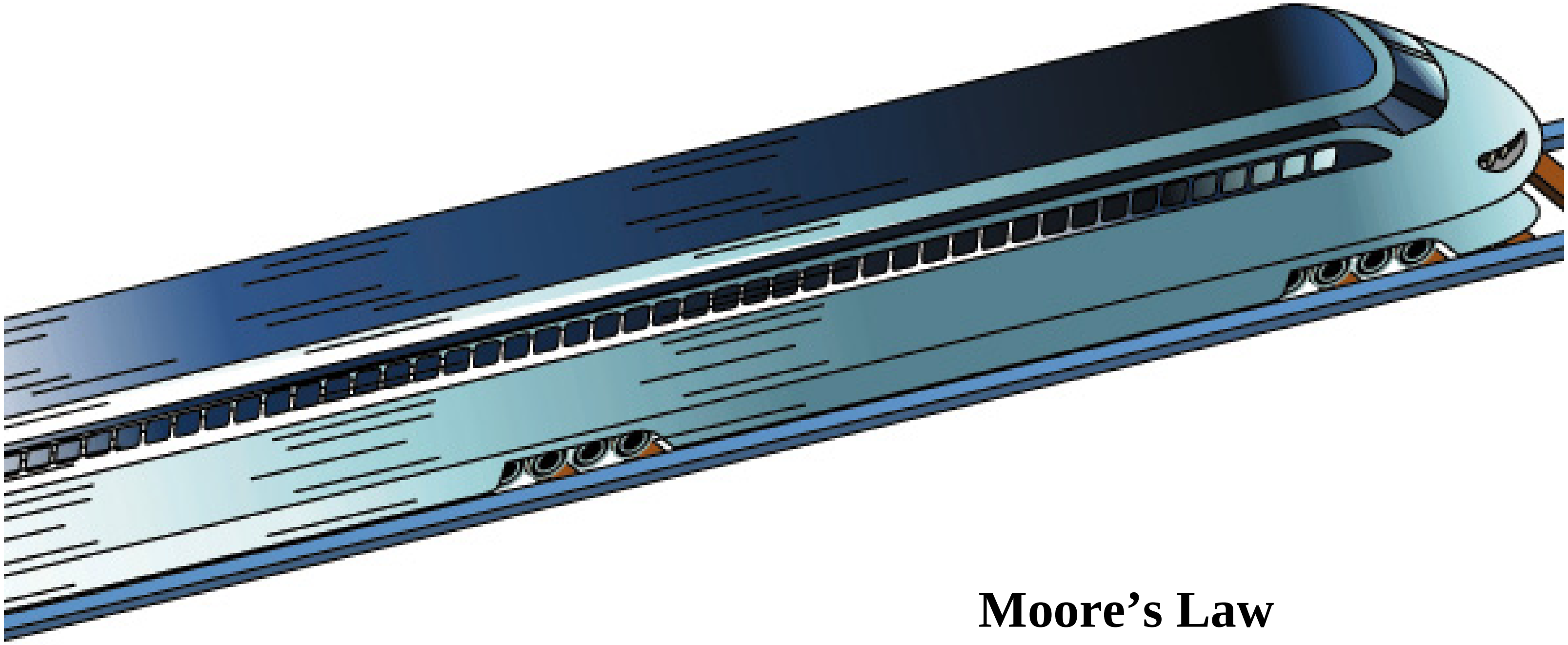
Design Reuse: Can It Halt the SoC Train Wreck

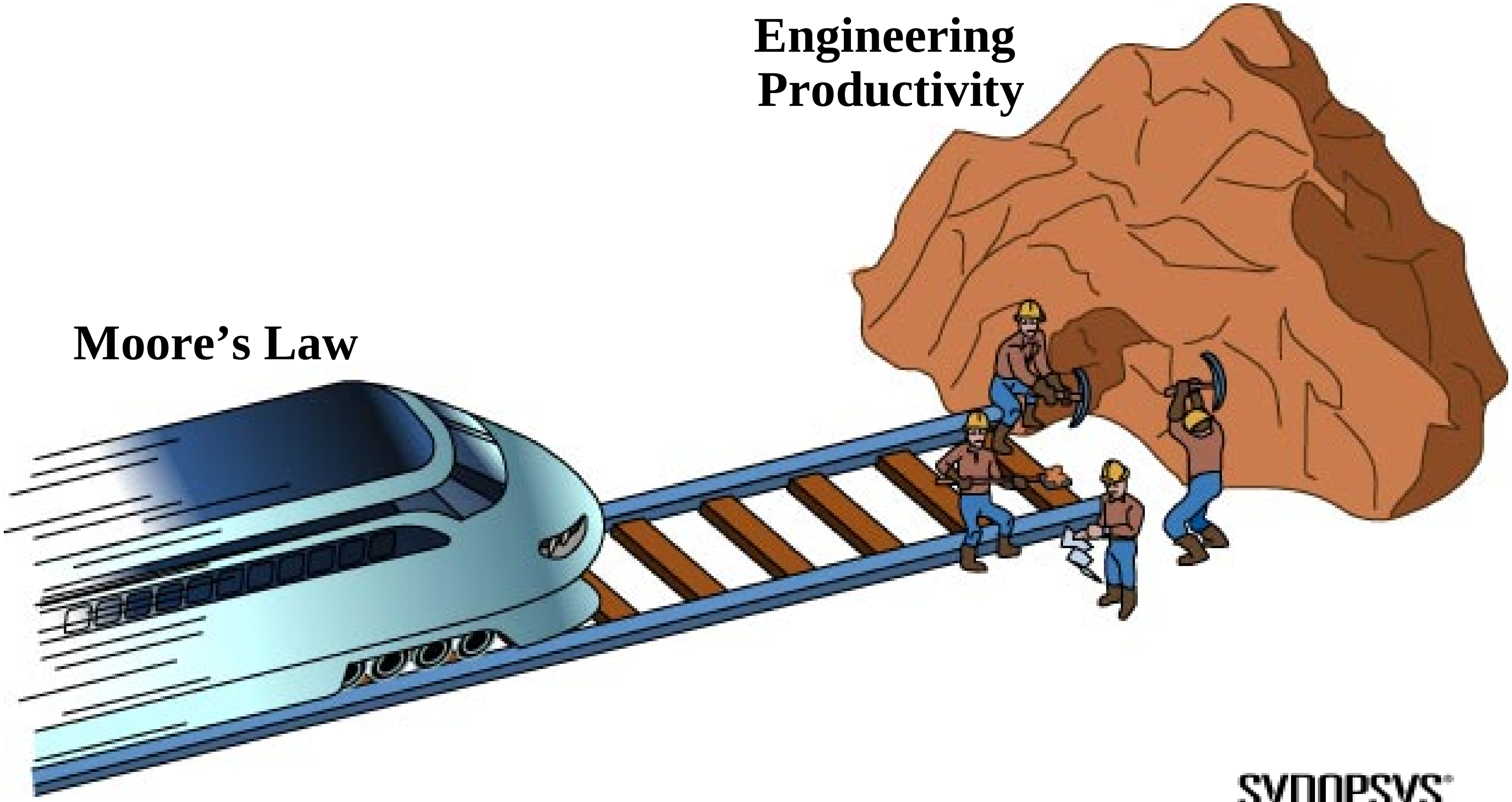
Mike Keating
VP of R&D
Design Reuse

The Irresistible Force



Moore’s Law

The Unmovable Object



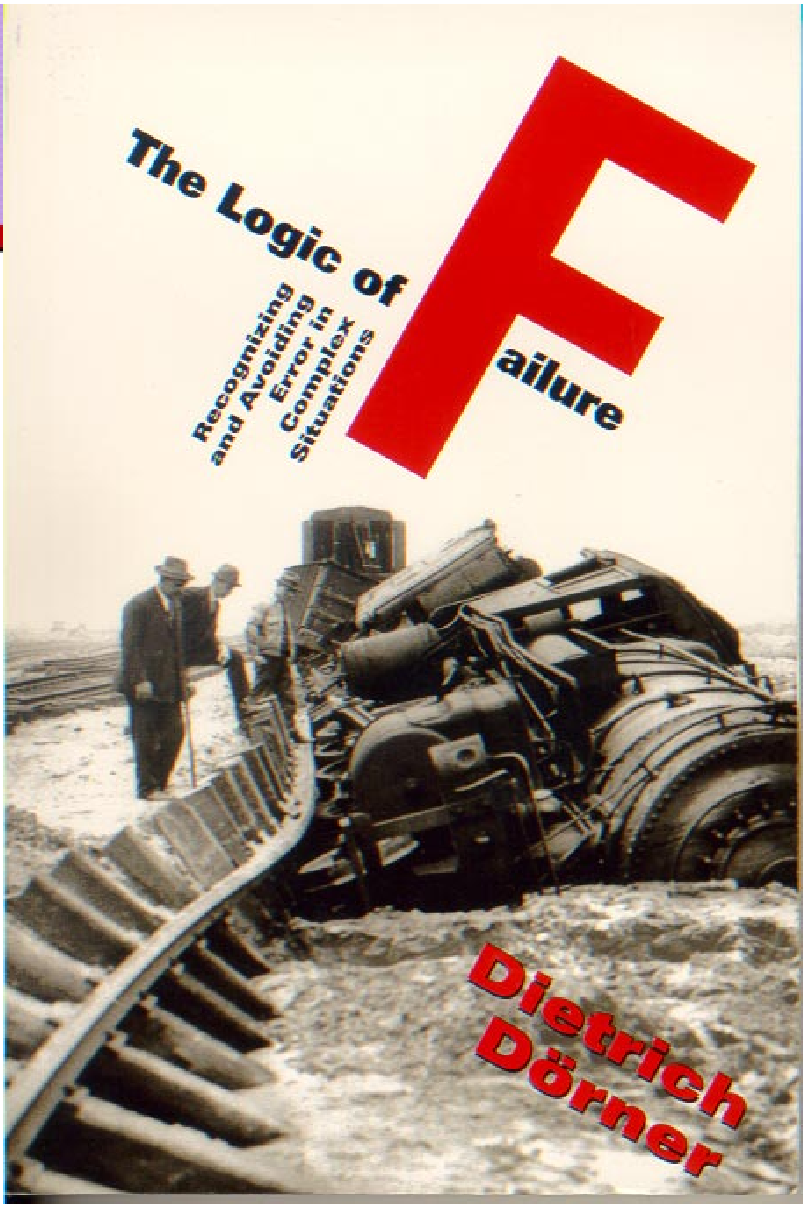
Engineering Productivity

Moore’s Law

Outline

- How big is the problem
- Internal Reuse
- Third Party IP
- Abdallah Tabbara’s questions
- Standard View of Design for Reuse

