

Key Features of the Design Methodology Enabling a Multi-Core SoC Implementation of a First-Generation CELL Processor

Designers' Forum Session 8D – 1/27/06

ASPDAC 2006

**Dr. Dac C. Pham
STI-DC Chief Engineer and Convergence Manager
IBM Systems and Technology Group
Austin, Texas**

Outline

- ❑ Design Goals
- ❑ Processor Design Challenges
- ❑ CELL Design Approach and Principles
- ❑ Key Features of the Design Methodology
- ❑ Implementation Details
- ❑ Conclusion

Outline

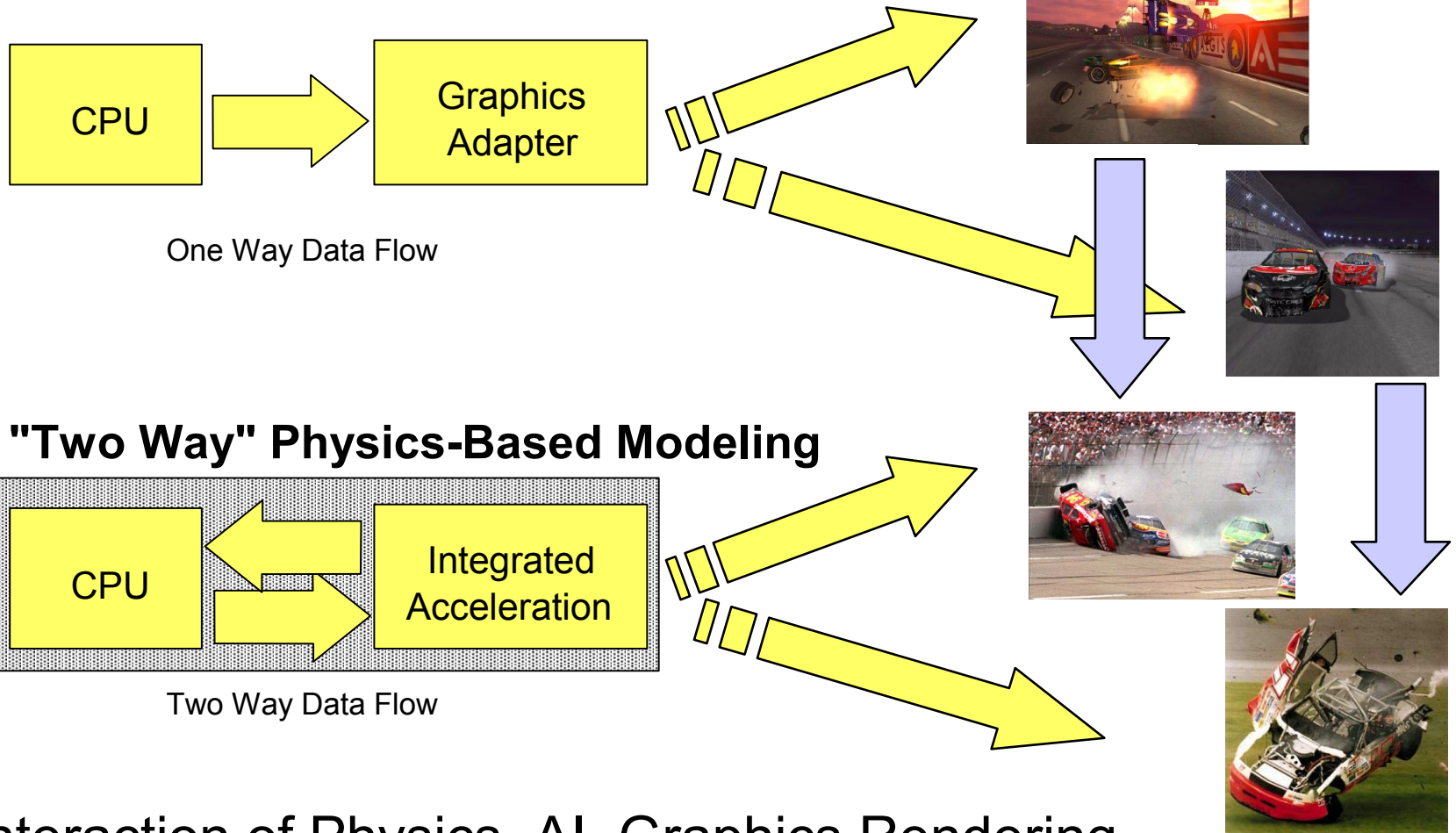
- **Design Goals**
- ❑ Processor Design Challenges
- ❑ CELL Design Approach and Principles
- ❑ Key Features of the Design Methodology
- ❑ Implementation Details
- ❑ Conclusion

Design Goals

- **Design for natural human interaction**
 - Realism requires Supercomputer attributes with extreme floating point capabilities
 - 2 TFLOPS in the new Playstation3 System
- **Set new performance standard**
 - Exploits parallelism while achieving high frequency
 - Multiple HF Cores
- **Foster innovation in Design & Methodology**
 - Holistic Design approach
 - Scalability and Flexibility through Modular design

Example: Digital Media Application

Current "One Way" Graphics Rendering



→ Interaction of Physics, AI, Graphics Rendering
...."Indistinguishable from Reality"

Outline

- Design Goals
- **Processor Design Challenges**
- CELL Design Approach and Principles
- Key Features of the Design Methodology
- Implementation Details
- Conclusion

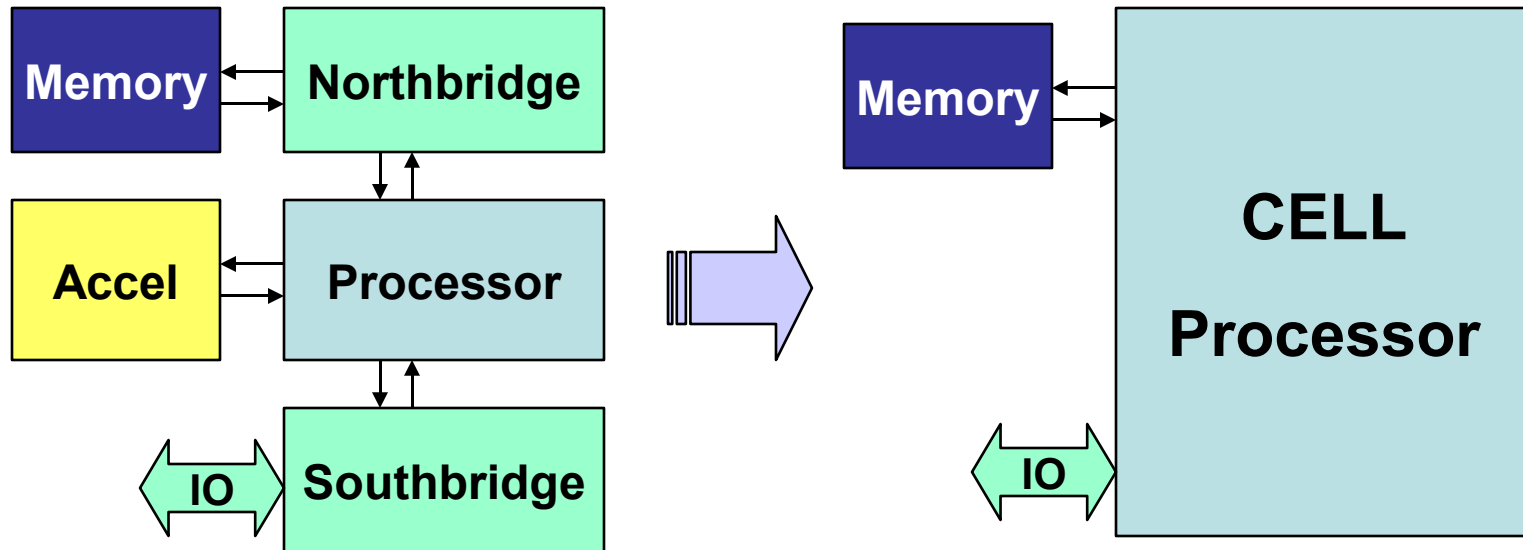
Processor Design Challenges

- Triple Constraints
 - Power
 - Frequency
 - Cost
- Design Trends
 - SoC and Giga Scale Integration
 - Multi-Core on a Chip
- Time to Market

Triple Constraints

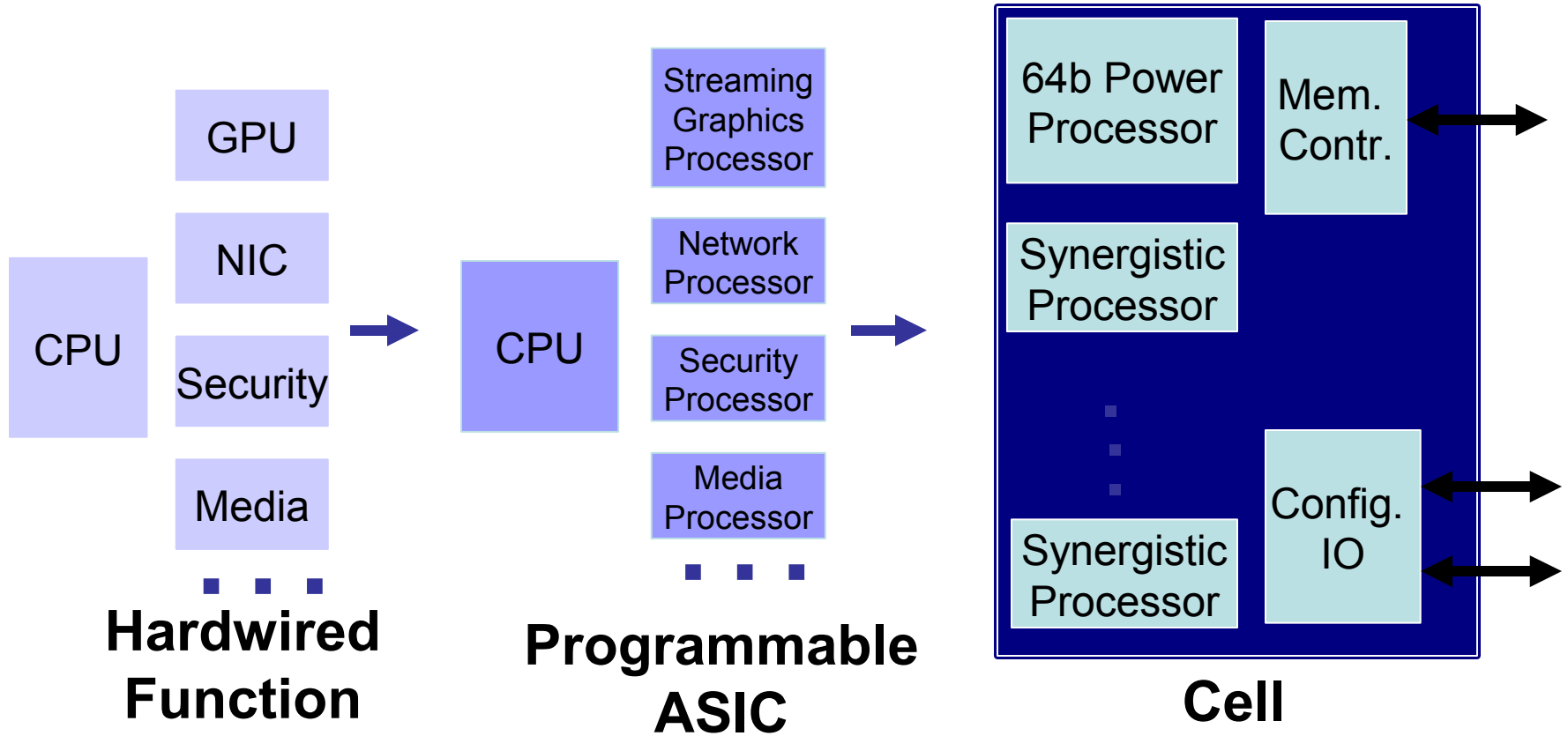
- Fundamental constraints: Power/Frequency/Cost
 - 10X Performance with Air-cooled Power possible through
 - Power Efficient design
 - Multi-Core
 - Cost Reduction through
 - Increase System Integration: Bridge Chip, Accelerators..
 - Leverage technology scaling: ¼ Billions transistors on single chip
 - Focus on Si Cost Reduction: Array redundancy, Test time, DFM, Partial Good...

