EE290 A: Advanced Topics in CAD
Component Based Design of Electronic Systems
Lecture 10

Professor Kurt Keutzer

Department of Electrical Engineering and Computer Sciences

University of California at Berkeley

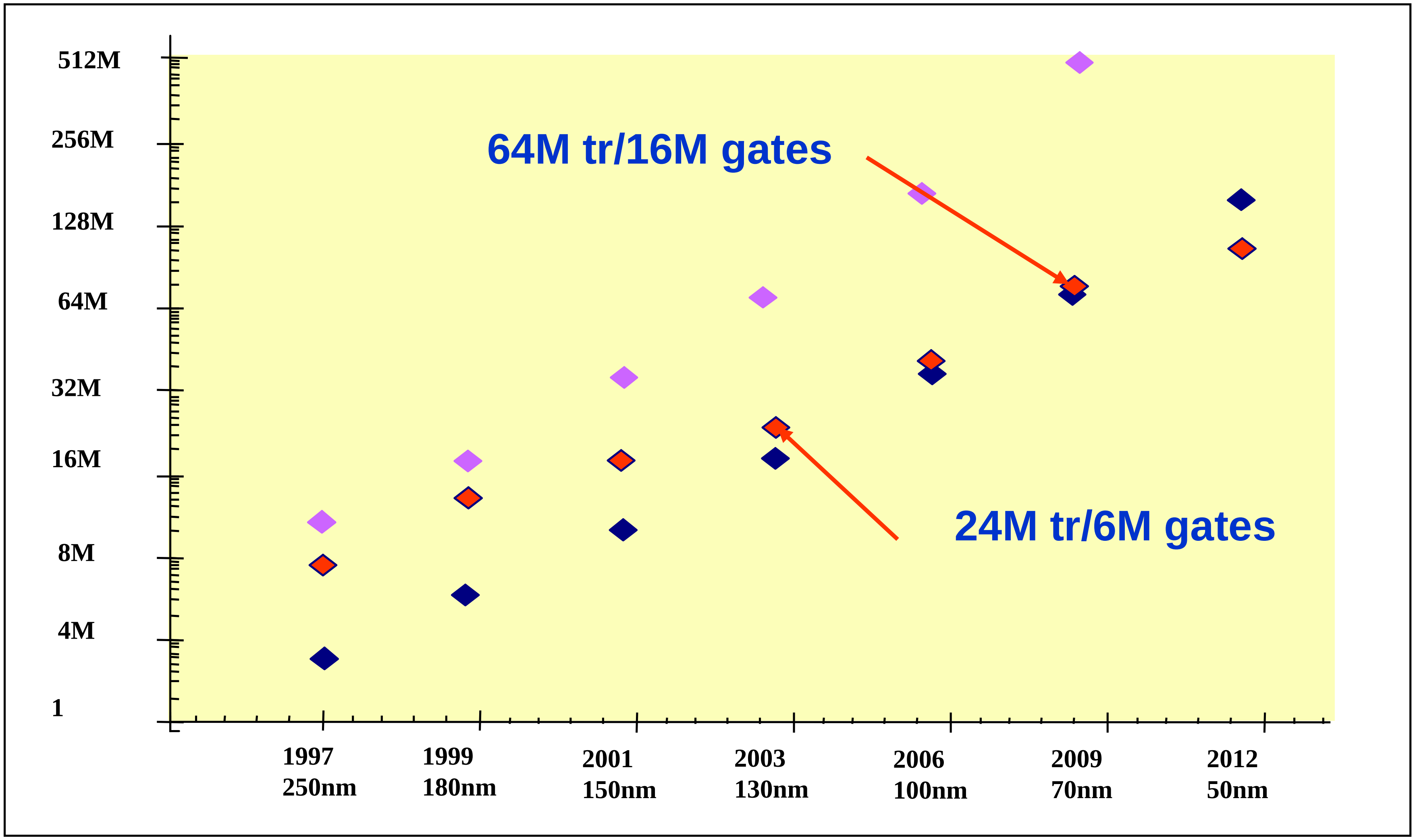
Spring 1999

Why components?

Why are components/IP blocks an attractive way to design electronic systems today?

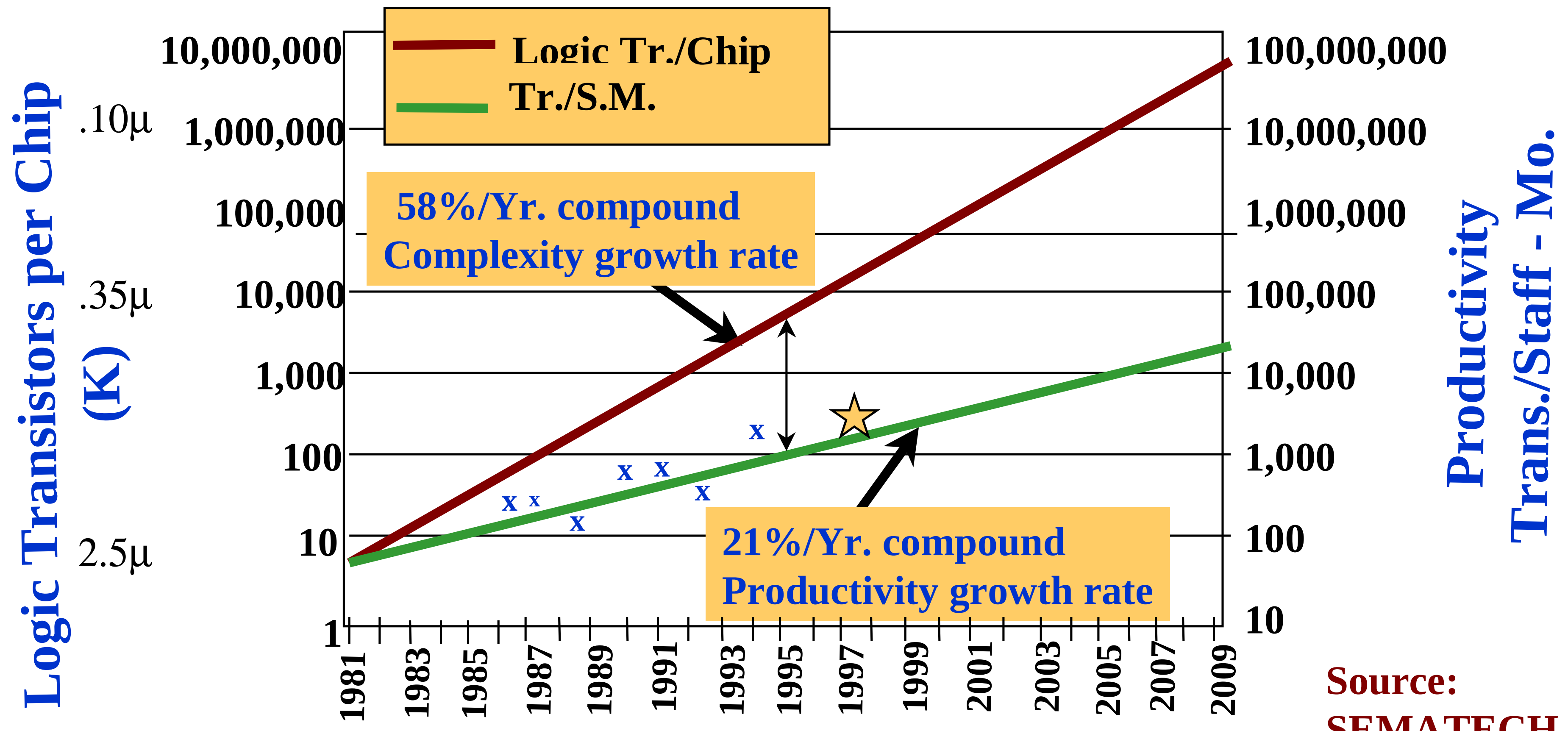
What are the alternative design methodologies?

