PART V

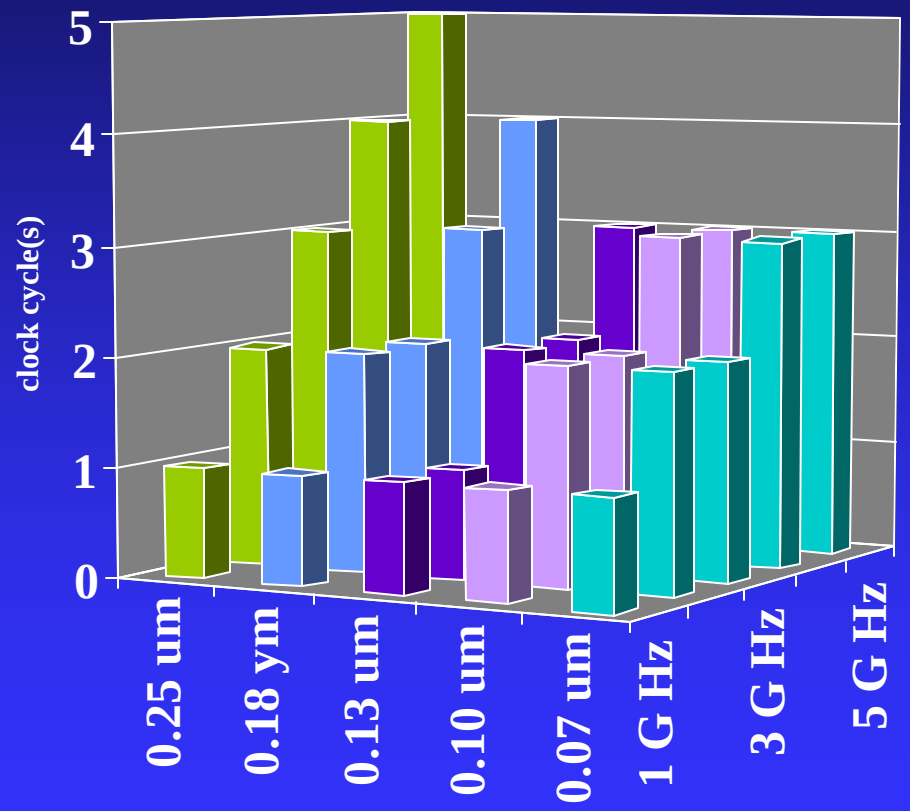
New Approaches to Harness Global Interconnects

Jason Cong
Computer Science Department
University of California at Los Angeles
Email: cong@cs.ucla.edu
Tel: 310-206-2775
http://cadlab.cs.ucla.edu/~cong

Part V Outline

- Interconnect-Centric Design Flow
- Interconnect Performance Estimation Models
  - IPEM for optimal wiresizing
  - IPEM for wiresizing and buffer insertion
- Interconnect Planning
  - Physical hierarchy generation
  - Floorplan/coarse placement with interconnect planning
  - Interconnect architecture planning
- Concluding Remarks

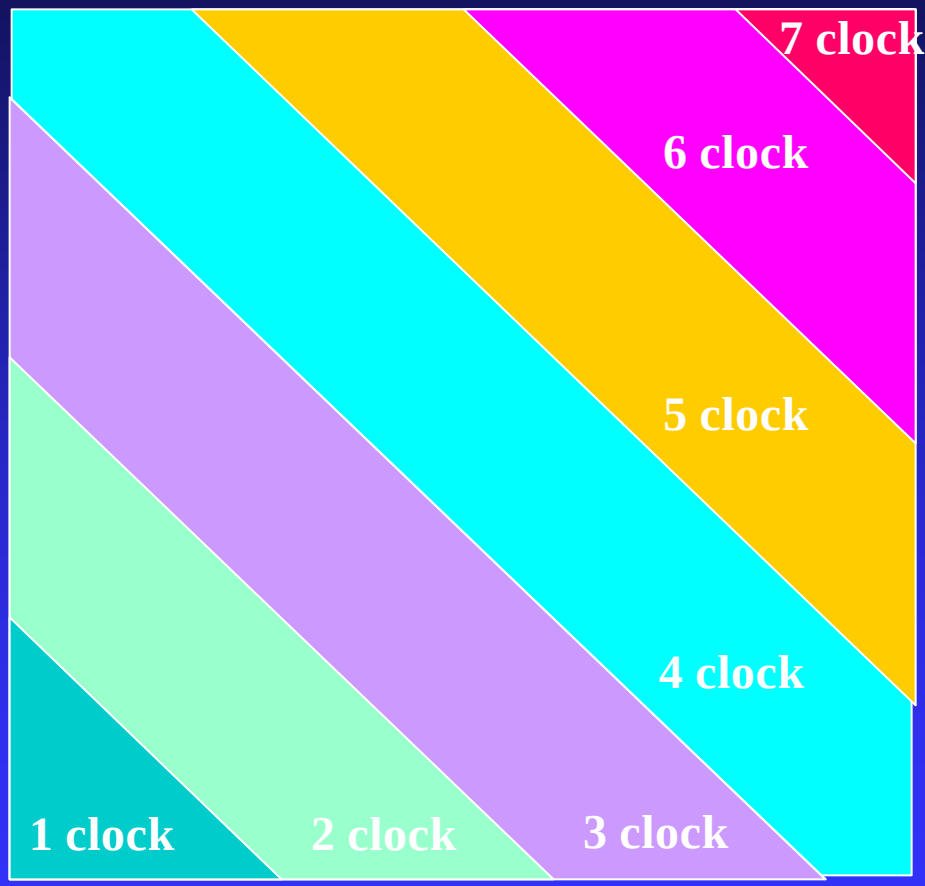
Clock cycles required for traveling 2cm line under BIWS (buffer insertion and wire sizing)



Estimated by IPEM
On NTRS'97 technology

Driver size: 100x min gate
Receiver size: 100x min gate
Buffer size: 100x min gate

How Far Can We Go in Each Clock Cycle



- NTRS'97 0.07um Tech
- 5 G Hz across-chip clock
- 620 mm^2 (24.9mm x 24.9mm)
- IPEM BIWS estimations
  - Buffer size: 100x
  - Driver/receiver size: 100x
- From corner to corner:
  - 7 clock cycles

Two Important Implications

- Interconnects determine the system performance

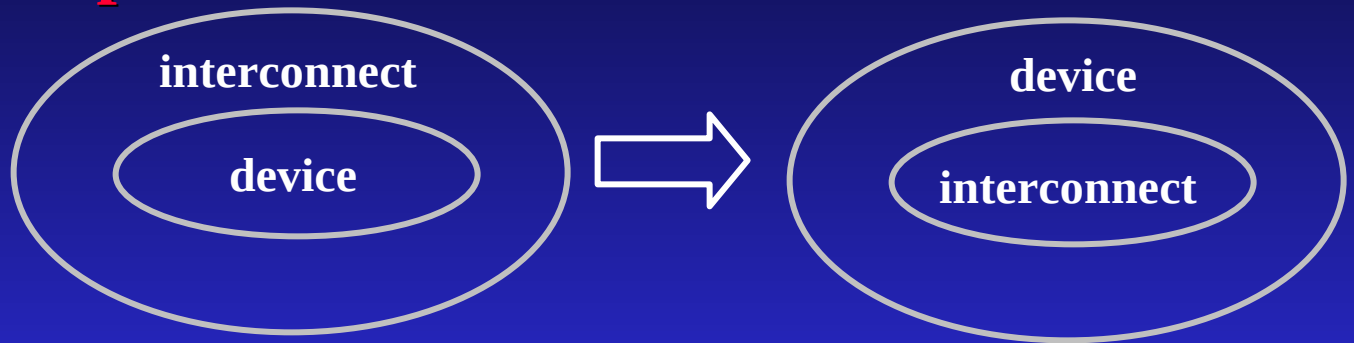
  Interconnect/communication-centric design methodology

- Need multiple clock cycles to cross the global interconnects in giga-hertz designs