Problems of Scale



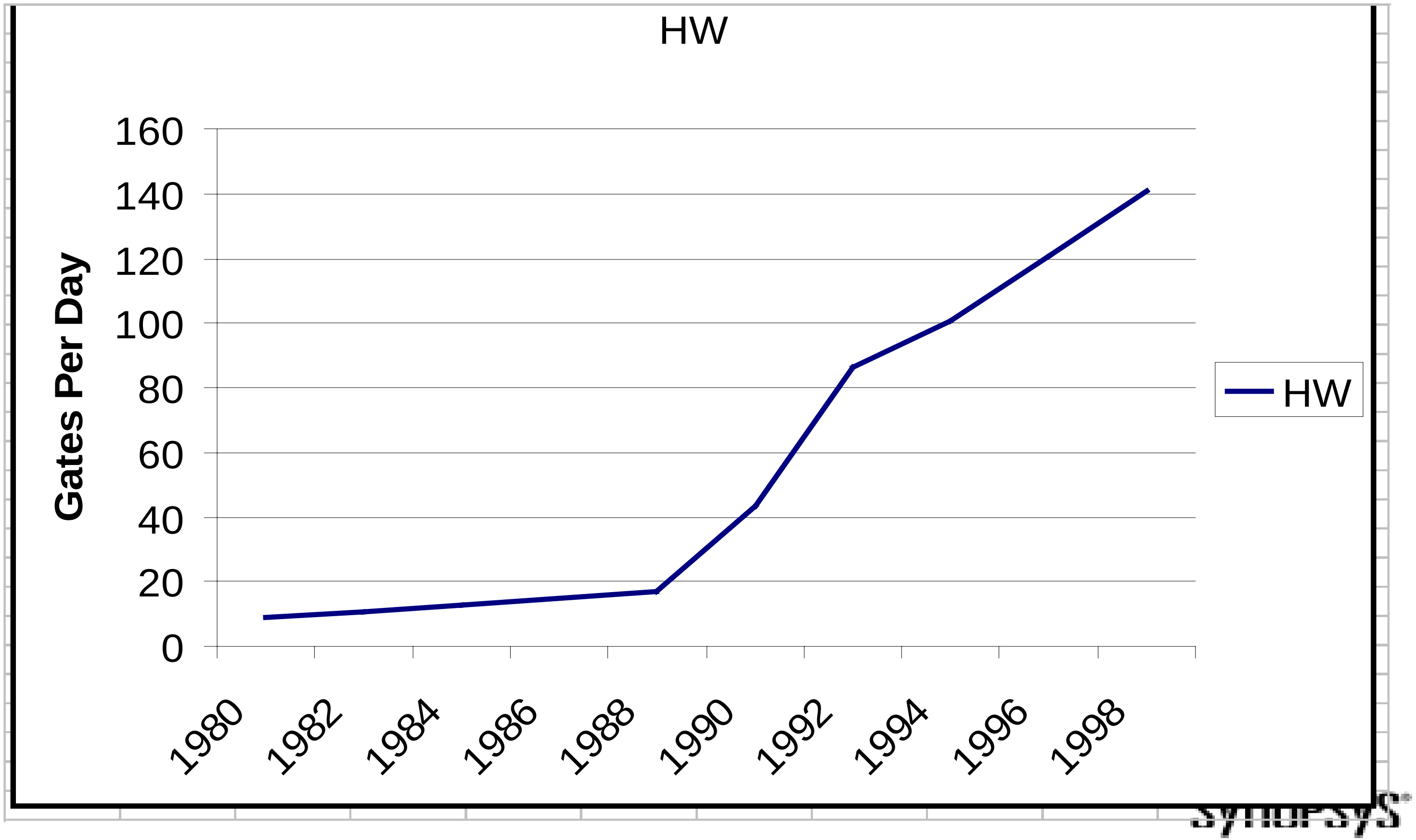
It is human nature to underestimate exponential growth

Plate Tectonics—
Shifts in IC Landscape

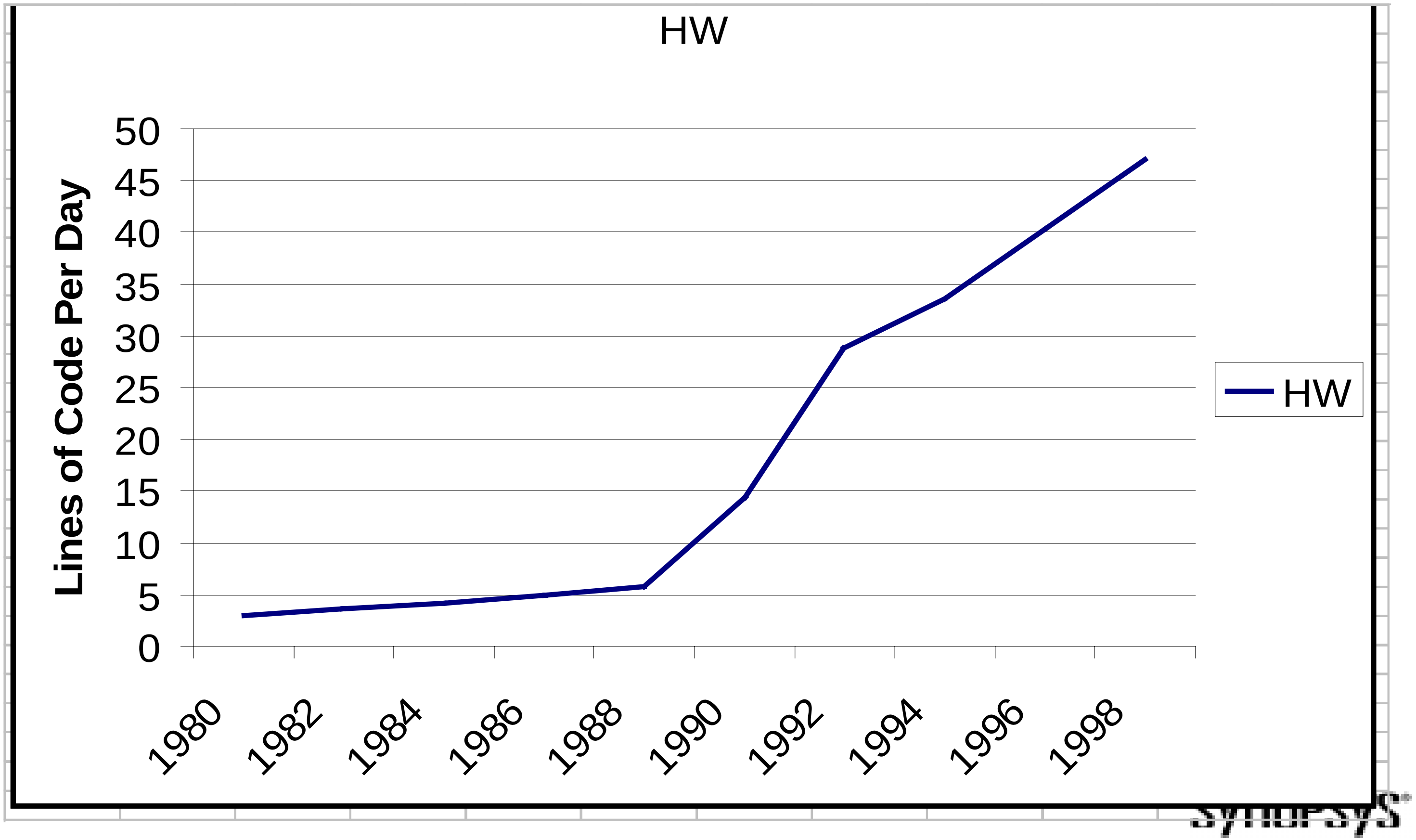
- System houses
  - –Using higher abstraction to deal with Moore’s Law
  - –Want to hand off specifications to semiconductor house
- Semiconductor Houses
  - –Doing many of the large SoC designs today
  - –Can they track Moore’s Law?