NRTS: Raw Silicon Capability



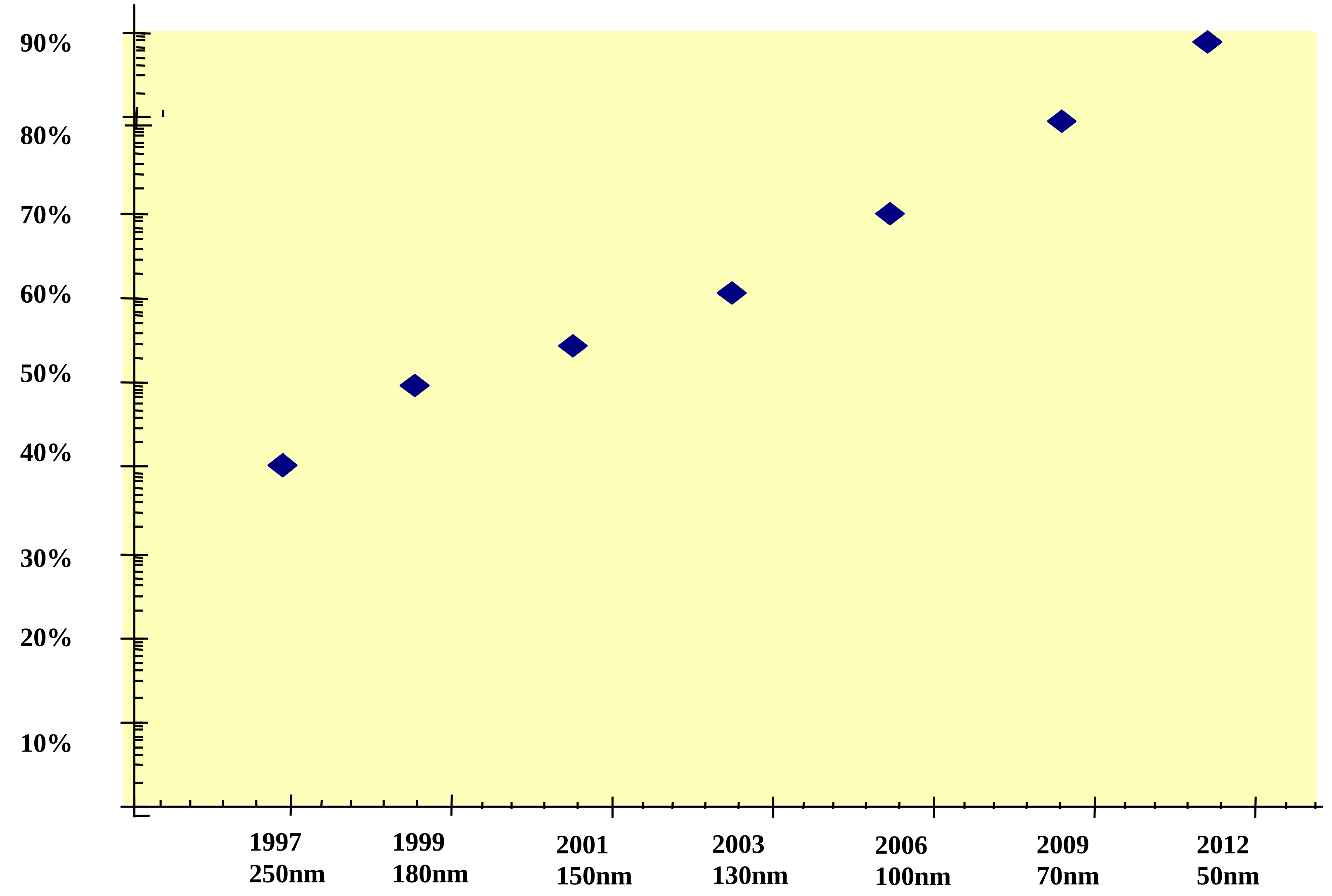
Total microprocessor tr. ◆ Microprocessor logic tr.cm2 ◆ ASIC logic tr. cm2 ◆

Design Productivity



Source: SEMATECH

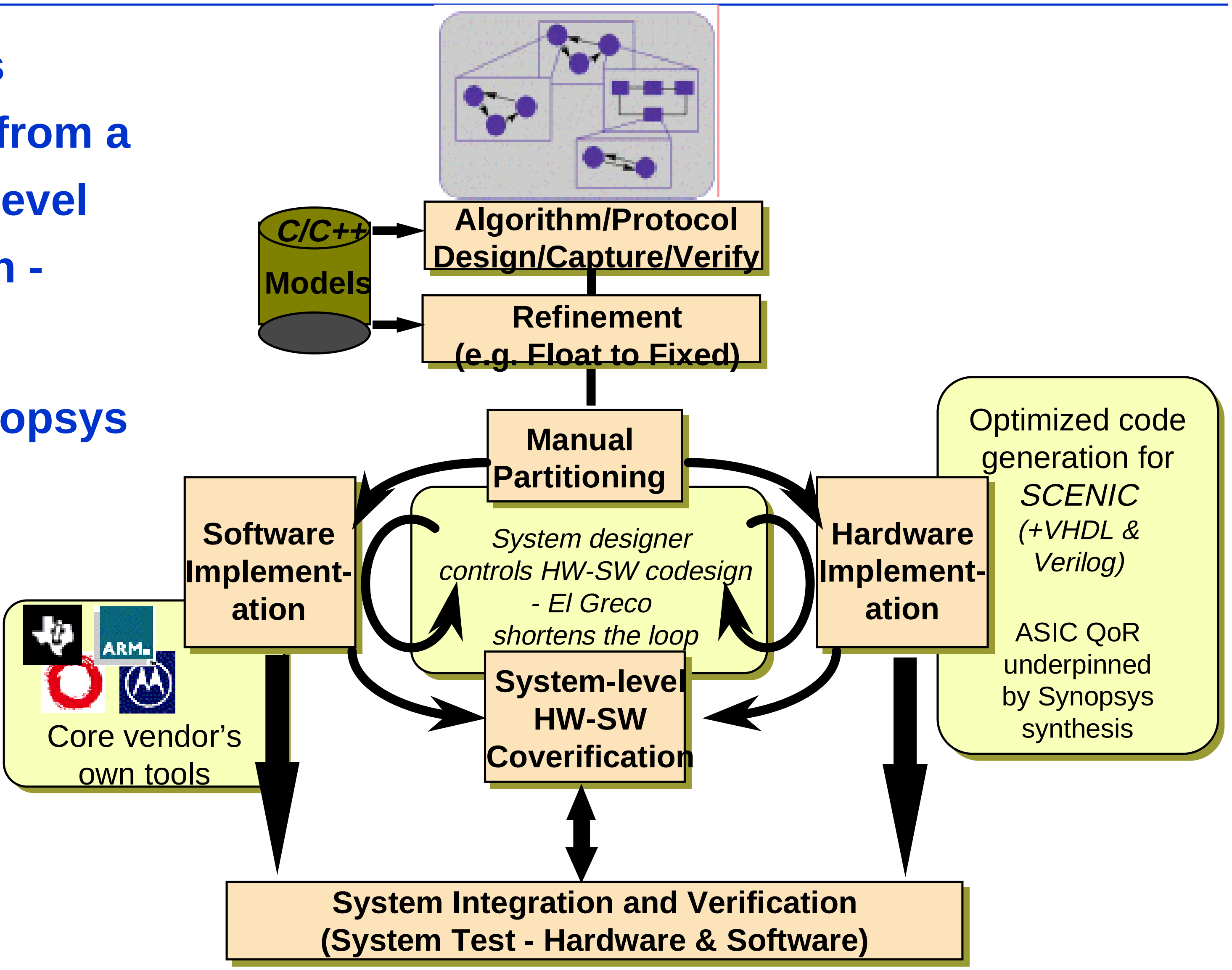
NRTS: Prediction - Increasing design reuse



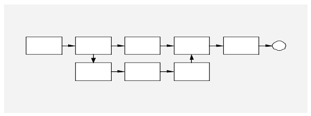
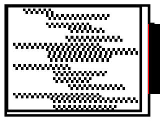
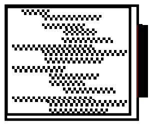
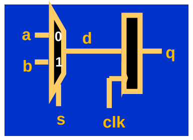
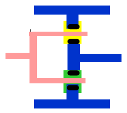
Alternative: El Greco design flow

Alternative is synthesis from a very high-level description -

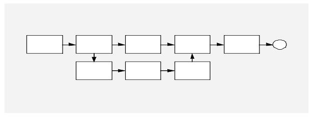
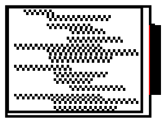
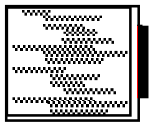
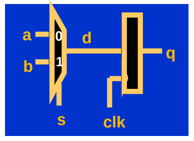
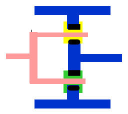
e.g. El Greco/Synopsys


C/C++ Models
Algorithm/Protocol Design/Capture/Verify
Refinement (e.g. Float to Fixed)
Manual Partitioning
Software Implement-ation
Hardware Implement-ation
System designer controls HW-SW codesign - El Greco shortens the loop
System-level HW-SW Coverification
Core vendor's own tools
ARM
Optimized code generation for SCENIC (+VHDL & Verilog)
ASIC QoR underpinned by Synopsys synthesis
System Integration and Verification (System Test - Hardware & Software)

Design Productivity by Approach

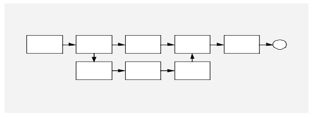
		GATES/WEEK (Dataquest)
DOMAIN SPECIFIC		8K - 12K
BEHAVIORAL		2K - 10K
RTL		1K - 2K
GATE		100 - 200
TRANSISTOR		10 - 20

To Design, Implement, Verify ...

		10M tr/2.5M gates Staff Months	24M tr/6M Staff Months	64M tr/16M Staff Months
		62.5	150	400
Beh		125	300	800
RTL		625	1500	4000
gate		6250	15,000	40,000
tr		62,500	150,000	400,000

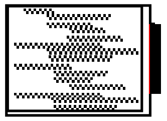
Even for 10M transistors ...

