



Silicon Integration  
Initiative, Inc.

# Low-Power Design and Verification

**Steven E. Schulz**

*President and CEO*

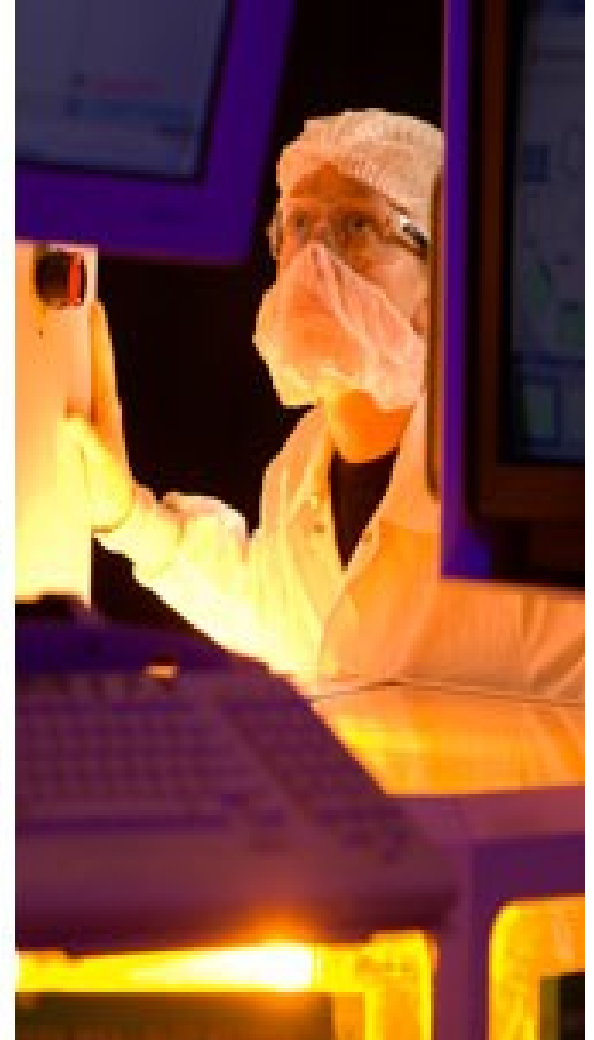
*Silicon Integration Initiative, Inc.*

*January 24<sup>th</sup>, 2008*

**Si2 - Innovation Through Collaboration**



# Today's Agenda



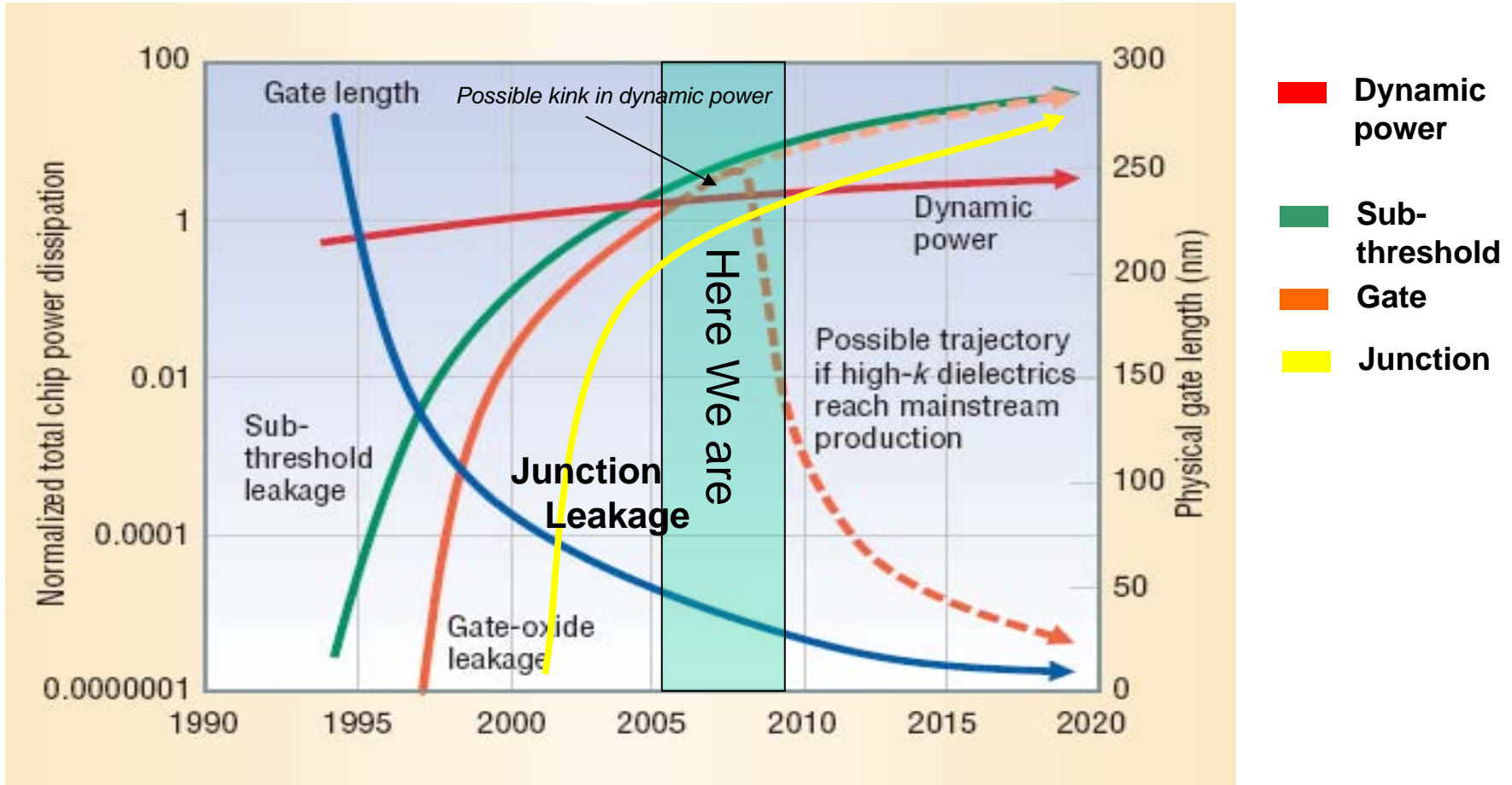


# Today's Agenda

- Why Low-Power Now?
- Design and Verification Flow Challenges / Reqs
- Common Power Format Introduction / Examples
- Industry / Market Adoption and Silicon Benefits
- Introduction to the Low Power Coalition
- 2008 Roadmap / Plans
- Q & A