System Trends toward Integration



- Increased integration is driving processors to take on many functions typically associated with systems
 - Integration forces processor developers to address off-load and acceleration in the design of the processor
 - Integration of bridge chip functionality

Giga Scale Integration



→ Need an innovative Design Methodology for High Frequency Multi-Core SoC

Time To Market

- Schedule predictability → Fast Convergence process
 - HLD Exit to 1st Tape-out in 9 months
- Bring-up readiness → Robust Verification process
 - OS boots in 24hr
- First Hardware to Production Ramp → Focus on bring up strategy and DFM
 - 6 months

Outline

- Design Goals
- Processor Design Challenges
- **CELL Design Approach and Principles**
- Key Features of the Design Methodology
- Implementation Details
- Conclusion

Holistic Design Approach

- Design
 - Cover all aspects of the design
 - Circuits, Cores, Chips, System, Software
- Development process
 - Fast Convergence
 - Top Down / Bottom Up
 - Early Design Planning / Final Convergence
 - Adaptability and Scalability
 - For long duration projects need to allow for refinement of ideas
- Organizational structure
 - Building the best processor development team spans across the globe
 - Enable Learning and Adaptive to changes in market

Design Principles

- Anticipate the future
 - Understand future technology limitations
 - Memory Wall, Power Wall, Frequency Wall
 - Reprogrammable elements
 - Work loads not available
 - Value proposition between uProc and ASIC design
 - High performance needs
 - Use the best technology in the design
 - New architecture, SOI, high speed interfaces, new package
 - Recognize tradeoffs
 - i.e. power, area, frequency triple constraint

Design Principles

- Design for flexibility
 - Modular design approach
 - Allow late changes from customer
 - Reuse of custom elements
 - Circuits Elements, Arrays, Cores
 - Flexibility in system design
 - Effects from board level effecting chip layout
 - Large changes in customer plans
 - Tradeoff in cost vs. flexibility

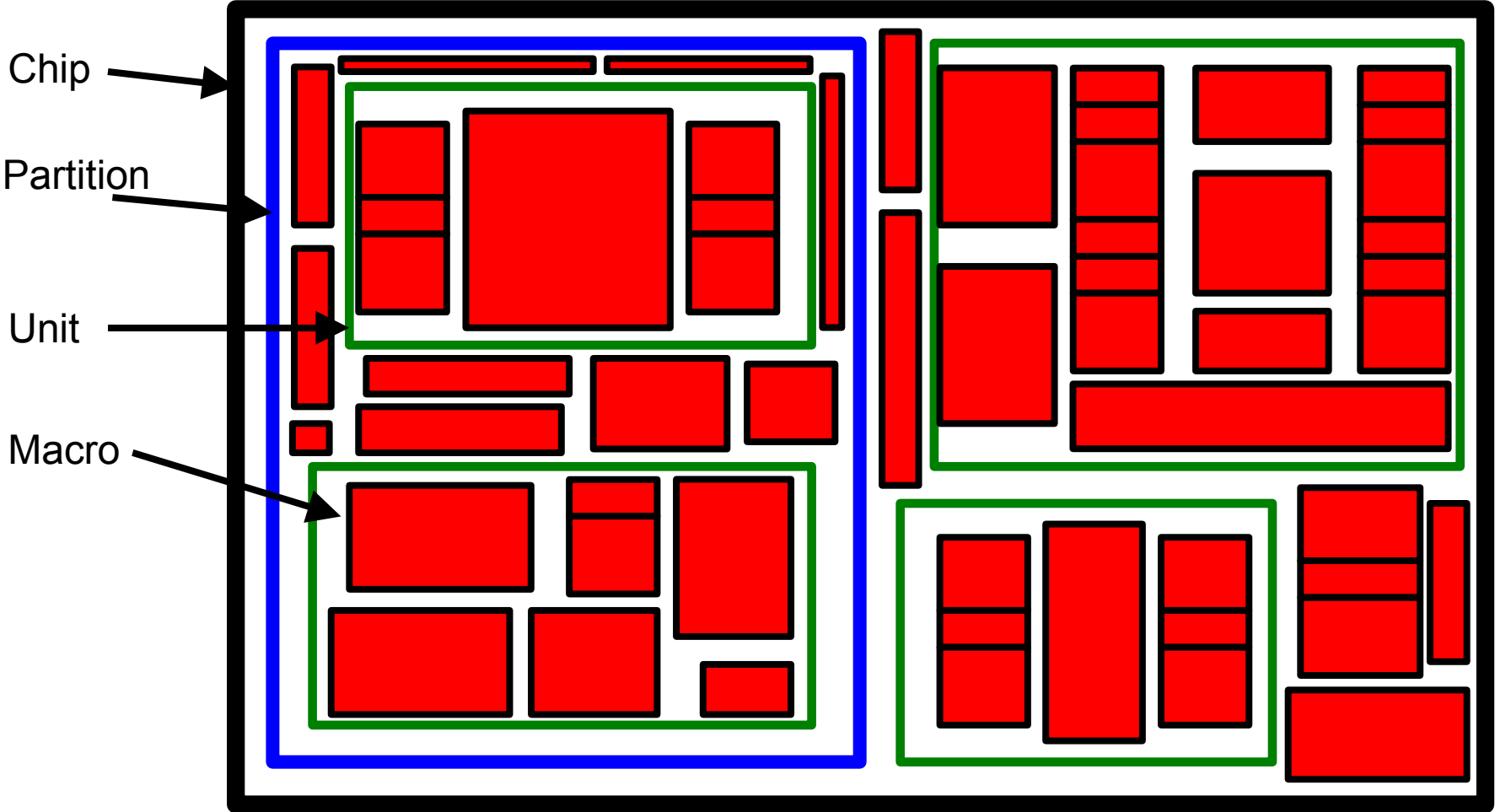
Outline

- ❑ Design Goals
- ❑ Processor Design Challenges
- ❑ CELL Design Approach and Principles
- **Key Features of the Design Methodology**
- ❑ Implementation Details
- ❑ Conclusion

Design Methodology Philosophy

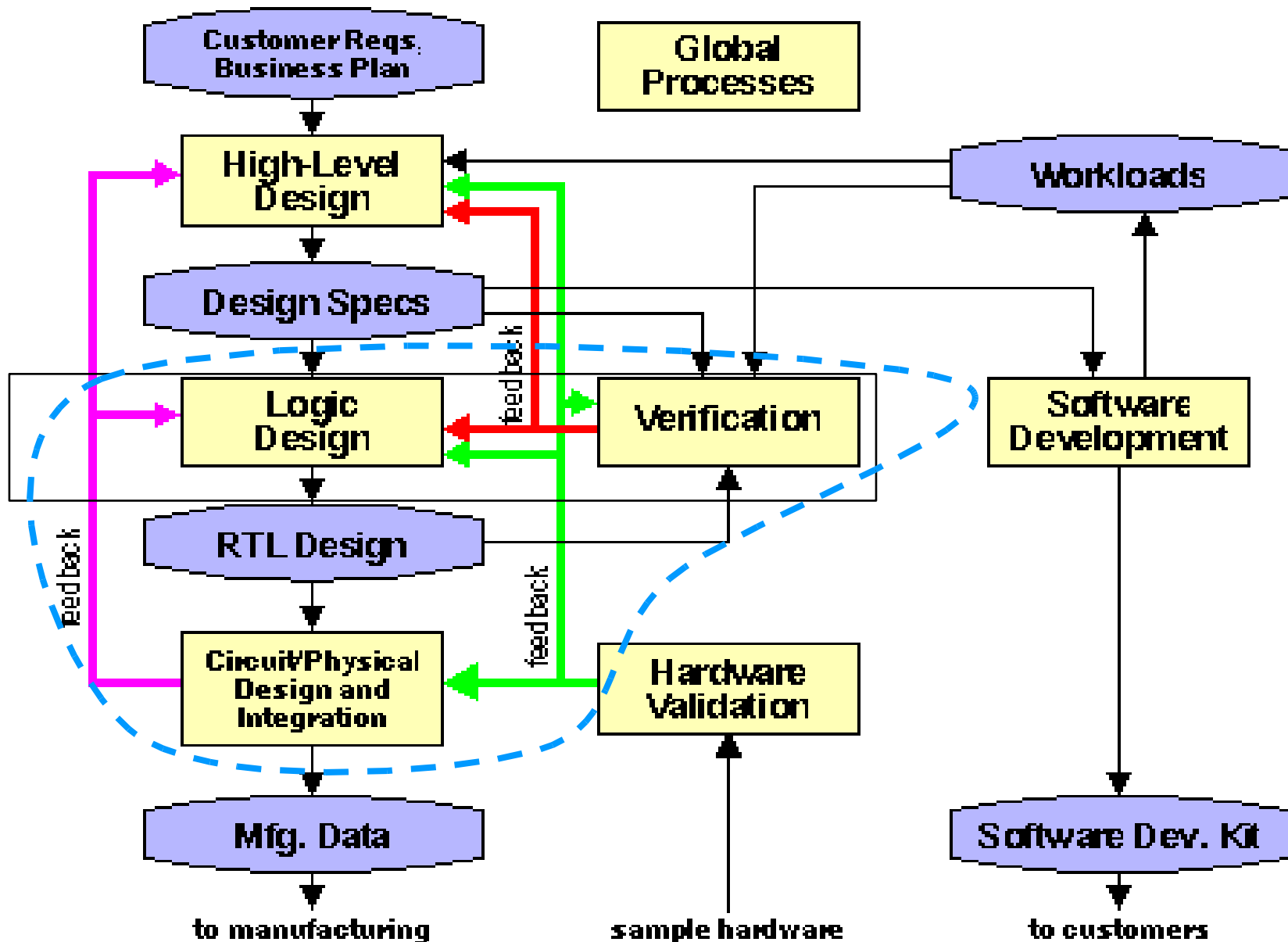
- Microarchitecture definition must go hand-in-hand with physical floorplan definition – wire delays are major component of performance
- “Divide and Conquer”
 - Chip hierarchy: macros, units, islands, partitions and chip
 - Macro is lowest level floorplannable object
 - Physical partitioning represented in RTL
 - Each level of hierarchy verified independently (DRC, LVS, Equivalence checking)
- Formal Equivalence Checking required between RTL and schematic
 - Latch points must match – no retiming
 - Performed hierarchically up to the chip level
- VHDL drives physical design
- Derived data is audited