The SoC Battle is being fought in the Semiconductor Companies

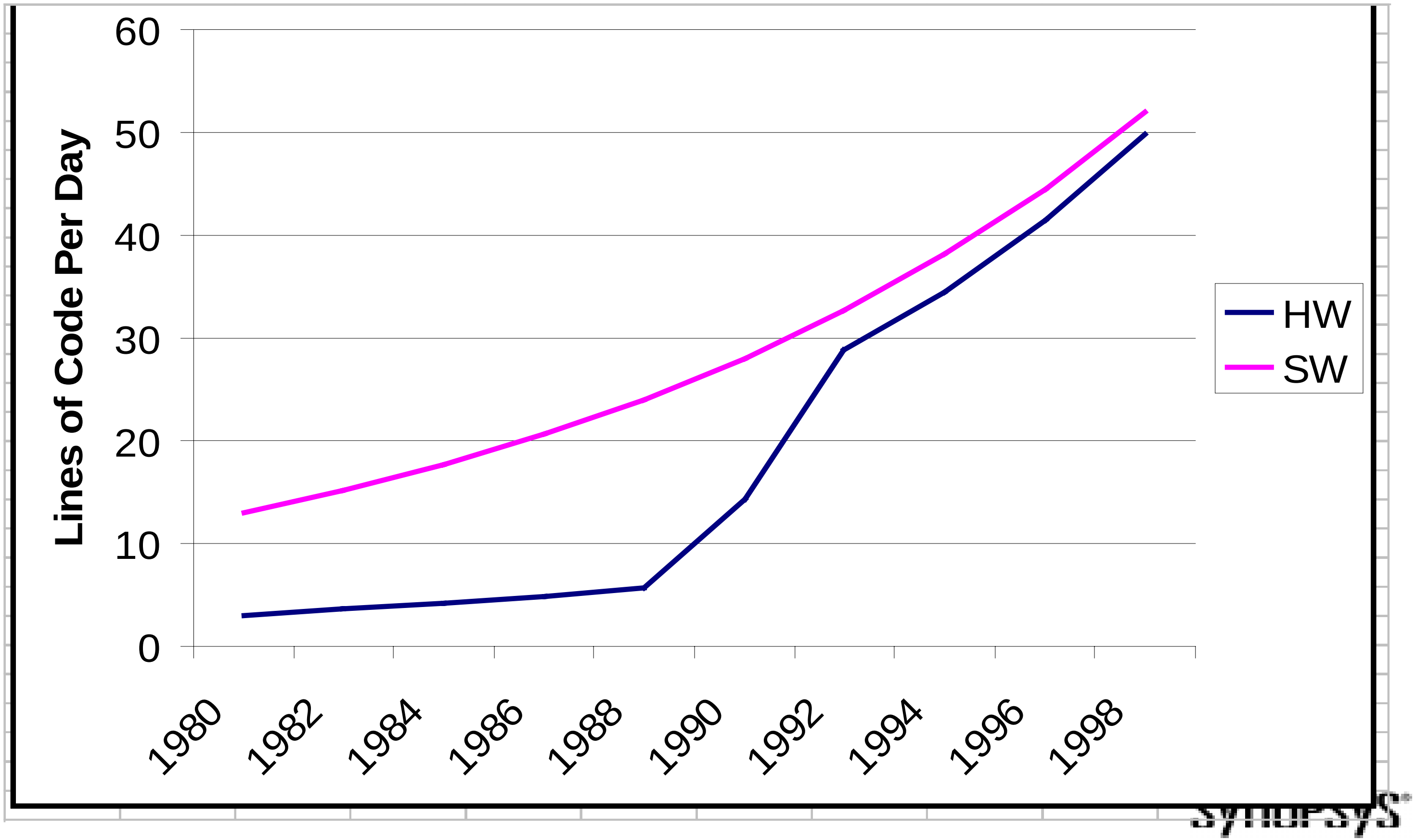
HW Productivity - Gates



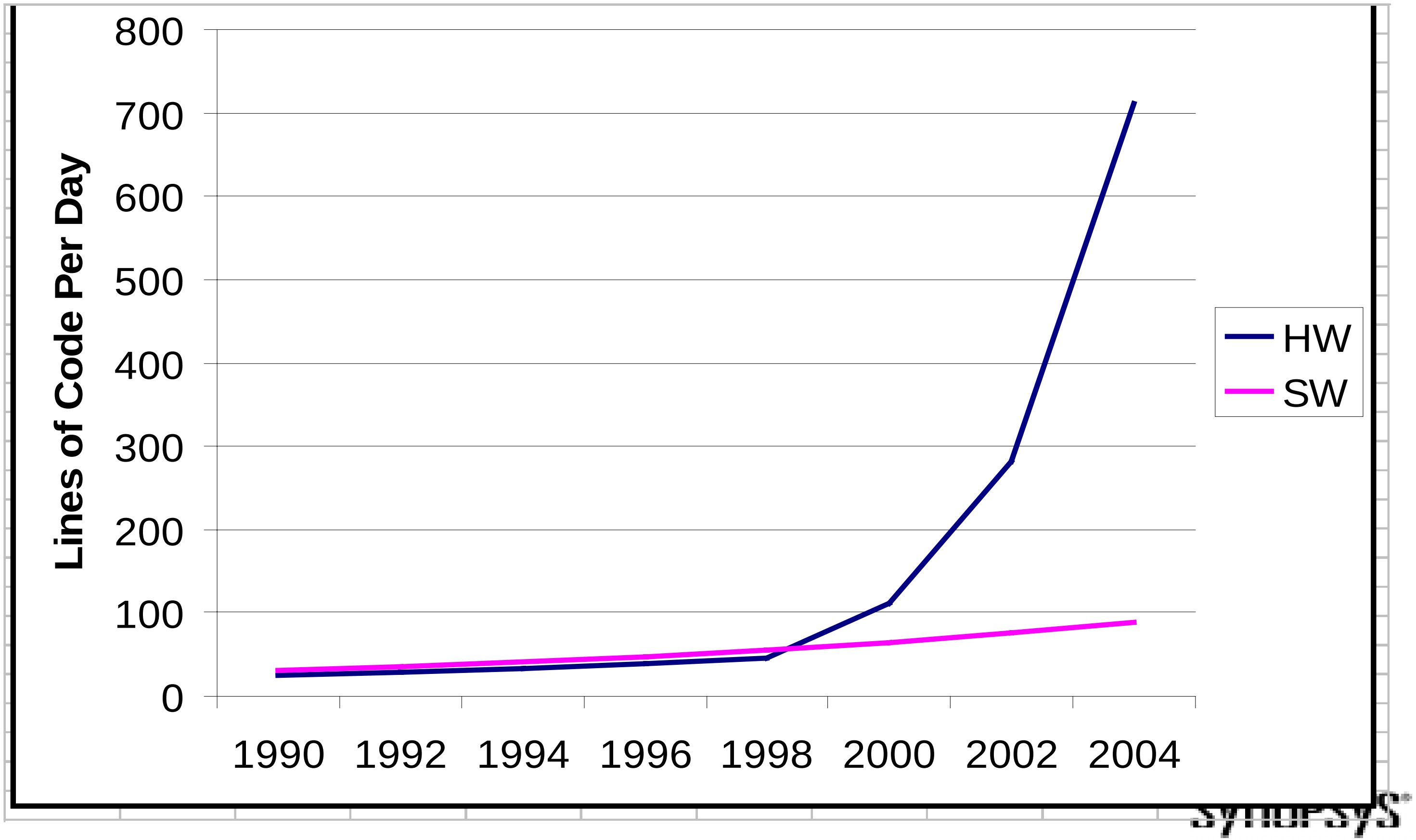
HW Productivity - LOC



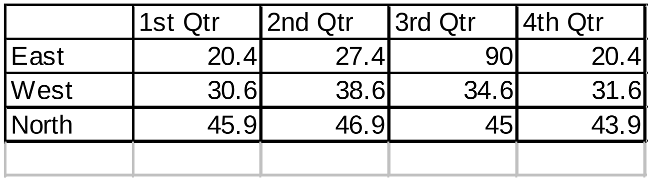
HW and SW Productivity



HW and SW Productivity

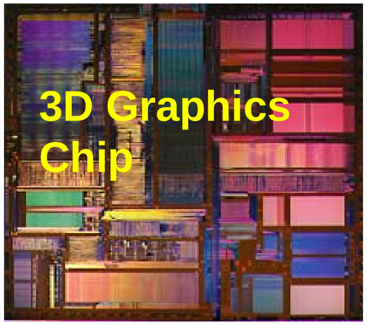


HW and SW Code Size



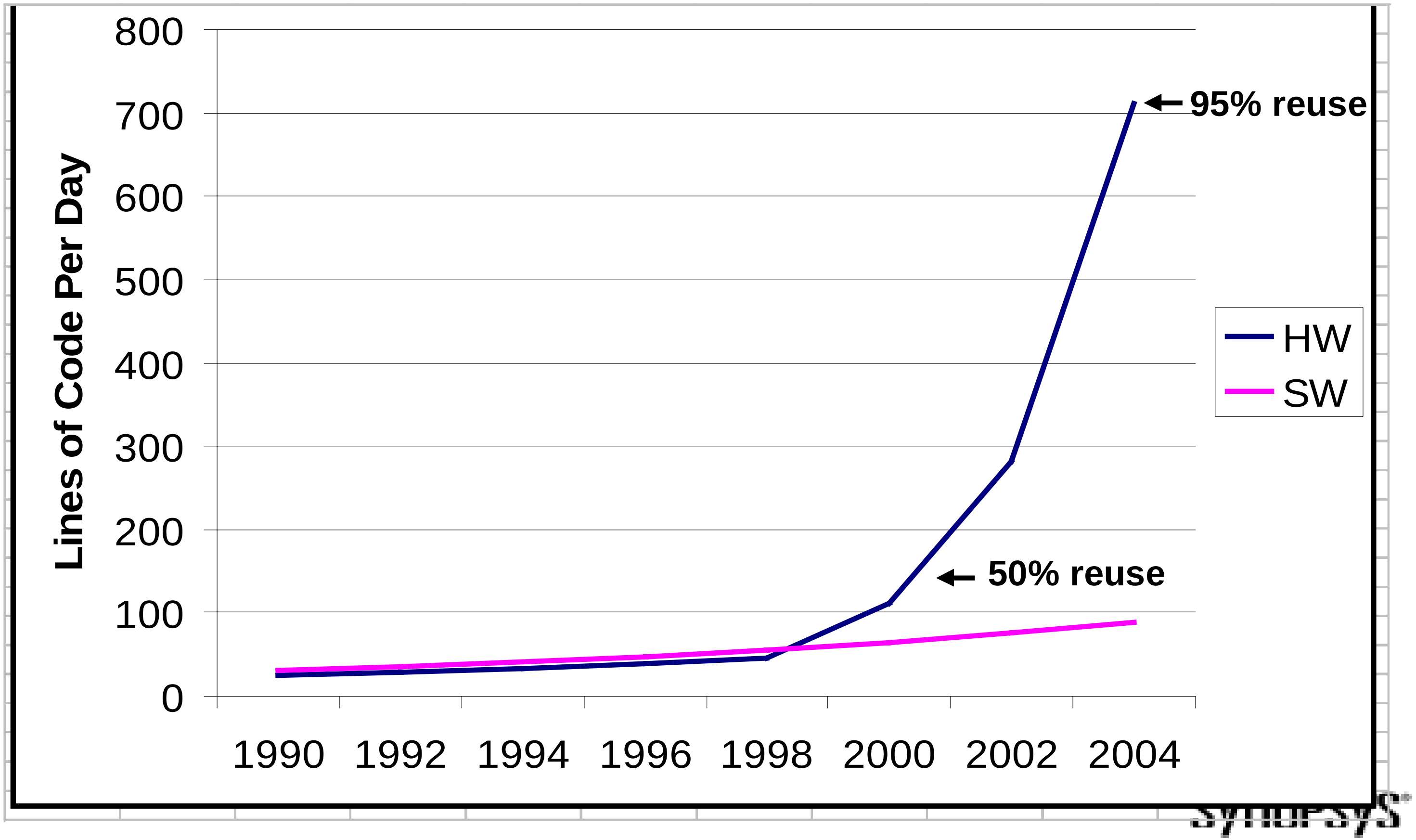
	1st Qtr	2nd Qtr	3rd Qtr	4th Qtr
East	20.4	27.4	90	20.4
West	30.6	38.6	34.6	31.6
North	45.9	46.9	45	43.9