  Pipelining/retiming on global interconnects

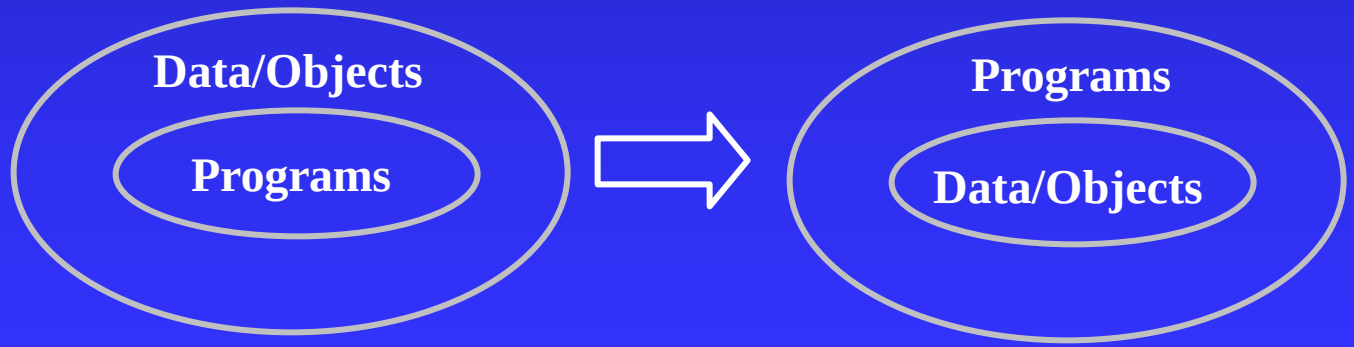
Interconnect-Centric Design Methodology

- **Proposed transition**

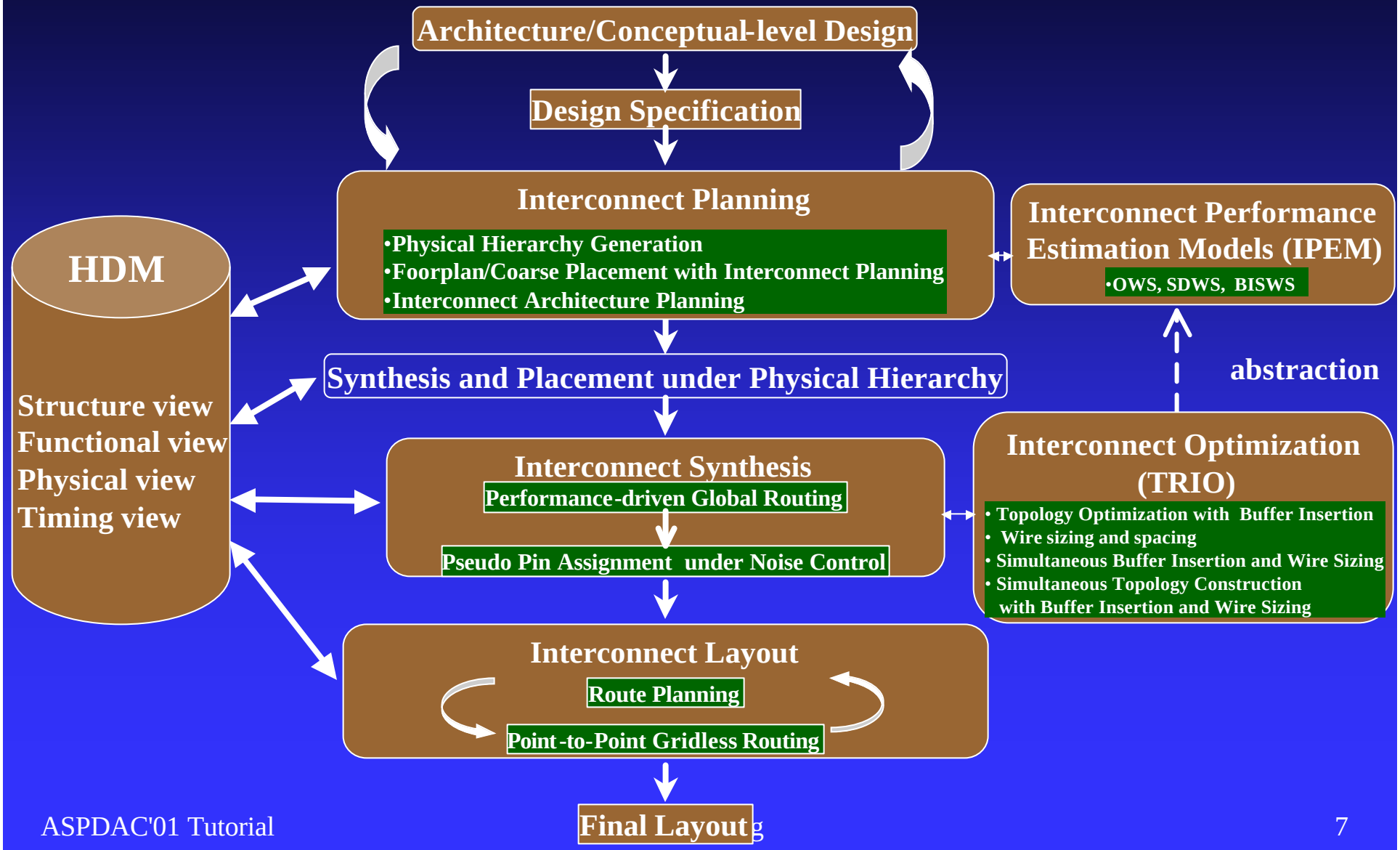


device/function centric **interconnect/communication centric**

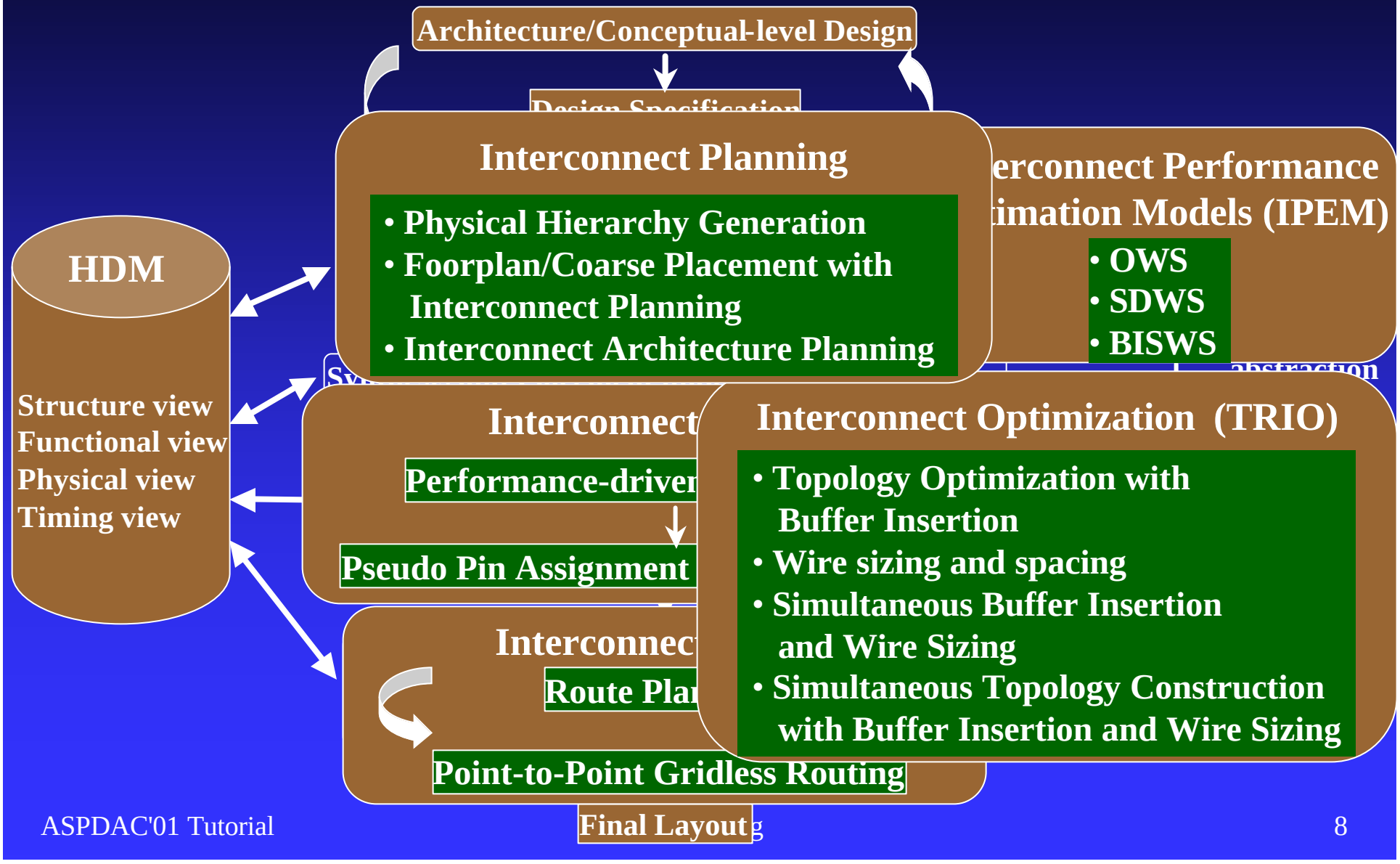
- **Analogy**



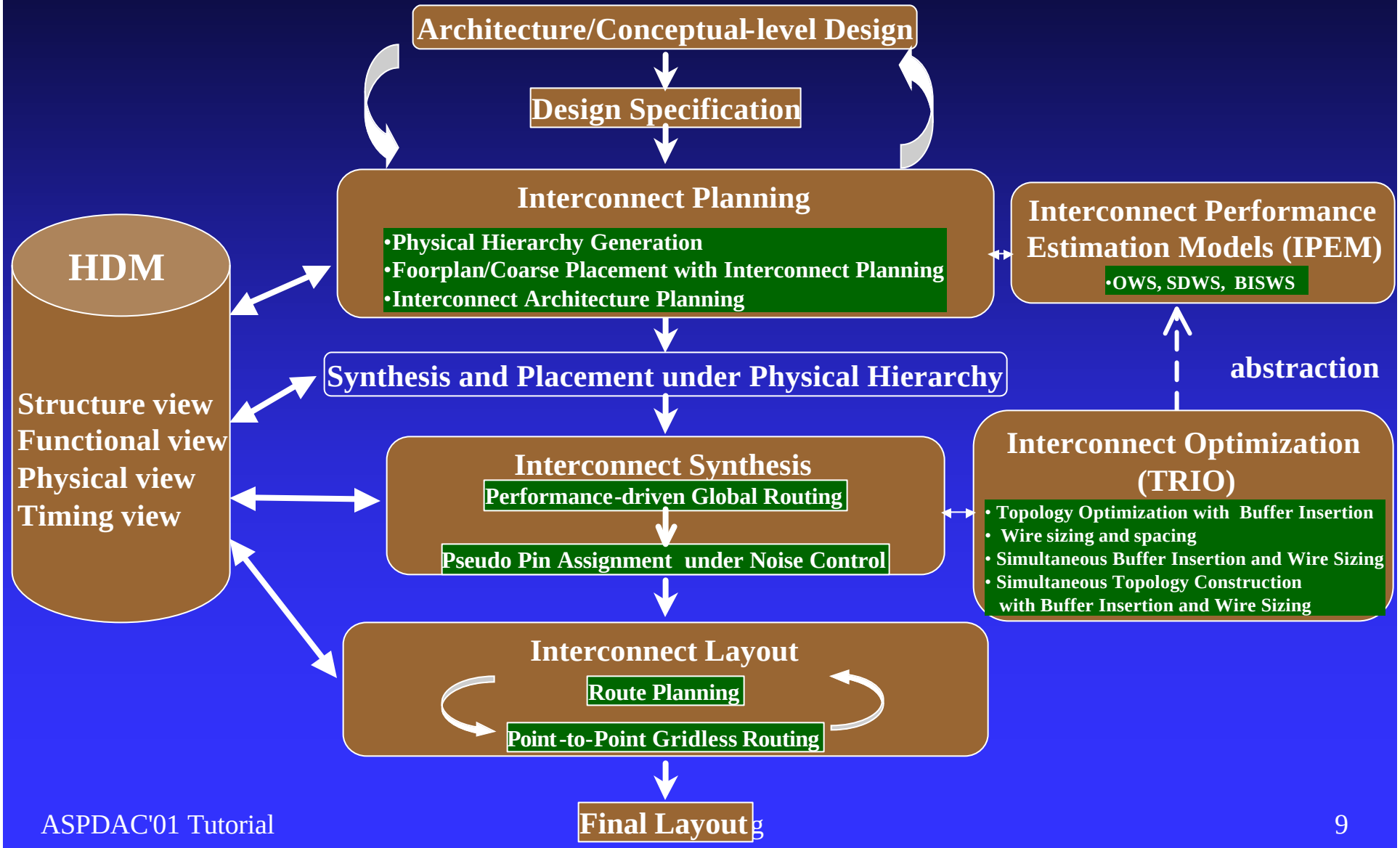
Interconnect-Centric IC Design Flow Under Development at UCLA



Interconnect-Centric IC Design Flow Under Development at UCLA



Interconnect-Centric IC Design Flow Under Development at UCLA



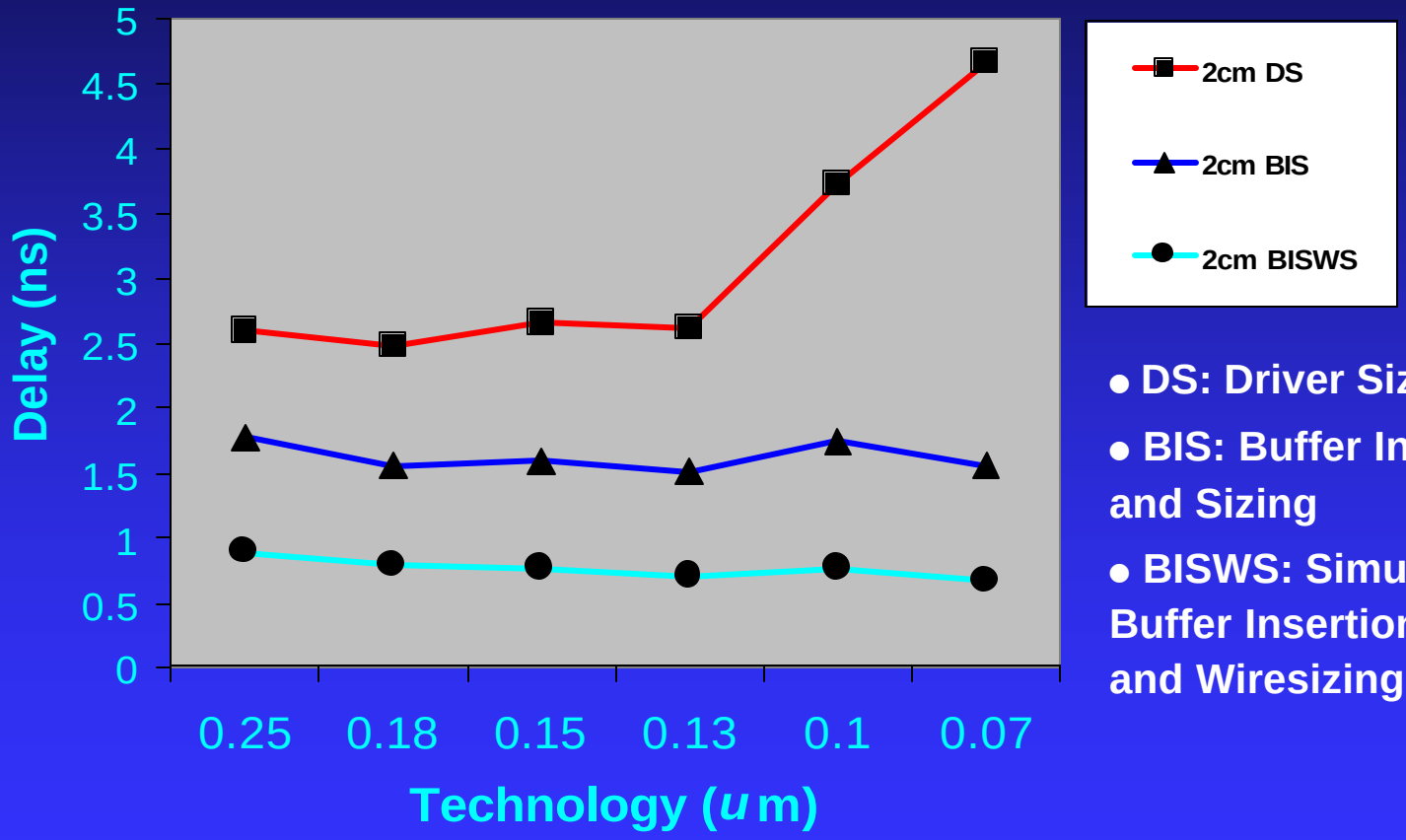
Part V Outline

- Interconnect-Centric Design Flow
- Interconnect Performance Estimation Models
  - IPEM for optimal wiresizing
  - IPEM for wiresizing and buffer insertion
- Interconnect Planning
  - Physical hierarchy generation
  - Floorplan/coarse placement with interconnect planning
  - Interconnect architecture planning
- Concluding Remarks