## 2001 International Technology Roadmap for Semiconductors



$$P = ACV^2f + VI_{leak}$$

## *Mobile applications trends:*

- *Leakage is significantly increasing due to process scaling*
- *Active power increases due to application integration (with the subsequent exponential increase in leakage). Current density is also on the increase.*
- *Active leakage is now a significant portion of SoC active power budget.*
- *Sleep mode techniques need to be enhanced and enabled in a consistent fashion throughout the design flow*
- *We need a concerted effort applied to leakage minimization at the micro-architectural, system and software level.*
- *Process variation now limits how much we can voltage scale and how we do our power accounting, and therefore new strategies need to be developed to capture these constraints, and enhance our current scaling approaches/methodologies.*

# Ultra-Low-Power Applications

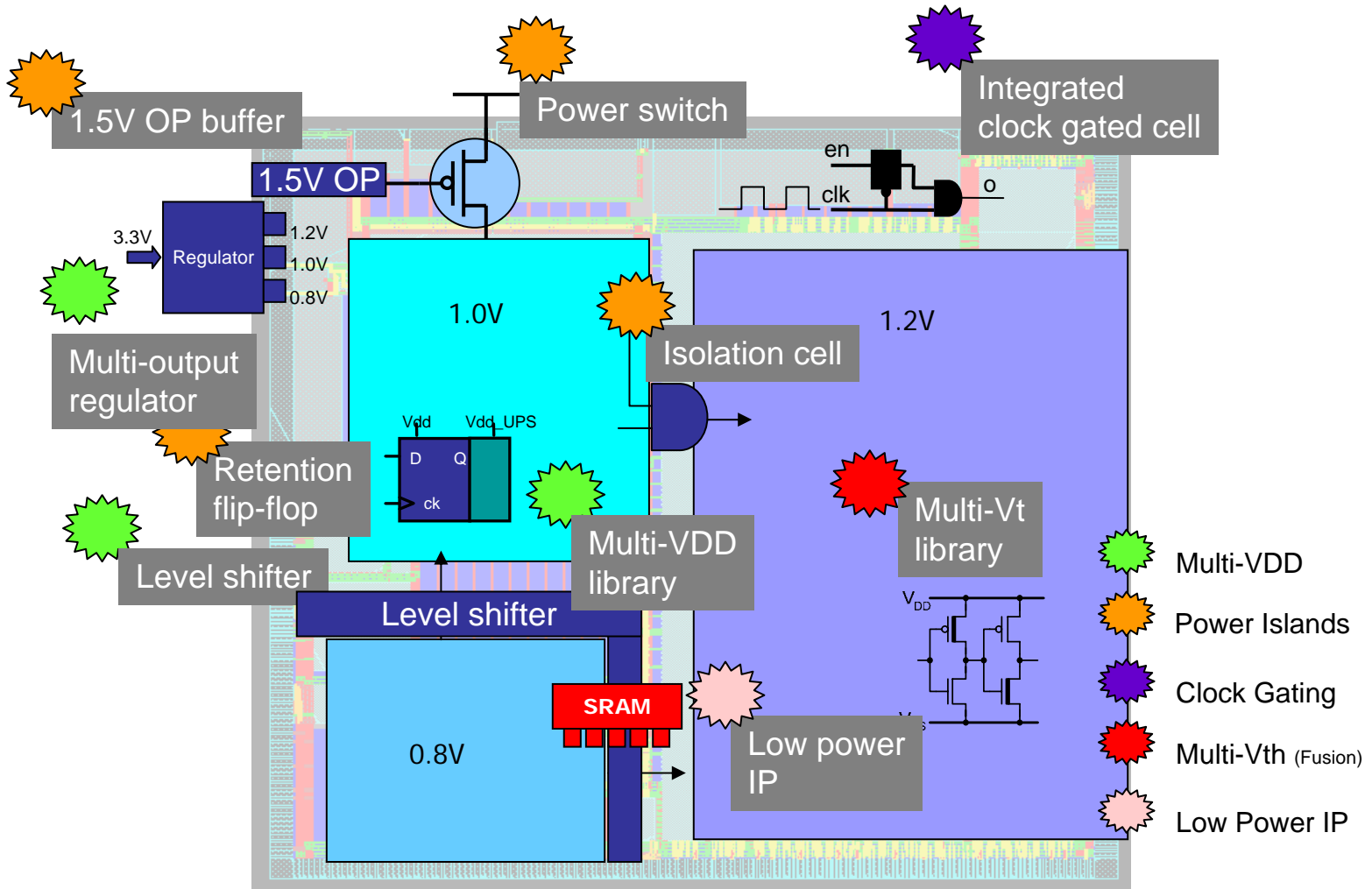
## Addressing a Wide Range of Requirements

Application Segment	User Expectations	Application Requirements
<b>Cellular Phones</b>	Receive and make voice calls, send messages, take, send and receive photos and videos, listen to music, play games for extended period before recharging battery	Medium to High-P erformance Long-Use time Long Stand-by Time
<b>Digital Cameras</b>	Take many photos, quickly review them and take more photos without worrying about battery life	Moderate to high-performance Long Play Time Stand-by irrelevant
<b>Hand-held Gaming</b>	Play games for long time before replacing batteries	High-P erformance Long Play Time Stand-by irrelevant
<b>Personal Digital Assistants</b>	Do everything a laptop can for extended period before having to plug into a wall	High-P erformance Long Use Time Long Stand-by Time
<b>Portable Media Players</b>	Listen to music, watch movies for long time without interruption to recharge	Moderate to high performance Long Play Time Stand-by Irrelevant
<b>Wireless LANs (802.11)</b>	Use the wireless device as if it is connected to a wire and to a wall	Moderate Performance Long-Use Time Low Stand-by

# Processor design for Power Efficiency: Different needs for different markets

- Server market:
  - Defining property: Server processors are rarely idle.
  - Power goal: Increase MIPS/Watt in Power State C0 (ACPI).
- Mobile market:
  - Defining property: Laptop processors are mostly idle.
  - Power goal: Reduce power in C2/C3 power states.
- Techniques:
  - Clock gating
  - Multiple power domains
  - Multiple threshold voltages
  - Headers/footers
  - Operand Isolation (holding cell inputs stable when output is unused)
  - Dynamic voltage and frequency scaling
  - And others...