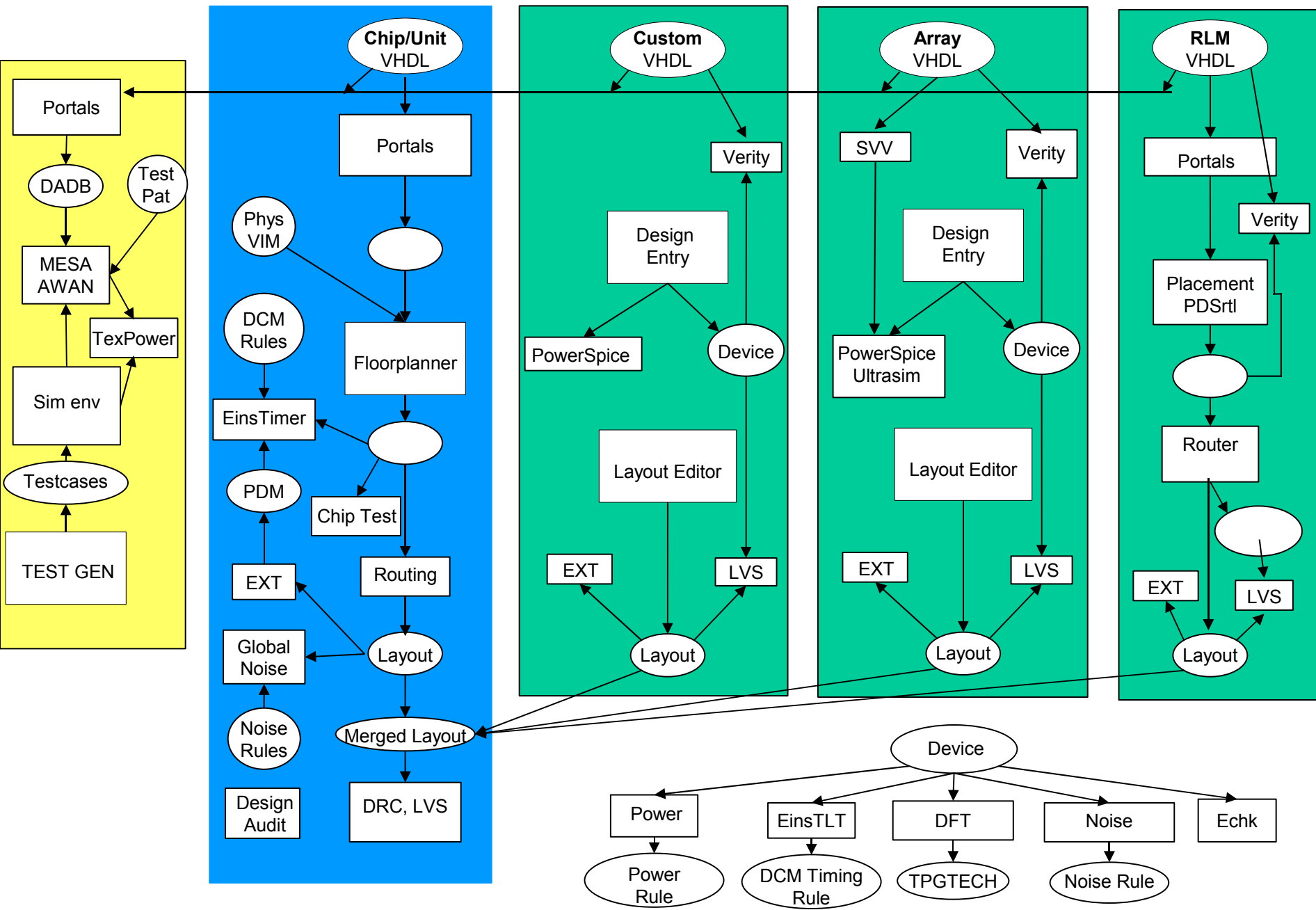
Schematic Illustration of Design Hierarchy



Chip (Partitions, units, macros) → Partitions (units, macros)
→ Unit (macros) → Macro (transistors)

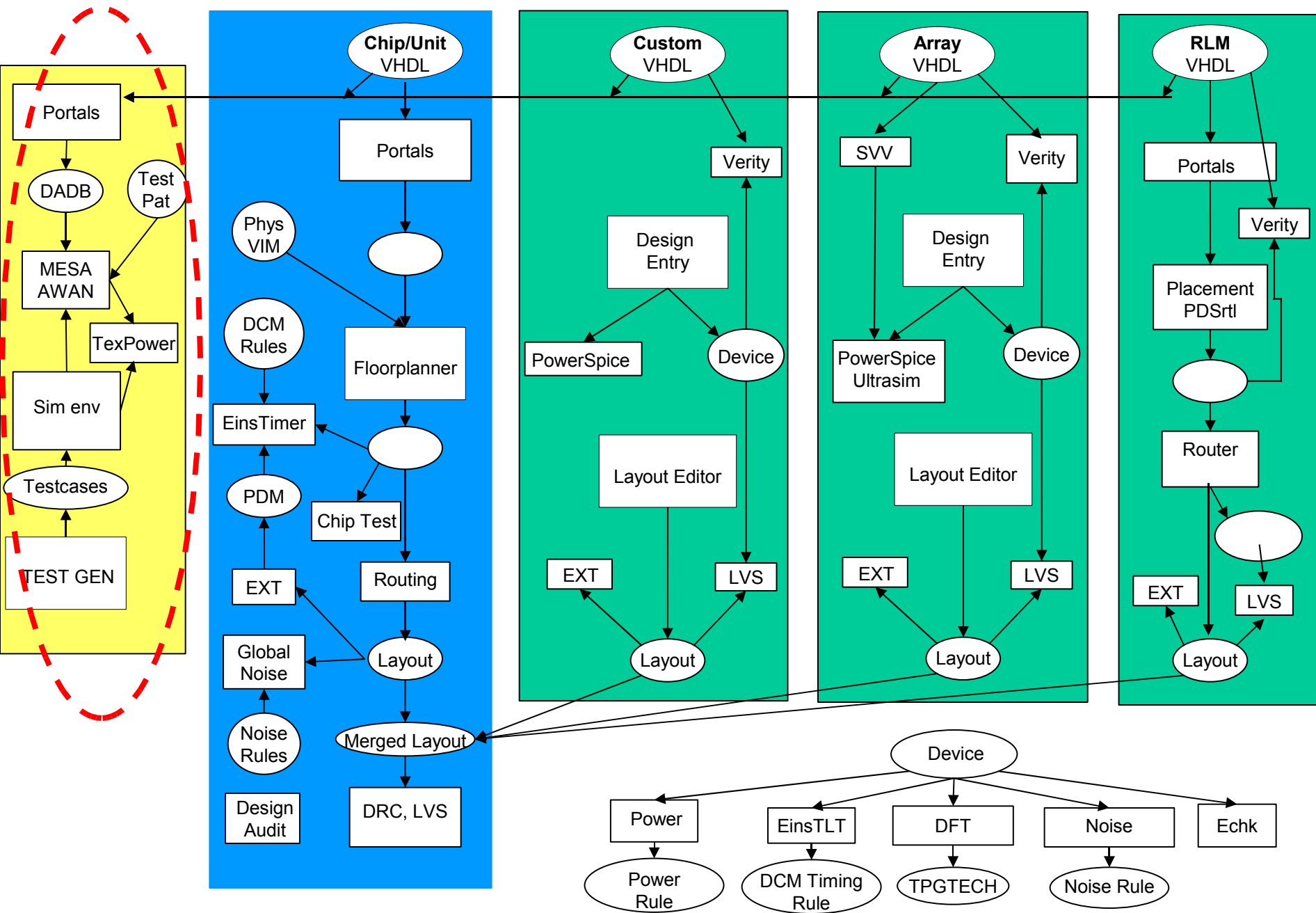
STI Development Process





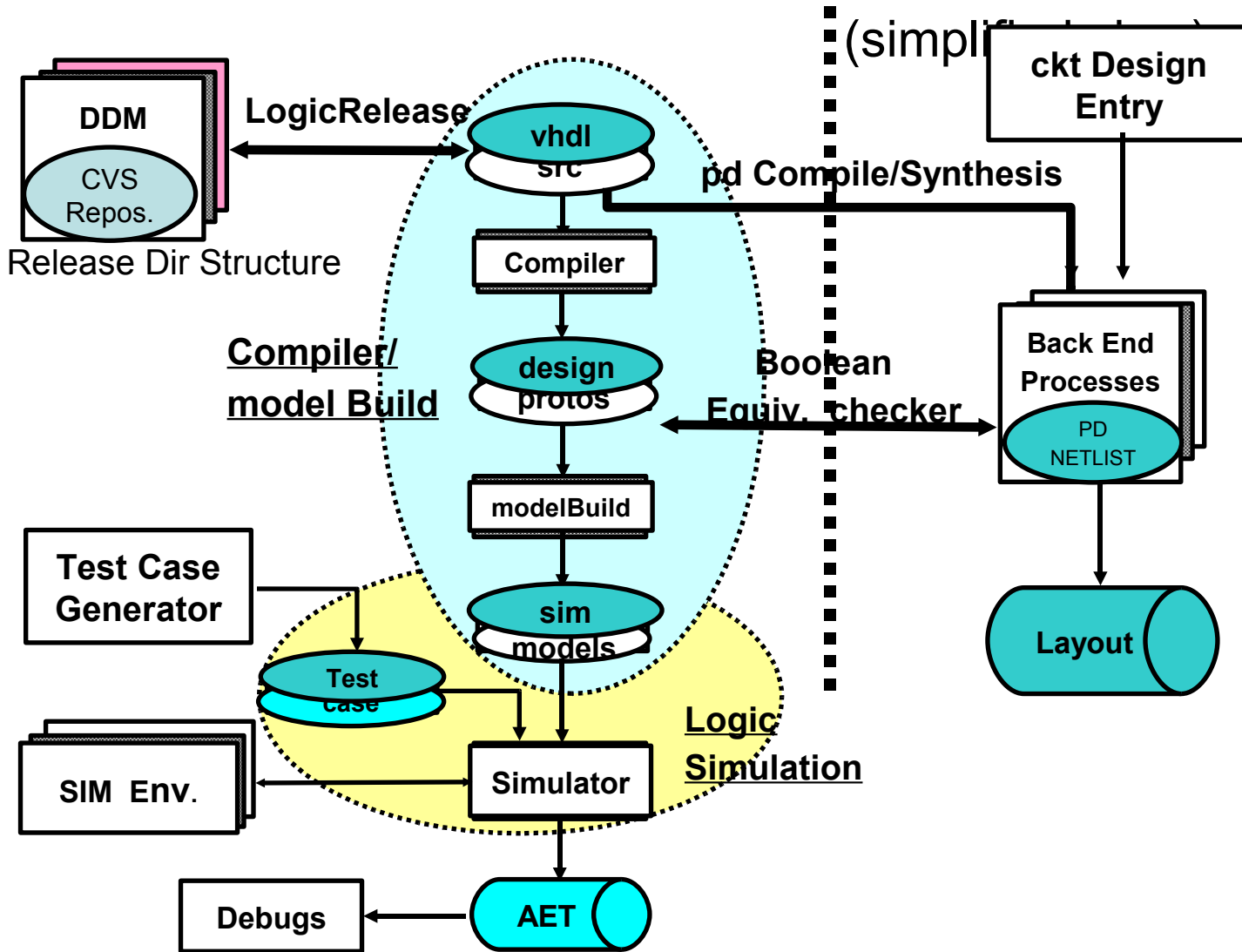
Design Data Management

- There are hundreds of designers at multiple remote design sites working on a chip design creating thousands of circuits/units → Need Design Data Management through a process call Audit and Promote to insure Right First Time.
 - With so many people and so much data there needs to be a way to verify that every check has been run on every piece of data that is going on the chip → this process called Audit
 - Over the course of the chip development, snapshots of the chip data are going to be needed so that different design teams can work with data that is of a certain quality. A **level** can be created to identify that data → this process called Promote