Staff Months



62.5

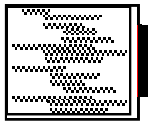
Beh



125

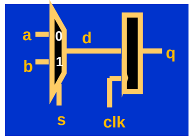
Implementations here are often not good enough

RTL

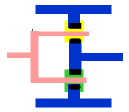


625

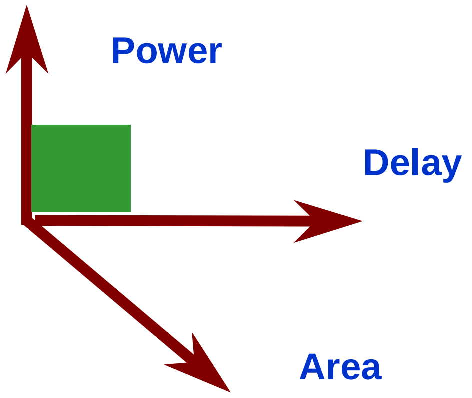
Because implementations here are inferior/ unpredictable



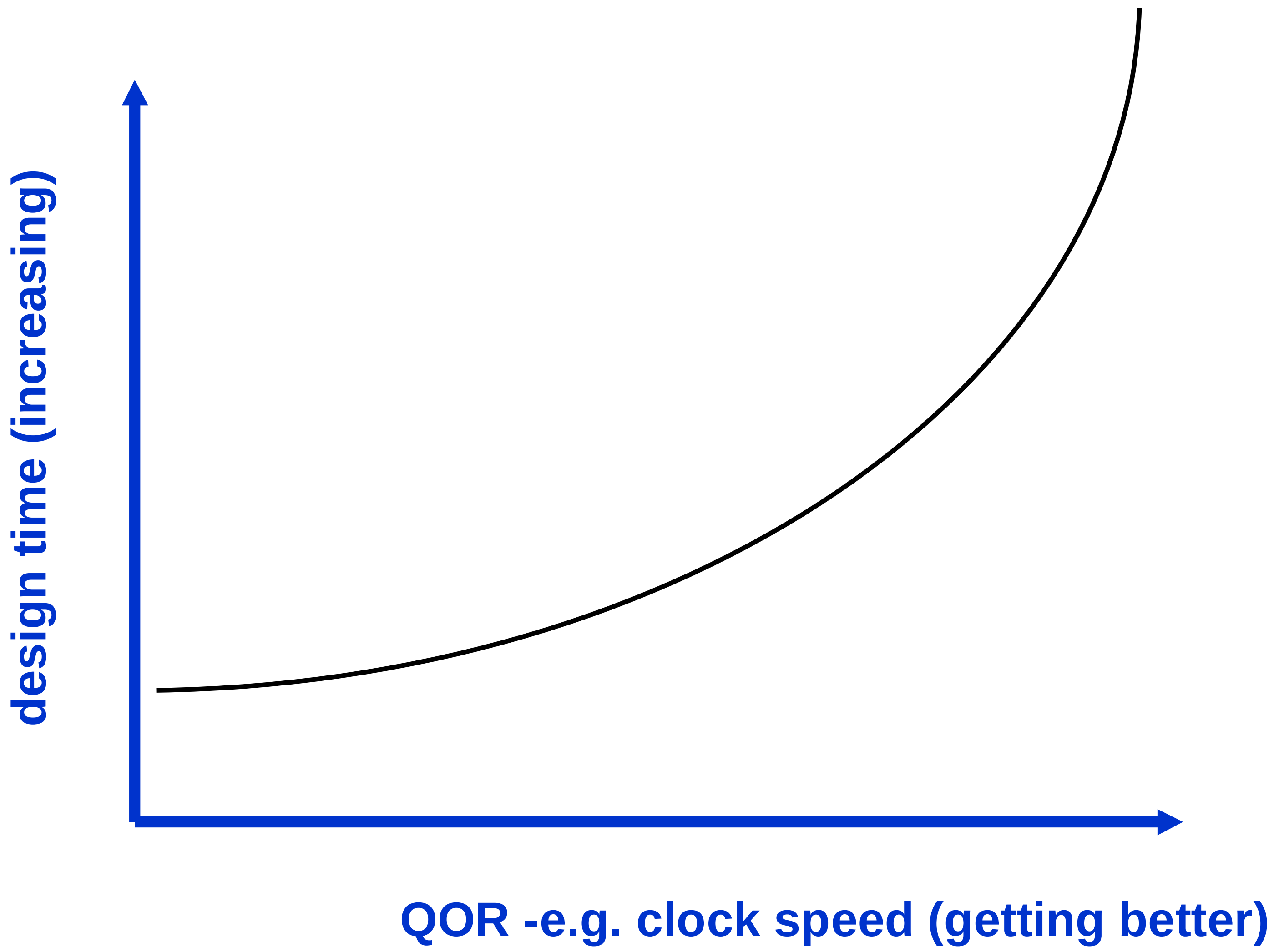
6250



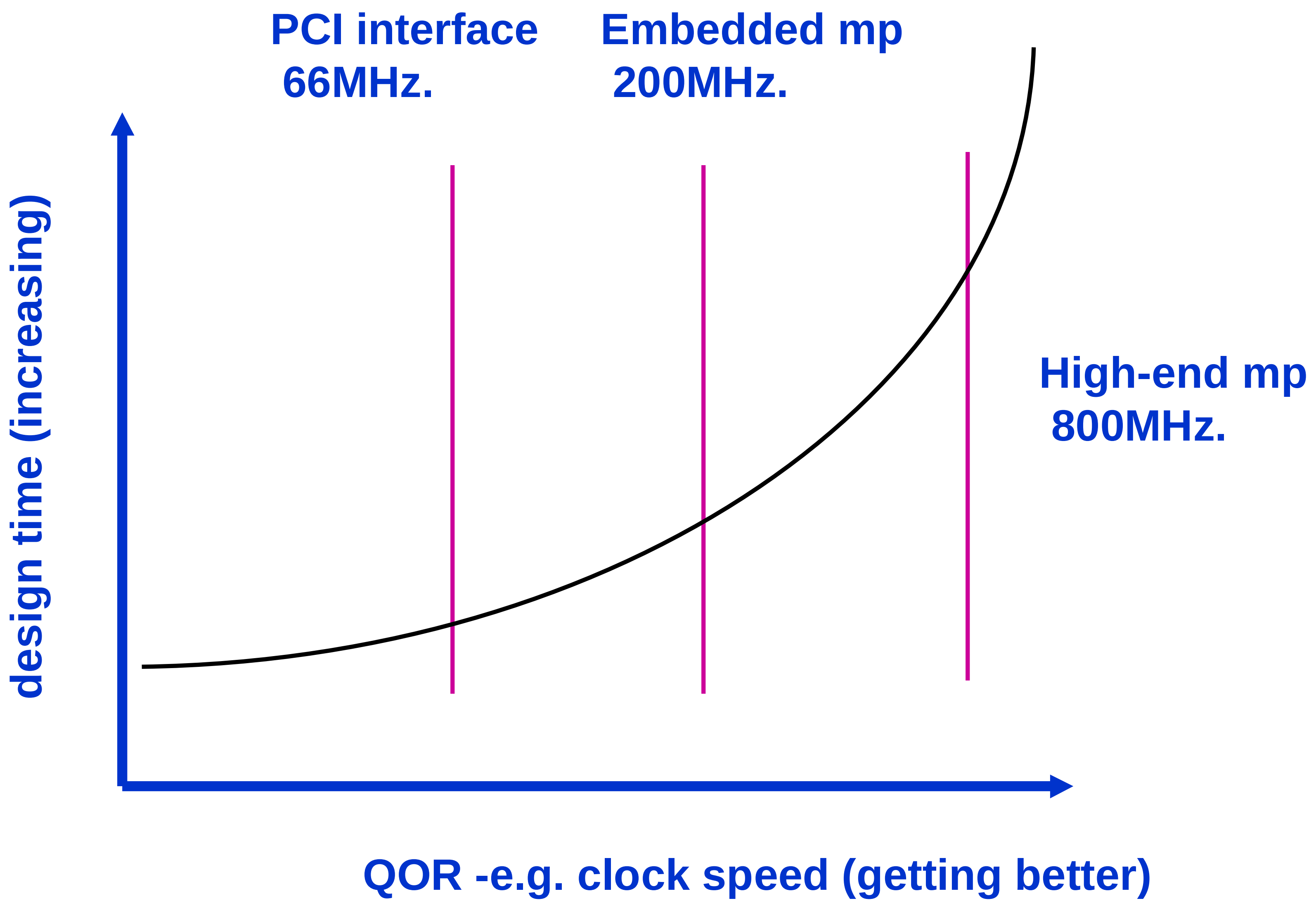
62,500



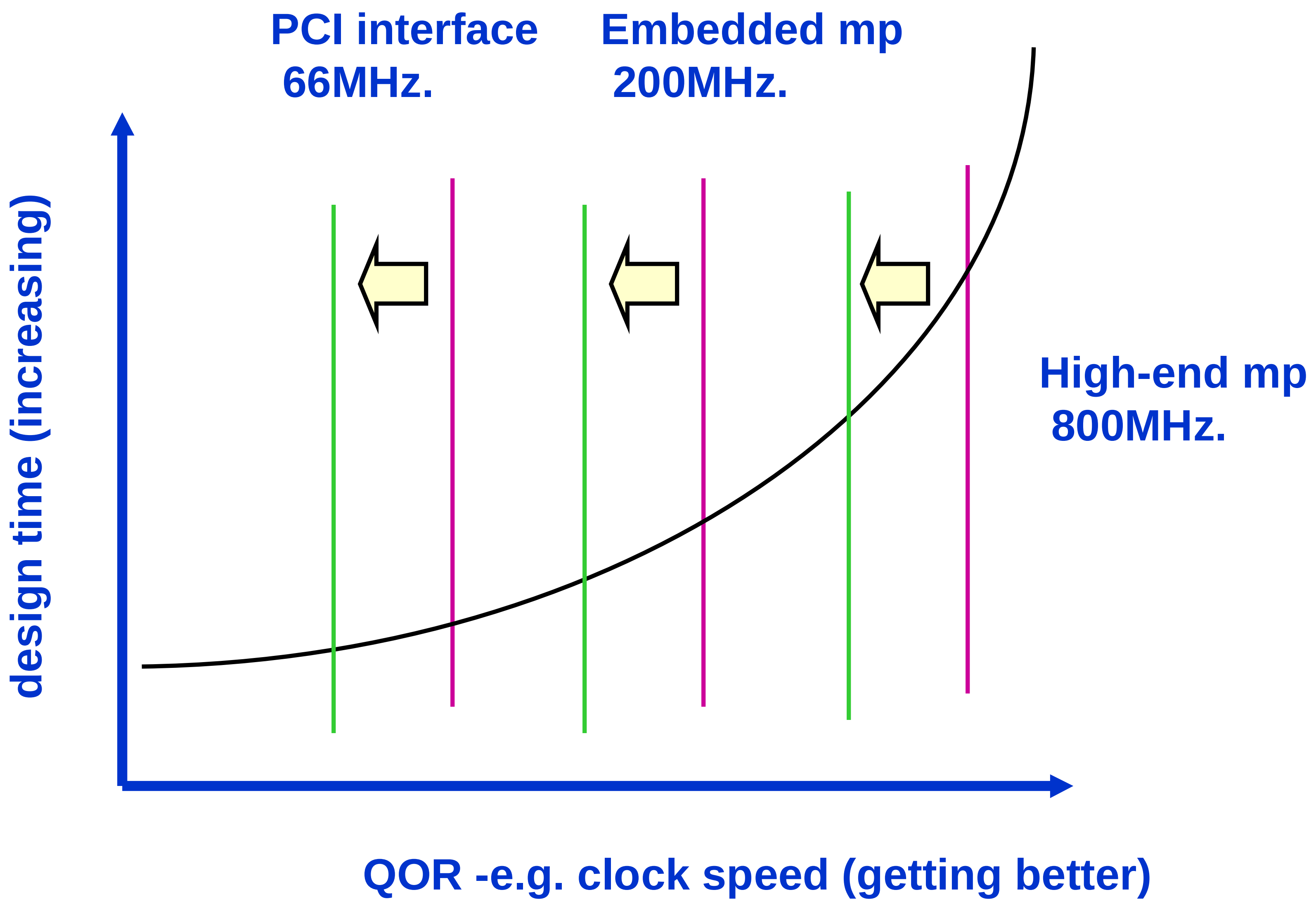
Productivity vs. QOR



Productivity vs. QOR



After only one process generation



Moral of the Story

Each process generation will

- Make it twice as easy to realize a fixed (e.g. 66MHz.) timing spec
- Make it twice as easy to realize a fixed (e.g. 100 mm) area requirement
- Make time-to-market requirements 20% more stringent