# PowerSmart™ -- Low Power Design Methodology



# Power analysis challenges: More complex than timing analysis

- It is pattern-dependent.
  - Circuit and gate-level power analysis require good RTL-level patterns for accurate results.
- It is a balancing act. (power efficiency)
  - Performance per watt (efficiency) is the metric, not Watts. Need to find blocks or nets that consume power without appropriate performance benefit.
  - Many tools sort blocks and nets by total power consumption not performance/watt. (E.g. clock nets burn a lot of power, but we already knew that)
- It is an aggregate (time and space) and a user-defined constraint.
  - Power analysis types: average power (for budgeting & package selection), energy (for battery life), peak power (IR drop analysis), etc.
  - E.g. Briefly higher localized power consumption can be tolerated for package selection, unless it exceeds limits.
- It requires coordination of data from physical design, gate design, RTL, and verification domains.
  - It requires knowledge in all these domains to cross-check results.
- Must allow for accuracy to be improved over time.
  - Detailed circuit-level power analysis data often comes too late in the design cycle.

# Other related issues: DFT and Timing

- Are scan paths hooked up in the RTL? Are they simulated in the Verilog? How are they verified?
- How do you analyze power consumption in scan mode?
- Timing also needs to know about the multiple voltage domains and operating points.
- Need to work on timing and power in one environment to achieve correct optimization and trade-offs.

# Current state ...

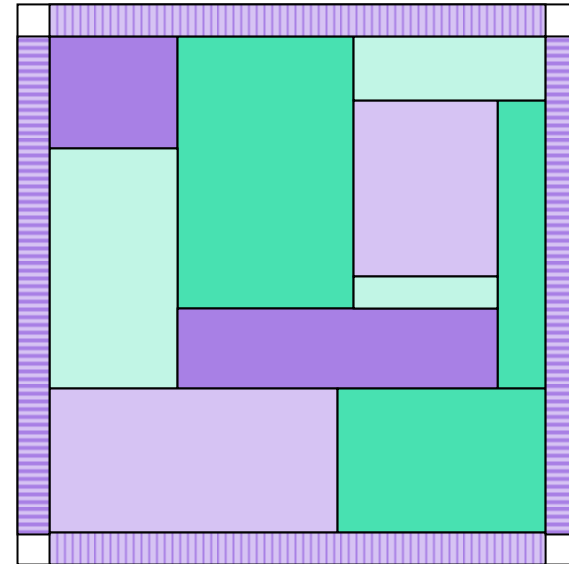
- Debugging capabilities are very poor
  - Capacity issues
  - Complexity issues
  - Reporting weak and misleading
- Functional correctness difficult to verify
- Tools are mostly in Gatelevel, should be in RTL
  - Important is to have accuracy for RTL or otherwise it is not useful
- All tools using different description for PM
  - PM configurations currently having thousands of statements in SoC level
  - No automation; It is designers responsibility to verify that all definitions are done correctly
  - Because updates for these definitions are done quite seldom, it is difficult to keep in mind complex configurations
  - There is no automation for PM definitions verification
- Design hierarchy presentation varies in configuration files between tools, also between RTL and gate in same tool. Syntax is effected by scripting languages like perl and tcl

## ***The verification flows need to enable:***

- *a voltage aware simulation method for logic problems due to voltage island partitioning*
- *a method for full design multi-voltage domain analysis and reporting*
- *a vector-less rule driven analysis of architecture, RTL, and gate correctness*
  - *a method for equivalence checking (i.e. across voltage states )*
- *a method that captures Island ordering*
- *a method that incorporates early detection of micro-architecture sequence errors*

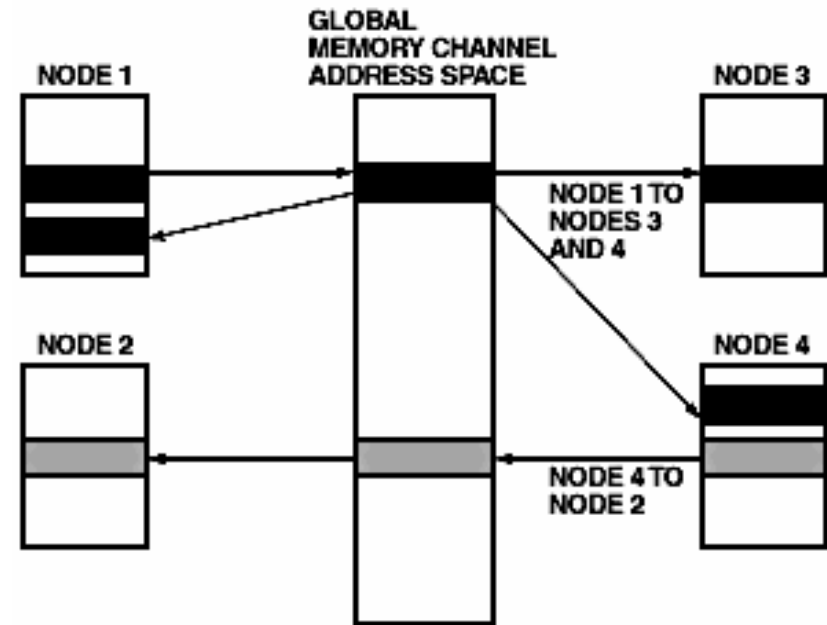
# System Aspects require differing views

- **Layout**
- Algorithm
- Bus Architecture
- Implementation
- Power
- Source
- Temperature
- Refinement
- Security
- Address Space
- Documentation