Front end Tools & Design Environment Flow

PD Environment

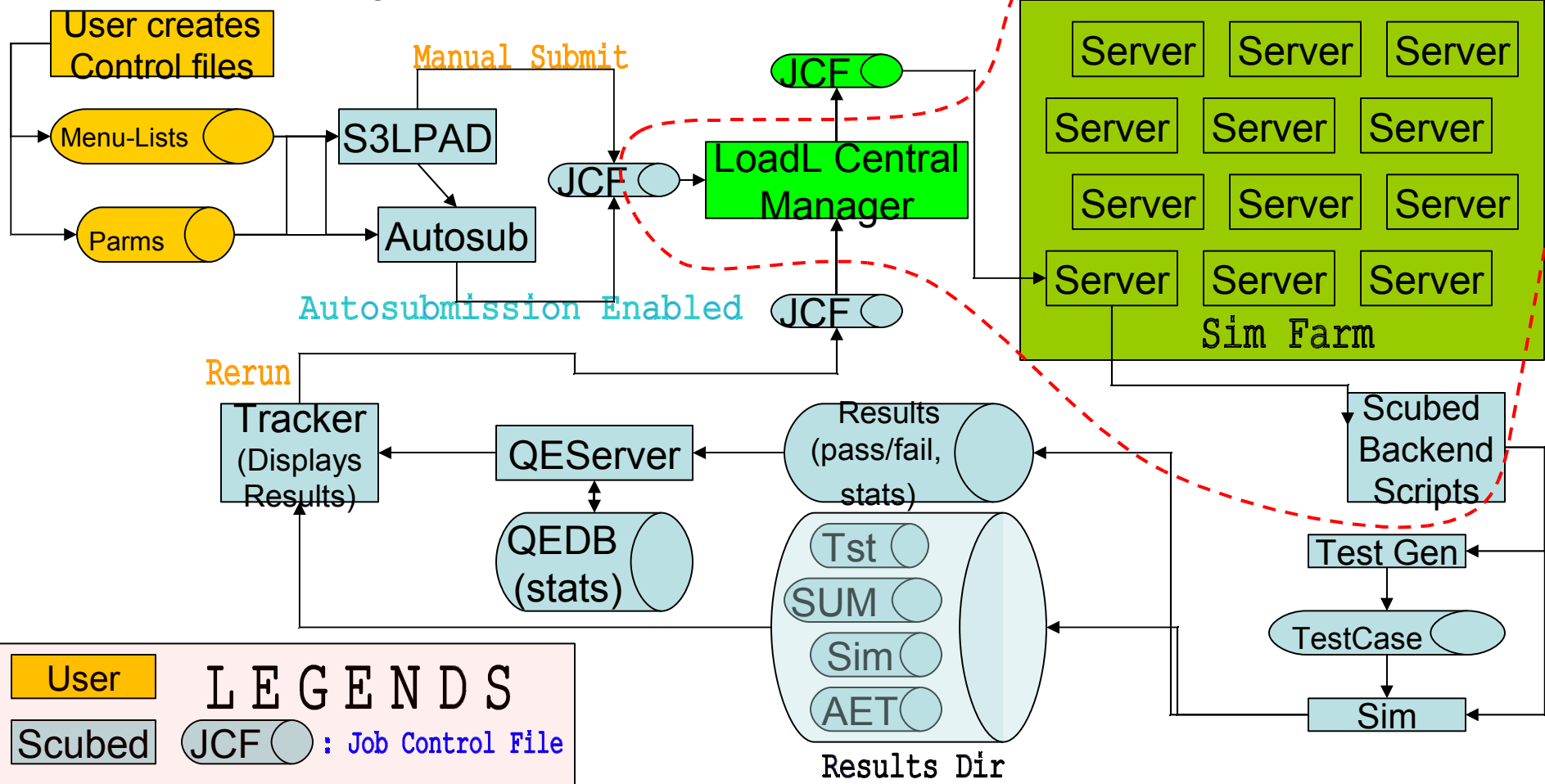


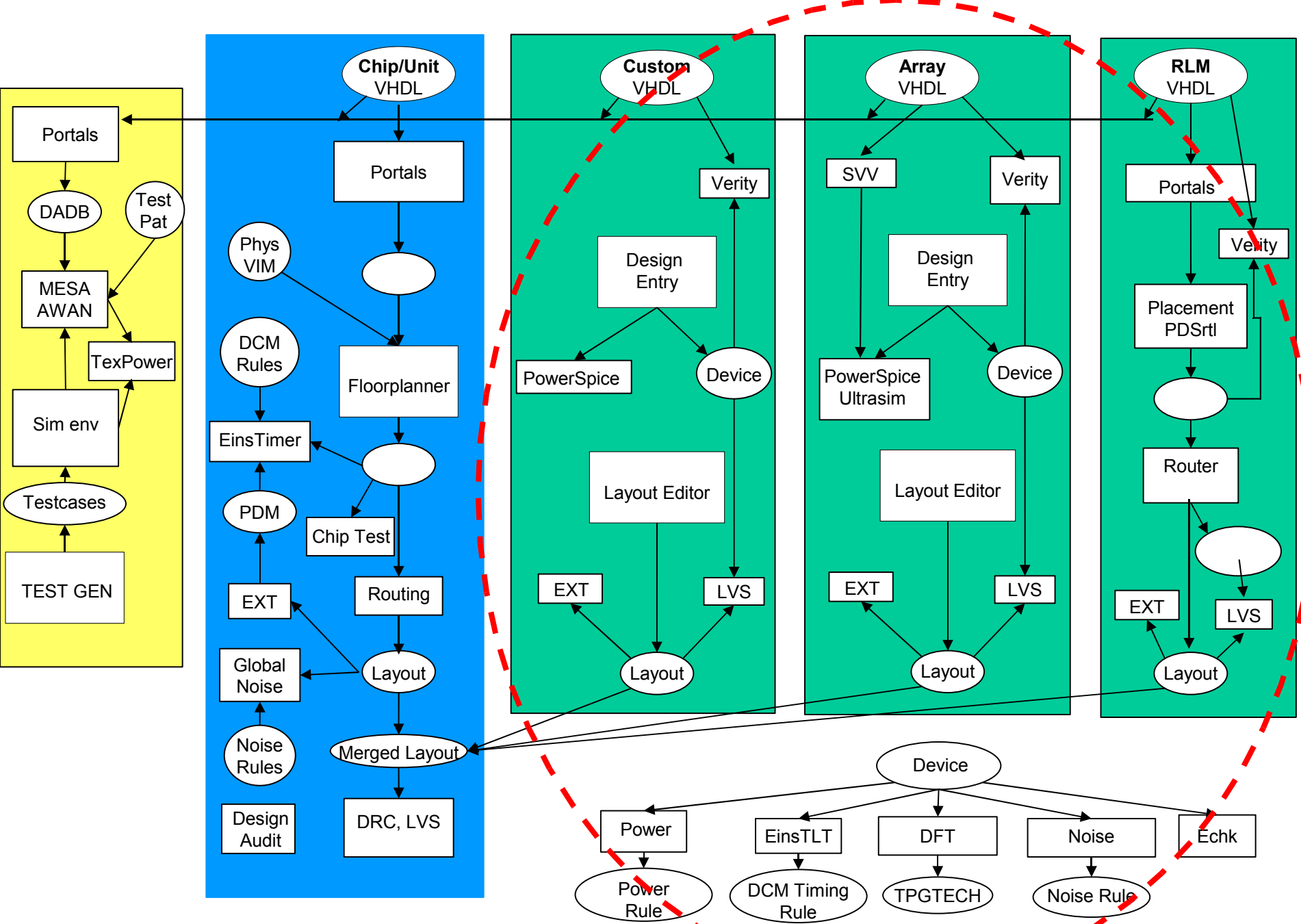
Hierarchical Verification

- Top Down Specification / Bottom up Implementation
 - Plan out all environments needed to create a quality chip
 - Break design into partition, island, unit, and block levels
- Test Generation: provide simulation with good stimulus
 - Pre-generated and on-the-fly stimulus creation
 - Static and random generation
- Model Build, Simulation, and Analysis
 - Bug Spray is an extension of VHDL including constructs to define coverage metrics, assertions, and design stimulus.
 - Model build for functional checking on a cycle by cycle basis
 - AWAN and ET4 (Palladium) used for very long running tests, eg. POR sequence, OS Boot, Workloads, etc.
 - Scan Chain analysis
 - Asynchronous interfaces analysis
- Formal Verification

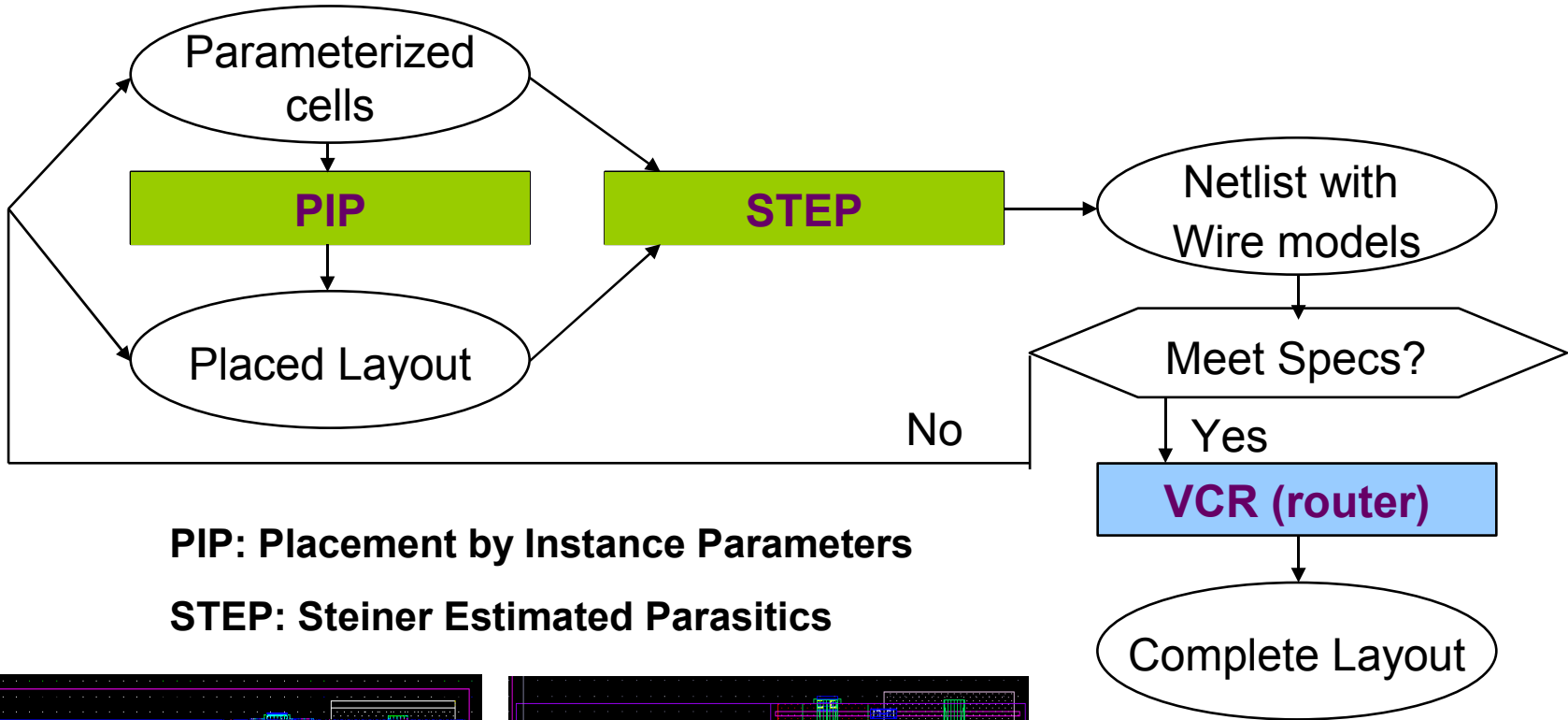
Batch Processing

- Load Leveler: Control software that allows access to the simfarm
- Scubed: Software automation and control tools that manage simulation regression runs and output