Time is always on the side of more productive EDA tools/methodologies, but current high-productivity synthesis methodologies are still not meeting QOR requirements

Why components? - summary

Increasing re-use of intellectual property blocks/components is the consensus because:

- Moore’s law continues to provide significantly greater silicon capability
- Significant productivity improvements are required
- Alternative methodologies are not coming forward - and the implementation quality does not meet requirements

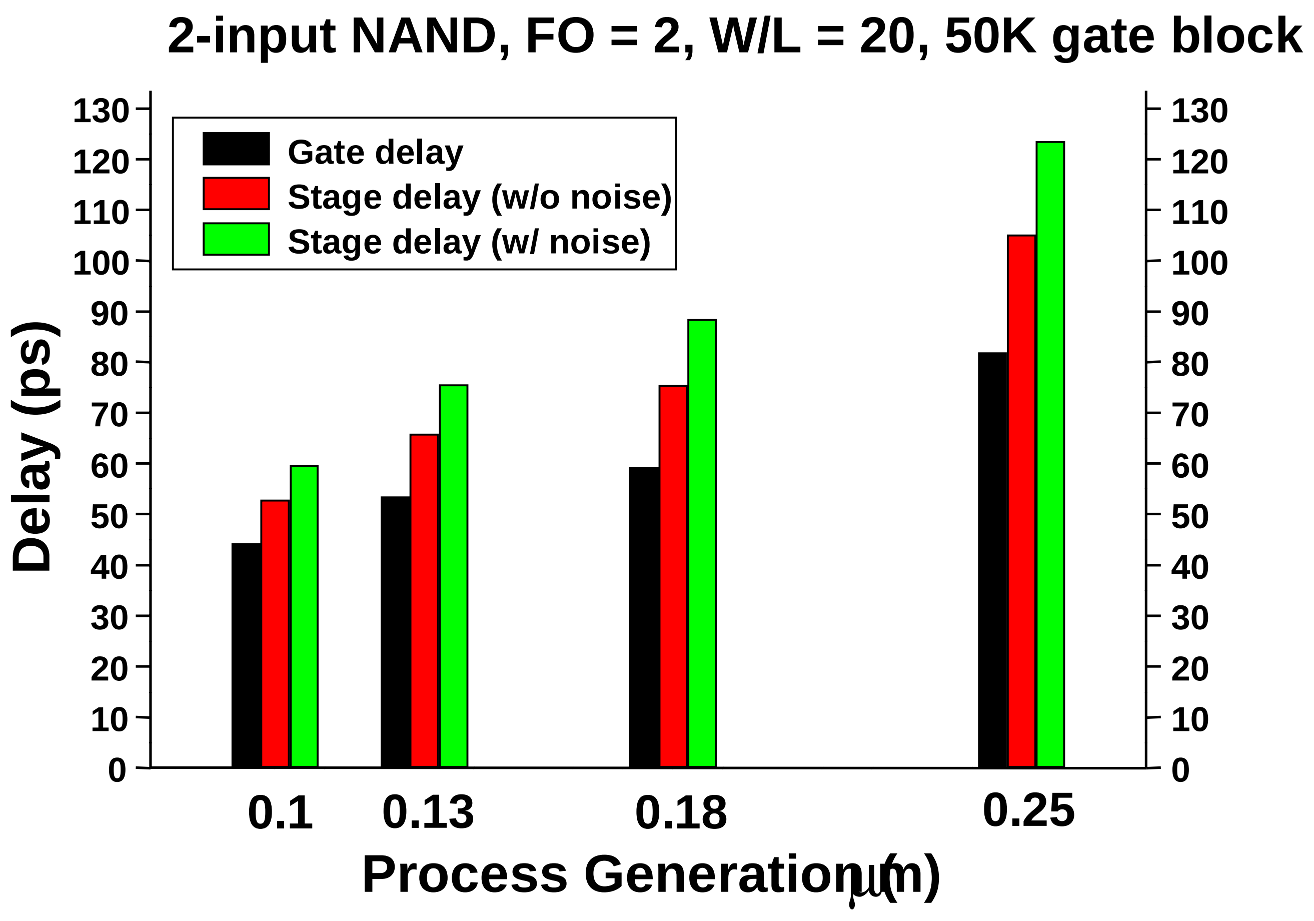
What type of components?

What type of components?

- **What size of component?**
- **What type/capability of component?**

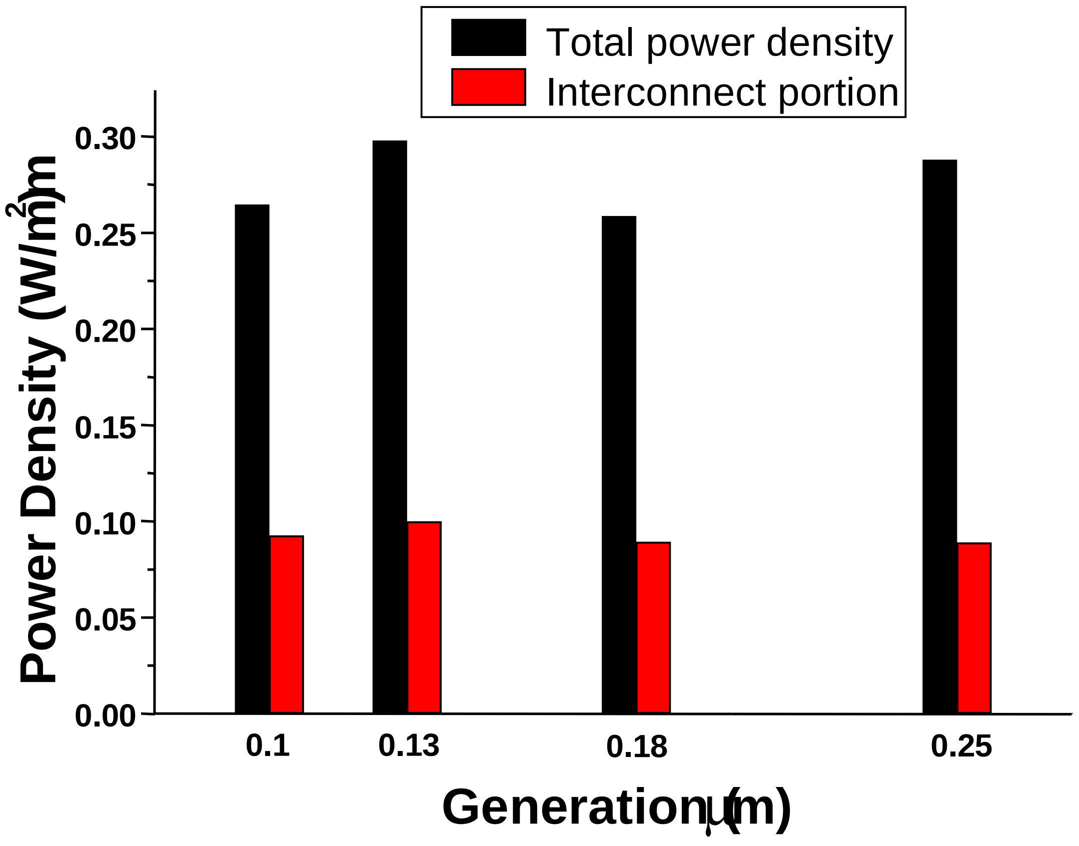
Delay degradation in DSM

- With scaling processes, interconnect delay becomes a decreasing portion of stage delay, *even with noise*