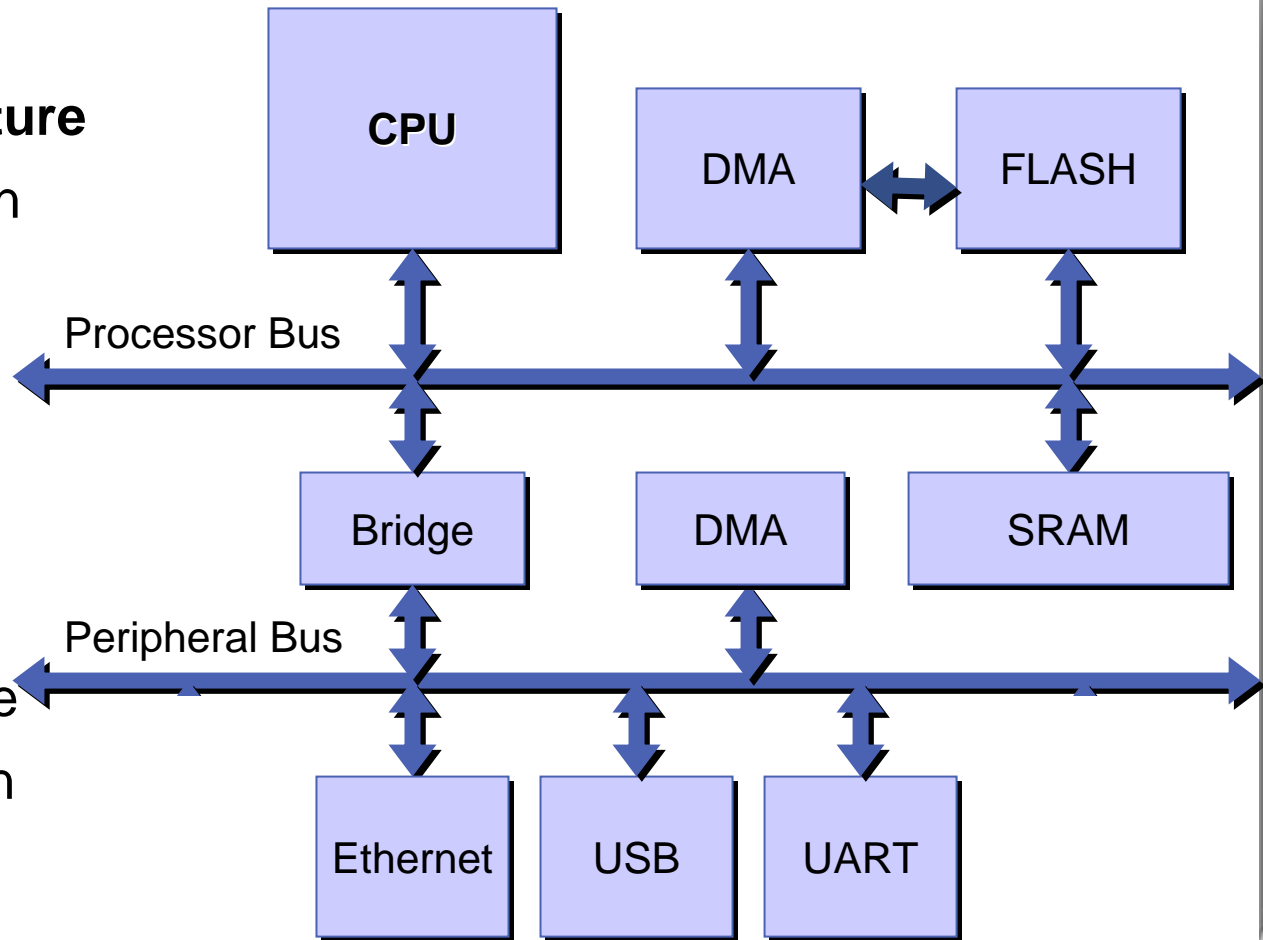
# Address space

- Layout
- Algorithm
- Bus Architecture
- Implementation
- Power
- Source
- Temperature
- Refinement
- Security
- **Address Space**
- Documentation



# Aspect View: Bus Architectural layout

- Layout
- Algorithm
- Bus Architecture**
- Implementation
- Power
- Source
- Temperature
- Refinement
- Security
- Address Space
- Documentation



# Hierarchical view of Energy Conservation

## Software Definitions

Dynamic system monitoring and intelligent control of energy savings, work load profiling, [dvfs], profiling and partitioning

## Platform Definitions

Power Trees/Voltage islands, Connectivity of components & consistent platform power modes, intelligent bus coding, dependency discovery/optimization

## Architectural Definitions

Heterogeneous processing resource optimization: MCU, DSP, accelerators, functional processing units, memory usage optimization

## Design Definitions

Hardware support for voltage islands, power gating, low-power idle modes, SRPG, AWB, DVFS, DPTC, clock gating

## PROCESS node Definitions

Transistor design, Vt Optimization, memory bitcell design. Special circuits, libraries, custom and analog blocks, SOI

Thanks to Milind Padhye, Freescale Semiconductor, Austin Wireless Design Center

# Low Power Design Needs

- Support Low Power Design Techniques thru the entire design flow using a single file format.
  - Design Representation
    - Accurately define and capture the low power design intent, modes and constraints.
  - Design Implementation
    - Floorplan and power grids.
    - Common constraints for all tools (Synthesis, APR, timing, DFT)
    - Design analysis tools with single power constraints.
    - Accurate power estimation and measurements
  - Design Verification
    - Voltage oriented simulators
    - Various static power technique modeling and simulations.
    - Silicon validation and correlation.