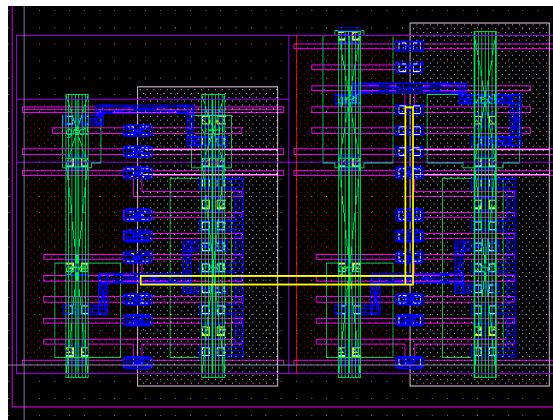
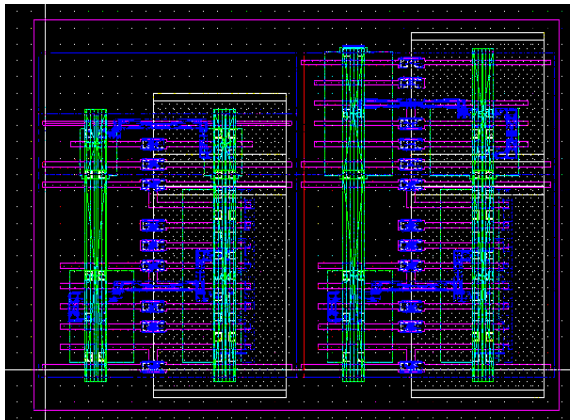