Dynamic Power Analysis, 50K blocks

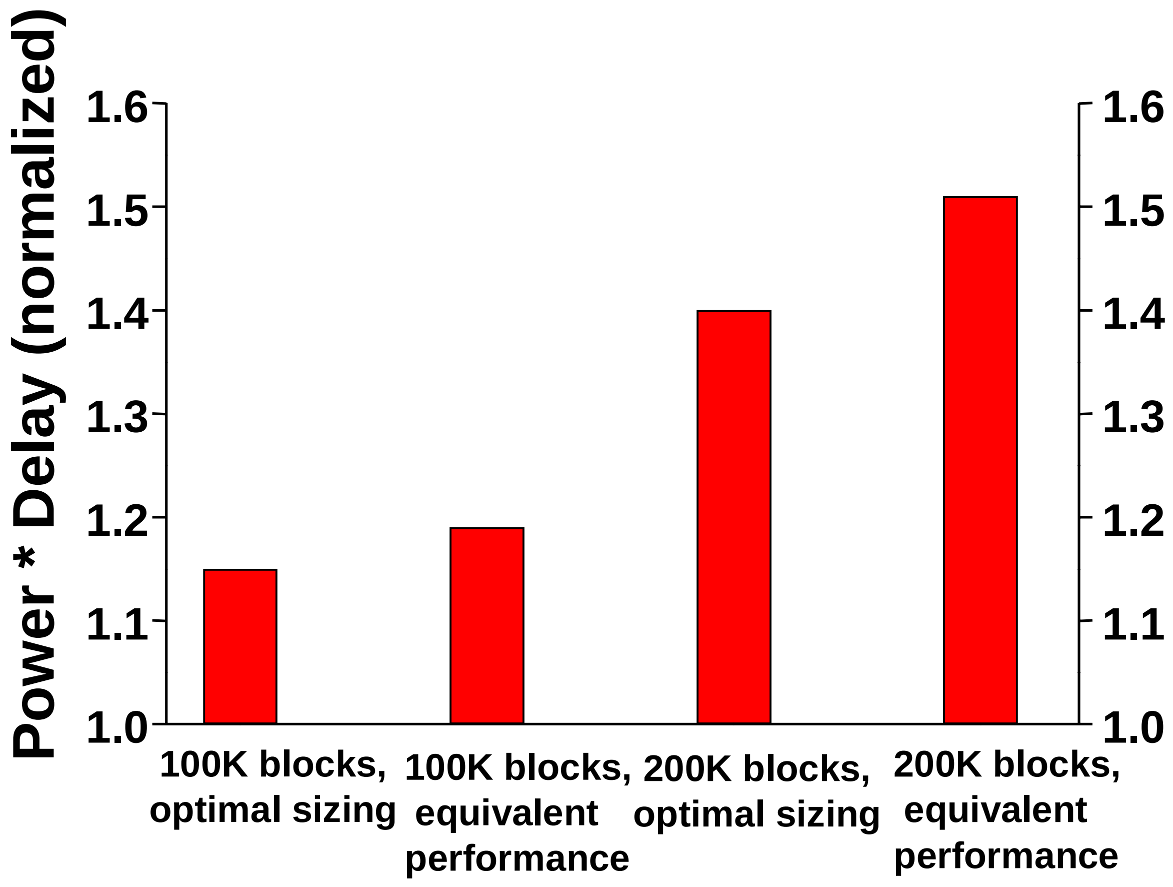
Parameters:	**Packing density from NTRS**
	Switching activity = 0.15
	Device sizing set at W/L = 20
	Routing density set at 0.4 (M1/2) and 0.2 (M3/4)



- **50K blocks**
- **Analysis focuses on dense std. cell logic parts of an ASIC**
- **Frequency set according to NTRS**
- **Power density is roughly constant through scaling**

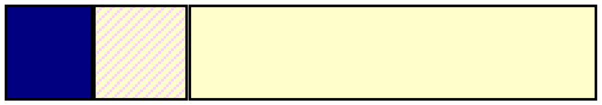
Dynamic Power Analysis, 100K & 200K Blocks

Parameters:	**Packing density from NTRS**
	Switching activity = 0.15
	Device sizing set at W/L = 20
	Routing density set at 0.4 (M1/2) varied from 0.2 to 0.3 (M3/4)



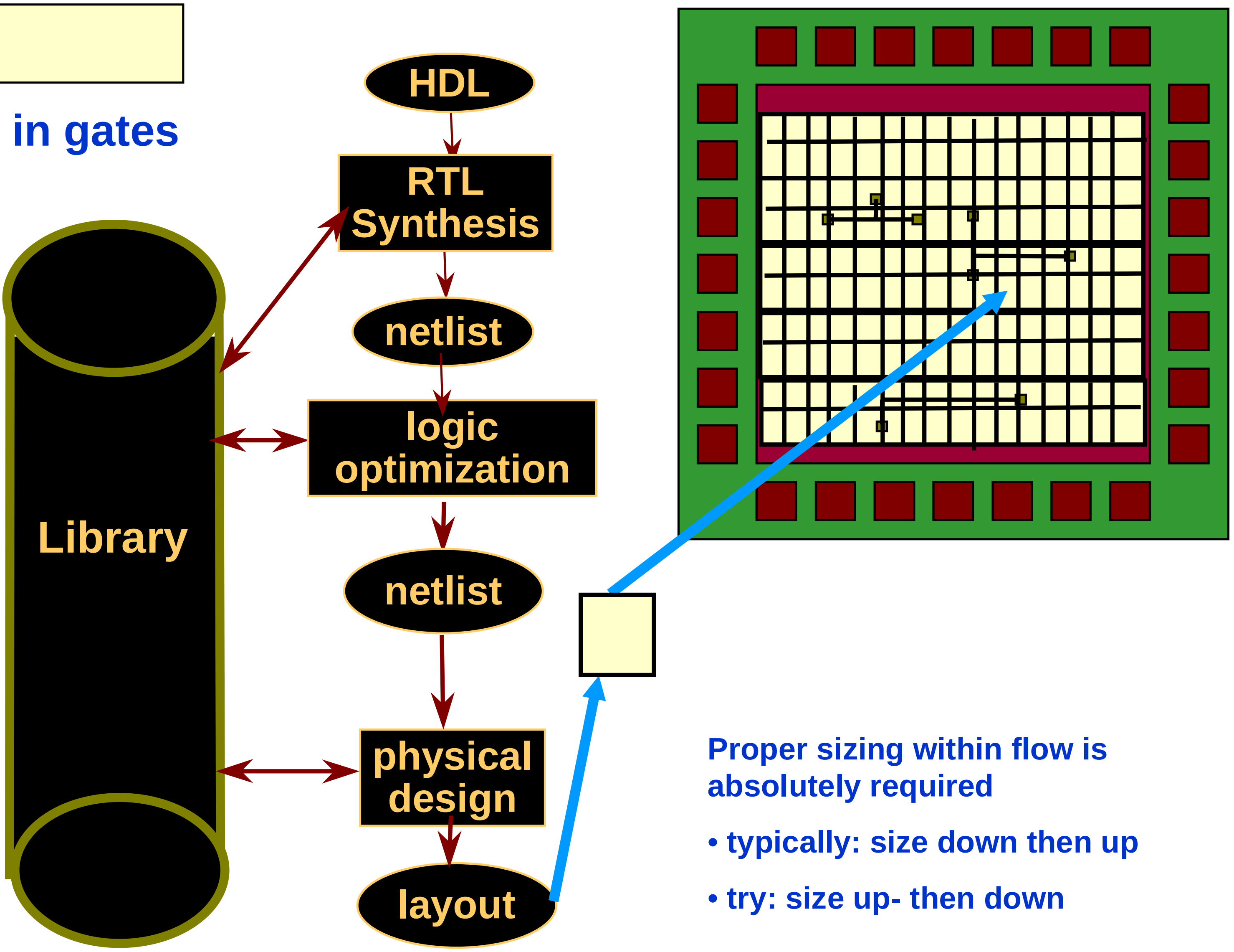
- **Various block sizes**
- **Normalized to 50K gate block performance**
- **Numbers are relatively independent of technology generation**

Within a 50K - 100K Module



75 - 100% delay in gates
3.5μ - 1.0μ
1980 - 1990
.18μ - 0.1μ
1998 - 2005

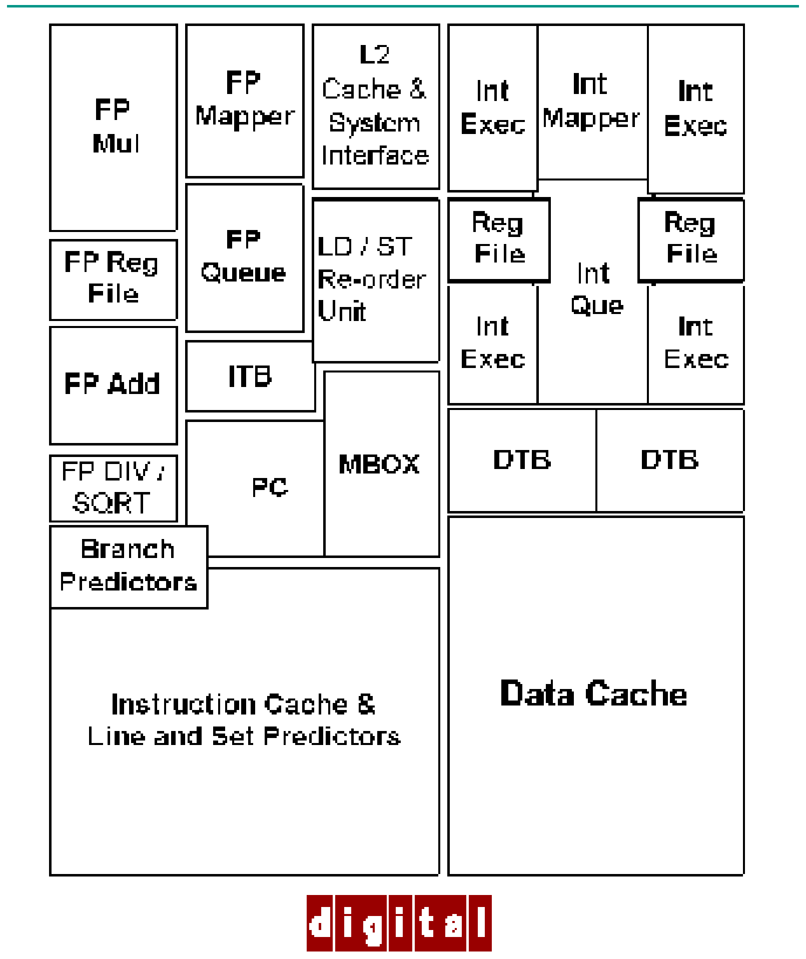
This flow should work OK for blocks of 50-100K gates and should continue to work OK in the future