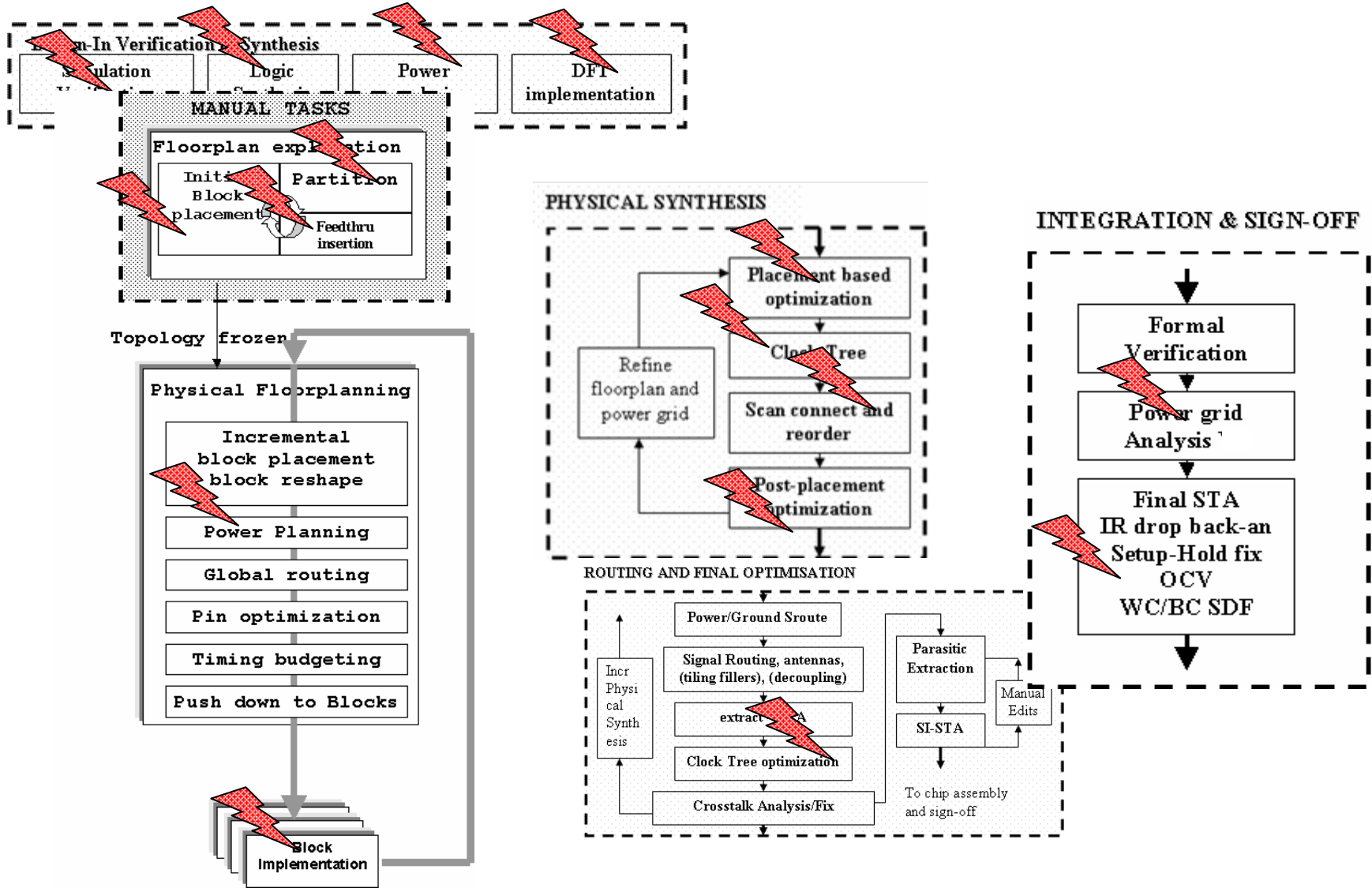
# Low power implementation : What's new ?

- ▶ Becoming mainstream:
  - For 65nm and below , Low power is crucial for low/high performance.
- ▶ So far:
  - For dynamic power
    - Reducing power dissipation source when not needed.
    - Minimize switching capacitances.
  - For static power
    - Use of multiple  $V_t$ (s) synthesis / optimization
- ▶ More recently:
  - Reducing supply reduces power, but also makes circuit slower. To meet both chip performance requirements and power goals, use voltage islands and voltage and frequency scaling.
  - Leakage can also be addressed by suppressing current when not needed.
  
  - Island of voltages increases the difficulty on implementation techniques.
  - Intrusive on functionality
  - Impact across design tasks ( Design-In and Implementation )

# Design implementation challenges

- ▶ New cells and their use model
  - Level Shifters
  - Retention logic
  - Isolation logic
  - Micro Switches
- ▶ Impacts at all levels of the design flow
  - Interface logic design, partitioning
  - Verification of power modes
  - Checks on interfaces between Power domains
  - Placement of IP in context voltage islands
  - Floorplanning with switches, Irdrop across switches, transient behavior.
  - DFT
  - Verification (STA, LVS, analysis)
- ▶ Conceptual shift : Power nets become functional signals

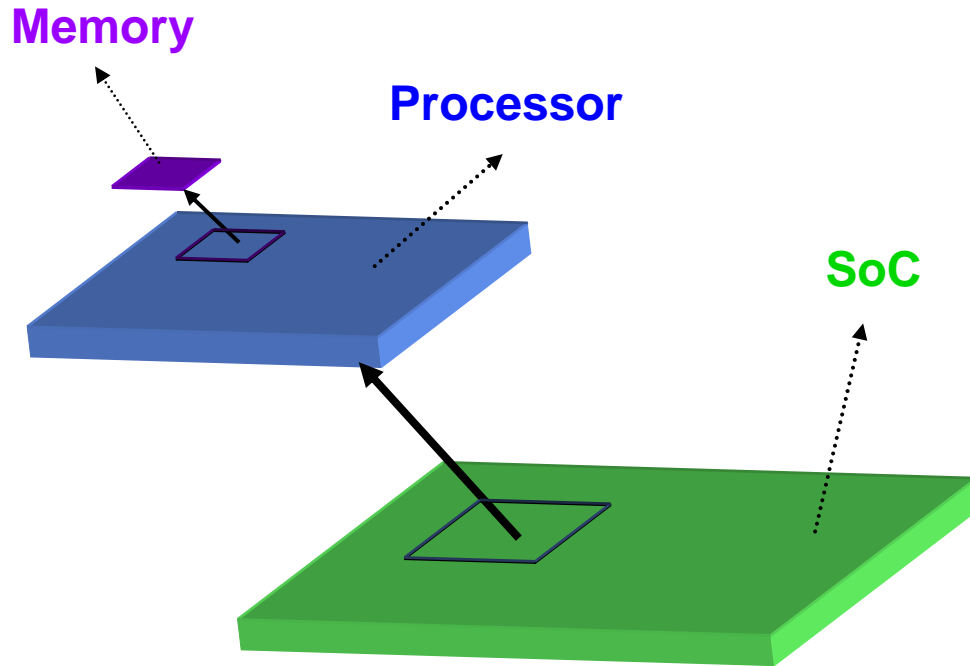
# Methodology and design flow impacted



# Short Term need ( 2) – Fill hole in Verification

- ▶ Low leakage design techniques have created a real paradigm shift.
- ▶ Power and ground nets are now becoming functional nets.
- ▶ They are not all explicitly in RTL or netlist levels.
- ▶ Proper connection of any other functional nets is verified by functional simulation....against the RTL or netlist.
- ▶ Being able to verify the power down modes , retention, recovery at power-on, etc in the context of RTL simulation is becoming mandatory.
- ▶ Verification tools should be power modes aware.