Layout Automation



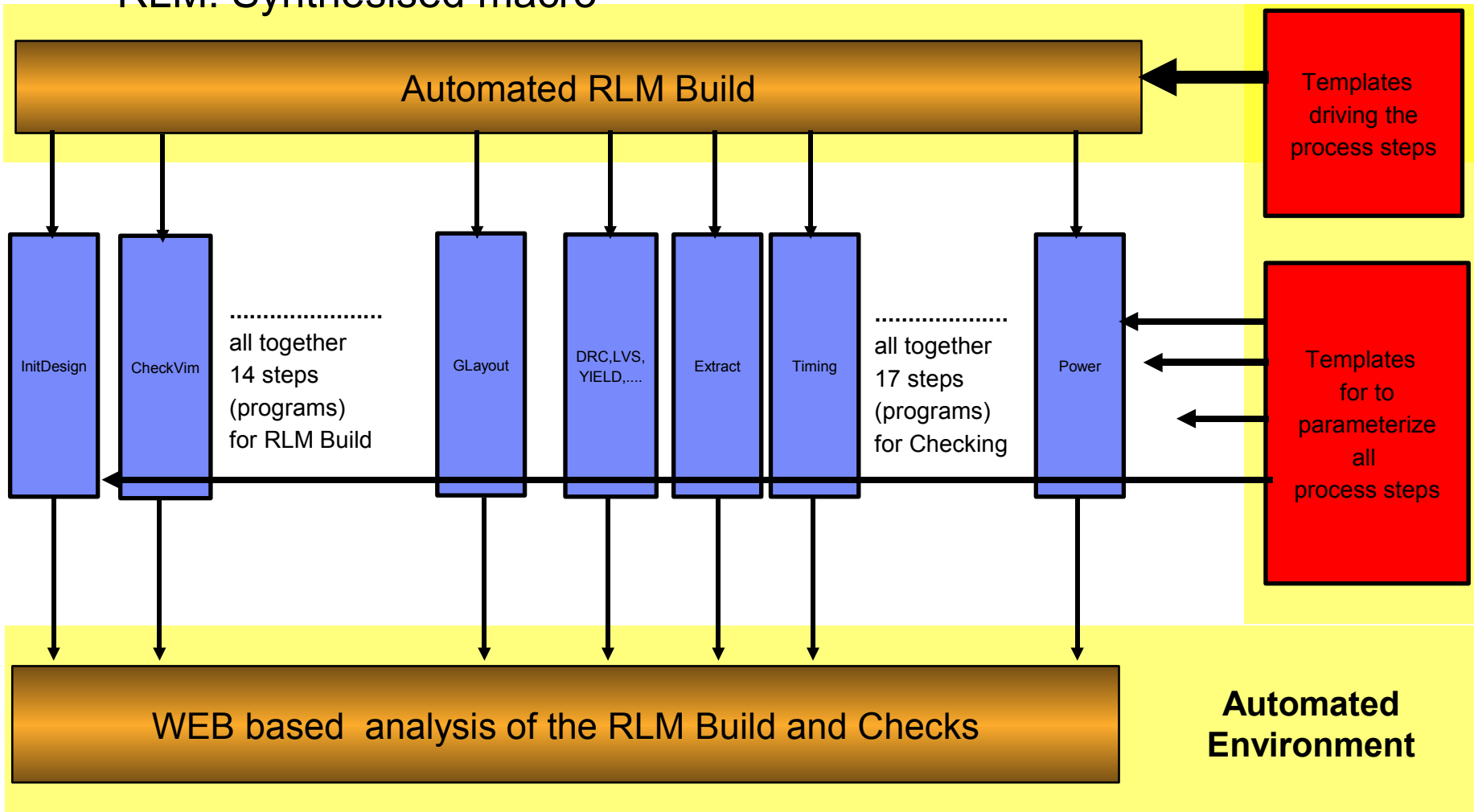
PIP: Placement by Instance Parameters

STEP: Steiner Estimated Parasitics

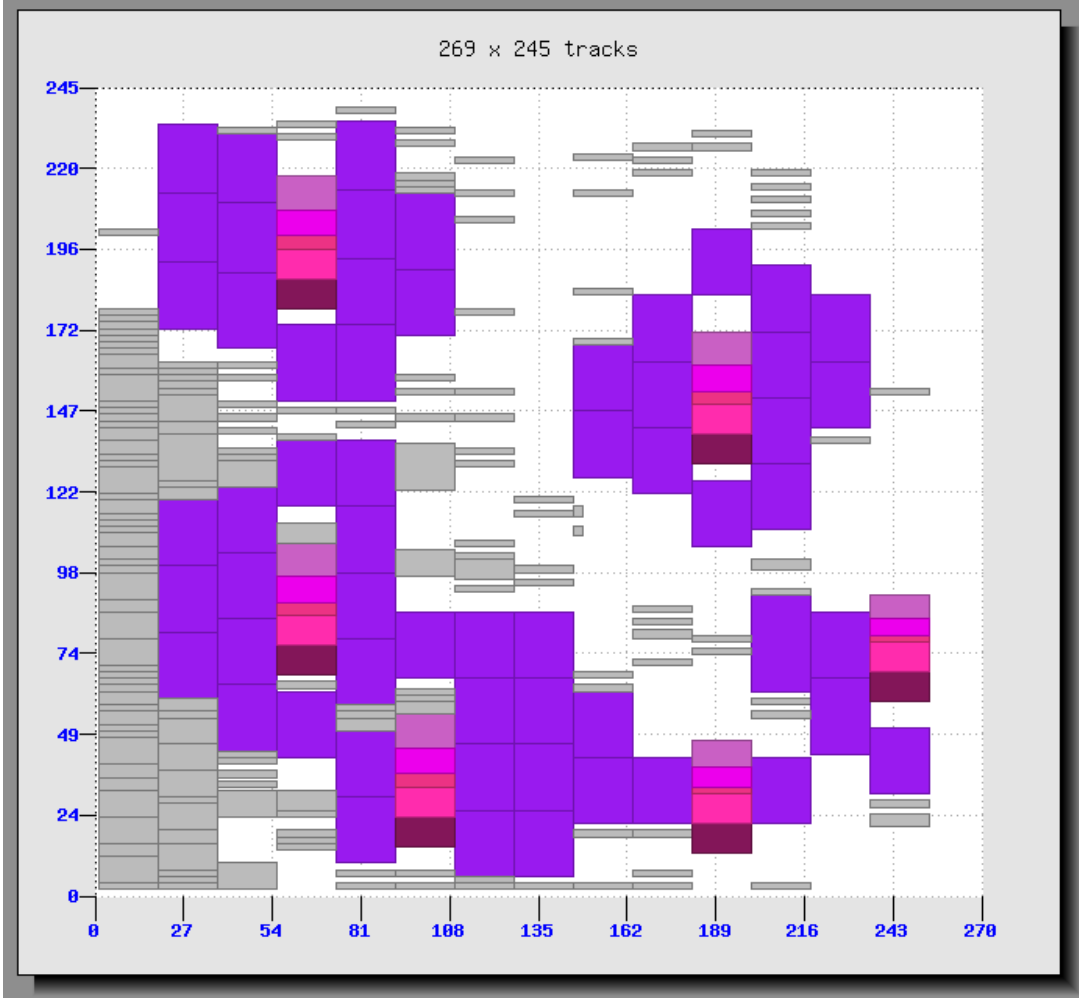


Automated RLM Build Process

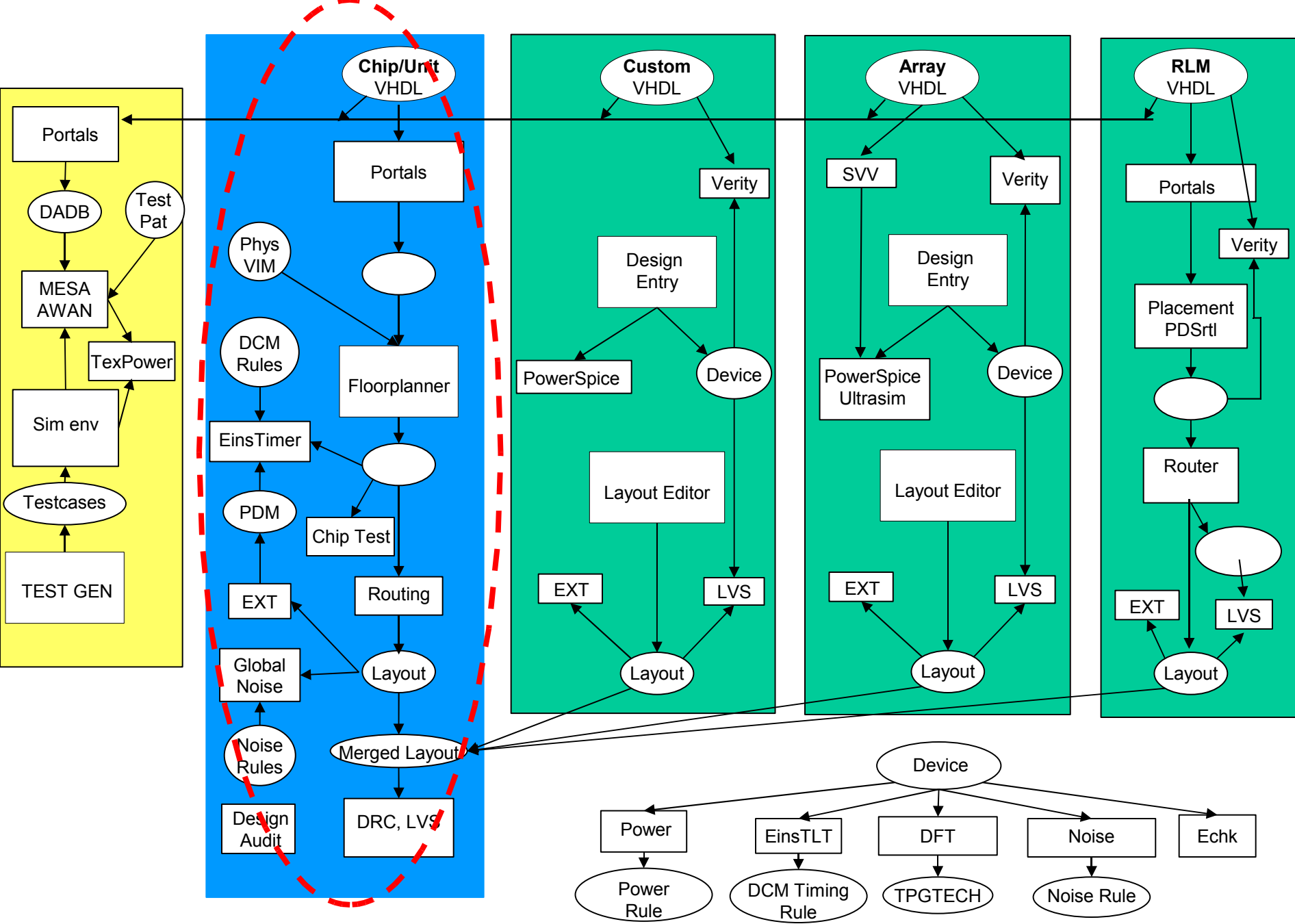
RLM: Synthesised macro



Example



latches – purple
lcb (local clock buffers)
clustered in the center
lcb have stacking order
defined in srule
(srule= synthesis rule)



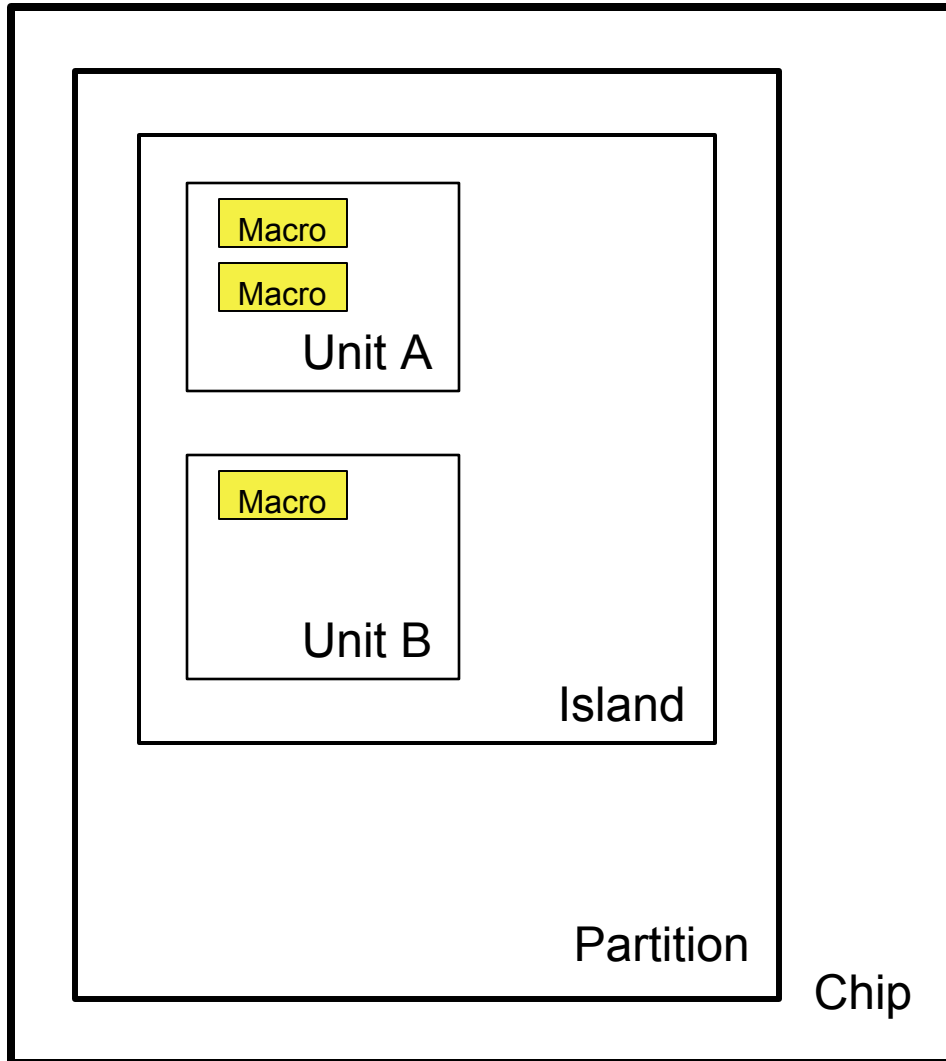
Integration Flow

- VHDL To Finished Layout
- Common Code And Methodology Infrastructure With RLM
- Additional Steps Unique To Unit Construction
 - Generate Power Busses
 - Buffer Planning/Insertion
 - Generate hierarchy design constraints
 - Decap Insertion
 - Unit Clock Router, minimize power
 - Routing with noise awareness, wire bending
 - Generate Power and Redundant Vias
 - Verification and Analysis: Extraction, Timing, IREM, Noise, Meth Check, Density Check, Yield Rule Check, DRC/LVS, Verity
- Saved Parameters For Each Design Making Rebuild Simple
 - Use Of Existing Designs As Template For New Designs

Timing Practices – “Fast Convergence”

- Macro partitioning encouraged to be on timing/latch boundaries
 - Macro designer owns full cycle with distribution
- Unit/Partition/Chip level static timing done early and often - progressively improving accuracy
 - Start with shell rules -> schematic based rules -> layout extracted rules
 - Steiner routes -> add wire codes -> 3D extraction -> noise uplift
- All latches treated as hard timing boundaries, no transparency
 - Forces designers to budget and meet 11FO4 cycle early
- Transistor level static timing required for all macros
 - Timing rules/abstractions created for each macro to be used in higher level timing runs

Hierarchical Timing Approach



Timing at 4 Levels of Hierarchy:

1. Unit (eg: sfx)
2. Island (eg: spu core)
3. Partition (eg: spc)
4. Chip

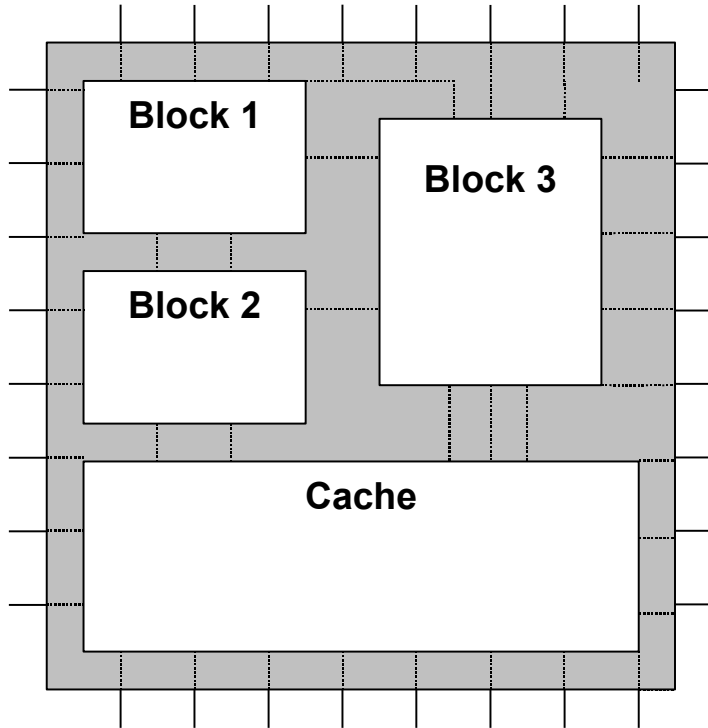
Hierarchical approach breaks down larger problem into manageable pieces (Units)

Chip Timing run times all paths across all hierarchies.

Internal Macro Timing Closed via EinsTLT but ALL paths visible in chip run

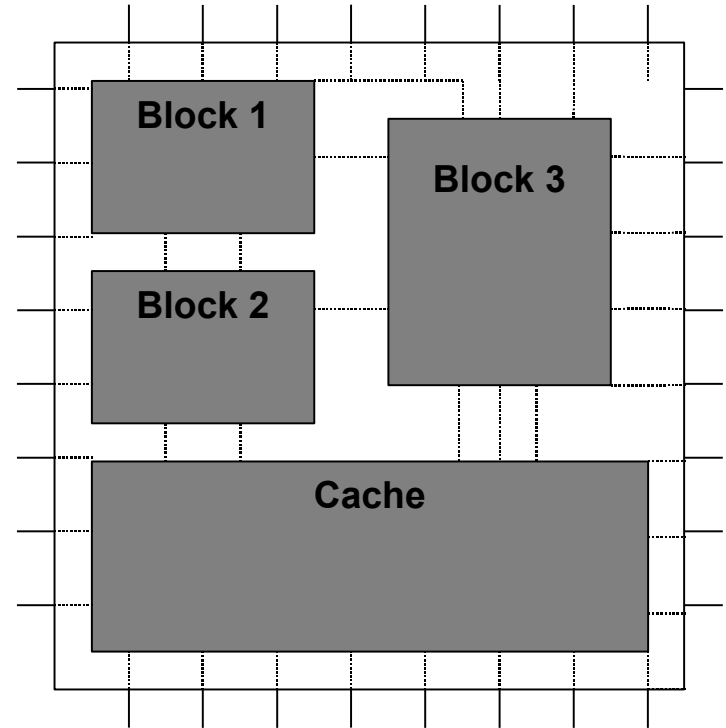
Noise Analysis

Macro Analysis



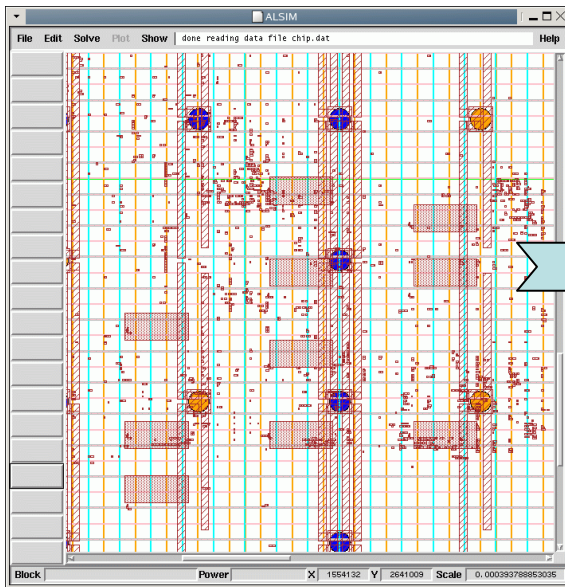
Noise analysis with focus on transistors and wires in macros

