaustive corner timing"**
- **Very low probability regions included**
  - Performance limited by the most limiting path at the most limiting corner
- **Virtually the same parametric yield can be obtained with 3σ coverage within the "circle"**
- **Statistical timing permits some parameters to be worst-cased and others to be treated statistically**

How statistical static timing analysis (SSTA) works

- **Deterministic: timing quantities are single numbers**



- **Statistical: timing quantities are probability distributions**

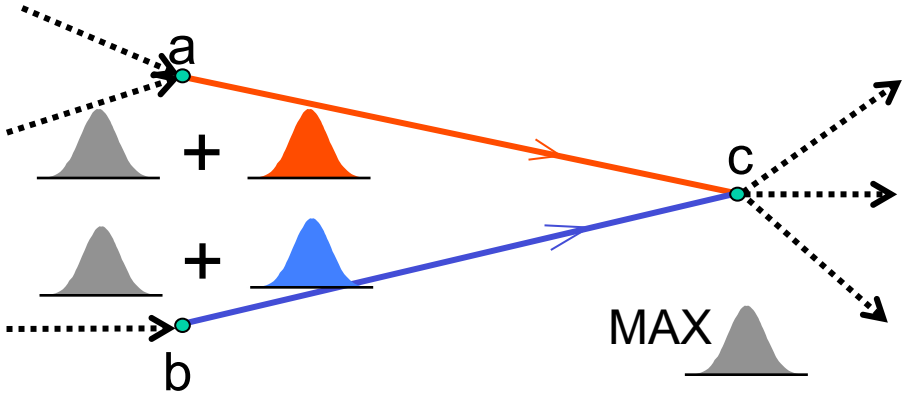


Chart courtesy Chandu Visweswariah

The first-order canonical form

$$a_0 + a_1\Delta X_1 + a_2\Delta X_2 + \cdots + a_n\Delta X_n + a_{n+1}\Delta R_a$$

Constant (nominal) value in the absence of variations

Sensitivities

Global random variables; these are probability distributions

Independently random uncertainty

- **All timing quantities computed and propagated in a parameterized form**
  - ATs, RATs, slacks, slews, guard times, PLL adjusts, delays, CPPR adjusts, coupling adjusts, assertions, …
  - The correlation between two canonical forms can be computed on-demand based on common dependence on global sources of variation
- **This is the key idea that makes statistical timing work**
  - Patent filed by IBM in September 2003