# For IP, context is key

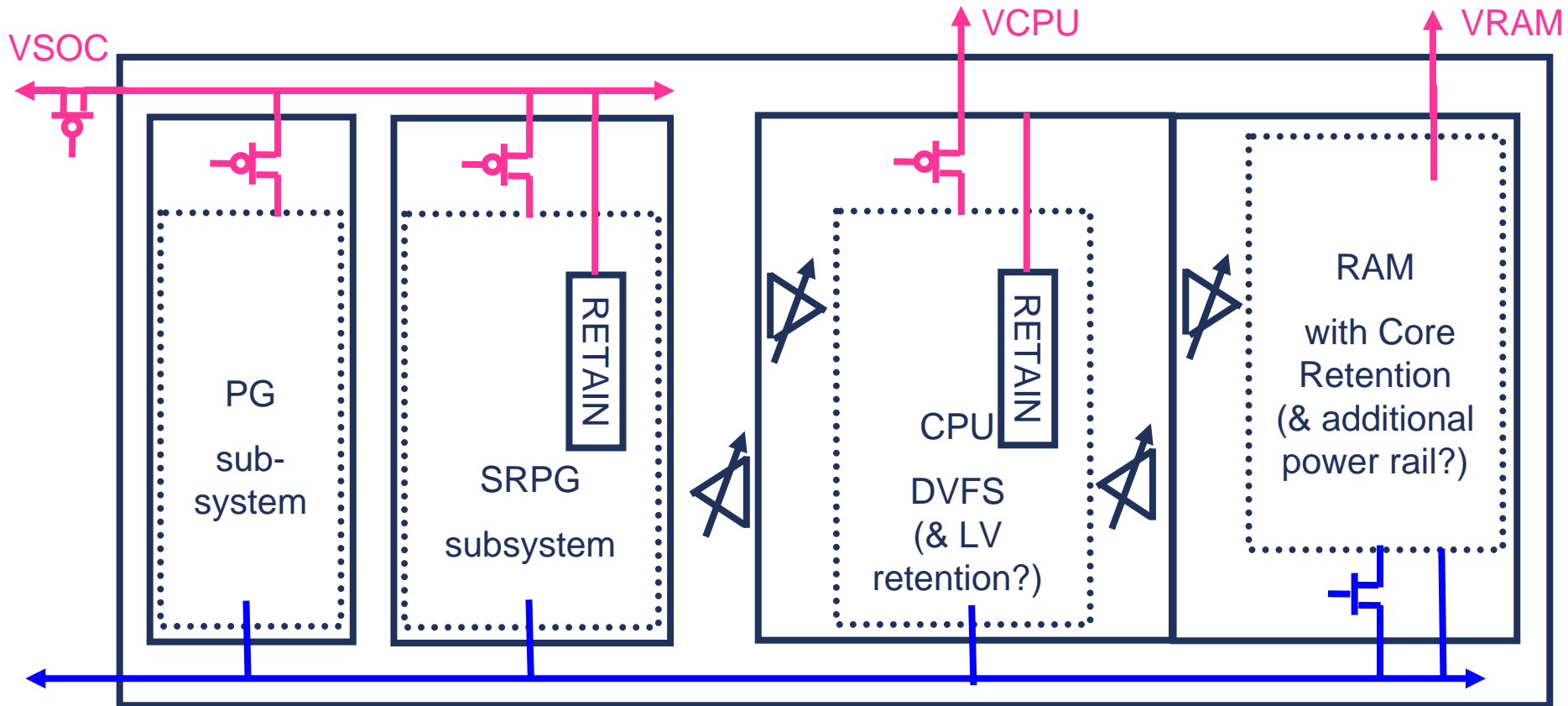


**Don't want formats limiting IP features**

- “Always-on” depends on context
  - Buffer within CPU
  - SoC buffer routed across CPU
- Characterization range is important
  - Cells, memory could be different
- Complex features
  - Multiple VDD, VSS pins
  - Multiple operating voltages
  - Voltage dependent behavior
  - Closed-loop behavior (tunable voltage)

# Canonical design to argue over.....

- Start with a realistic example to exercise interfaces and control
  - Power and Ground are signals – but not as we know them.....
  - Power Gating, Retention, (Dynamic) Voltage Scaling, Level shifters, Memory...
  - Isolation clamps across boundaries, a number of supply voltages
- e.g. a SOC with always powered logic plus:

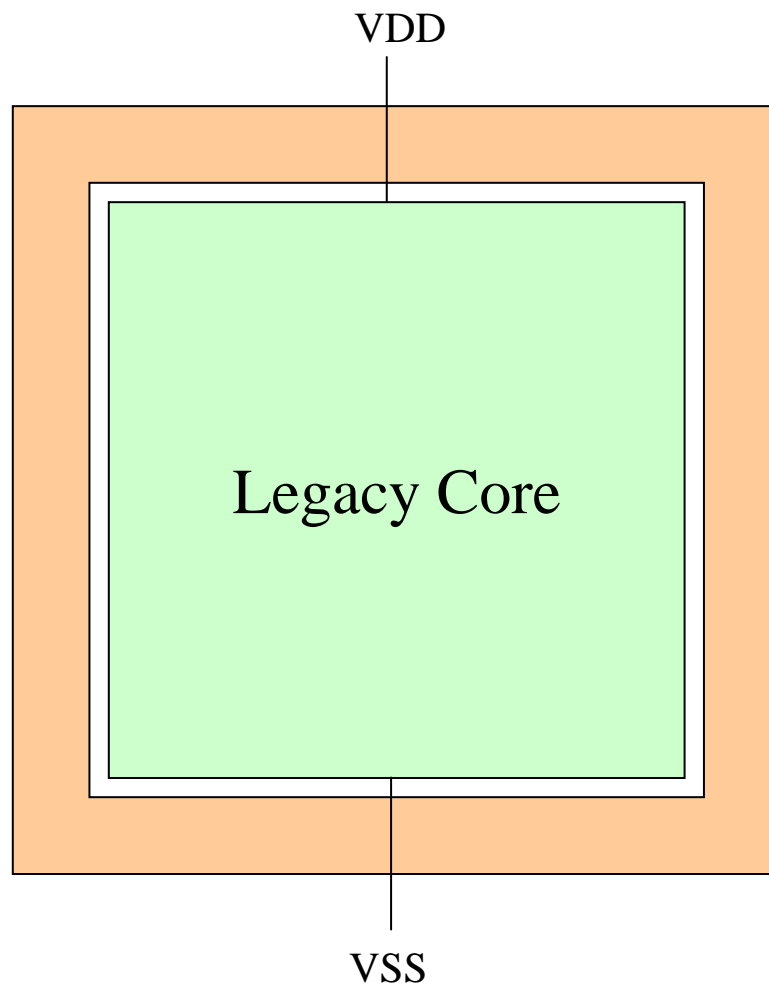


# Addressing power management challenges

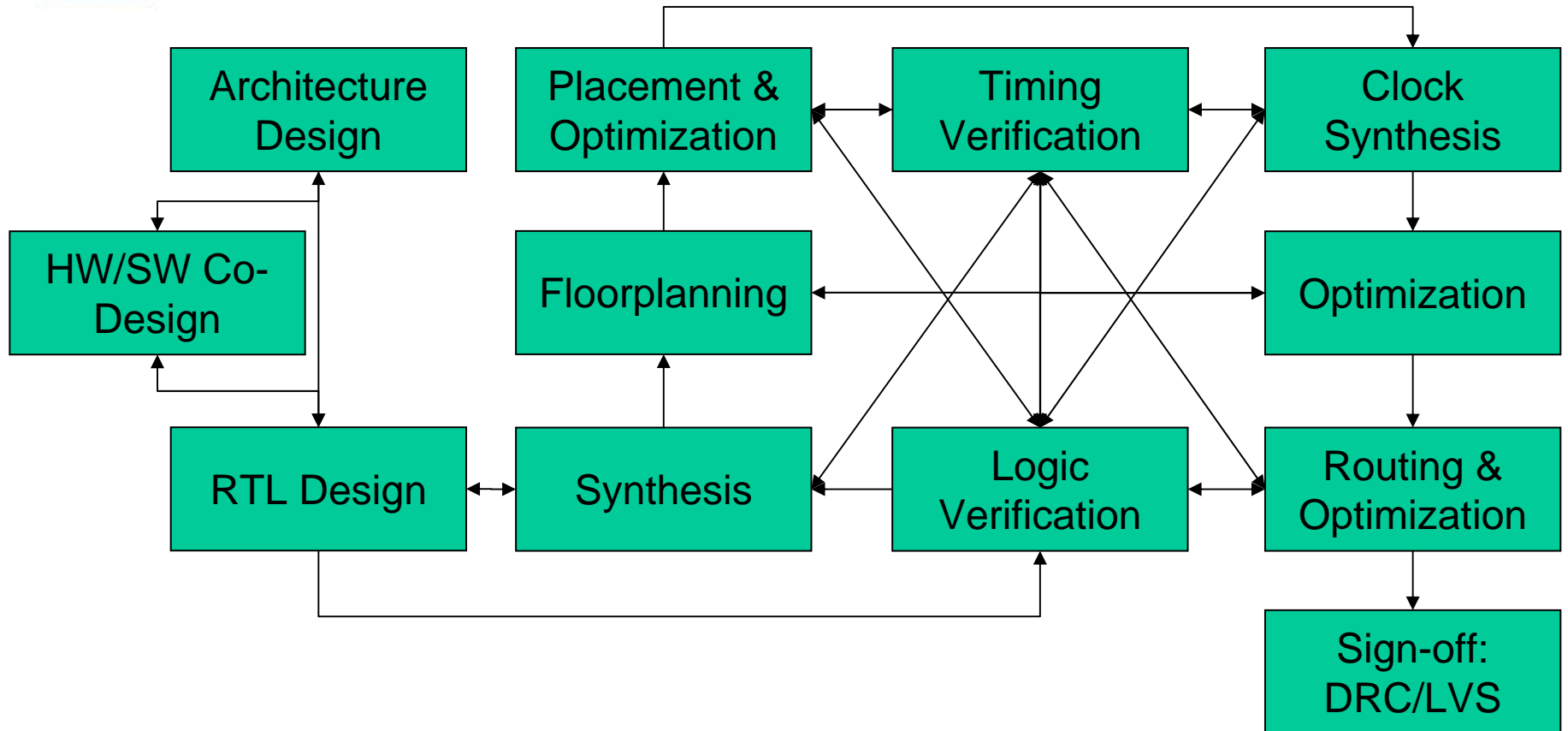
---

- Operational and Standby (leakage)
  - Active power + leakage
- Power gating/voltage scaling
  - On-chip – fast but with care to avoid  $dl/dt$  problems
  - Off-chip – may add latencies as long as 100's of microseconds
- Need to be able to quantify
  - Real-time cost (e.g. interrupt latency) in “wake-up” times
  - Energy cost functions getting into/returning from power saving states