Unit/Chip Analysis



Global analysis with focus on behavior of wires

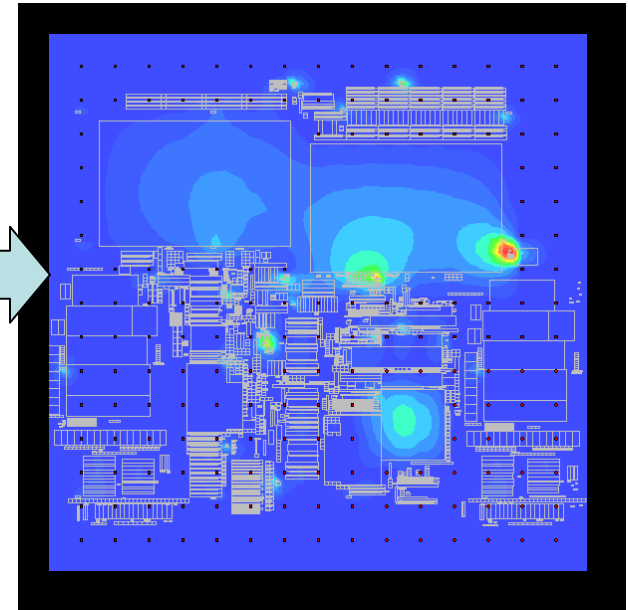
Power Grid Design/Analysis



Layout

ALSIM

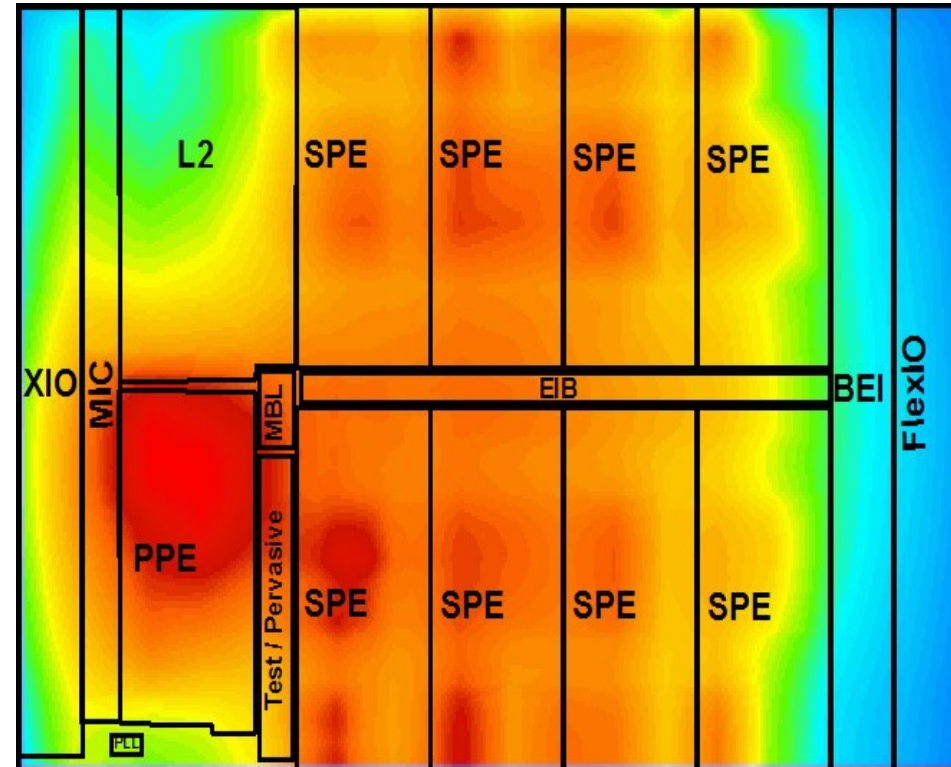
Block Spec.
Tech Spec.
...



Analysis Results

Hot Spot Analysis

- Extensive thermal analysis early in the design cycle
- Power maps created for use with package and heat sink models.
- Steady state and transient thermal behavior simulated
- Analysis feedback to chip floorplan and thermal sensor design



Outline

- ❑ Design Goals
- ❑ Processor Design Challenges
- ❑ CELL Design Approach and Principles
- ❑ Key Features of the Design Methodology
- **Implementation Details**
- ❑ Conclusion

Implementation Challenges

- Technology Scaling
 - Minimize cross chip variations in delay and leakage
 - Array bit cell stability, writability, yield
 - Growing impact of wire RC vs. device speed
- 11FO4 design within air-cooled power envelope
 - Power, Clock, Signal Distribution variation due to hot spots, inductance effects, etc
 - Multi Clock domains
 - Intra-Chip interconnections
 - Global Optimization with “triple constraints”: Frequency, Power, Cost (Die Size and Yield)

Circuit Design Practices

- Strict design guidelines to minimize design variations
 - Layout topology check and DFM rules for yield
 - Circuit topology and electrical checks
 - Global active clock pulse limiter for dynamic circuits
 - Hold time margin scale with clock path delay
- Reduce design sensitivity to technology leakage
 - Limited dynamic logic circuit usage
 - No Low-Vt devices
- Array yield focus
 - Array redundancy for bit cell stability fails
 - Reduced cell stress during read

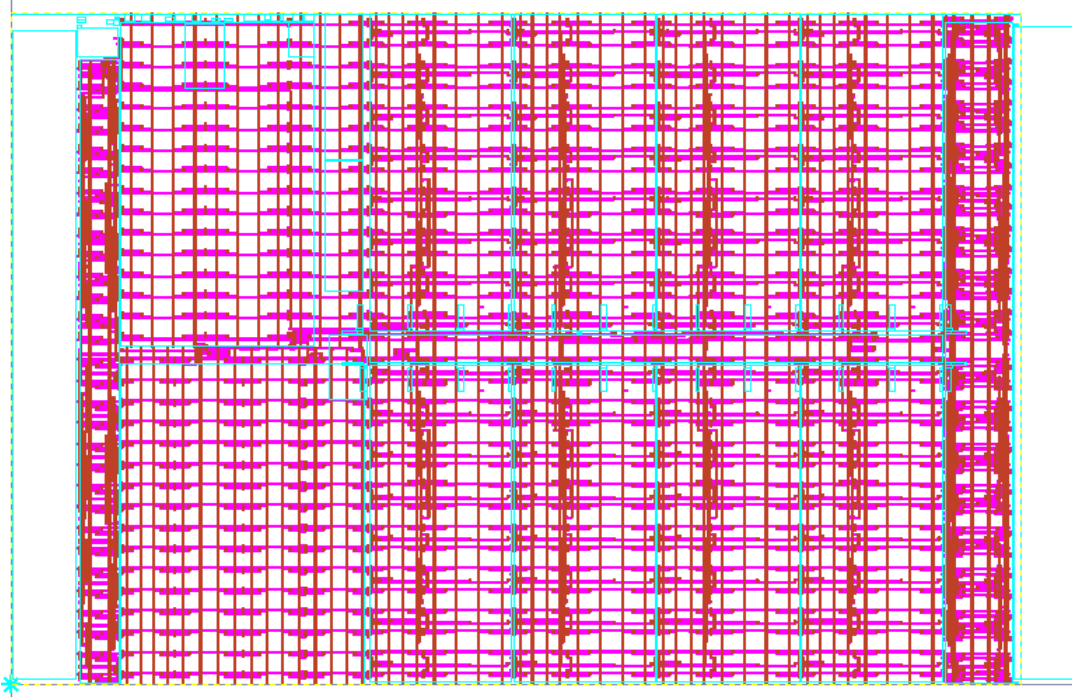
Power Management Practices

- Dynamic power is controlled by fine-grain clock gating
- Leakage power is managed by adding lower vt devices only where necessary
- Accurate power estimation
 - Macro level uses circuit simulation and generates a power rule (0-50% input switching)
 - Partition/Chip level uses behavior simulation with specific workloads and macro level power rules

Test / Pervasive Design Practices

- Distributed test functions
 - LBIST engine for cores
 - ABIST engine for arrays
- Distributed debug features
 - Common debug bus
 - Centralized trace array
- Centralized test and pervasive control
 - Common strategy for logic debug and performance monitoring
 - Monitor some activity externally
- Early focus on design bring up
 - At speed test (internal chip scan, ABIST, programmable LBIST)
 - On chip logic analyzer for debug
 - On chip performance monitor
 - Isolate, start, stop, step controls for lab debug.

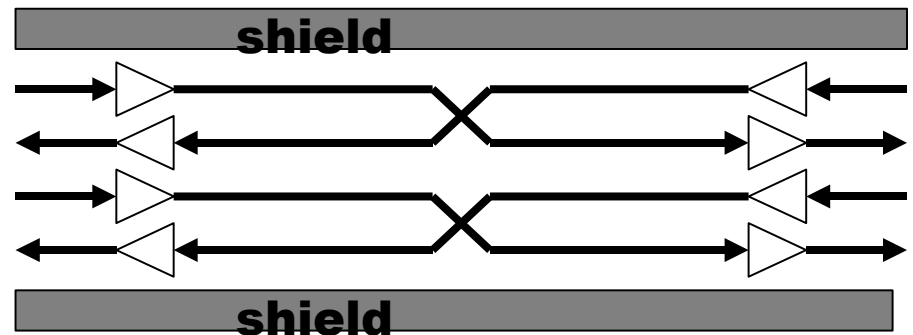
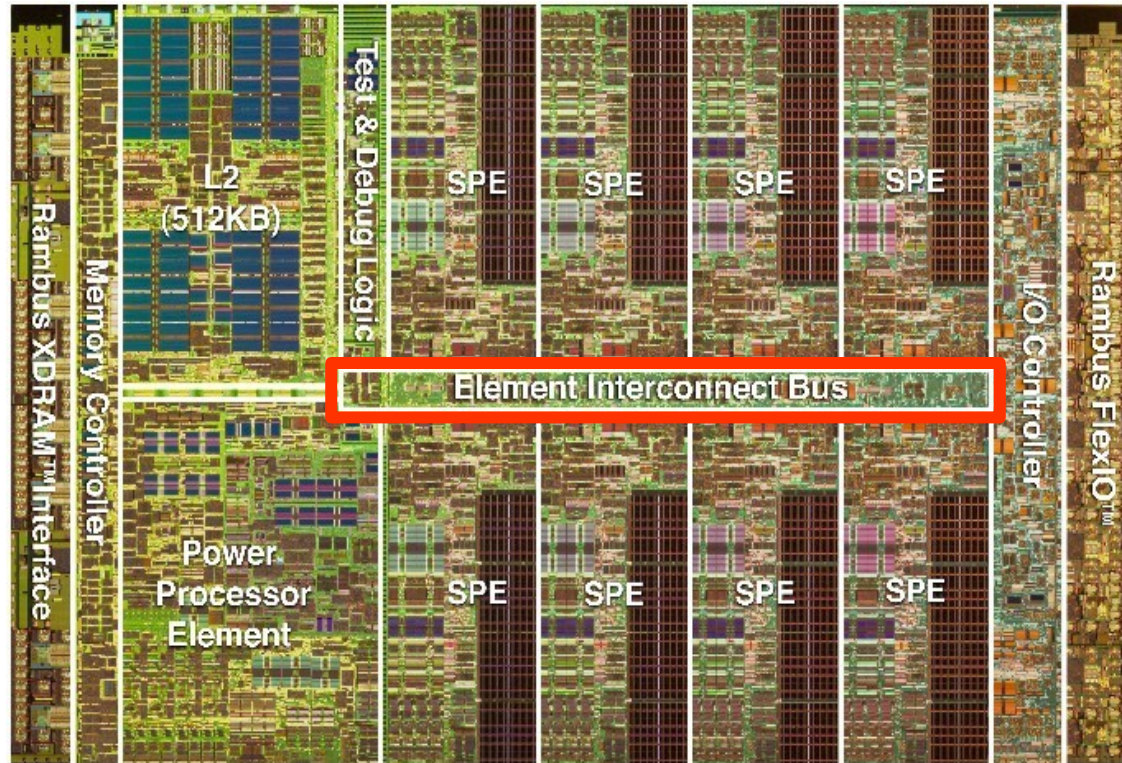
Clock Distribution Network