Proper sizing within flow is absolutely required

- typically: size down then up
- try: size up- then down

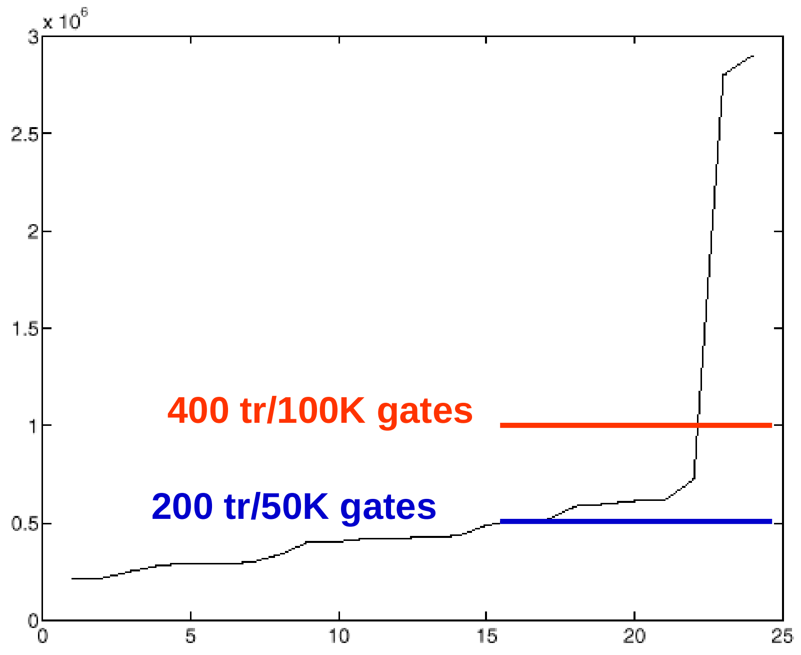
Is 100K too small? Look at Dec Alpha

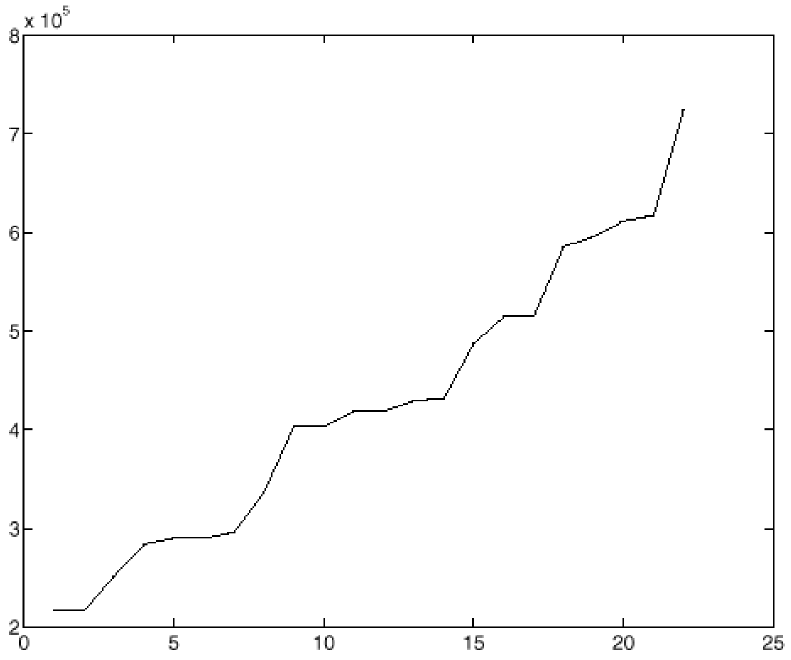


Details (Alpha 21264)

Unit	#	Aspect Ratio	# Transistors
Instruction cache	1	0.73	2.9M *
ITB	1	0.56	284k
PC	1	0.91	488k
Branch Predictor	1	0.53	337k
Data cache	1	0.82	2.8M *
DTB	2	0.74	419k
MBox	1	0.61	586k
LD/ST Reorder Unit	1	0.78	612k
L2 Cache/System IO	1	0.79	596k
Integer Exec	2	0.75	290k
	2	0.54	404k
Integer Queue	1	0.5	617k
Integer Reg File	2	0.91	217k
Integer Mapper	1	0.71	432k
FP div/sort	1	0.57	252K
FP add	1	0.97	429k
FP Queue	1	0.81	515k
FP Reg File	1	0.67	296k
FP Mapper	1	0.81	515k
FP mul	1	0.61	725k
uP	24	0.81	15.2M

Details of Block Size





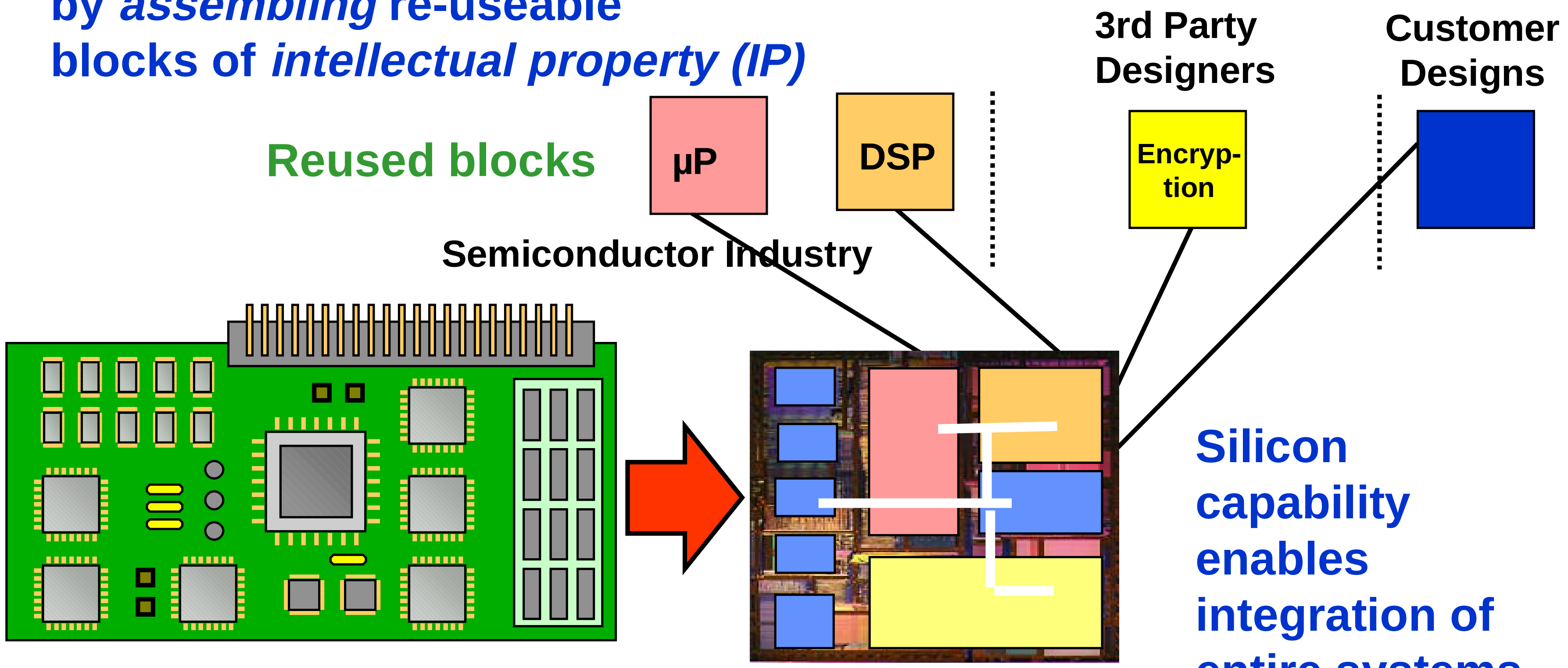
A. Tabbara - UC Berkeley

Components: Next Generation SSI?

- Gary Smith, Principal Analyst at Dataquest writes:
- ‘‘The amount of confusion within the CAD community has been the big surprise in the effort to develop System Level Integration (SLI) design methodology. … How do you design a million gate IC ? … Fortunately Kurt Keutzer gets it. The first thing we needed to know was the optimum size of a basic SLI library element. The work, at Berkeley, now tells us it’s 50,000 gates. So we have defined today’s SSI. Now the challenge is to develop the 200 to 400 basic library elements needed to really do SLI design. ‘’

Design Paradigm: Re-useable IP

Achieve required design productivity by *assembling* re-useable blocks of *intellectual property (IP)*



Silicon capability enables integration of entire systems on a single die

What type of component? Reuse at what level?