= 300k lines of code



= 300k lines of code

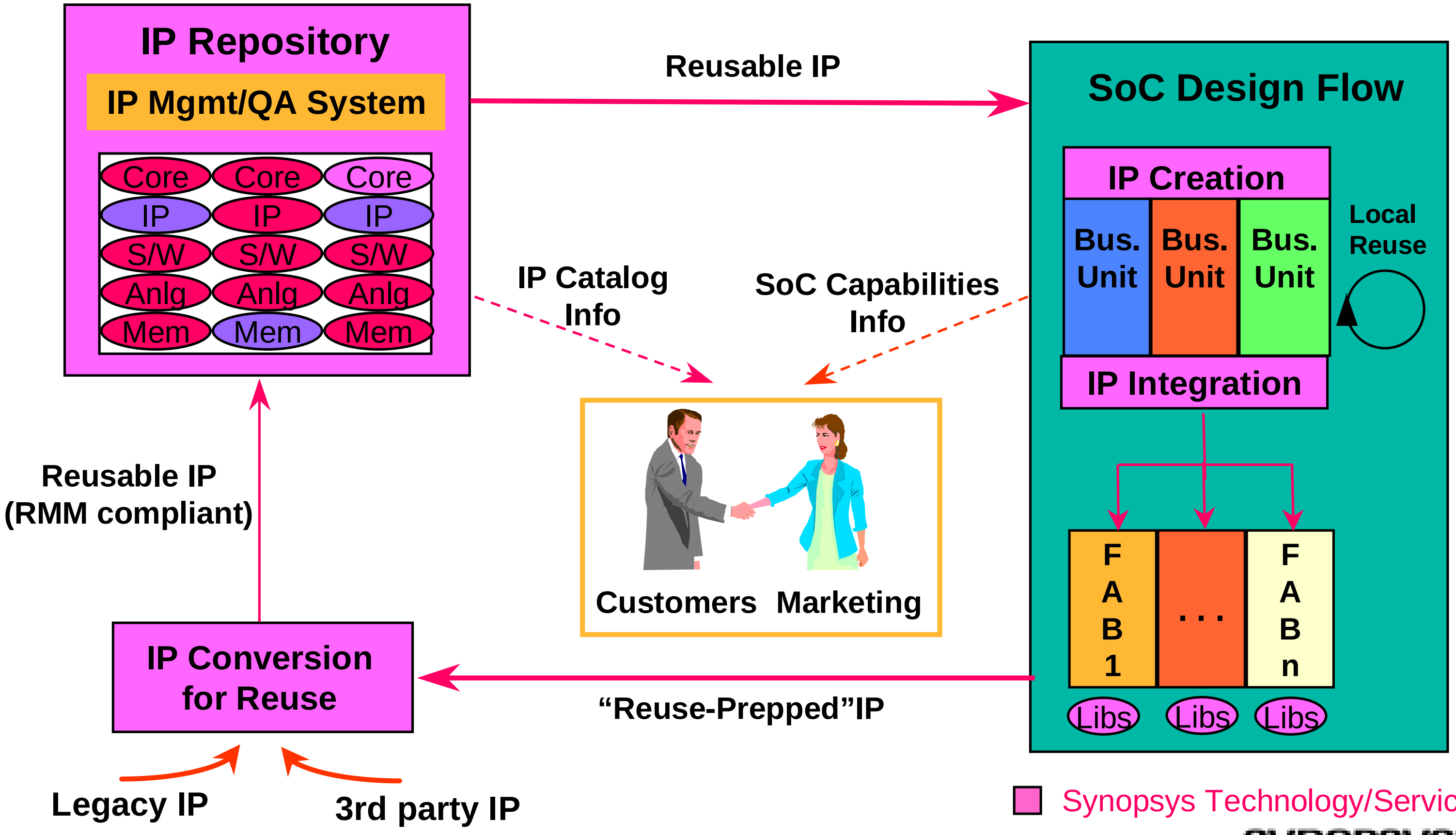
The Inflection Point



We can’t follow software any more

How Will the Semiconductor Companies Achieve This Level of Reuse?

Consensus on the Model



IP Repository

IP Mgmt/QA System

Core	Core	Core
IP	IP	IP
S/W	S/W	S/W
Anlg	Anlg	Anlg
Mem	Mem	Mem

Reusable IP

SoC Design Flow

IP Creation

Bus. Unit

Bus. Unit

Bus. Unit

Local Reuse

IP Integration

FAB 1

. . .

FAB n

Libs

Libs

Libs

IP Catalog Info

SoC Capabilities Info

Customers

Marketing

Reusable IP (RMM compliant)

IP Conversion for Reuse

“Reuse-Prepped”IP

Legacy IP

3rd party IP

■ Synopsys Technology/Services

Reuse Experience to Date

- Assigned CAD team to investigate reuse
  - Planning large investment in tools, infrastructure
  - Major focus on integration flow for SoC designs
  - Excellent coding and design guidelines
- IP is still of poor quality
  - Does not follow coding and design guidelines
  - IP is very difficult to integrate

We are not on track to meet our goals

SYNOPSYS®

What’s Going Wrong

- **Management issues**

- **Engineering issues**

Management View

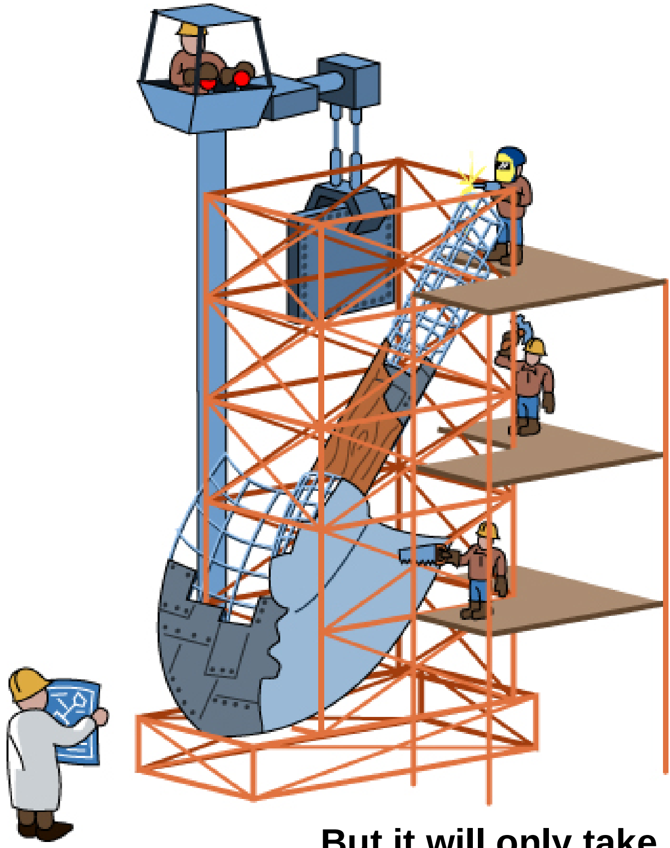
- **CAD can drive Reuse**

- **Reality: Reuse requires an integrated business, engineering, and CAD solution**

CAD-driven reuse initiatives will FAIL

SYNOPSYS®

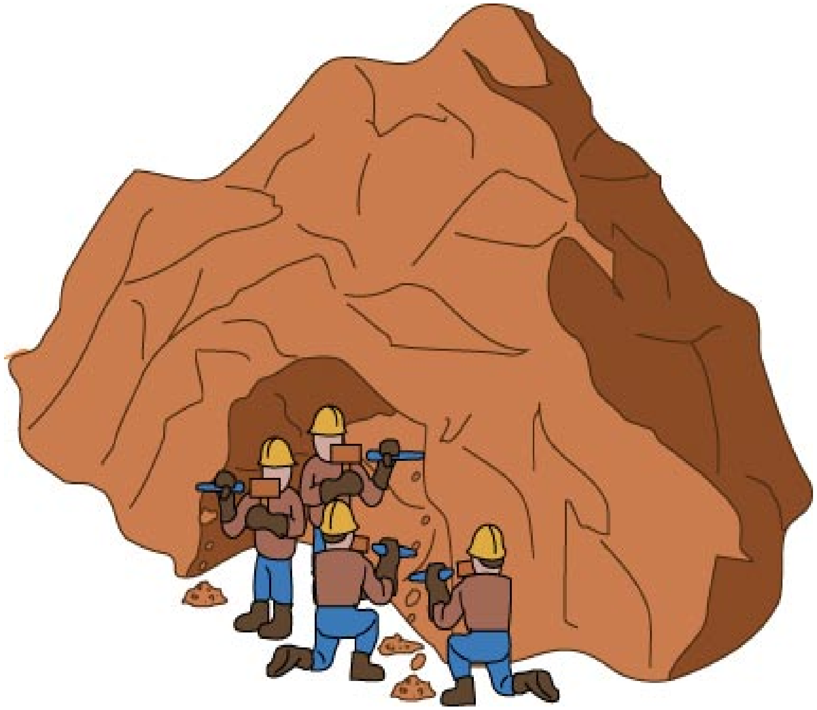
The CAD View



But it will only take 1 scoop!

- **Reuse is a tool and flow problem**
- **Reality: SoC design and reuse are about how we design blocks of IP**

Engineering View



But when we’re done, it will be a piece of art!

- Full custom design is key to success
- Reality: Good RTL, SC design is key to meeting TTM, QOR

What Do We Do About It

- We have a vision of the goal
- We need to create a vision of the path
- Engineers execute a greedy algorithm

Culture change must be incremental

Step 1: The value of Good Design

- Three fundamental rules
  - Fully synchronous RTL designs
  - Rigorous, well-planned verification
  - Develop specification before design, update after design
- They accelerate the design process
  - Synchronous designs speed up timing convergence
  - Bottom-up verification speeds up chip verification
  - Good specs reduce iterations through the flow

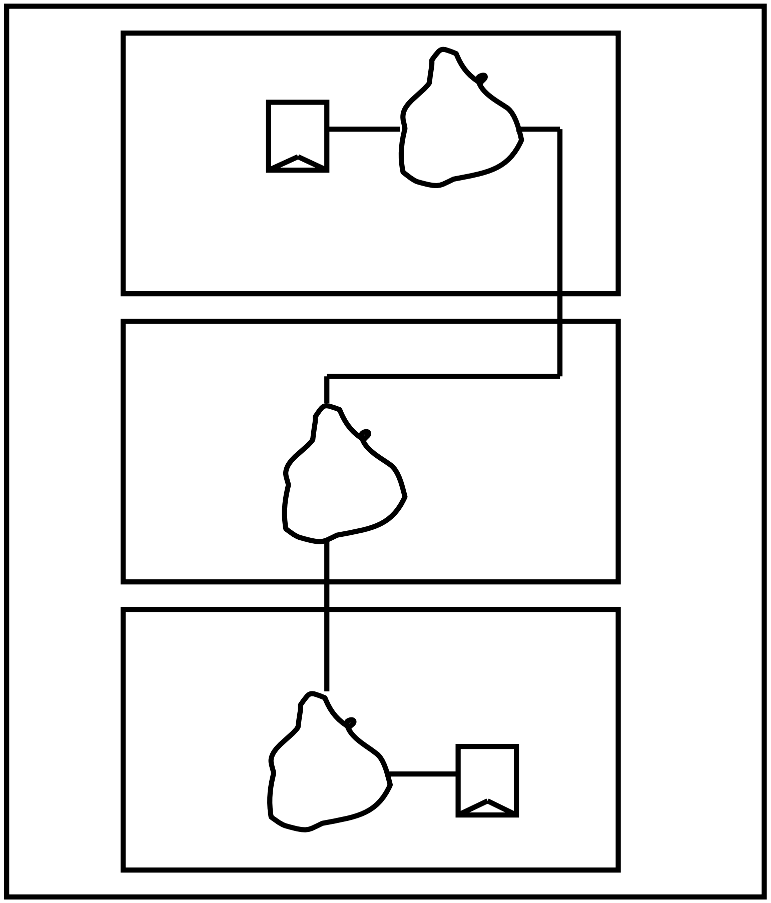
SYNOPSYS®

Case Study

- Redesign of processors
  - several processors from several companies
  - convert customer version to synthesizable

- Results
  - initially about 2x larger, higher power
  - use full custom memory, PC clock gating
  - within 10% in speed, area, power

Case Study - Importance of Locality



- **Without registers**
  - –timing is global
  - –optimization is flat
  - –runtimes are long
- **Result**
  - –more iterations
  - –slower iterations
  - –months wasted

Short list of key design rules

- Fully synchronous design
  - Register outputs (inputs also, if possible)
  - Flip flops, not latches
  - Single clock, single edge
  - No false paths, multi-cycle paths
  - No state-dependent timing
  - Use synchronous rams only

Short list of key design rules

- Use a library that has been through the flow before
- Verify as early and as completely as possible
  - use code coverage tools, aim for 100%
  - use industry standard testbenches and methodology
  - real code, random code, corner testing