Interconnect Performance Estimation

- Introduction & Motivation
- Problem Formulation
- Interconnect Delay Estimation Models under Various Layout Optimizations
- Application and Conclusion

Impact of Interconnect Optimization on Future Technology Generations



- DS: Driver Sizing only
- BIS: Buffer Insertion and Sizing
- BISWS: Simultaneous Buffer Insertion/Sizing and Wiresizing

Complexity of Existing Interconnect Opt. Algorithms

- 2cm line, W=20, B=10, segment every 500um
- Use best available algorithms:
  - Local Refinement (LR)
  - Dynamic Programming (DP)
  - Hybrid of DP+LR

LR DP+LR DP

Algorithm	OWS	BI+OWS	BIWS	BISWS
Delay (ns)	4.5	1.6	1.02	0.81
CPU (s)	0.06	0.42	4.5	12.4

(HSPICE needs additional 60 seconds!)

Needs for Efficient Interconnect Estimation Models

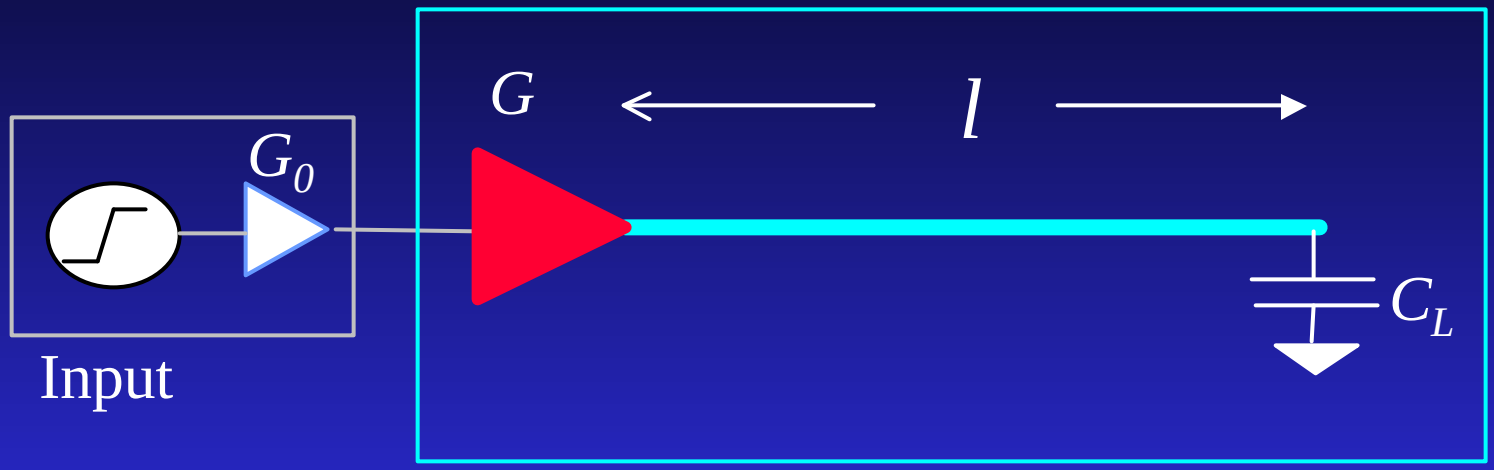
- **Efficiency**
- **Abstraction to hide detailed design information**
  - **granularity of wire segmentation**
  - **number of wire widths, buffer sizes, ...**
- **Explicit relation to enable optimal design decision at high levels**
- **Ease of interaction with logic/high level synthesis tools**

Interconnect Performance Estimation Modeling

[Cong-Pan, ASPDAC'99, TAU'99, DAC'99]

- Develop a set of interconnect performance estimation models (IPEM), under different optimization alternatives:
  - Optimal Wire Sizing (OWS)
  - Simultaneous Driver and Wire Sizing (SDWS)
  - Simultaneous Buffer Insertion and Wire Sizing (BIWS)
  - Simultaneous Buffer Insertion/Sizing and Wire Sizing (BISWS)
- IPEM have
  - closed-form formula or simple characteristic equations
  - constant running time in practice
  - high accuracy (about 90% accuracy on average)

Problem Formulation



- R_{d0} driver effective resistance of the input stage G_0
- R_d driver effective resistance of G
- l interconnect wire length
- C_L loading capacitance

➡ **What is the optimized delay?**

Do not run TRIO or other optimization tools !

Parameters and Notations

- **Interconnect**
  - c_a area capacitance coefficient
  - c_f fringing capacitance coefficient
  - r sheet resistance
- **Device**
  - t_g intrinsic gate delay
  - c_g input capacitance of the minimum gate
  - r_g output resistance of the minimum gate
- **Based on 1997 National Technology Roadmap for Semiconductors (NTRS'97)**

Delay/Area Estimation under OWS

- **Closed-form delay estimation formula**

$$T_{ows}(R_d, l, C_L) = \left[\frac{a_1 l}{W^2(a_2 l)} + \frac{2a_1 l}{W(a_2 l)} + R_d c_f + \sqrt{R_d r c_a c_f l} \right] \cdot l$$

where

$$a_1 = \frac{1}{4} r c_a , \quad a_2 = \frac{1}{2} \sqrt{\frac{r c_a}{R_d C_L}}$$

$W(x)$ is Lambert's W function defined as $we^w = x$

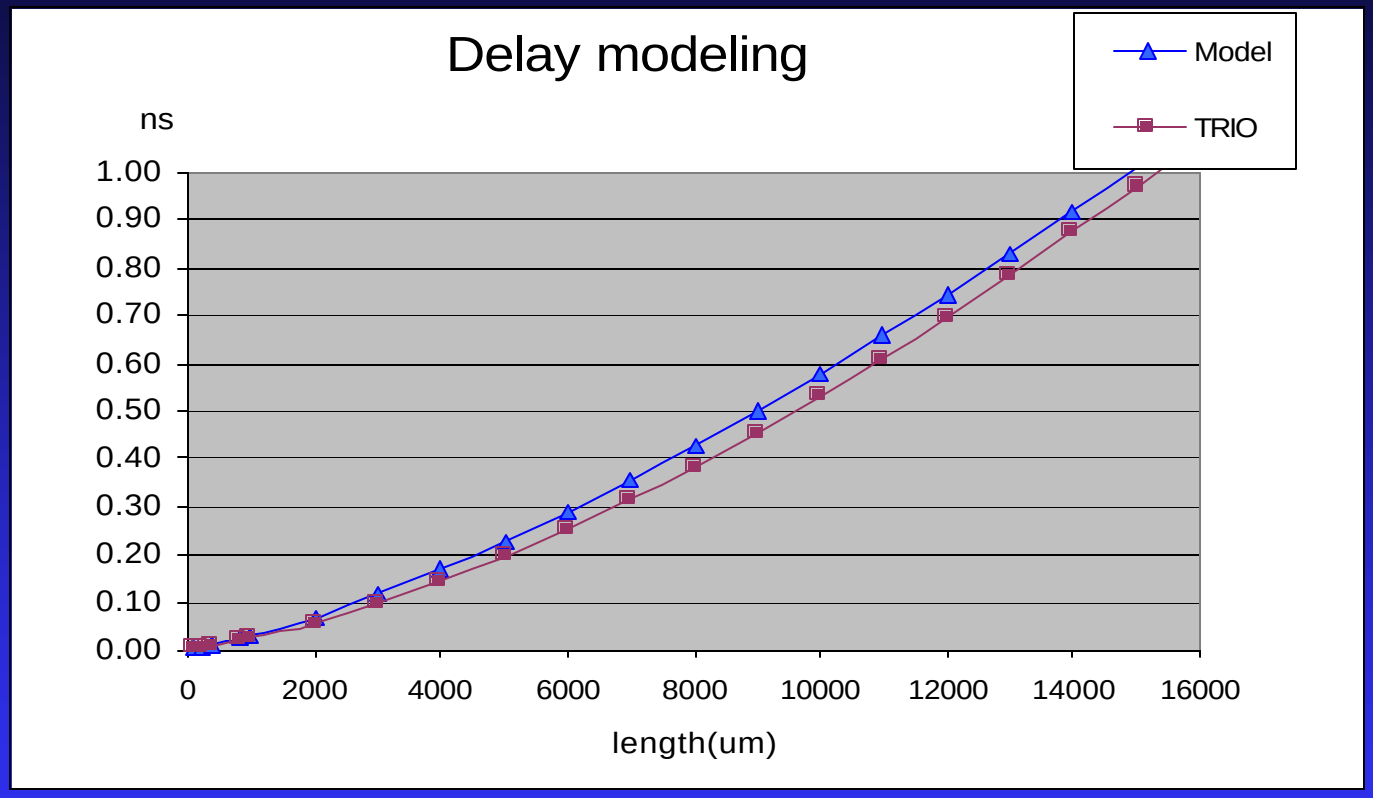
- **Closed-form area estimation formula**