Chart courtesy Chandu Visweswariah

Timing closure needs

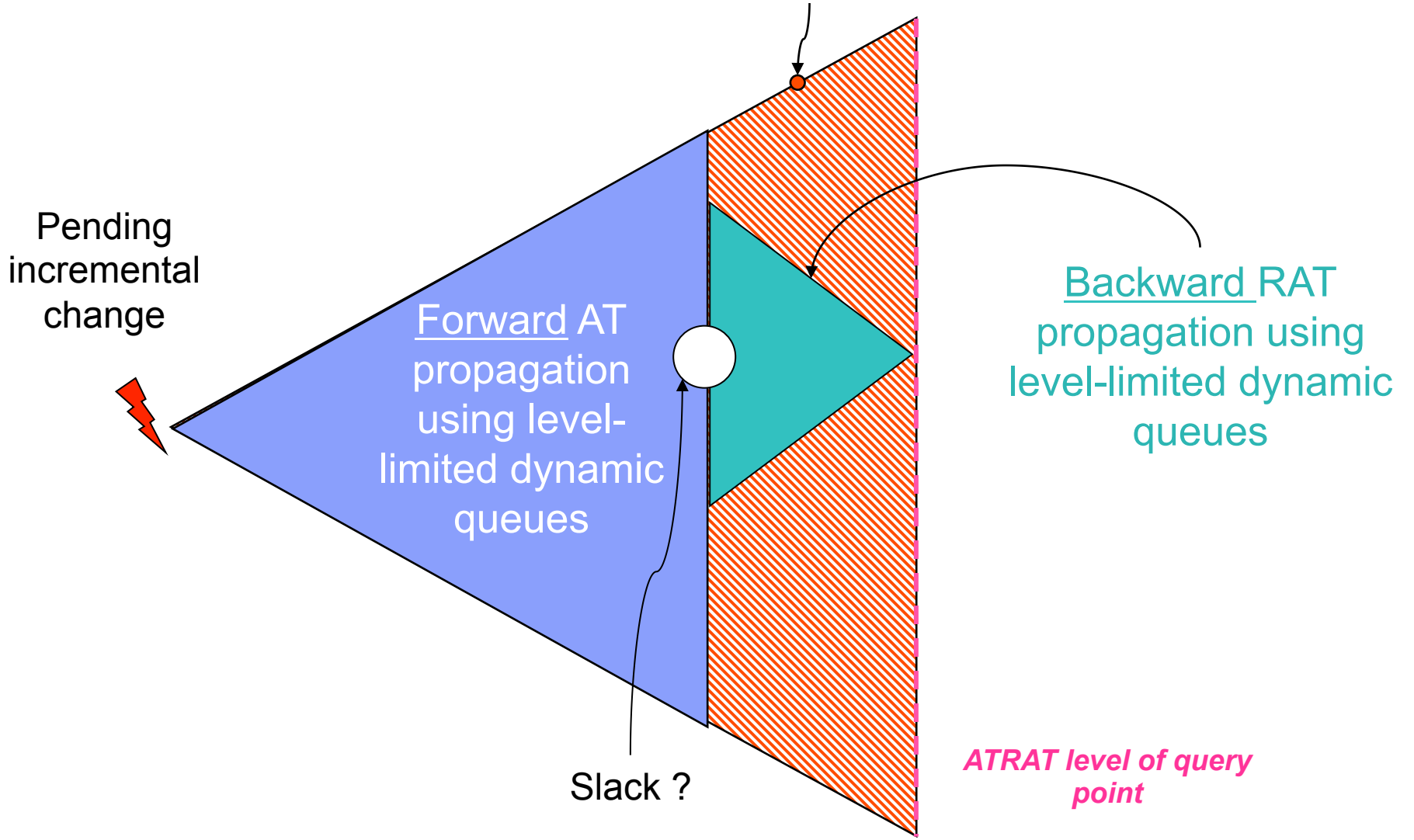
- Incremental analysis
  - Fine grain capabilities: Respond quickly in the inner loop of optimization
  - Coarse grain capabilities: Periodically redo CRPR, noise, slack steal, etc.

- Integration
  - Modular plug-and-play architecture
  - Integration into optimizers: we have a physical synthesis tool and a low-perturbation routing-aware polishing tool with our single golden timer integrated at the C++ level;