Entire sub-system

- e.g. MPEG

Large module

- DCT, motion estimation

Sub-module

- Filter, multiplier

Primitive component

- gate, full-adder

``We want to reuse at the highest level that we can.’’ -
Lance Mills, HP

Type/Functionality of Components

Video: MPEG, DVD, HDTV

Audio: MP3, voice recognition

Processors: CPUs, DSPs, Java

Networking: ATM, Ethernet, ISDN, FibreChannel, SONET

Bus: PCI, USB, IEEE 1394

Memory: SRAM, ROM, CAM

Wireless: CDMA, TDMA

Communication: modems, transceivers

Coding: speech, Viterbi, Reed-Solomon

Display drivers/controllers: TFT

Other: sensors, encryption/decryption, GPS

Power PC core: 3.1mm^2 in 0.35µ

ARM Core: 3.8 mm^2 in 0.35µ

MPEG2 Decoder: ~65k gates

PCI Bus: ~8k gates

Ethernet MAC: ~7k gates (soft)

RSA Encryption: ~7k gates

Niraj Shah

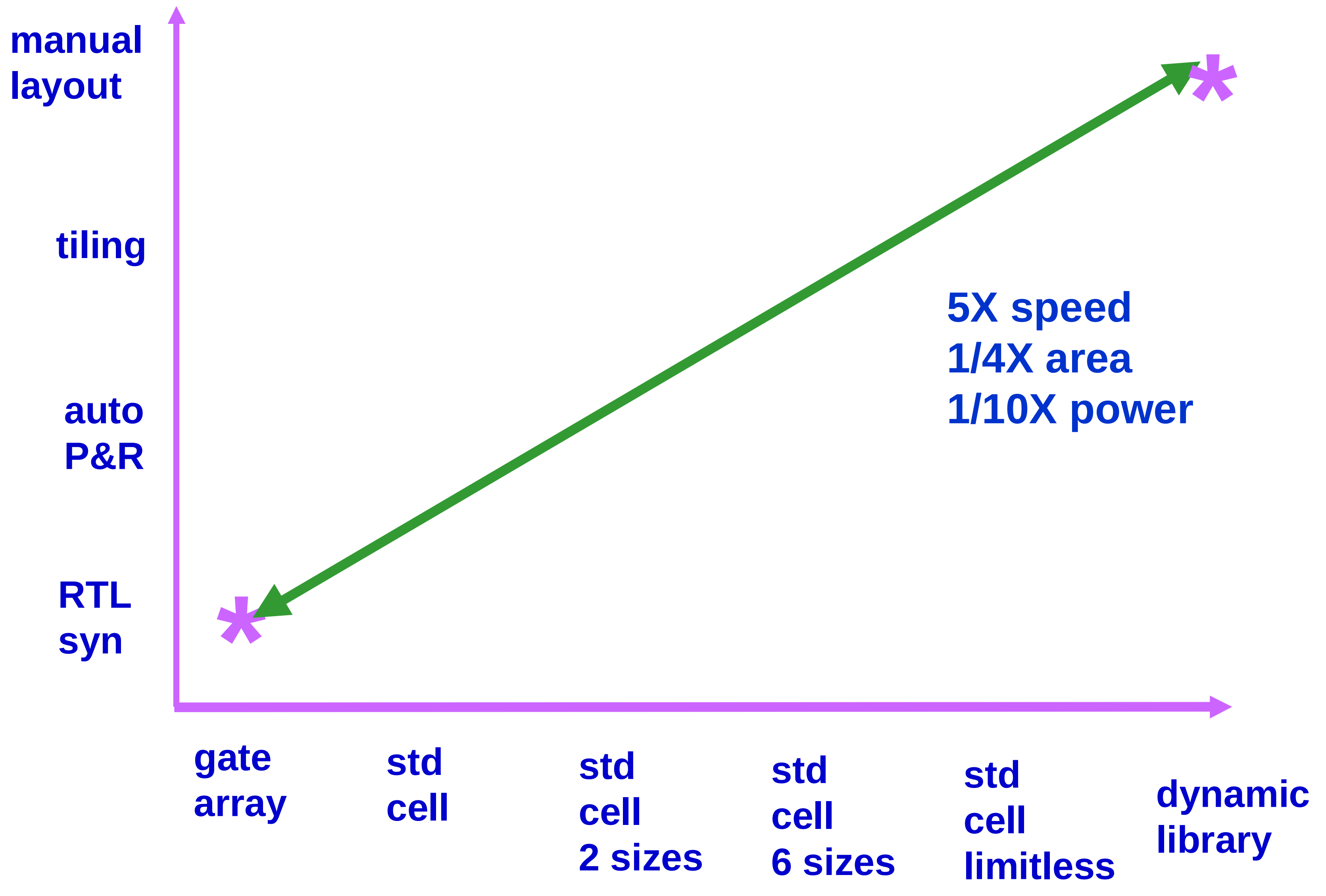
How will components be implemented?

What are the most promising implementation media?

- **SW**
  - **SW running on a standard processor - MIPS R4000**
  - **SW running on a tailored processor - TI TMS320C54**
  - **SW for a configurable processor - XTENSA**
  - **SW for an application-specific processor - C-Cube MPEG decoder**
- **HW**
  - **hard - actual layout**
  - **firm - netlist**
  - **soft - synthesizable RTL model in VHDL?/ Verilog?**

The aim of this course is to make us the authority on these questions!

How much do we lose in hard vs soft IP?



What do we gain with soft vs. hard IP?

FAB portability - soft, firm IP is migratable to multiple processes

Modifiability - soft IP can be user modified (a plus?)

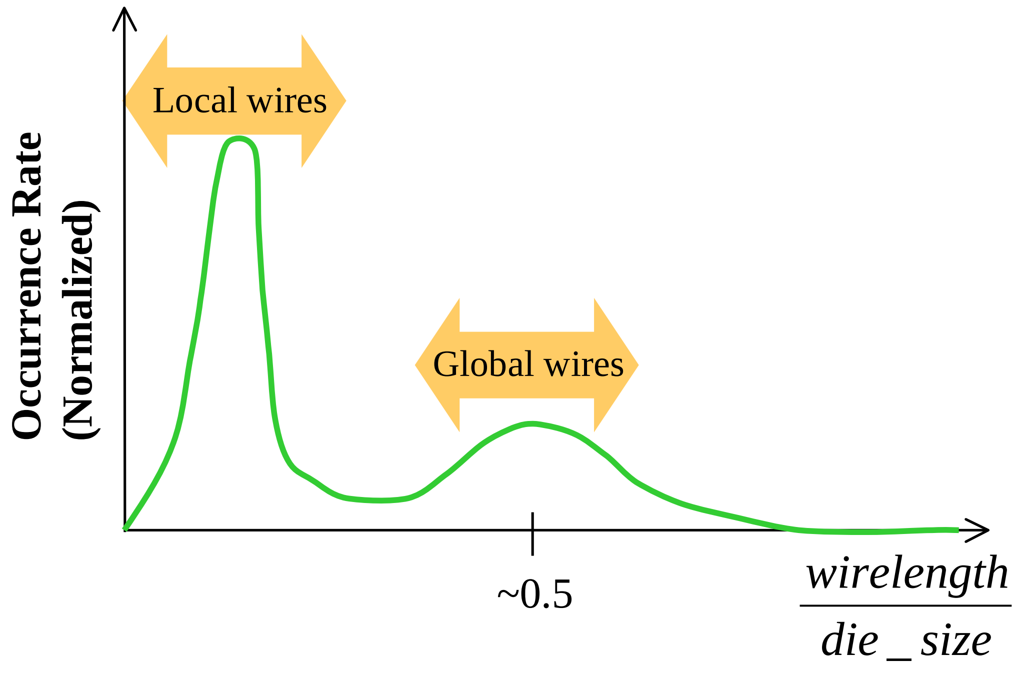
Process migratability - a soft, firm IP design can more easily be migrated to the next process generation

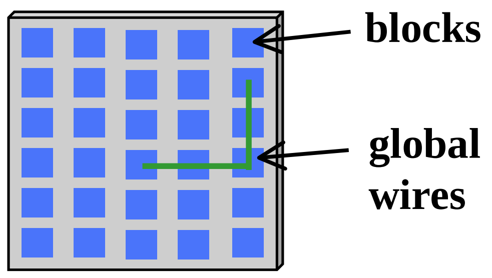
Wider range of QOR of implementation

Interconnect Complexities

Interconnect effects play a major role in the increasing costs for large hard-block design styles

Without new Circuit Fabrics and the flexibility they offer, interconnect problems will significantly impact performance and cost for emerging IC technologies

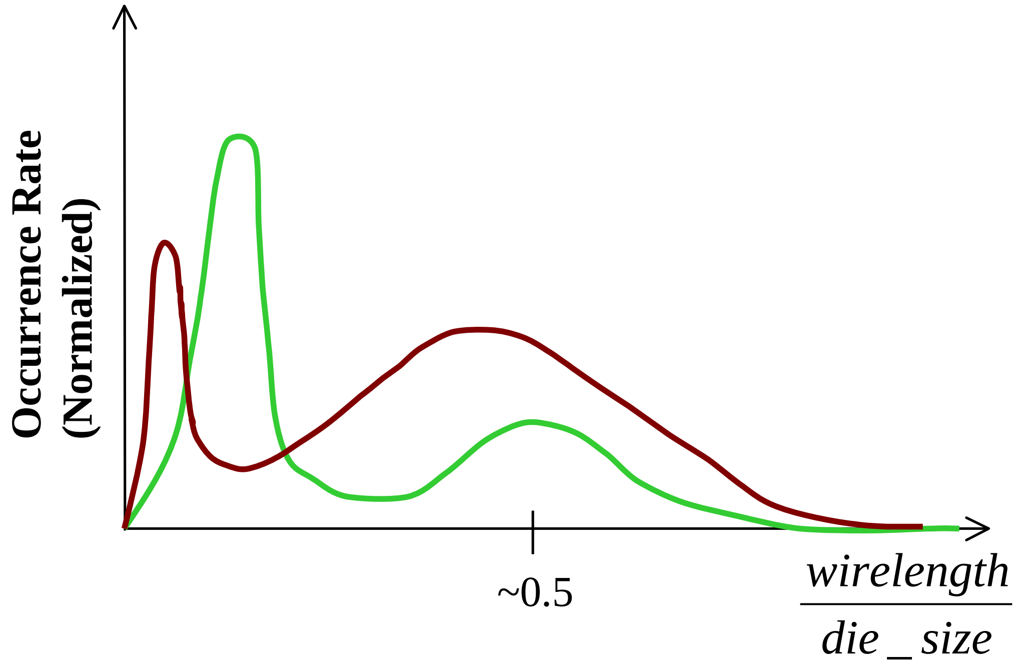


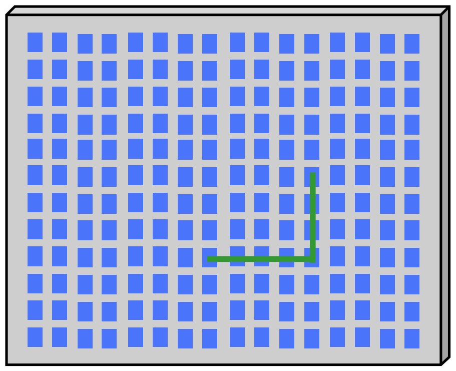


Technology Scaling

Block sizes cannot grow as rapidly as chip sizes since block design becomes increasingly more difficult --- each block is a chip design over multiple configurations