$$A_{ows}(R_d, l, C_L) = \sqrt{\frac{r(c_f l + 2C_L)}{2R_d c_a}} \cdot l$$

Property of DEM-OWS

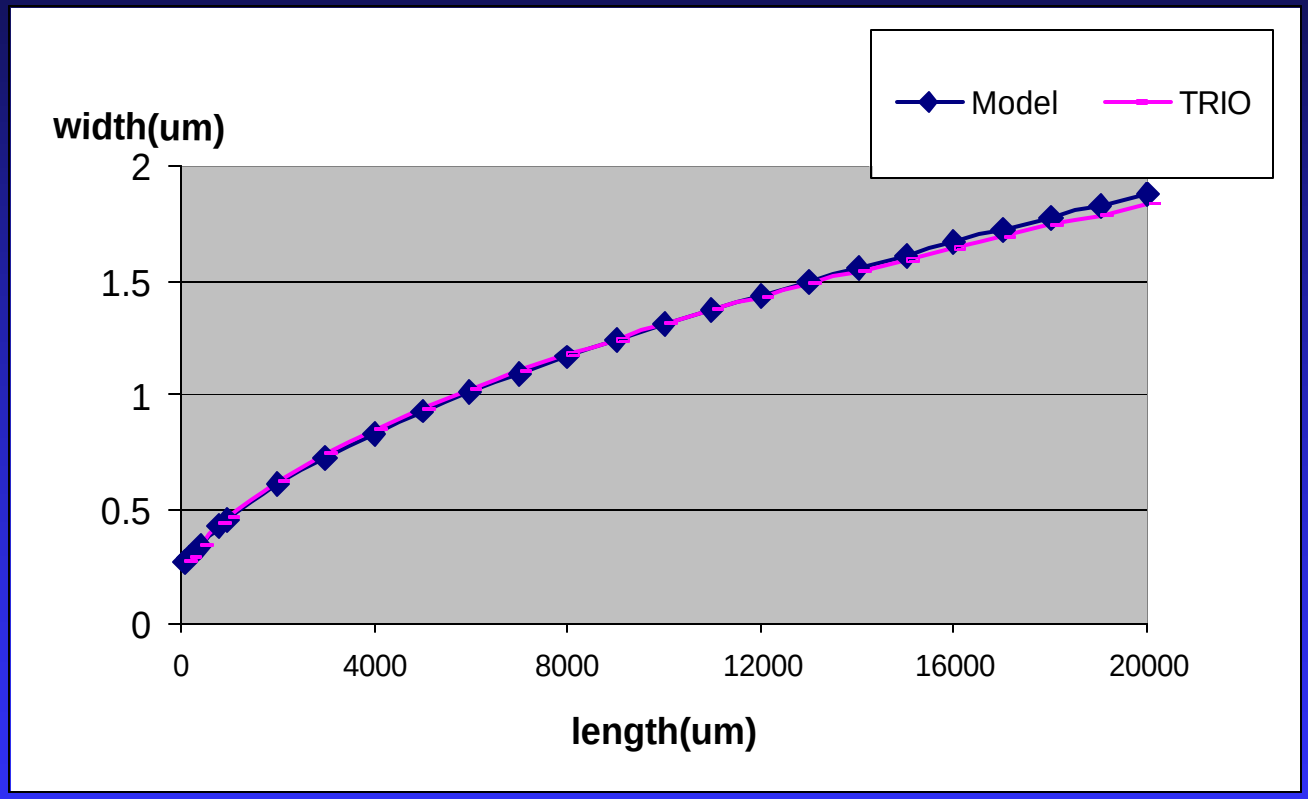
- **Theorem: T_{ows} is a sub-quadratic, convex function of length l**
- **Note: Without wiresizing, wiring delay μ l^2, as used in some previous layout-driven logic synthesis systems, such as [Ramachandran et al., ICCAD-92], is no longer accurate!**
- **Closed-form DEM-OWS will serve as a basis for deriving SDWS, BIWS and BISWS**

Comparison of IPEM-OWS vs. TRIO



- **0.18um, $R_d = r_g/100$, $C_L = c_g$ x 100**
- **For expt., max wire width is 20x min, wire is segmented in every 10um**

Area Estimation for OWS



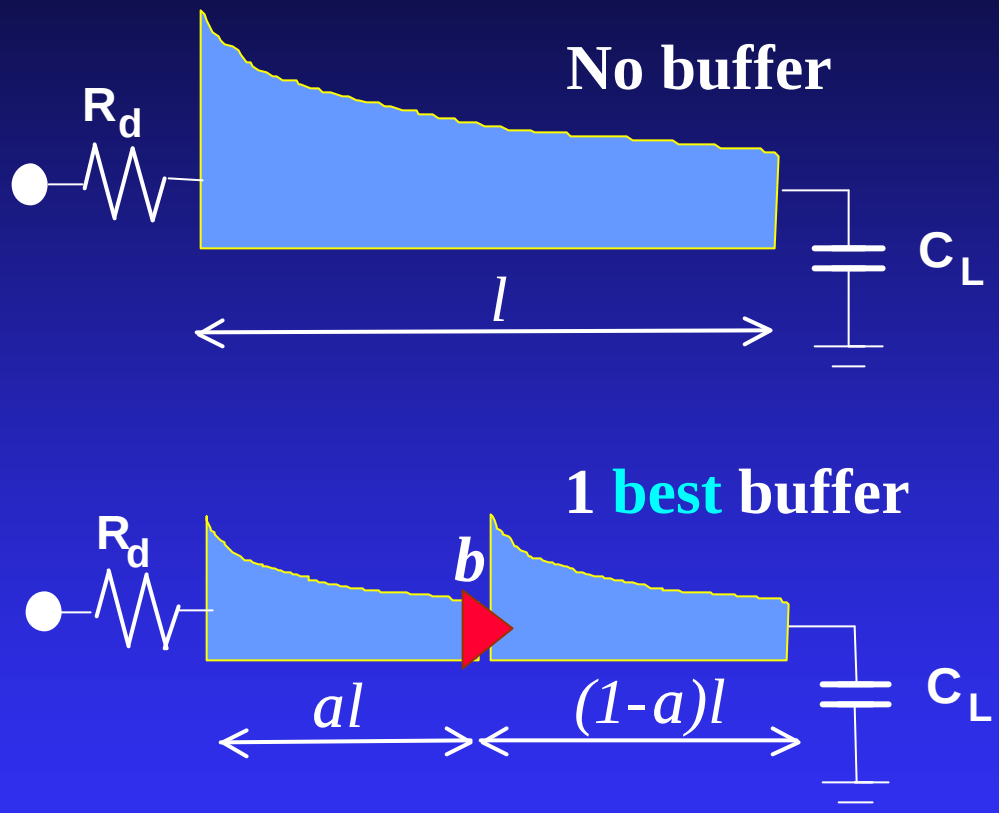
Critical Length for BI under OWS

$T_{ows}(R_d, l, C_L)$

Solve for l, => critical length l_{crit} (b, R_d, C_L)

- Computed by bisection method

- Constant time in practice



$$T_{1biws}(R_d, l, C_L) = \min_{0 \le a \le 1} \{T_{ows}(R_d, al, C_b) + t_g + T_{ows}(R_b, (1-a)l, C_L)\}$$

Critical Lengths $l_{crit}(b, R_b, C_b)$

Decrease

Technology (um)	*0.25*	*0.18*	*0.15*	*0.13*	*0.10*	*0.07*
b=10x	4.12	3.80	3.97	3.61	2.92	2.08
b=50x	6.40	5.81	6.01	5.51	4.45	3.30
b=100x	7.47	6.83	7.04	6.39	5.30	3.91
b=200x	8.65	7.92	8.14	7.43	6.35	4.49
b=500x	9.98	9.10	9.30	8.57	7.13	5.21

unit: mm

Min. WS	2.52	2.23	2.14	1.94	1.50	1.43

- Cf. [Otten ISPD'98, Otten-Brayton DAC'98] (uniform wire width)
- Denote $l_c = l_{crit}(b, R_b, C_b)$

"Logic Volume" within l_c

- Defined as the number of min 2-input NAND gates that can be packed within the area of $l_c/2 * l_c/2$

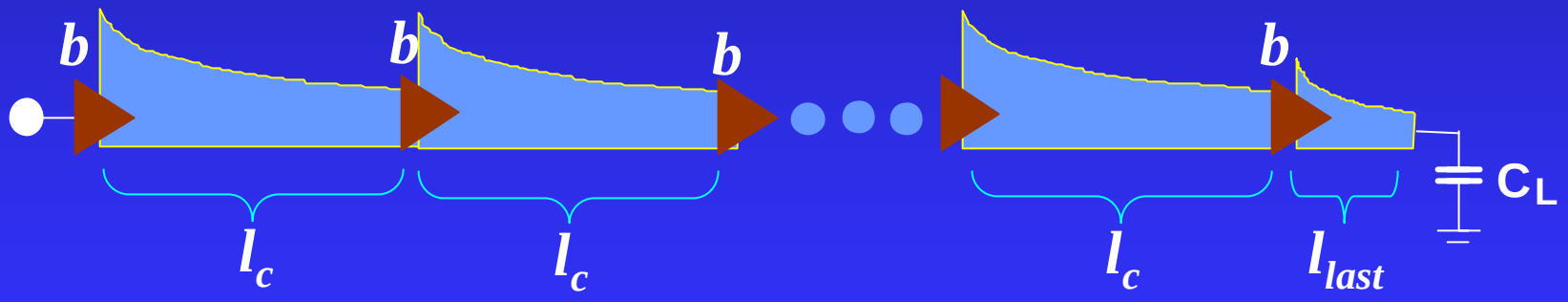
Technology (um)	*0.25*	*0.18*	*0.15*	*0.13*	*0.10*	*0.07*
2-NAND (um^2)	7.80	4.04	3.00	2.18	1.28	0.64
b=10x	0.55	0.89	1.31	1.49	1.66	1.69
b=50x	1.31	2.09	3.01	3.48	3.87	4.25
b=100x	1.79	2.88	4.13	4.68	5.48	5.97
b=200x	2.4	3.88	5.52	6.33	7.87	7.88
b=500x	3.19	5.12	7.21	8.42	9.93	10.6

unit: million

Increase

Property of BIWS

- **Theorem:** For BIWS, the distances between adjacent buffers are the same, and equal to l_c -- the critical length.
- **Proof**: based on the convexity of T_{ows}



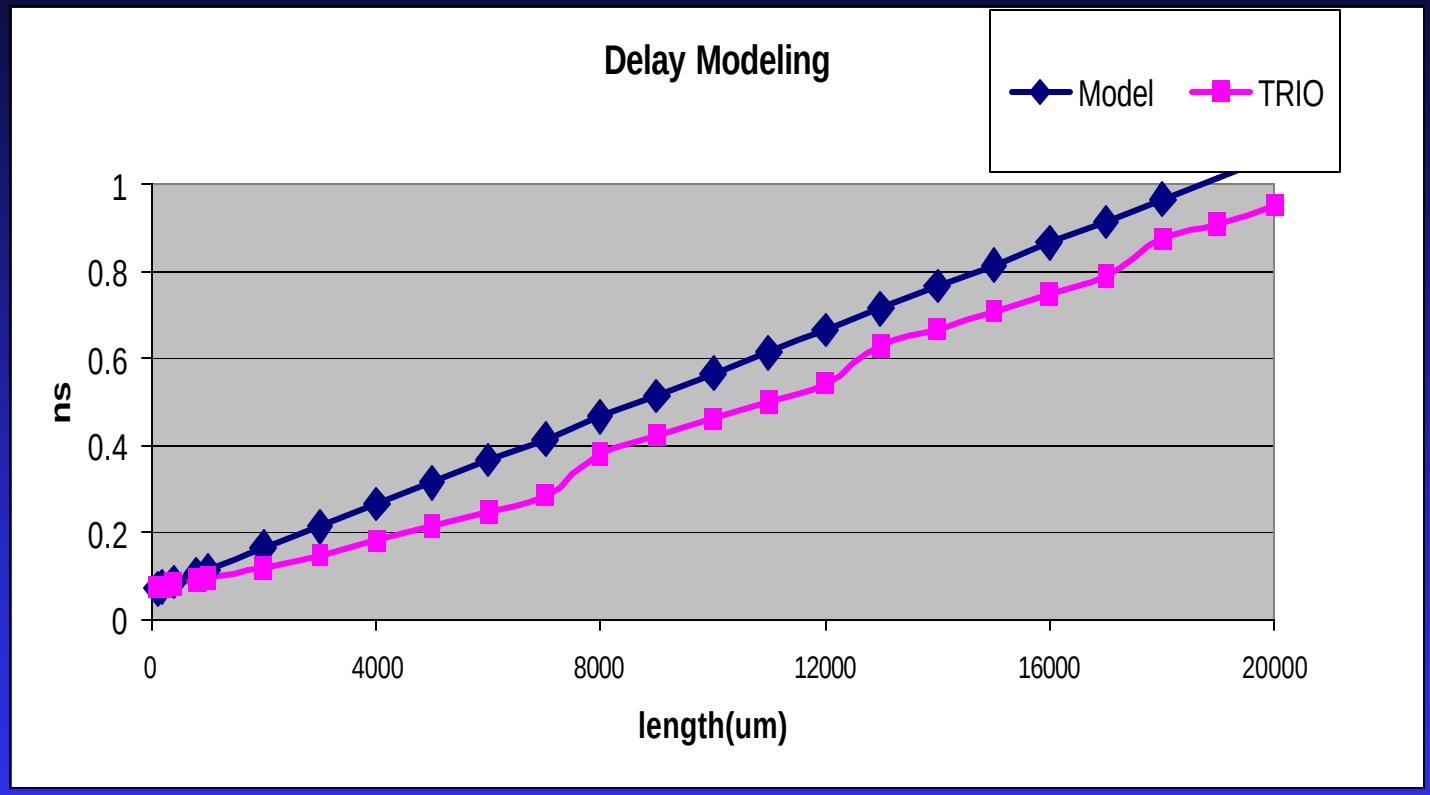
IPEM for BIWS

- Original long interconnect is divided into él/l_cù stage
- The stage number is proportional to l
- Each stage of length l_c has delay $T_{ows}(R_b, l_c, C_b)$
- ➡ Linear DEM for BIWS

$$T_{biws} = t_{biws} \cdot l + t_g$$

t_{biws} is the slope, and can be obtained from $T_{ows}(R_b, l_c, C_b)$

IPEM for BIWS vs. TRIO



- *0.18um,* $R_{d0} = r_g/10$, $C_L = c_g$ x *10,* **buffer type is 100 x min.**
- **For expt., max. wire width is 20x min. width, wire is segmented in every 100um.**