SYNOPSYS®

Step 2: Extend to reuse

- Good designs can be reused as is
- More architecture work can extend uses
- Additional packaging can facilitate integration
  - better documentation, scripts, verification
  - package existing designs for easier integration

SYNOPSYS®

Step 3: Lower the Cost

- Include architecture work as part of system design
- Document design as it is being done
- Use tools to lower packaging costs

SYNOPSYS®

Step 4: The Rest of the Path

- Standardize processes, buses
- Automate compliance checking
- Infrastructure to support standards
- Create and import more IP
- Develop infrastructure to manage IP

Once the practicality and value of reuse is demonstrated, this will happen on its own

SYNOPSYS®

Managing the Culture Change

- Don’t dictate reuse—facilitate it
  - Need to show that reuse can accelerate projects
    - Pilot projects to show true cost/benefit of design for reuse
    - Start small, use good third-party IP to show value
  - Find design managers who are committed to design discipline
  - Can force the first time; after that, engineers must drive on their own
- Develop an organizational structure/economy that rewards design for reuse

Synopsys’ Role

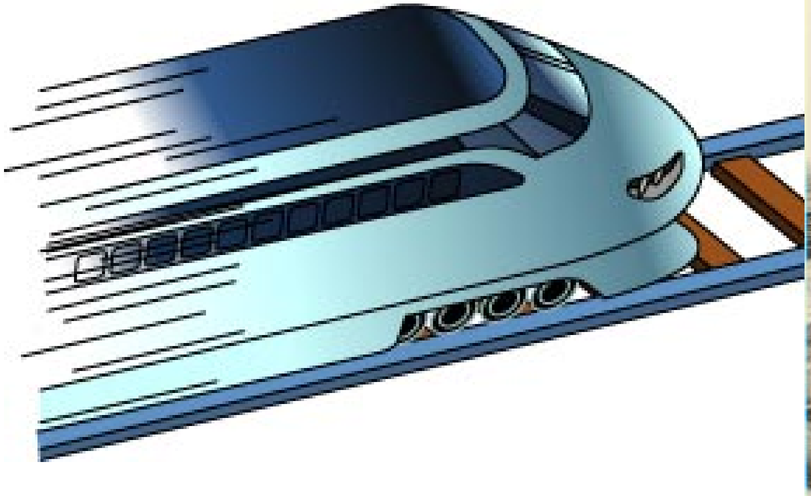
- Examples of quality IP
  - Foundation—thousands of customers
  - 8051, PCI—state-of-the-art ease of use
- Kick-starting internal reuse by redesign services
  - A7S—synthesizable ARM, including working silicon
- Reuse tools
  - AppBuilder, CoreMill, VeraCore
  - Helping engineers to develop, package, and use IP
- Documenting the Methodology—RMM

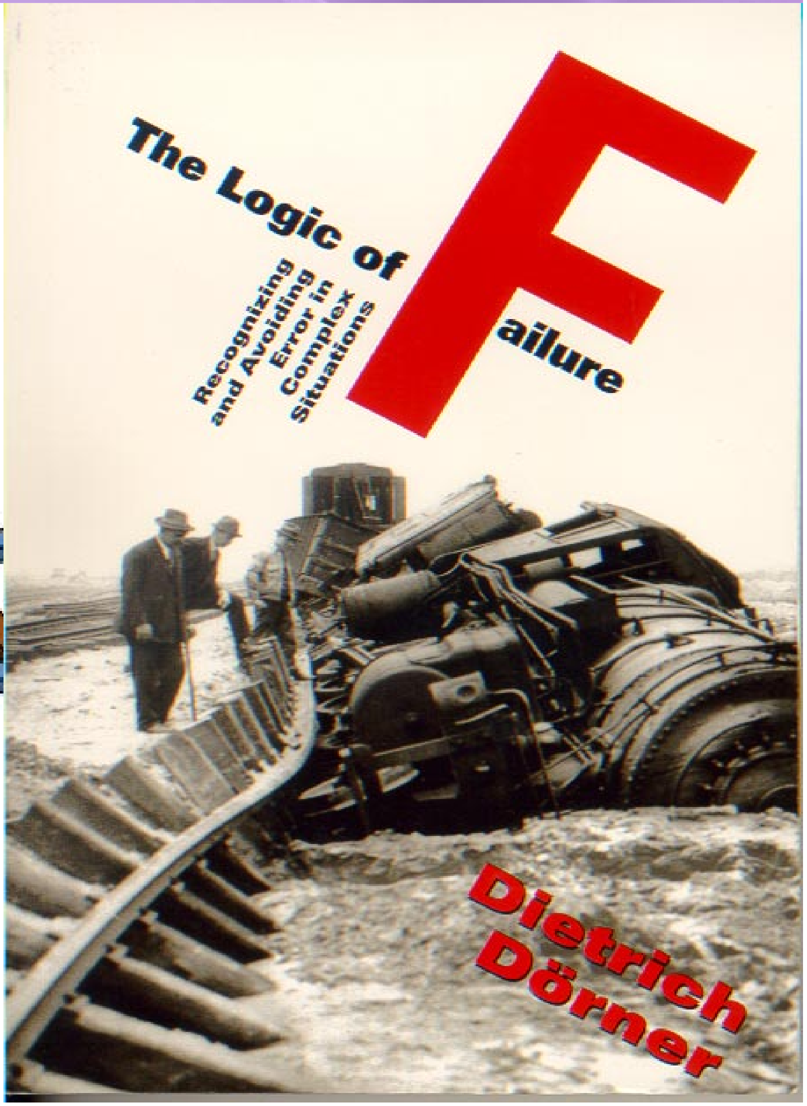
Summary—
Journey of a Thousand Miles

- **The SoC/reuse battle is urgent and critical**
- **It will be won or lost by engineering**
- **It means changing the engineering culture**
- **Culture change needs to be incremental**

We must take the right first steps NOW

SYNOPSYS®



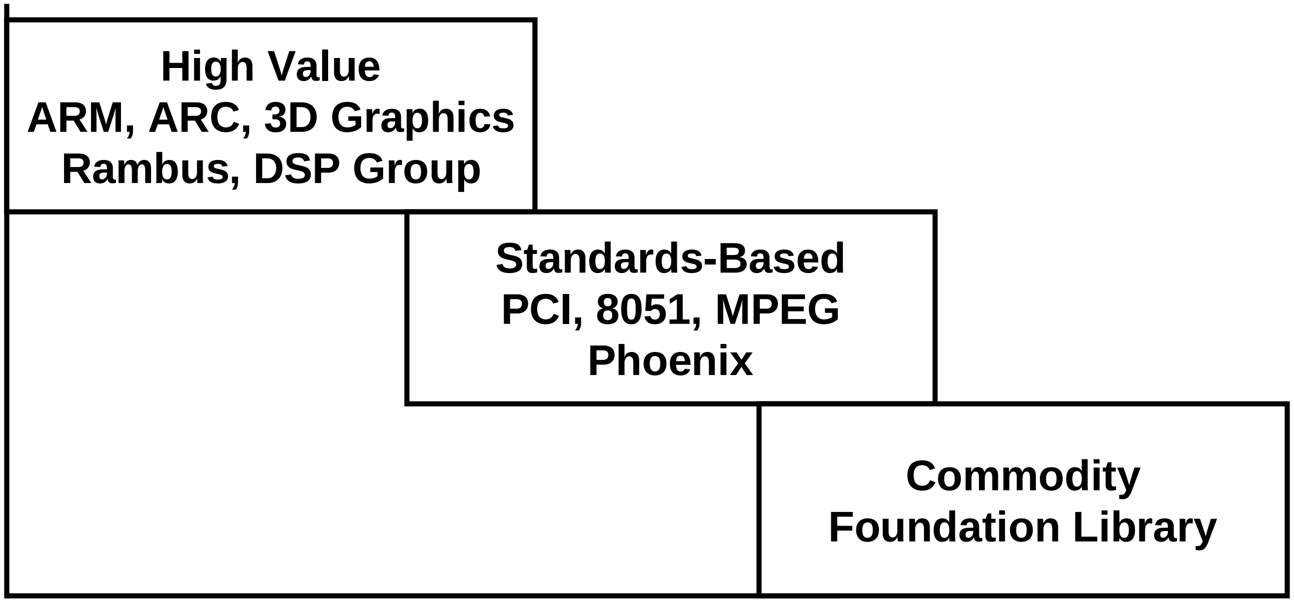

The Logic of Failure
Recognizing and Avoiding Error in Complex Situations
Dietrich Dörner

Supplemental Material

Taxonomy of IP

- Domain Independent
  - –ARM, PCI
- Domain Specific
  - –MPEG
- Application Specific
  - –Specialized block for Sega game

Third Party IP Landscape

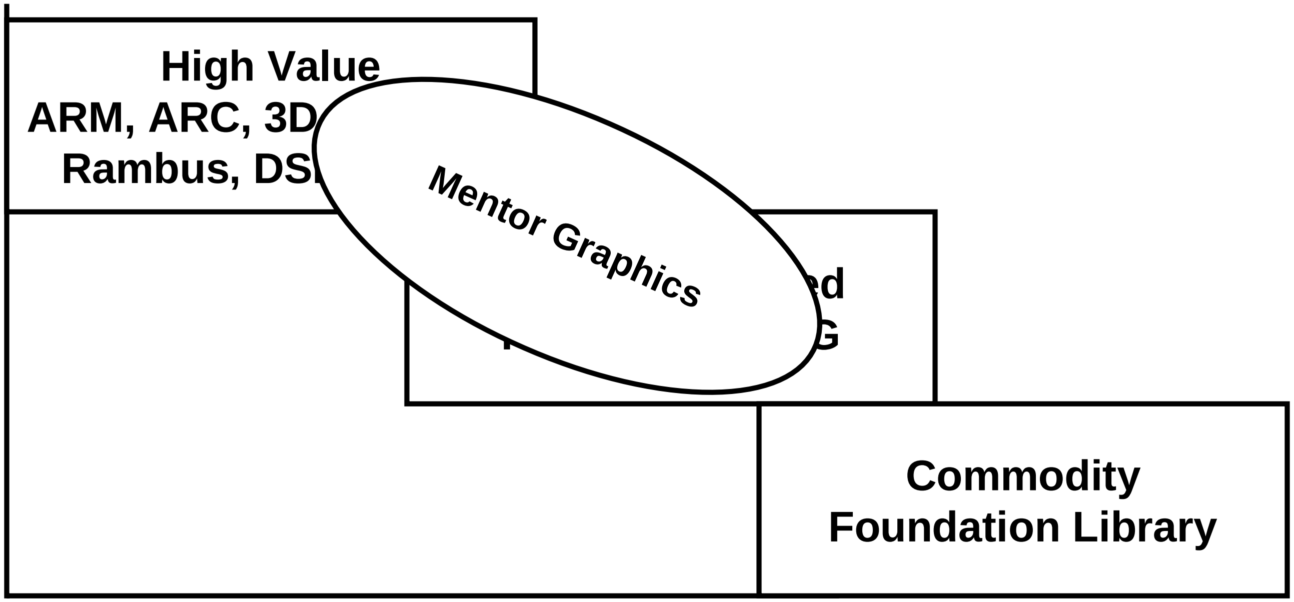


High Value
ARM, ARC, 3D Graphics
Rambus, DSP Group

Standards-Based
PCI, 8051, MPEG
Phoenix

Commodity
Foundation Library

Third Party IP Landscape



What’s Happening with 3rd Party IP

- ARM, DSP Group, Rambus, Snps doing well
- Phoenix profitable but not growing
- Rest are struggling