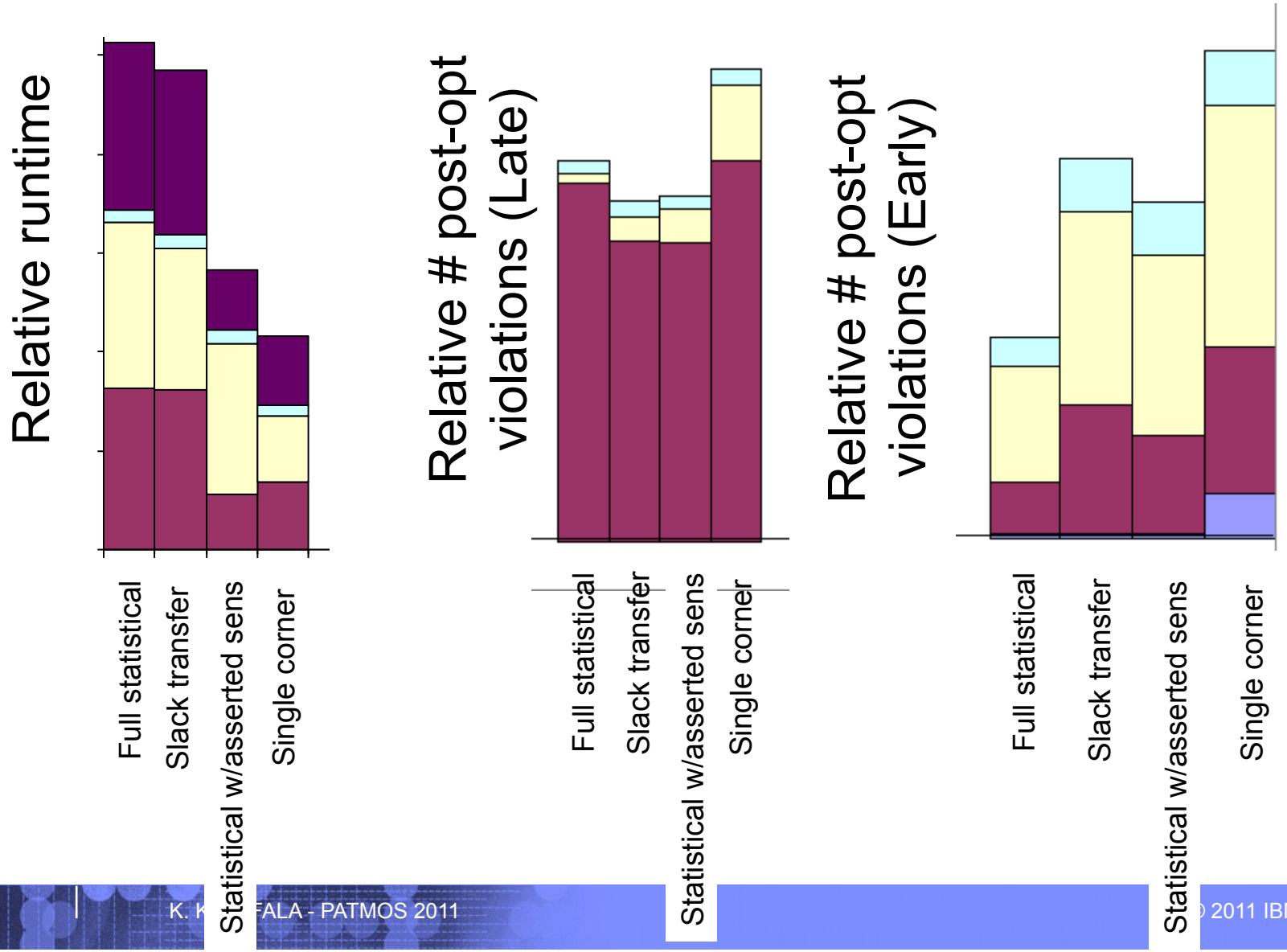
- Approximation modes
  - Sign off accuracy is expensive ➔ need cheaper modes which are less accurate, but much faster
  - Asserted vs. computed statistical sensitivities, pin vs. path slew mode, local retiming queues

Incremental dynamic multi-threaded SSTA

Additional dynamic queue processing for RAT dependency on AT



Optimization results using different approximation modes



Wrap up - Static timing analysis for microprocessor design

- Mix of both transistor and std cell design
- Multi-level hierarchy
- Use of both SOI and bulk technologies
- Domino logic
- Noise on delay and glitch analysis
- Use of clock grids, trees, and combinations thereof

Watson™

System z®

IBM POWER7 ®

Power your planet

Introducing new systems, software and solutions
for businesses of all sizes

Smarter Systems for a Smarter Planet

Wrap up - Static timing analysis in an ASIC design environment

- **ASIC timing sign off needs**
  - Statistical timing for process coverage and SVB enablement
  - Support for multiple clock domains, with frequencies in the multi-GHz
  - High speed serial link timing support
  - Support for other complex embedded IP (e.g., eDRAM)

Sampling of designs that have been successfully released through the 45nm design system

Design	Die Size (mm)	Max. Clock	SVB	Placeable Objects	# of Serdes Channels	eDRAM
Chip A	14.5x14.5	640 MHz	2-bin	7.9 M	18	Yes
Chip B	17.5x17.5	850 MHz	2-bin	15.2 M	49	Yes
Chip C	15.3x15.3	1.6 GHz	16-bin	17.9M	16	Yes
Chip D	26.7x21.8	3.0 GHz	Custom	18.4 M	54	No
Chip E	16.7x24.5	2.3 GHz	Custom	29.1 M	50	Yes

Summary

- **The post exponential era is upon us**
  - Single threaded performance gains being replaced by large scale integration of multiple cores
  - At the same time, we need to deal with increasing levels of SoC integration, impacts of process variability, etc.

- **To succeed at 22nm and below, STA must**
  - Be highly parallelizable
  - Leverage hierarchy and reuse
  - Account for process variability in a sustainable way (use SSTA)
  - Do all of the above in an incremental framework

- **(Speaker's Opinion) Any timing analyzer which cannot satisfy the above will eventually be doomed**