IPEM under BISWS

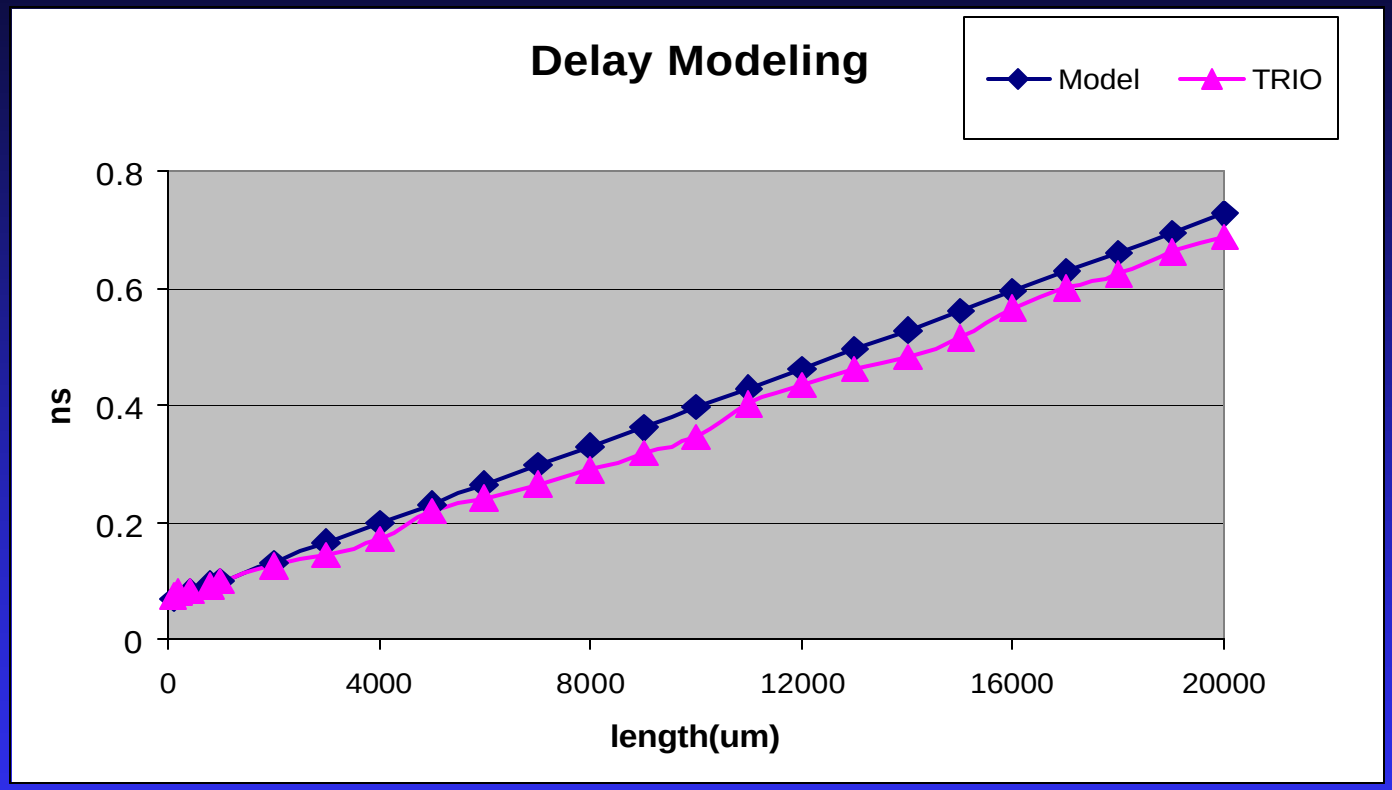
- Observations from extensive experiments:
  - Linear delay versus length
  - Internal buffers are about the same size
- Therefore, we estimate BISWS by the best BIWS from available buffer types
- Linear delay model for optimal BISWS

$$T_{bisws} = t_{bisws} \cdot l + t_g$$

where $t_{bisws} = \min_{b \in B} t_{biws}$, B is the buffer set

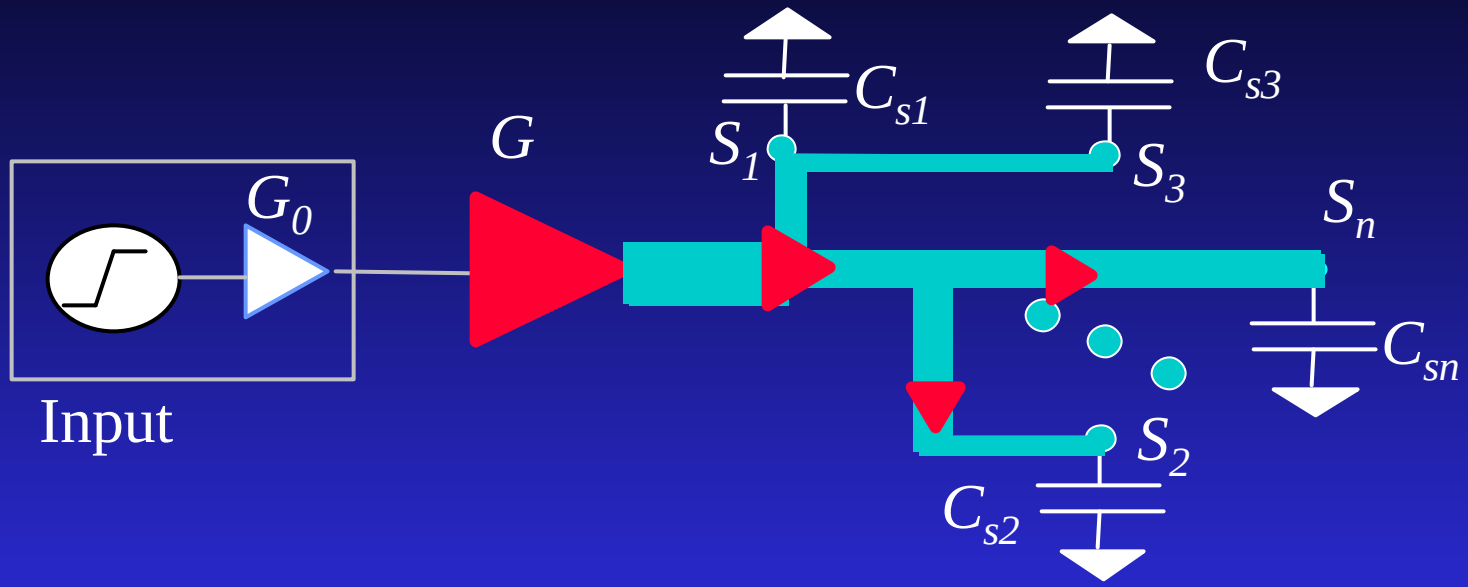
- Complexity O(|B|). Since the set B is normally less than 20, constant time in practice.

Comparison of IPEM for BISWS vs. TRIO



- *0.18um, $R_{d0} = r_g/10$, $C_L = c_g$ x 10*
- **For expt., max. allowable buffer/driver size is 400x min device; max. wire width is 20x min. width; wire is segmented in every 100um.**

IPEM for Multiple-Pin Nets



- **Estimation with different optimization objectives:**
  - **Minimize the delay to a single critical sink (SCS)**
  - **Minimize the maximum delay (defined as the tree delay) for multiple critical sinks (MCS)**
  - **Minimize weighted delay ...**

Some Applications of IPEM

- Layout-driven physical and RTL level floorplanning
  - Predict accurate interconnect delay and routing resource without really going into layout details;
  - Use accurate interconnect delay/area to guide floorplanning/placement
- Interconnect Architecture Planning
  - E.g. Wire width planning
- Floorplanning + interconnect planning
  - E.g. Buffer block planning
- Available from http://cadlab.cs.ucla.edu/~cong

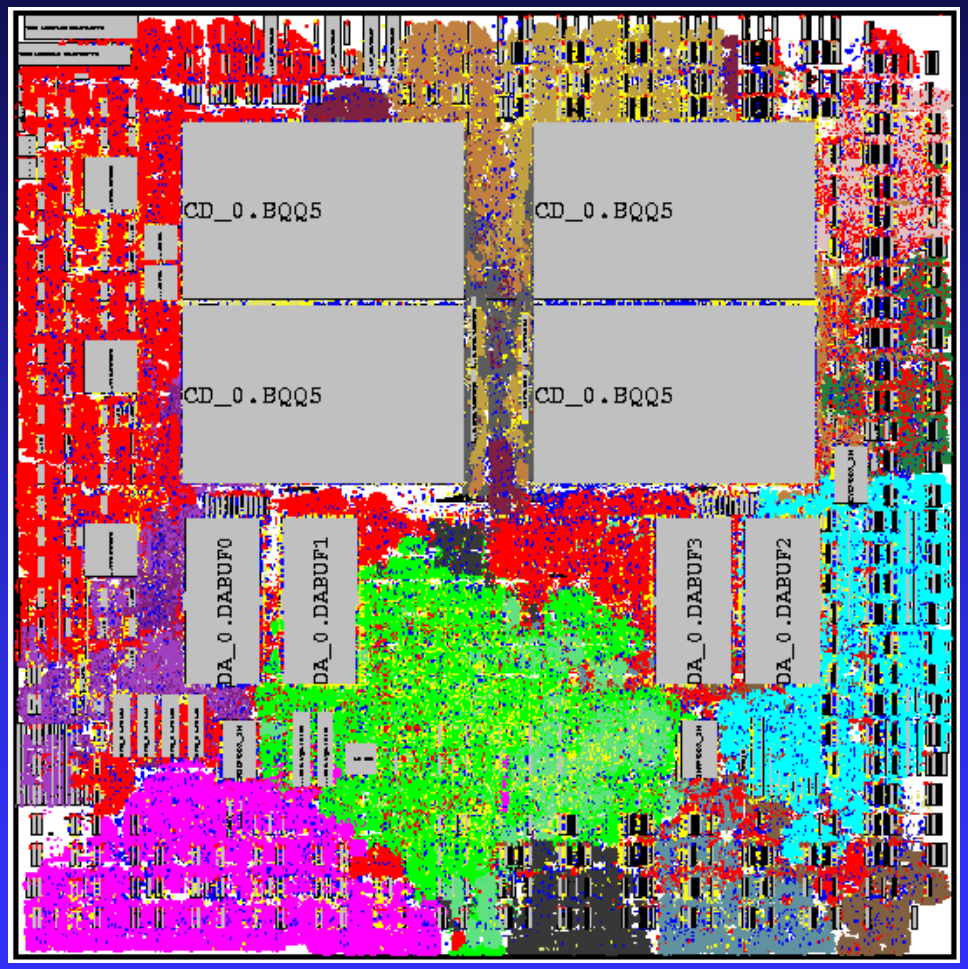
Part V Outline

- Interconnect-Centric Design Flow
- Interconnect Performance Estimation Models
  - IPEM for optimal wiresizing
  - IPEM for wiresizing and buffer insertion
- Interconnect Planning
  - Physical hierarchy generation
  - Floorplan/coarse placement with interconnect planning
  - Interconnect architecture planning
- Concluding Remarks