Why the Problems

- Not enough value for enough customers
- Most IP is hard to use
  - –Poor documentation
  - –Poor synthesis scripts
  - –Designs focused to one customer's needs
- Standards-based IP is not differentiated
  - –can't sell enough at $100k to justify investment in quality

SYNOPSYS®

Technical Issues

- what is a good component in terms of:
- - size and complexity
- - interface size and complexity
- - hard vs. firm vs. soft

- **what sort of interface specification is good for components**
- **issues that arise are:**
- **- synchronous vs. asynchronous timing at the chip level**
- **- register bounding of IPs**
- **- how robust are the specifications and components typically**
- **-**

- how good are the estimates for IPs in terms of power, performance, area
- - how do components vendors ensure that components will work correctly
- when incorporated into a design

Design for Reuse
The Standard Model

Mike Keating
July 28, 1998

Outline

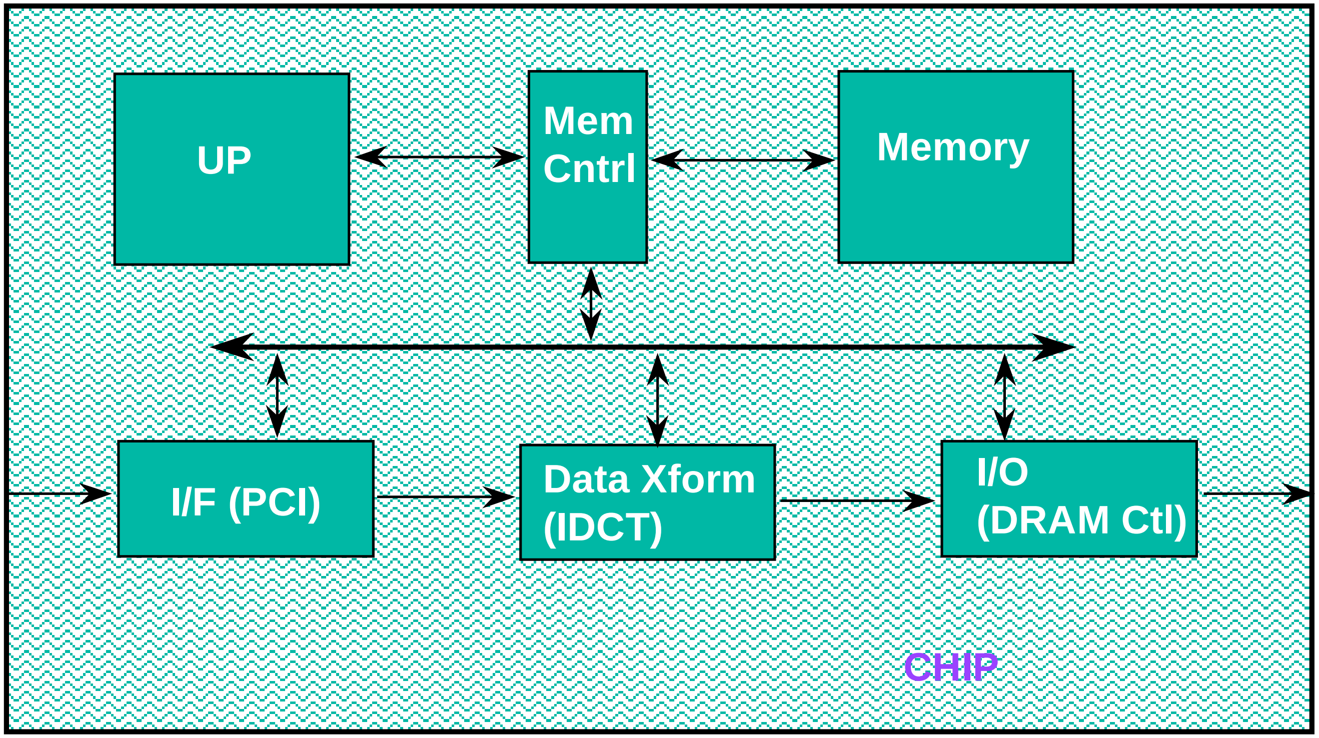
- Fundamental Goals and Approach
- Deliverables
- Architecture and External Interfaces
- Internals
- Manufacturing Test
- Summary

Fundamental Goal

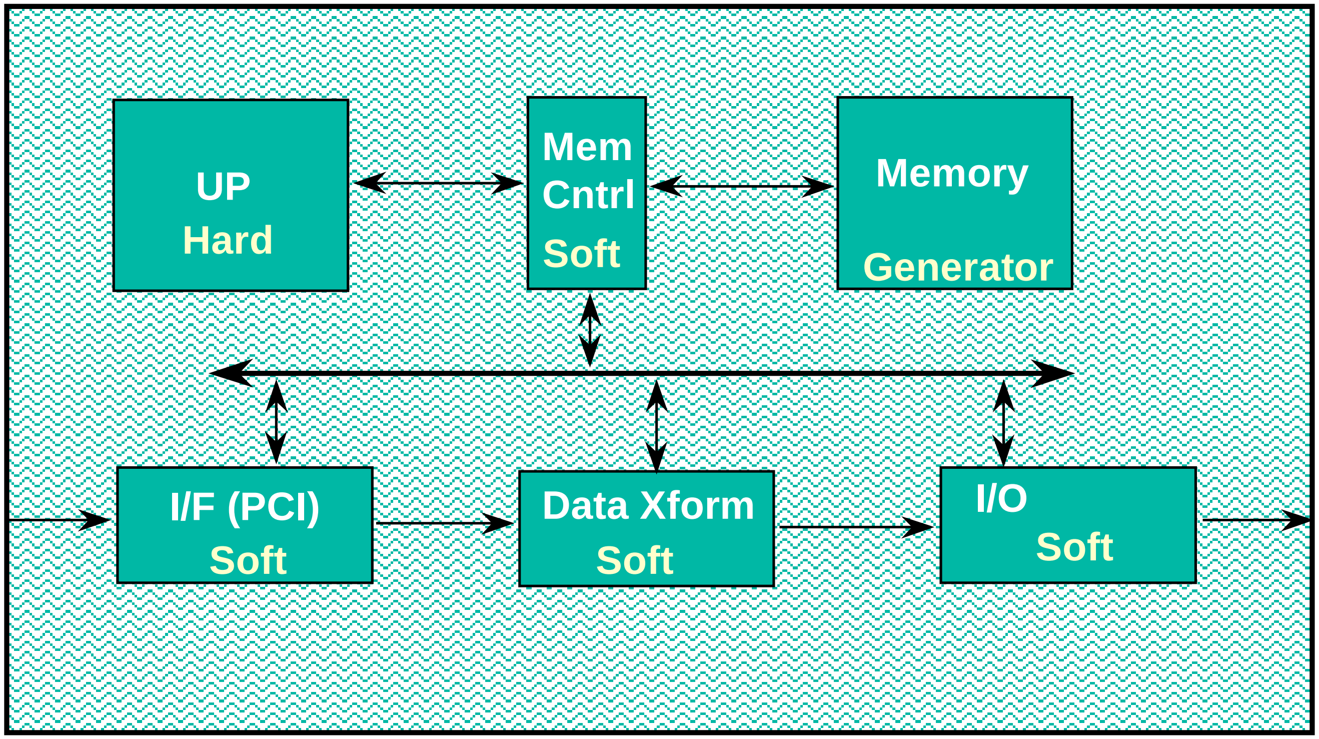
- **Enable team to get the right chips to market at the right time at the right cost**
- **Make SoC (ie, large chip) design much faster, easier**
- **Make development time, performance more predictable**
- **Solution: Reuse-based SoC Design**

SYNOPSYS®

SoC Canonical Design



SoC - Sources of Blocks



Fundamental Approach

- **Good IP design makes:**
  - **chip development time, performance predictable**
  - **chip integration effort linear in number of blocks**
- **Poor IP design makes:**
  - **chip development time, performance unpredictable**
  - **chip integration effort exponential in number of blocks**

SYNOPSYS®

- **THE SOC DESIGN BATTLE IS WON OR LOST ON THE QUALITY OF IP**

SYNOPSYS®

Corollary

- **More IP integration problems will be solved by improving IP than by improving the integration flow**

- **Rapid Hard IP integration tools are not as important as Hard IP standards (layer assignments, drc decks)**

Deliverables for IP

Soft vs Hard IP

- All Digital IP starts as soft IP
  - RTL model is reference implementation model
  - Specification can be written and/or executable (C++ or behavioral HDL)
  - Synthesizable RTL used to generate all hard views
  - Full custom blocks can be used to replace blocks in critical path

SYNOPSYS®

Soft vs Hard IP

- GDSII (Hard Version) is just another view
  - –Provided to expedite other designs on same process
  - –Porting is done via re-synthesis
  - –Full custom sections ported by porting tools or manually

IP Deliverables

- Documentation
- Behavioral models (processors, complex IP)
- Soft IP
  - RTL, Synthesis scripts, Verification suite
  - Works in integrators environment - simulator, synth, lib

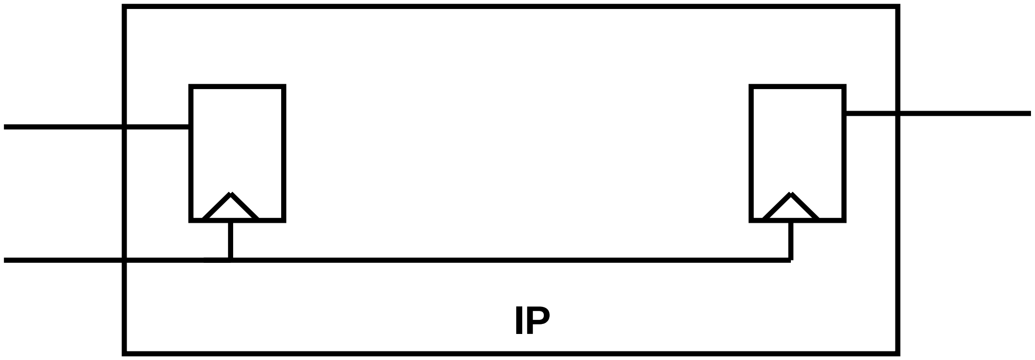
SYNOPSYS®

Deliverables

- Hard IP
  - GDSII compliant to process rules and standards
  - Integration models: simulation, timing, physical
  - Works in integrators environment - sim, physical tools
- Quality must be demonstrated
  - functional correctness must be demonstrated in silicon