If the blocks are inflexible, the *global* wiring problems begin to dominate all aspects of performance quality and system cost



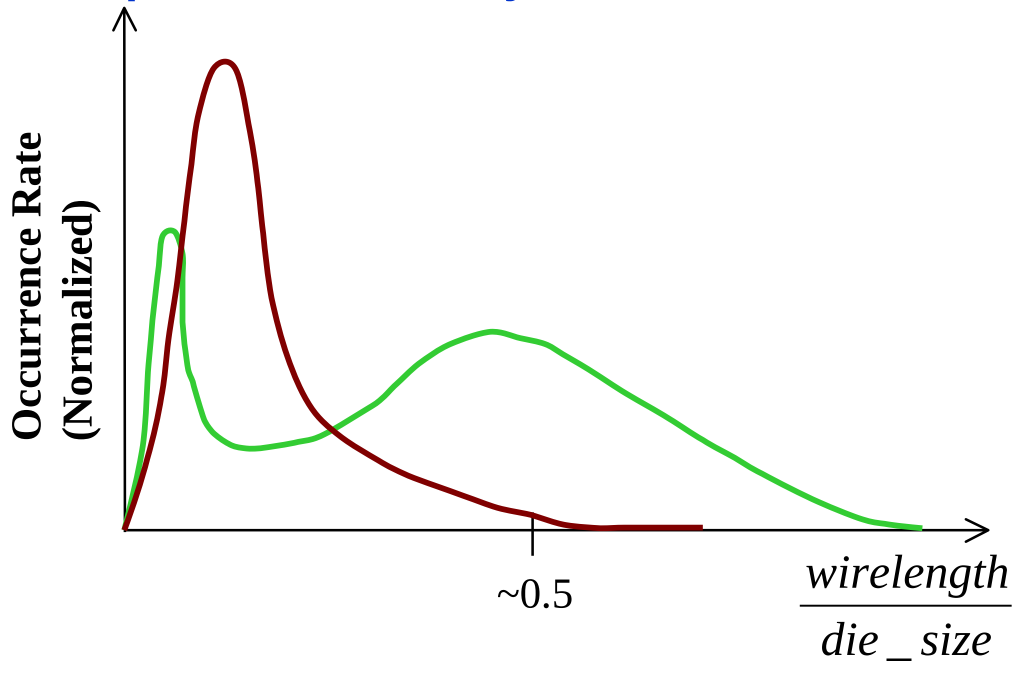


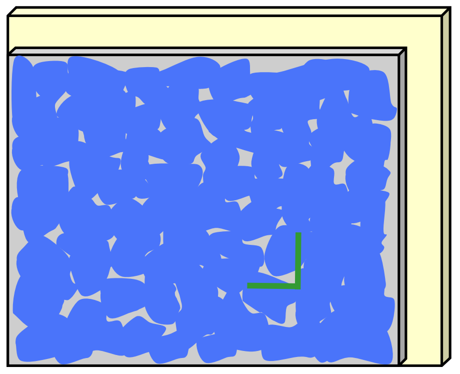
Larger chip with finer feature sizes

Softer *Fabrics*

With soft, flexible Fabrics, the system assembly can more thoroughly exploit the available technology

The interconnect problem is controlled via: soft boundaries for area re-shaping; re-synthesis and re-mapping for timing; smart wires; and top-down specified Fabric synthesis





Superior timing, power and cost

What methodology will prevail?

Objected oriented

- Top down
- Correct by construction
- Highly sophisticated user
- Intelligence of the system is more in the tools than the IP
- Interfaces are carefully tuned for application
- High performance oriented

Component-oriented

- Bottom up
- Robust and error tolerant
- Relatively unsophisticated user
- Intelligence is in the IP, tools are relatively simple
- Standard ``plug and play'' interfaces for a broad range of applications
- High productivity oriented

Outline of issues

Why components?

- Raw silicon capability
- Design productivity

What type of components?

- What size of component?
- What type/capability of component?

How will they be implmented?

- Review of implementation alternatives

What methodologies are likely to succeed?

- Object-oriented vs. component-oriented

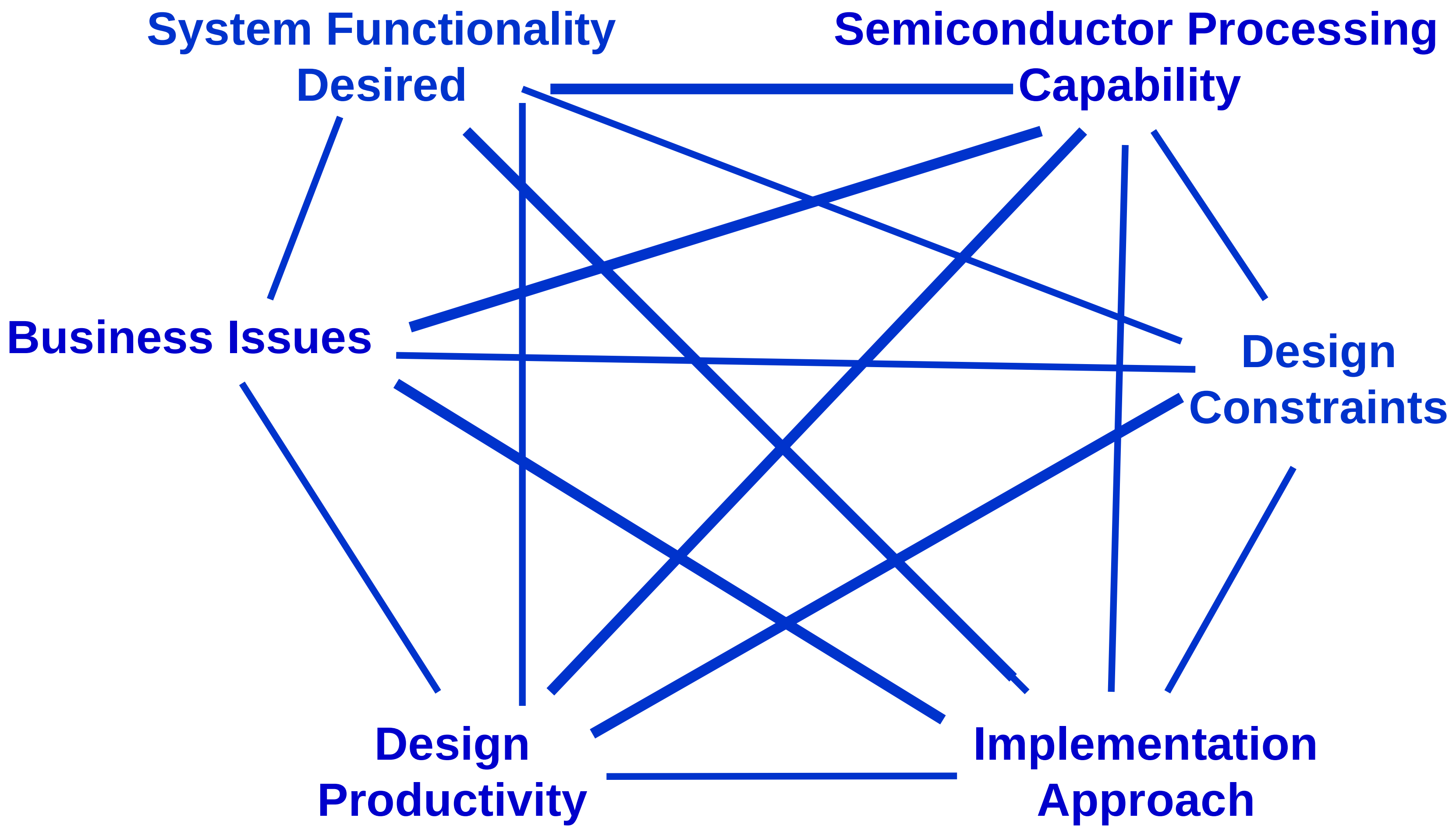
Who are the players?

- foundries,
- fabless semiconductor, 3rd party IP providers, vertical semiconductor,
- system companies

Which design styles are likely to predominate

- Time-to market (productivity)
- Features
  - Process portability
  - In-field up-gradabilty, programmability
  - Quality of results

Interrelationship of issues


System Functionality Desired
Semiconductor Processing Capability
Business Issues
Design Constraints
Design Productivity
Implementation Approach