Physical Hierarchy Generation

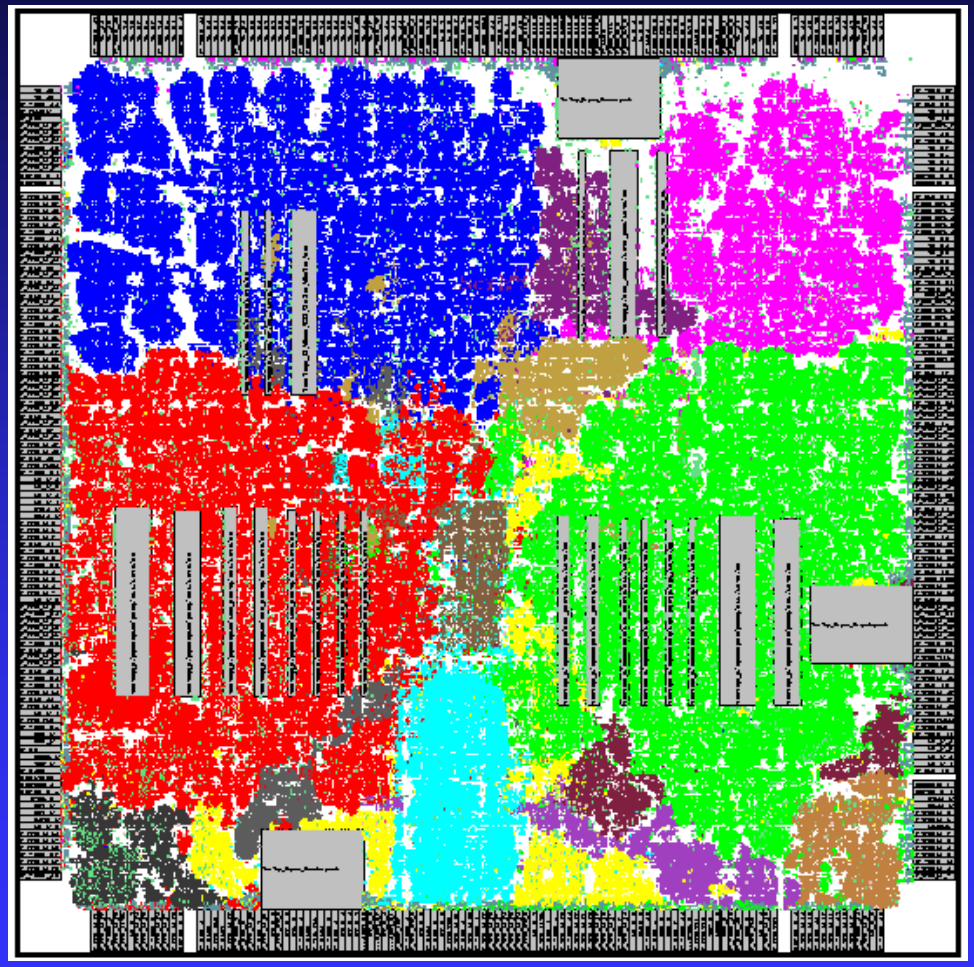
- Designs are hierarchical due to high complexity
- Design specification (in HDL) follows logic hierarchy
- Logic hierarchy may not be suitable to be embedded on a 2D silicon surface, resulting poor interconnect designs
  - RT-level floorplanning is a bad idea!
- Solution: transform logic hierarchy to physical hierarchy

Example of Logic Hierarchy in Final Layout



By courtesy of IBM (Tony Drumm)

Example of Logic Hierarchy in Final Layout



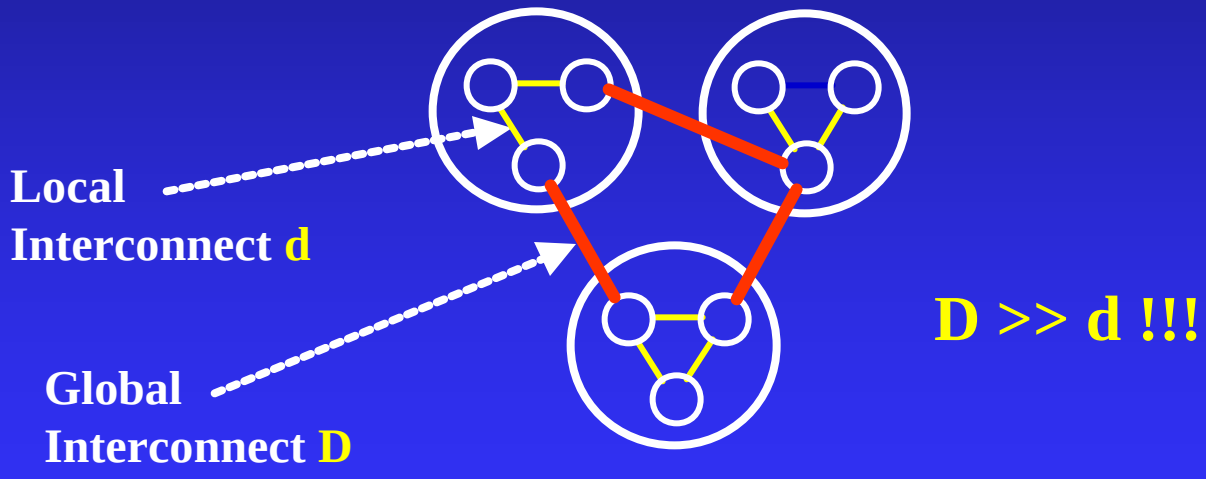
By courtesy of IBM (Tony Drumm)

Transform Logic Hierarchy to Physical Hierarchy

- Simultaneous partitioning, coarse placement, and retiming on the *flat* netlist to generate a good physical hierarchy
  - Synthesis will follow
- Use multi-level optimization to handle with the complexity

Role of Partitioning

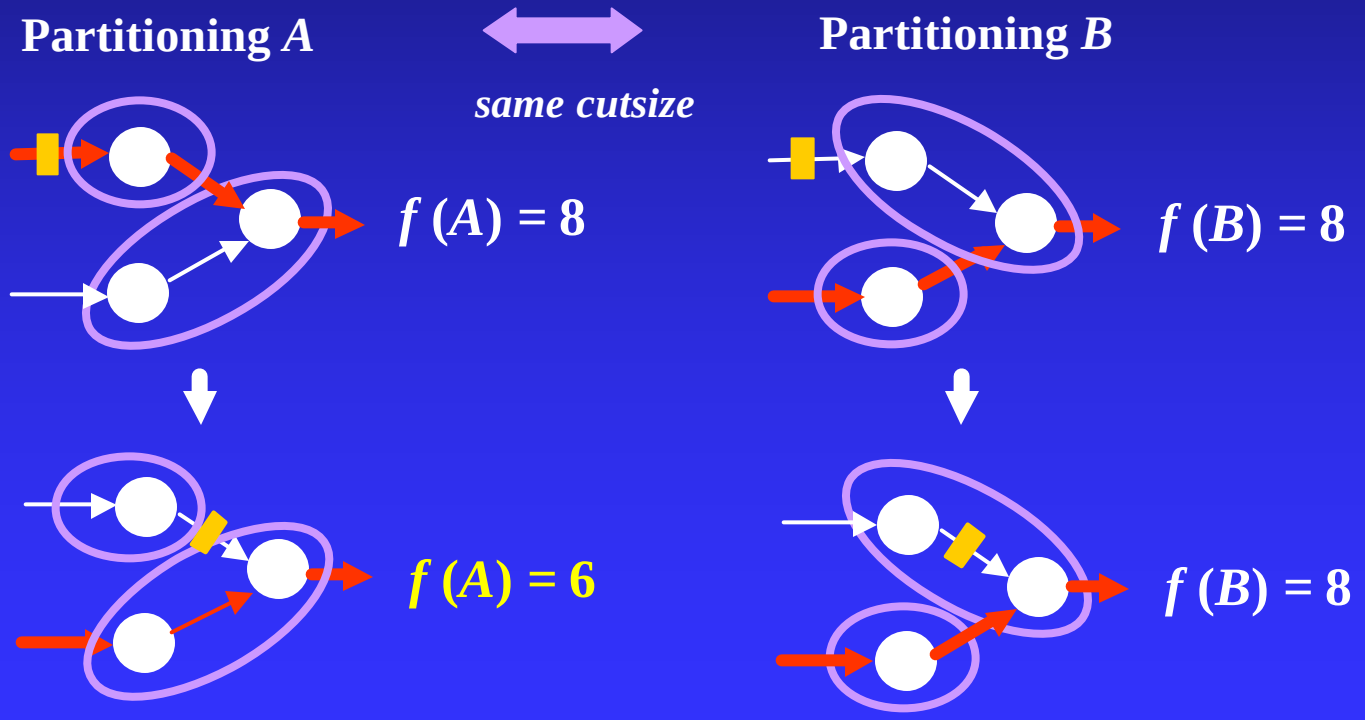
- **Importance of Partitioning:**
  - **Conventional view: enables divide-and-conquer**
  - **DSM view: defines global and local interconnects**



Need of Considering Retiming during Partitioning

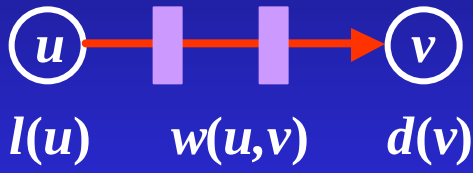
- Retiming/pipelining on global interconnects

- Multiple clock cycles are needed to cross the chip
- Proper partitioning allows retiming to hide global interconnect delays.

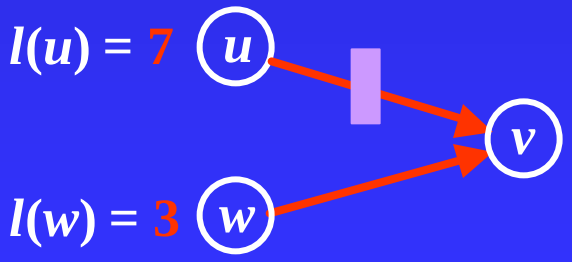


Sequential Arrival Time (SAT)

- **Definition [Pan et al, TCAD98]**
  - $l(v)$ = max delay from PIs to v after opt. retiming under a given clock period f
  - $l(v) = \max\{l(u) - f \cdot w(u,v) + d(u,v) + d(v)\}$



  - Relation to retiming: $r(v) = \lceil l(v) / f \rceil - 1$
  - Theorem: P can be retimed to f + max$\{d(e)\}$ iff l(POs) $\leq f$



$d(v) = 1, d(e) = 2, f = 5$

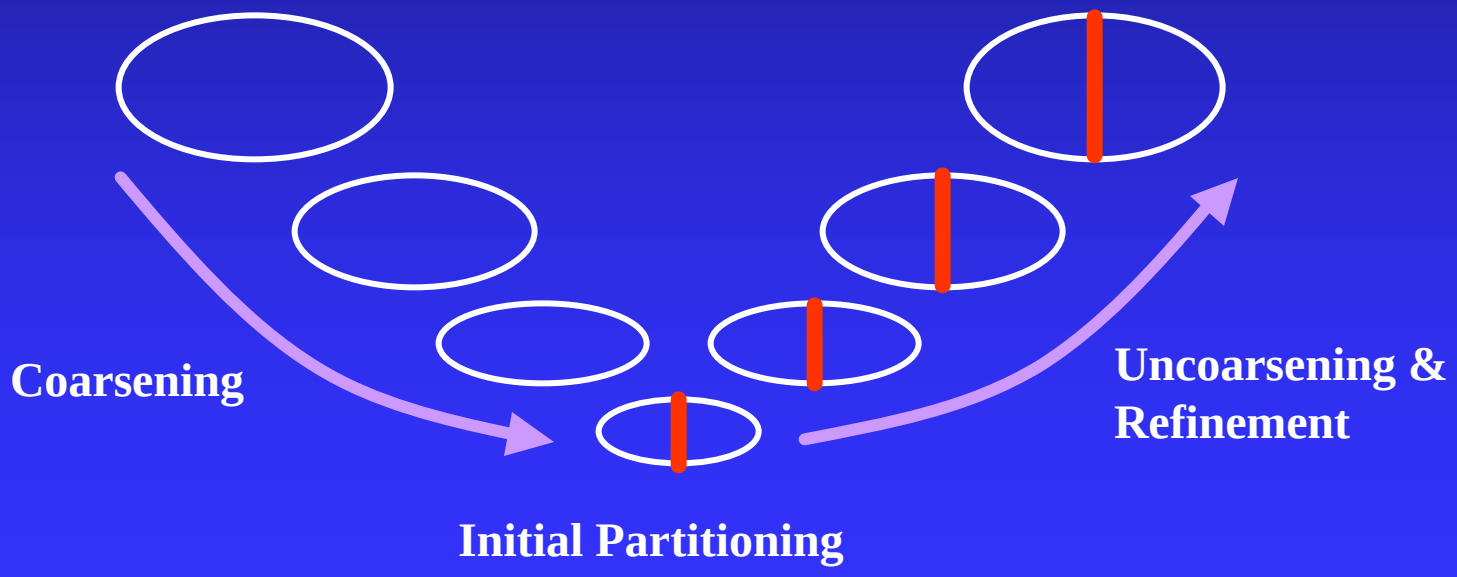
$l(v) = \max\{7-5\cdot1+2+1, 3+2+1\} = 6$