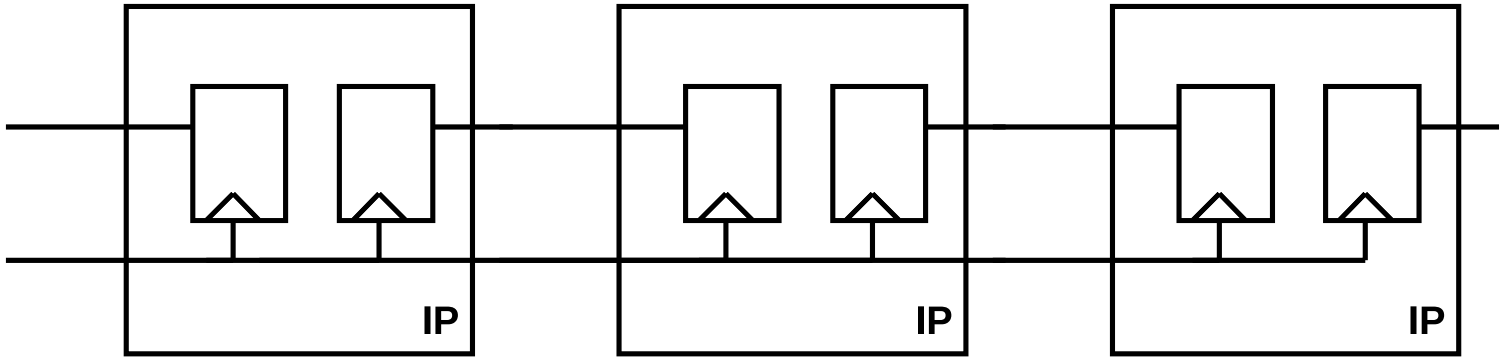
Design Guidelines for IP

Architecture and Interfaces



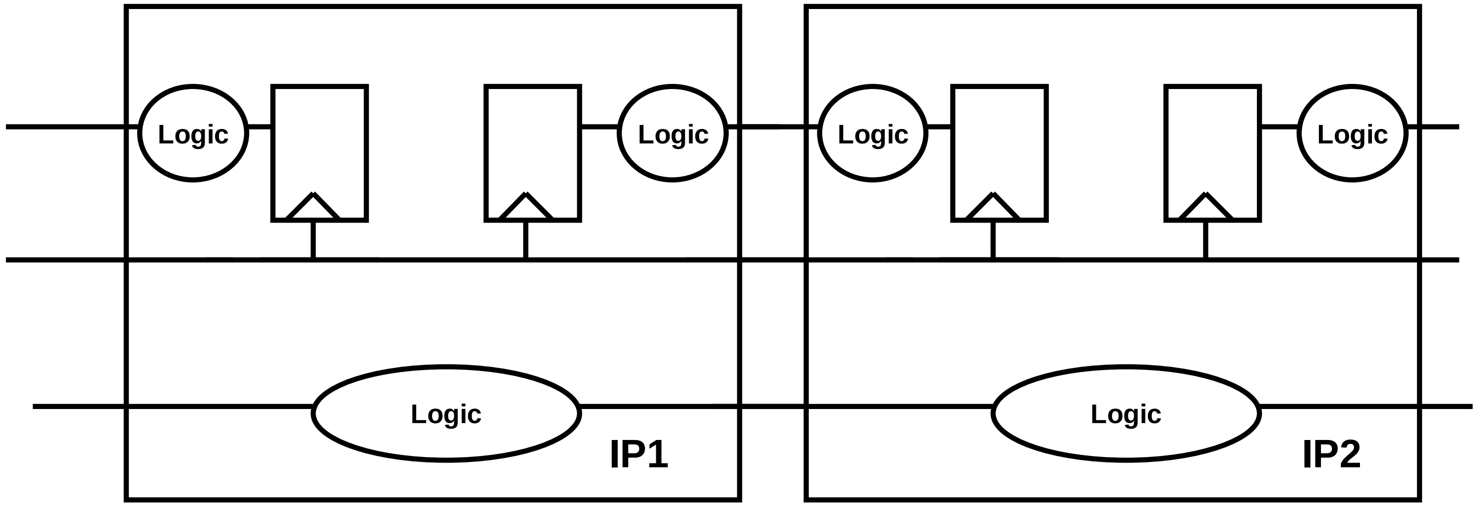
- All inputs go directly to flop input
  - assures consistent input timing, independent of state
- All outputs come directly from flop output
  - assures consistent output timing, independent of state

SoC Requires Scalability



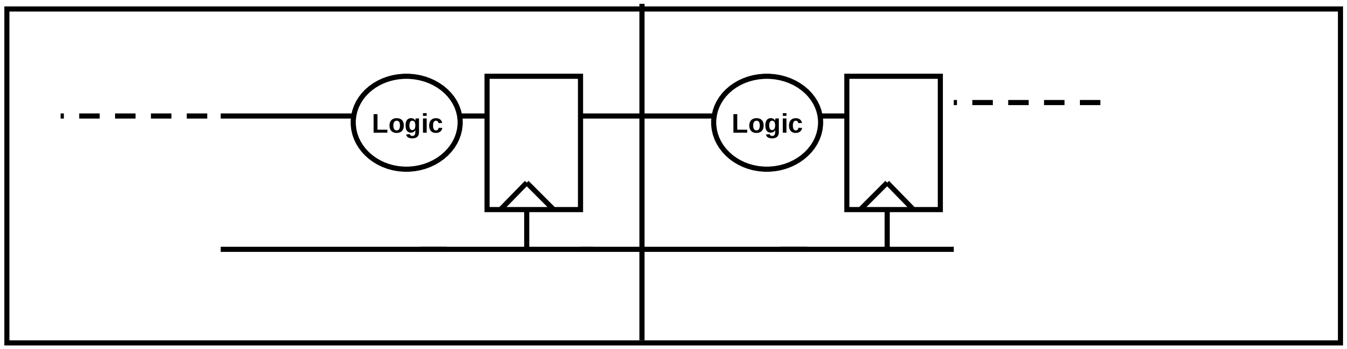
- **With this architecture:**
  - –**one full clock cycle for block-to-block signals**
  - –**chip performance limited only by block internal timing**
  - –**timing is simple, easy to analyze**
- **Architecture scales, allowing arbitrary blocks to plug and meet timing**

Designs that don’t Scale



- **Logic on inputs and combinational paths**
  - –**make chip level timing complex and IP-dependent**
  - –**top level timing may never converge**
  - –**chip performance limited by inter-block timing**

Internals -



- **Fully synchronous design**
  - –**Flop-based**
  - –**Single edge of single clock (wherever possible)**
  - –**Exceptions partitioned separately for easy analysis**
- **Assures portable designs that work with tools**
  - –**Synthesis, timing analysis, timing driven P&R**

SYNOPSYS®

Low Power Design for Reuse

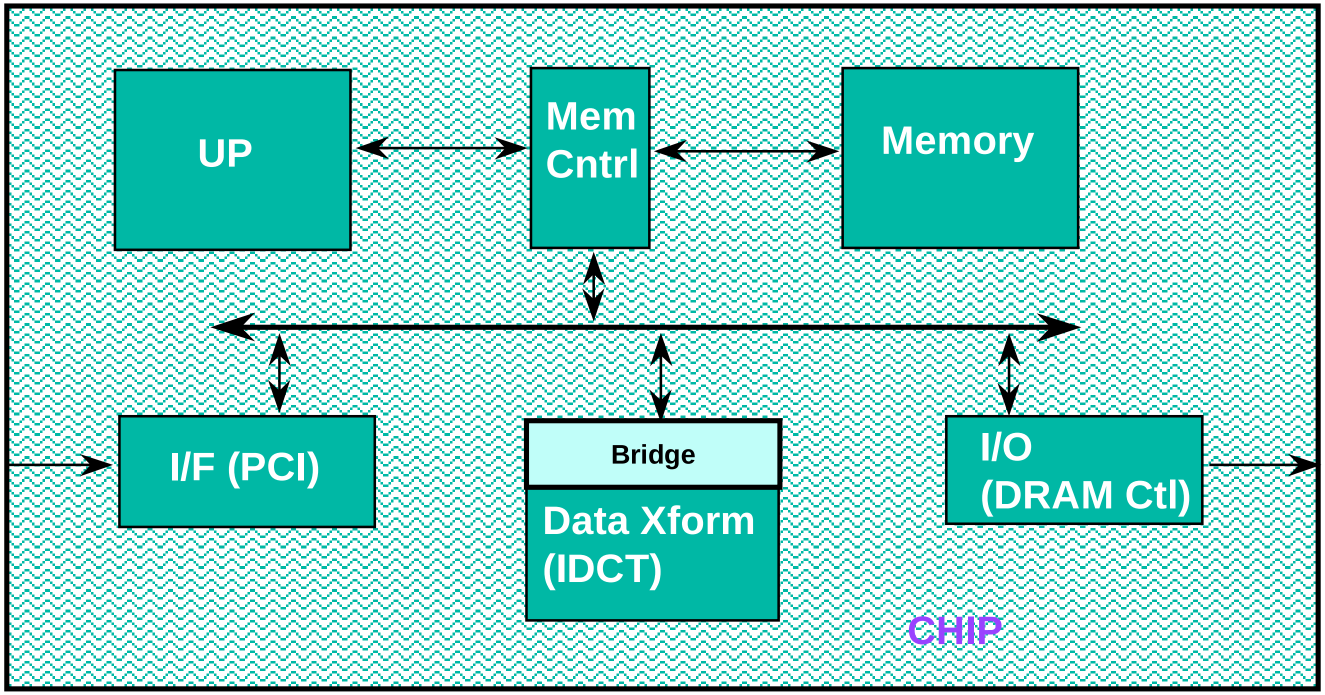
- **Lower supply voltage (P = CV^2f)**
  - **increase pipeline depth to compensate for slower speed**
- **Single clock + low power flop is lower power than Multiple clocks + latches**
  - **standard fully synchronous design**

Low Power Design for Reuse

- Gate sizing
  - optimize drive strengths for lowest that meets timing
  - low power library + power compiler
- Clock gating
  - per register - use power compiler
  - per block - separate out gating circuit from rest of design

SYNOPSYS

Bus Interface Standards



SYNOPSYS

Manufacturing Test

Manufacturing Test Strategy

- Memory uses BIST
- Logic use full scan
  - soft IP is verified for scan testability
  - hard IP includes scan and scan control port
  - Logic BIST may be provided
- Processors that don’t use scan/BIST
  - must provide test structures and test control port
  - usually boundary scan