- Clock grids used final two layers of metal
 - Over 850 individually tuned buffers.
- Three clock networks each from separate PLL
 - Processor
 - Bus interface
 - Memory interface
- Main clock grid covers >85% of the chip
- Multiple clock frequency islands for second and third clock grids

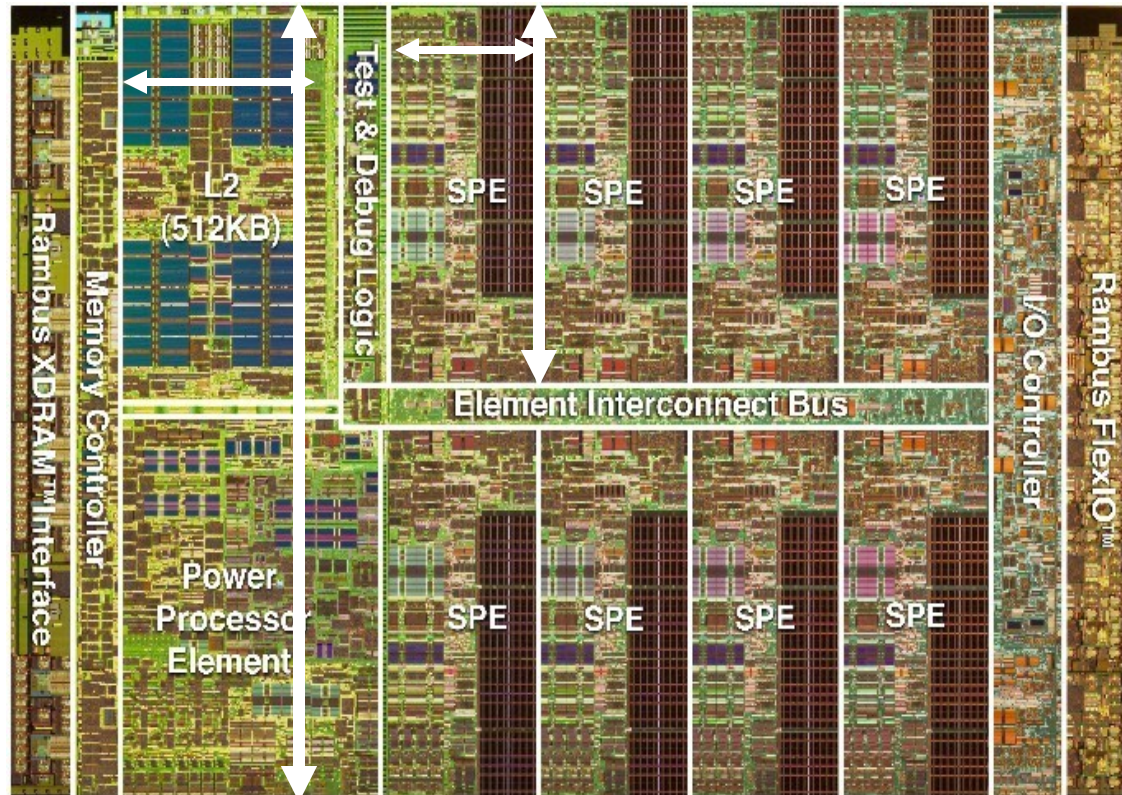
Engineering Busses

- EIB design
 - Four 128-bit data rings
 - 64-bit tag
 - Two twisted pairs of wires interleaved with shields.
- Engineering for signal integrity
 - 50% of global nets engineered
 - 32K repeaters added

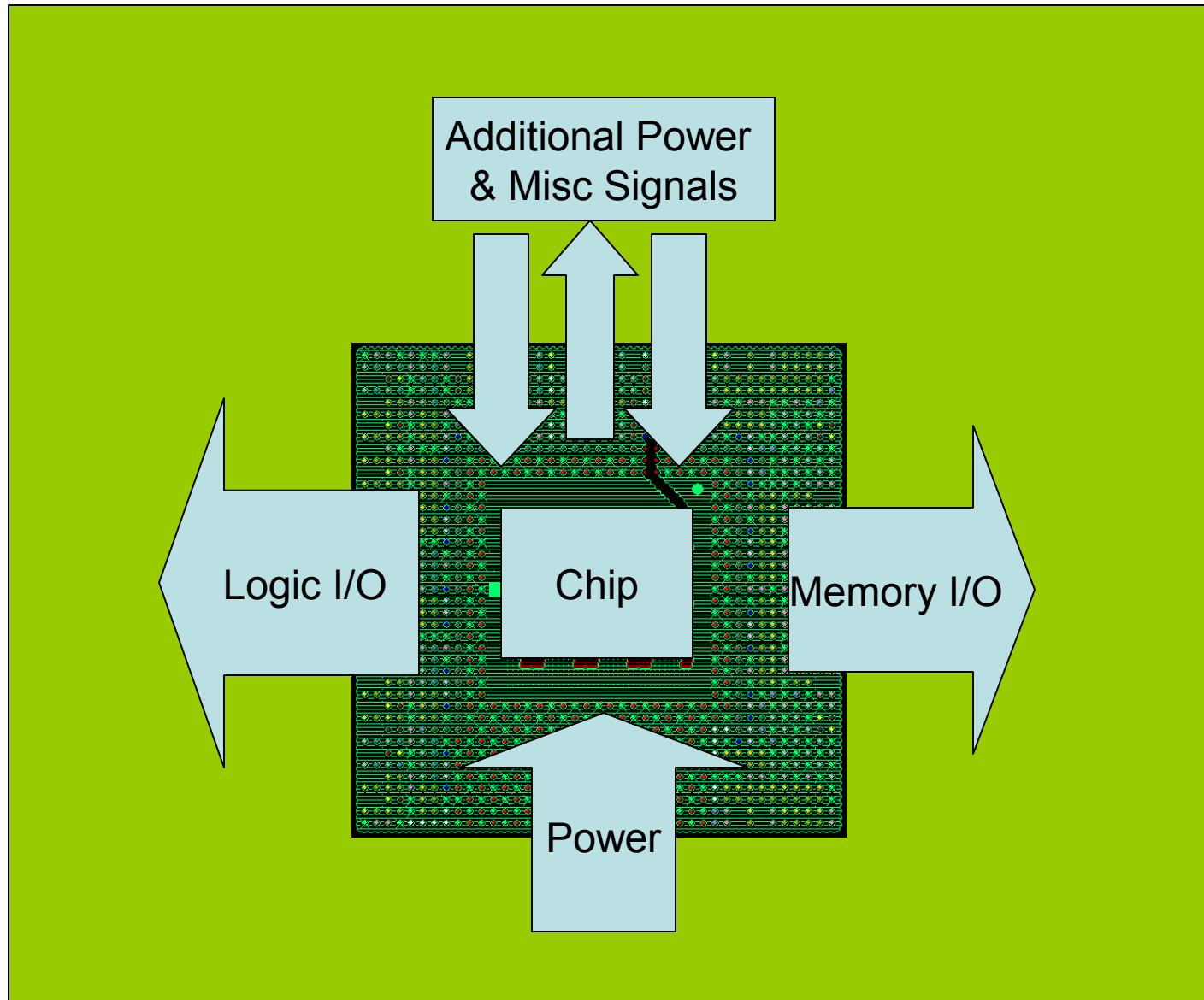


Chip Assembly

- Early design planning for optimum partition aspect ratio
- Modular construction for chip assembly
- At top level, 17 major physical partitions, 8912 discrete blocks.
- The chip total, across all levels of hierarchy, 177K blocks, 580K repeaters and 1.55M signal nets
- 241M transistors in 90nm SOI technology (8 levels of copper interconnect +1 local interconnect layer)



Board Influences



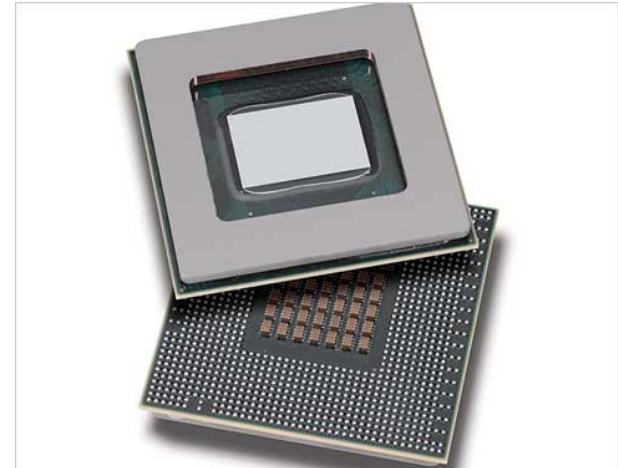
Outline

- ❑ Design Goals
- ❑ Processor Design Challenges
- ❑ CELL Design Approach and Principles
- ❑ Key Features of the Design Methodology
- ❑ Implementation Details
- **Conclusion**

Conclusions



©2005 Sony Computer Entertainment Inc. All rights reserved.
Design and specifications are subject to change without notice.



- The CELL processor, a multi-core design, was successfully implemented using
 - Innovative design methodology
 - Good design practices
 - Rules for modularity and reuse
 - Triple Constraints for optimum design point
- Correct operation has been observed with good Frequency range (over 4GHz)
- Sony/SCEI announced PS3 System in 5/05

Acknowledgments

- The co-authors: Hans-Werner Anderson, Erwin Behnen, Mark Bolliger, Sanjay Gupta, Peter Hofstee, Paul Harvey, Charles Johns, Jim Kahle, Atsushi Kameyama, John Keaty, Bob Le, Sang Lee, Tuyen Nguyen, John Petrovick, Mydung Pham, Juergen Pille, Stephen Posluszny, Mack Riley, Joseph Verock, James Warnock, Steve Weitzel, Dieter Wendel
- The deep collaboration and the many contributions from the entire Sony-Toshiba-IBM team who worked tirelessly side-by-side on the design of this processor.
- The Executive management teams of the three companies who provided management oversight and created the right business conditions for this project