Simultaneous Partitioning/Placement with Retiming

- Minimize SAT during partitioning/placement
- Apply optimal retiming to the resulting solution (best suitable for retiming)
- Partitioning/placement with retiming can be applied recursively to generate physical hierarchy
- Good news: SAT can be computed efficiently (linear time in practice, quadratic time in the worst case)
- Difficulty: Flattened netlist can be very large!
  - Solution: use multi-level method

Multi-level Partitioning

- Iterative coarsening (clustering) to generate a multi-level hierarchy
- Initial partitioning on the coarsest level
- Iterative de-clustering and refinement

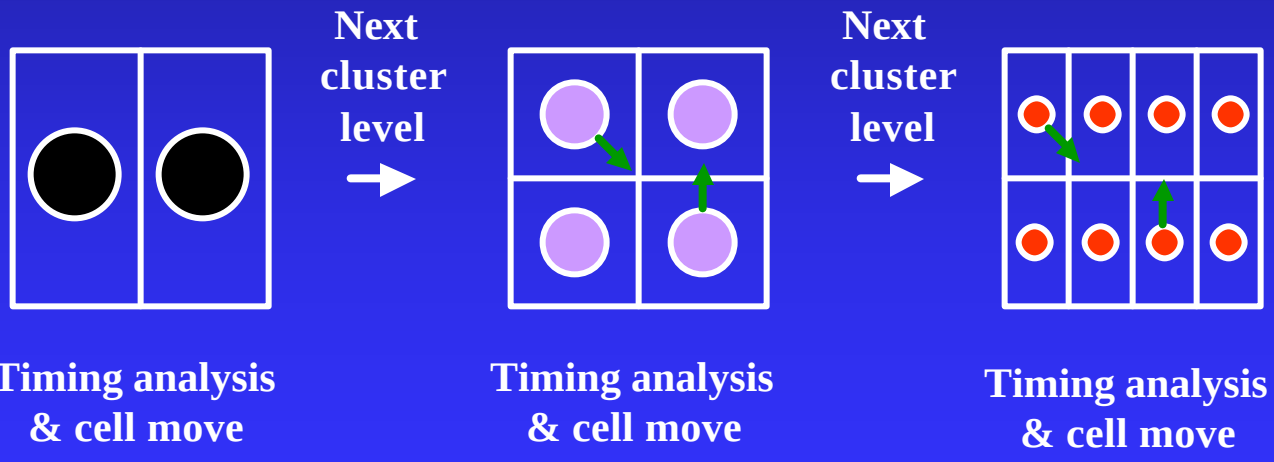


Hierarchical Approach vs Multi-Level Approach

- Hierarchical approach: higher-level design *constrains* lower-level designs
  - Not sufficient information at higher-level
  - Mistake at higher level is impossible or costly to correct
- Multi-level approach: finer-level design *refines* coarse-level design
  - Converge to better solution as more details are considered

Example: Multi-Level Partitioning with Coarse Placement & Retiming

- Bottom-up multi-level clustering
- Top down cell move based multi-level partitioning
- Sequential timing analysis at each level [Cong and Lim, ICCAD00]

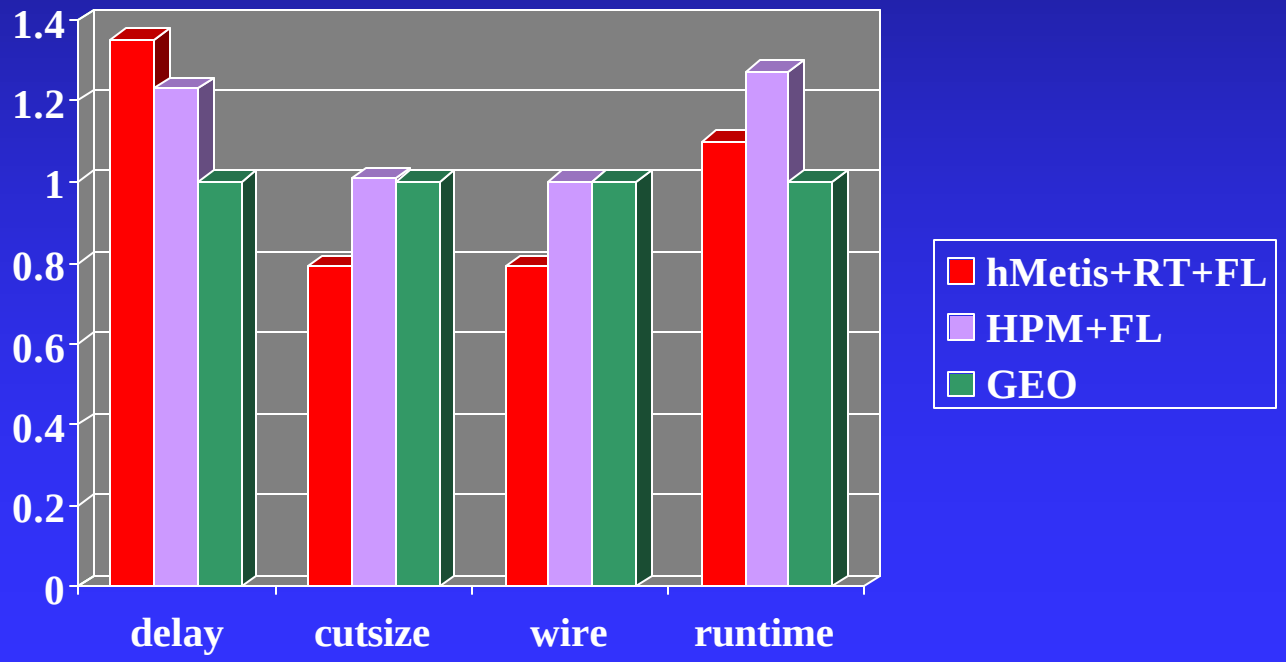


Success of Multi-Level Approach

- **First used to solve partial differential equations (multi-grid method)**
- **Successfully applied to circuit partitioning (hMetis [Karypis et al, 1997])**
  - **Best partitioner for cut-size minimization**
- **Successfully applied to physical hierarchy generation (HPM and GEO [Cong et al, DAC'00 & ICCAD'00])**
  - **30-40% delay reduction compared to hMetis**
- **Successfully applied to circuit placement [Chan et al, ICCAD'00]**
  - 10x speed-up over GordianL

Experimental Results

- Comparison with existing algorithms
  - hMetis [DAC97] + retiming + slicing floorplan [Algo89]
  - HPM [DAC00] + slicing floorplan [Algo89]
  - GEO: simultaneous partitioning + coarse placement + retiming
    Close to 40% delay reduction!



Interconnect Planning

- Physical Hierarchy Generation
- Floorplan/Coarse Placement with Interconnect Planning
  - Example: Buffer Block Planning in Floorplanning
- Interconnect Architecture Planning

Demand of Buffers in Nanometer Designs

- Need to insert buffers in long global interconnects for performance optimization

Technology (um)	*0.25*	*0.18*	*0.13*	*0.10*	*0.07*
#buffer per chip	5k	25k	54k	230k	797k

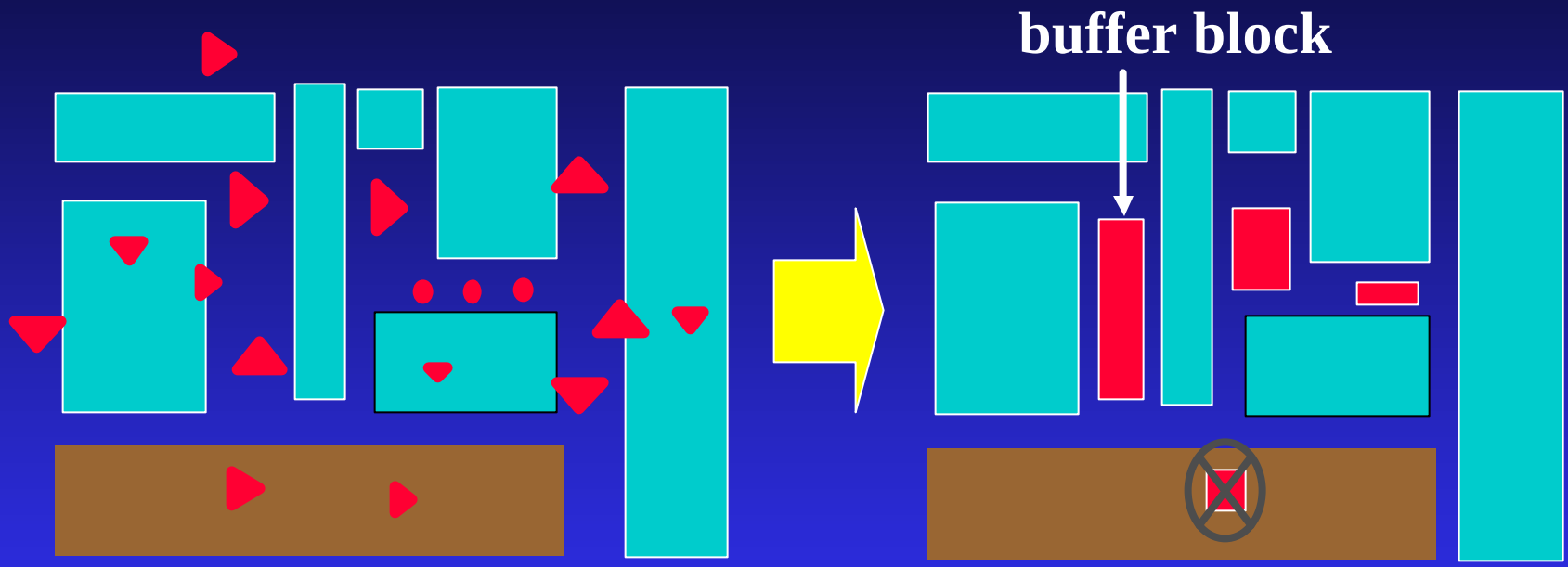
Source: [Cong'97, SRC Work Paper]

http://www.src.org/research/frontier.dgw

(Estimated based on NTRS'97 & [Davis-Meindl'97])

Buffer Block Planning Problem

[Cong-Kong-Pan, ICCAD'99]

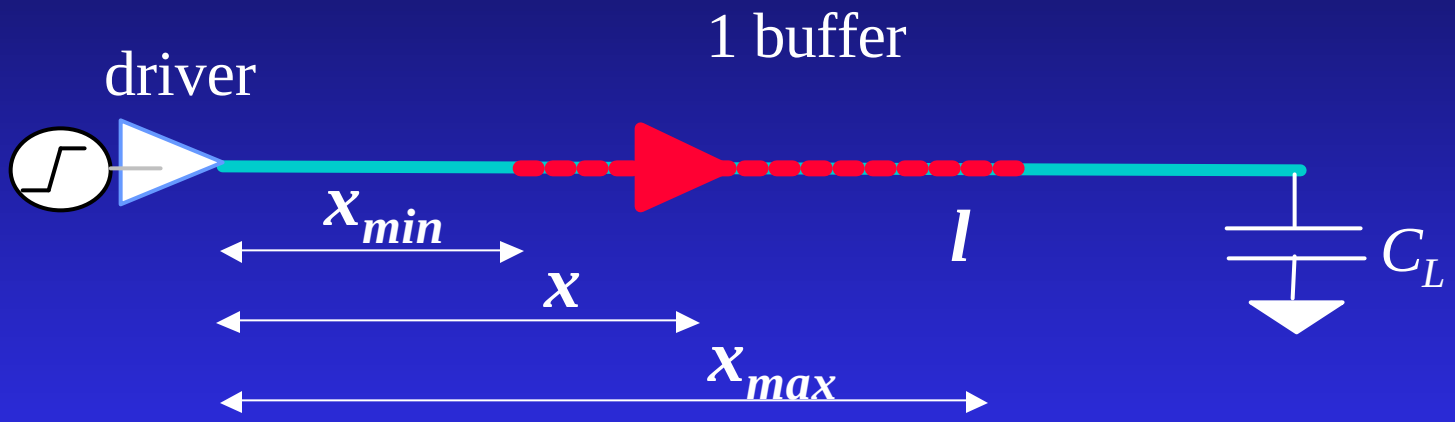


- Restriction from hard IP blocks
- Implications on P/G routing
- Impact on floorplan configuration

=> need to plan ahead for buffers.