Libraries and Memory

Prerequisites for Reuse

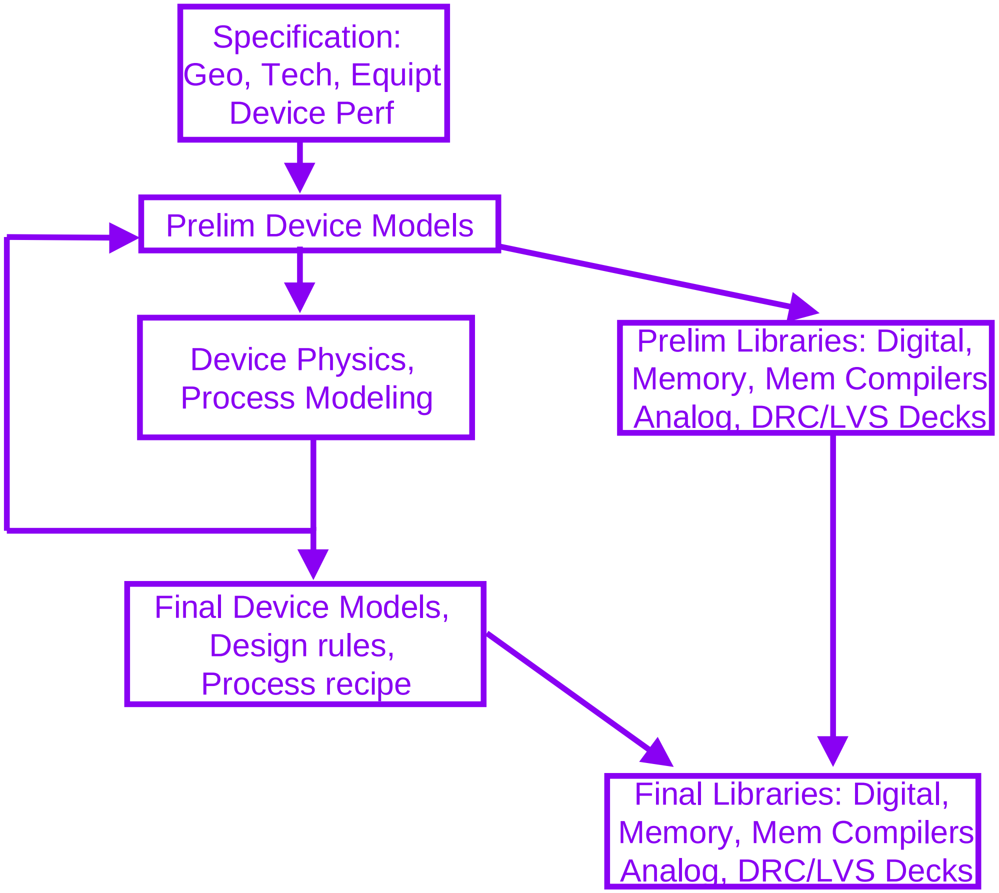
- Good libraries and Memory Compilers
  - synthesis-friendly, well supported
  - available early - BU's should not have to make their own
  - address the needs of the BU's
    - low power, high performance, high temp
    - SRAM, Flash
  - currently a mess, requires significant investment to fix

SYNOPSYS®

Prerequisites for Reuse

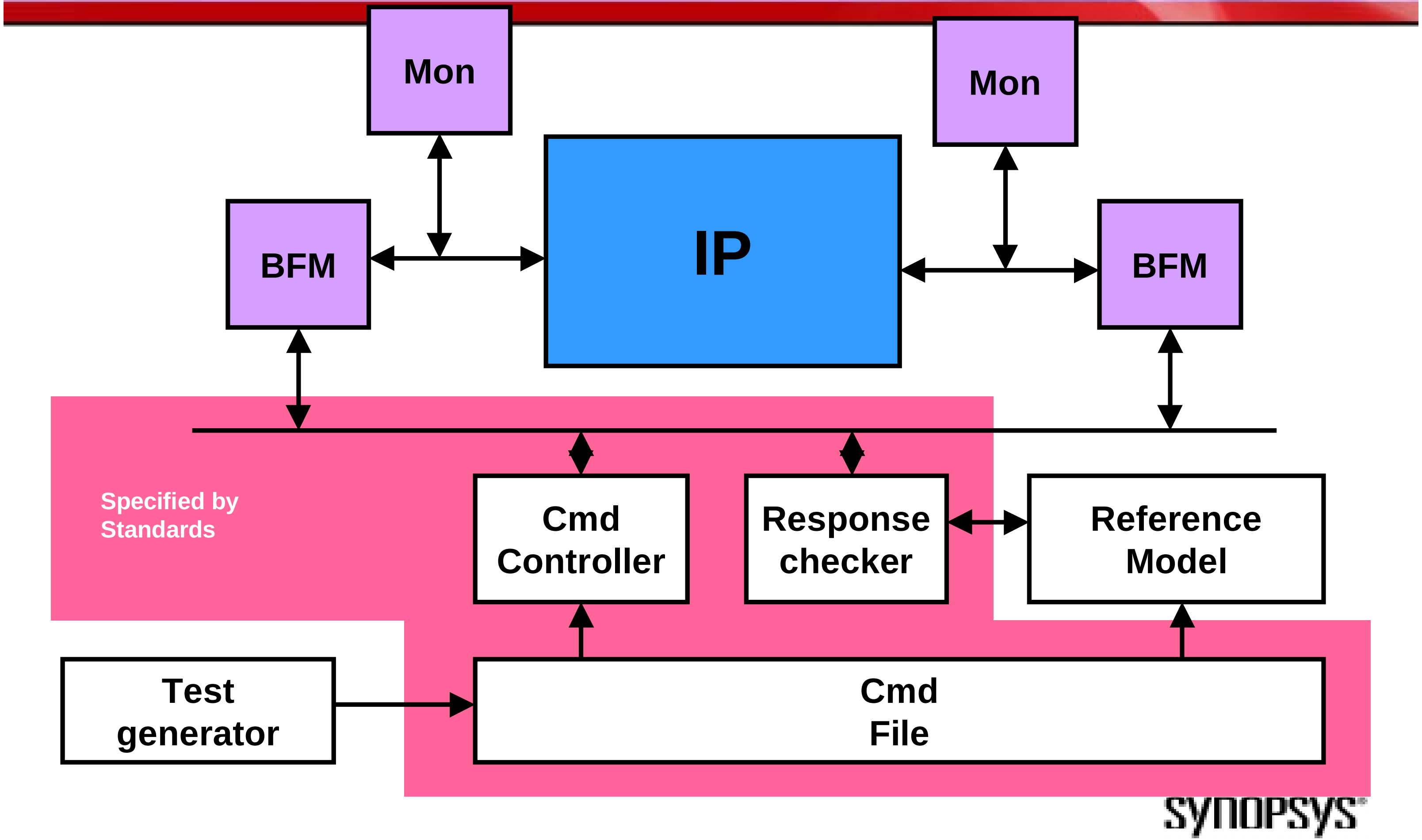
- Consistent physical design rules
  - necessary for integration of hard IP
  - standardized layer assignments
  - standardized drc, lvs decks

Foundry/Library Group

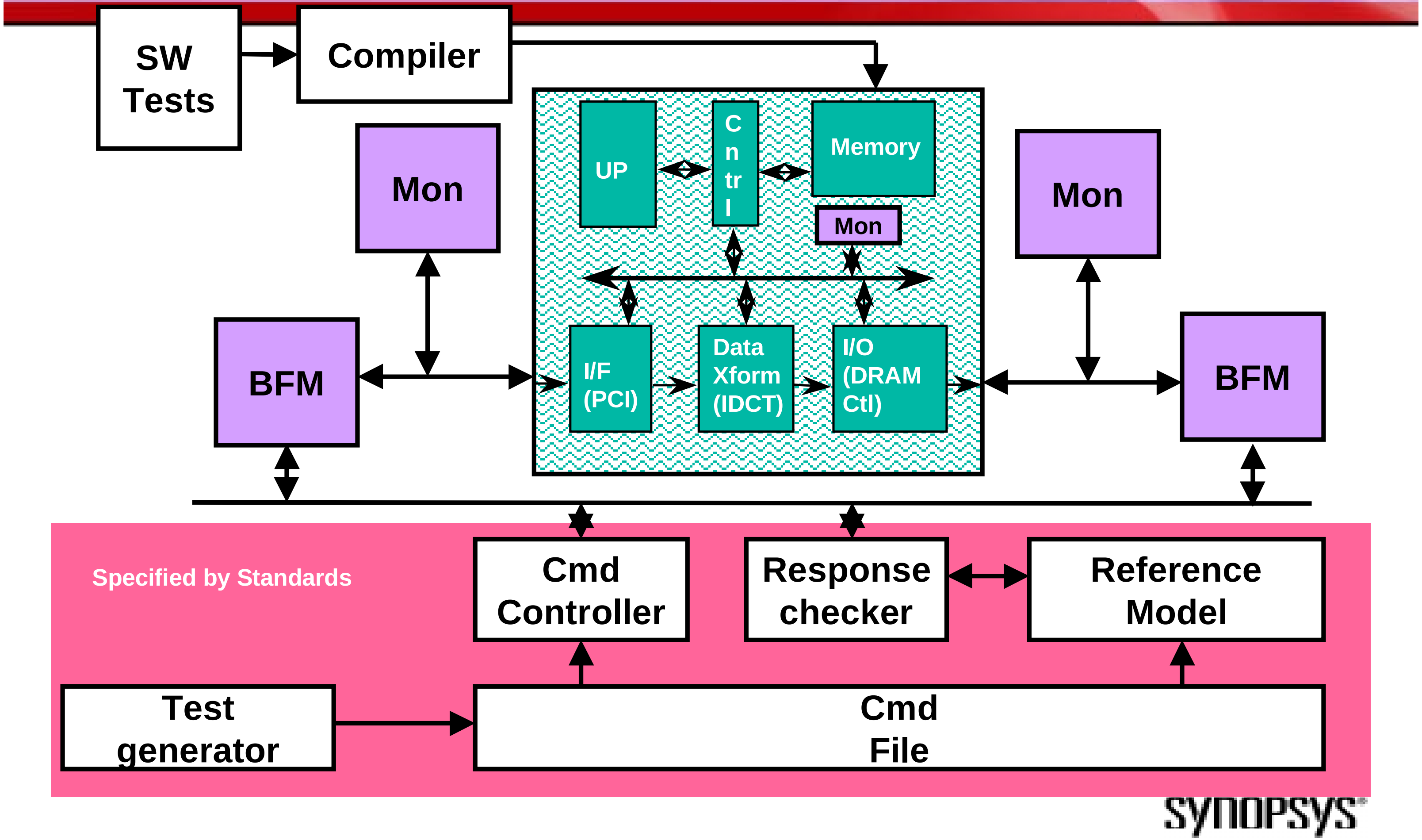


Functional Verification

Functional Test - IP



Functional Test - Chip



Summary

- Good IP design makes SoC design much easier
  - –Back end flow becomes straight-forward, consistent from design to design
  - –SoC design focuses on application and architecture
  - –Implementation focuses on functional verification

SYNOPSYS®

Summary

- Good IP provides simple, regular interfaces to chip
  - registered inputs, outputs
  - timing model is simple, not state dependent
  - manufacturing test is easy to interface to rest of chip