Optimal Buffer Location Can Be Relaxed

- **Closed-form formula of feasible region (FR) for inserting one buffer to meet delay constraint**



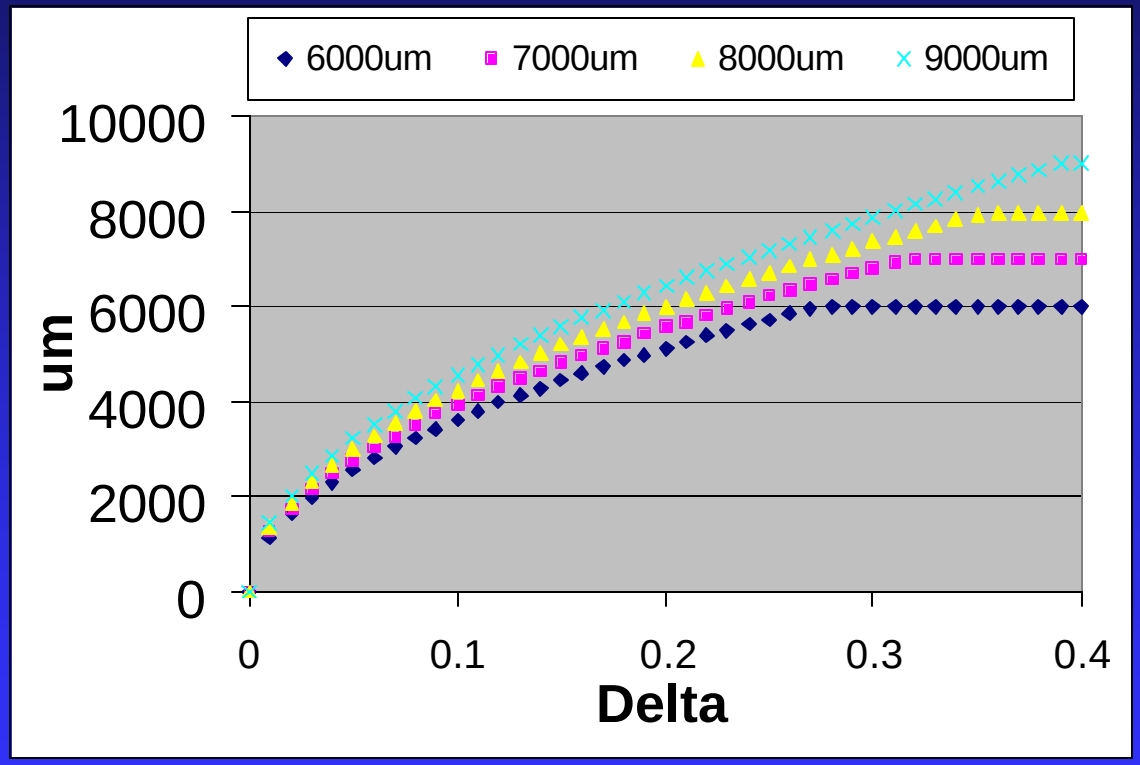
$$x \in [x_{\min}, x_{\max}]$$

$$x_{min} = MAX\left(0, \frac{K_2 - \sqrt{K_2^2 - 4K_1K_3}}{2K_1}\right)$$

$$x_{max} = MIN\left(l, \frac{K_2 + \sqrt{K_2^2 - 4K_1K_3}}{2K_1}\right)$$

Feasible Region (FR) Is Very Large

- Even under tight delay constraint, FR for BI can still be very large!



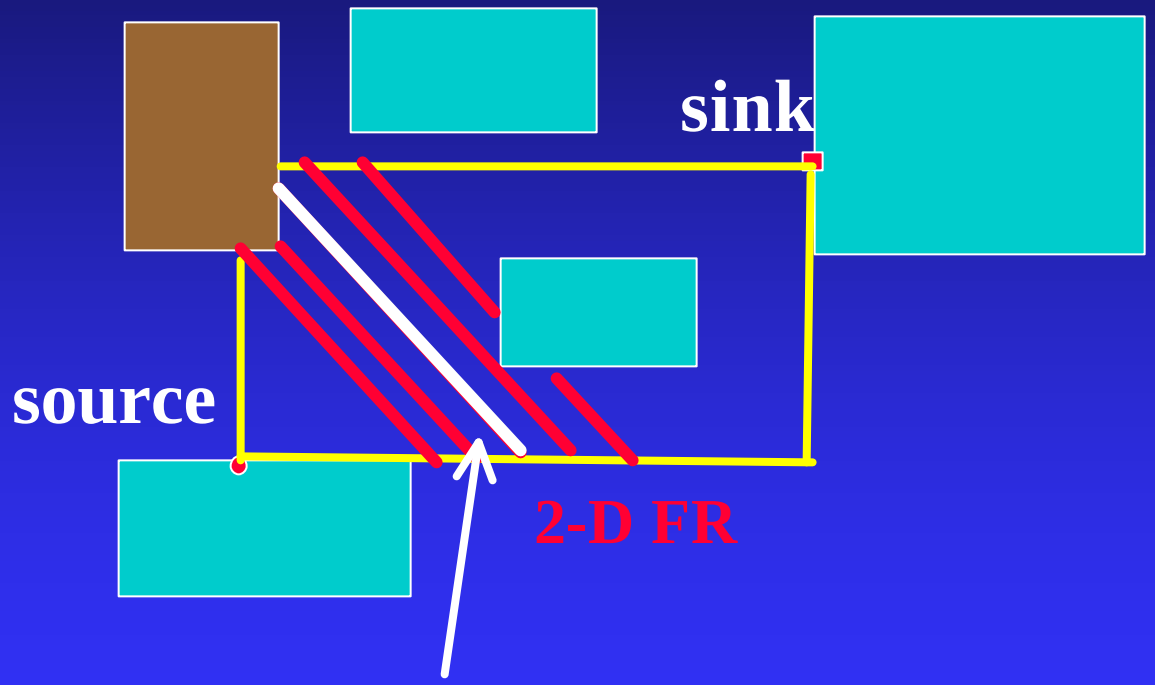
❖ Delay budget is (1+Delta) T_{opt} (the best delay by optimal buffer insertion)

Delta	FR
1%	19%
5%	43%
10%	60%
20%	86%

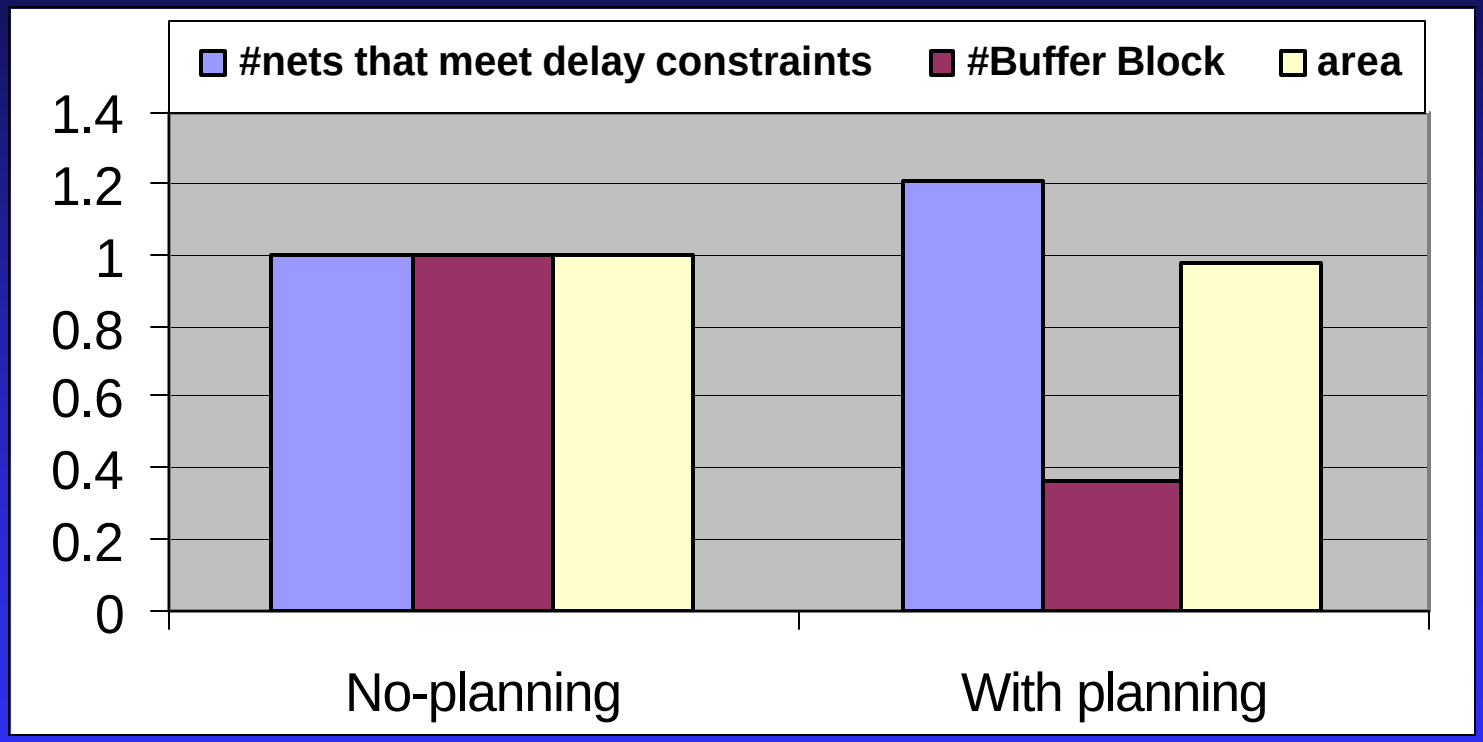
=> FR provides a lot of flexibility to plan buffer location

Extension: 2D Feasible Region

- FR extended to 2-dimension with obstacles



Experimental Results of Buffer Block Planning



Buffer block planning reduces # buffer blocks, better meets timing constraints, and use smaller area

Concluding Remarks

- High-performance designs in DSM technologies need carefully interconnect planning
- Efficient interconnect performance estimation models (IPEMs) are important for interconnect planning
- Top-level partitioning defines global and local interconnects, and impacts performance significantly
- Retiming and pipelining over global interconnects are necessary for multi-gigahertz designs
- A clever combination of partitioning and retiming can hide (some) global interconnect delays
- Buffer block planning help to reduce complexity while achieving good performance

Acknowledgments

- Thanks to Sung Lim, David Pan, and Xin Yuan at UCLA for their help with slides
- Thanks to SRC, MARCO/GSRC, and Intel Corp. for their supports of a number of research projects covered in this tutorial

- Updated slides in PDF file will be available at

  http://cadlab.cs.ucla.edu/~cong