# Legacy Core



# Traditional Design Flow

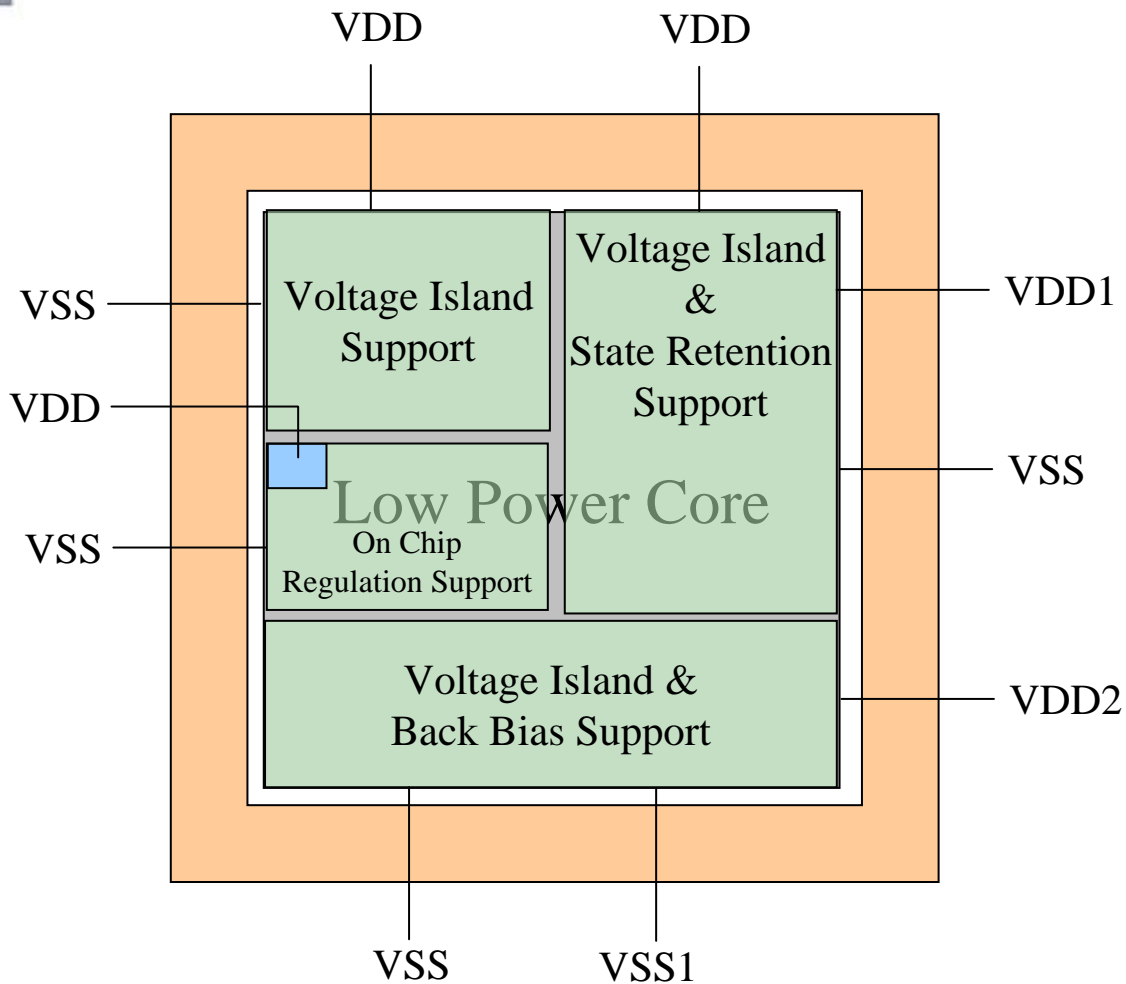




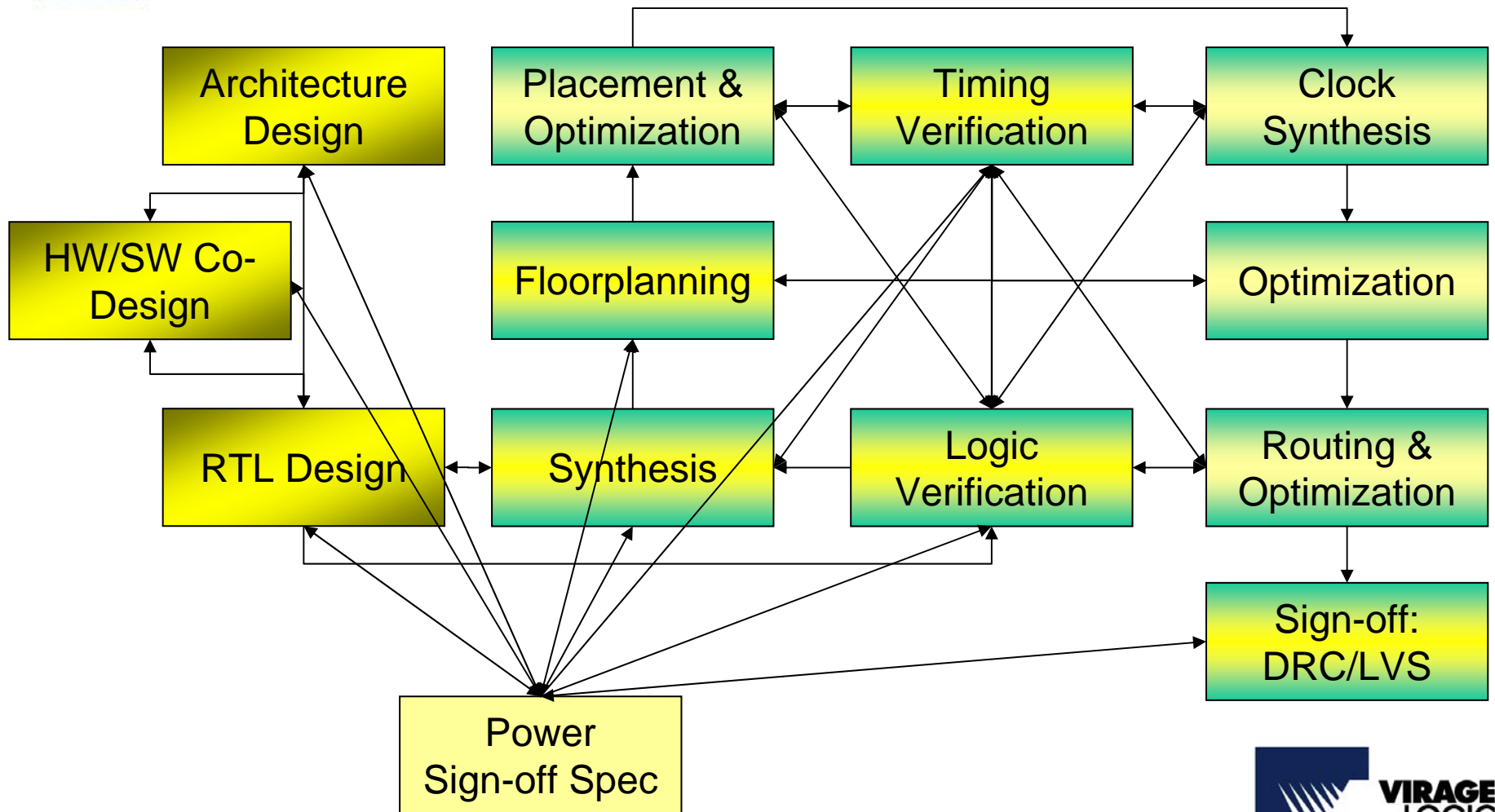
# Techniques Relevant To IP

- Right size libraries
  - Smaller transistors lead to smaller parasitics
  - Performance trade-off
- Multi-Vt libraries
  - Right Vt for the right paths at the right performance
  - Effectively used to control leakage
  - Increases the number of libraries needed to implement the design
- Voltage Islands
  - Requires updates to deal with multiple power supplies and associated conditions
  - Requires special level shifting components to implement
- Power Gating/On-Chip Regulation
  - Requires special power gating cells/regulation cells
  - Need to deal with “derived” power nets
  - Need to deal with POR cycle
- Substrate Bias
  - Requires dealing with multiple power supplies and possibly “negative” power supplies
  - Requires special level shifting components to implement

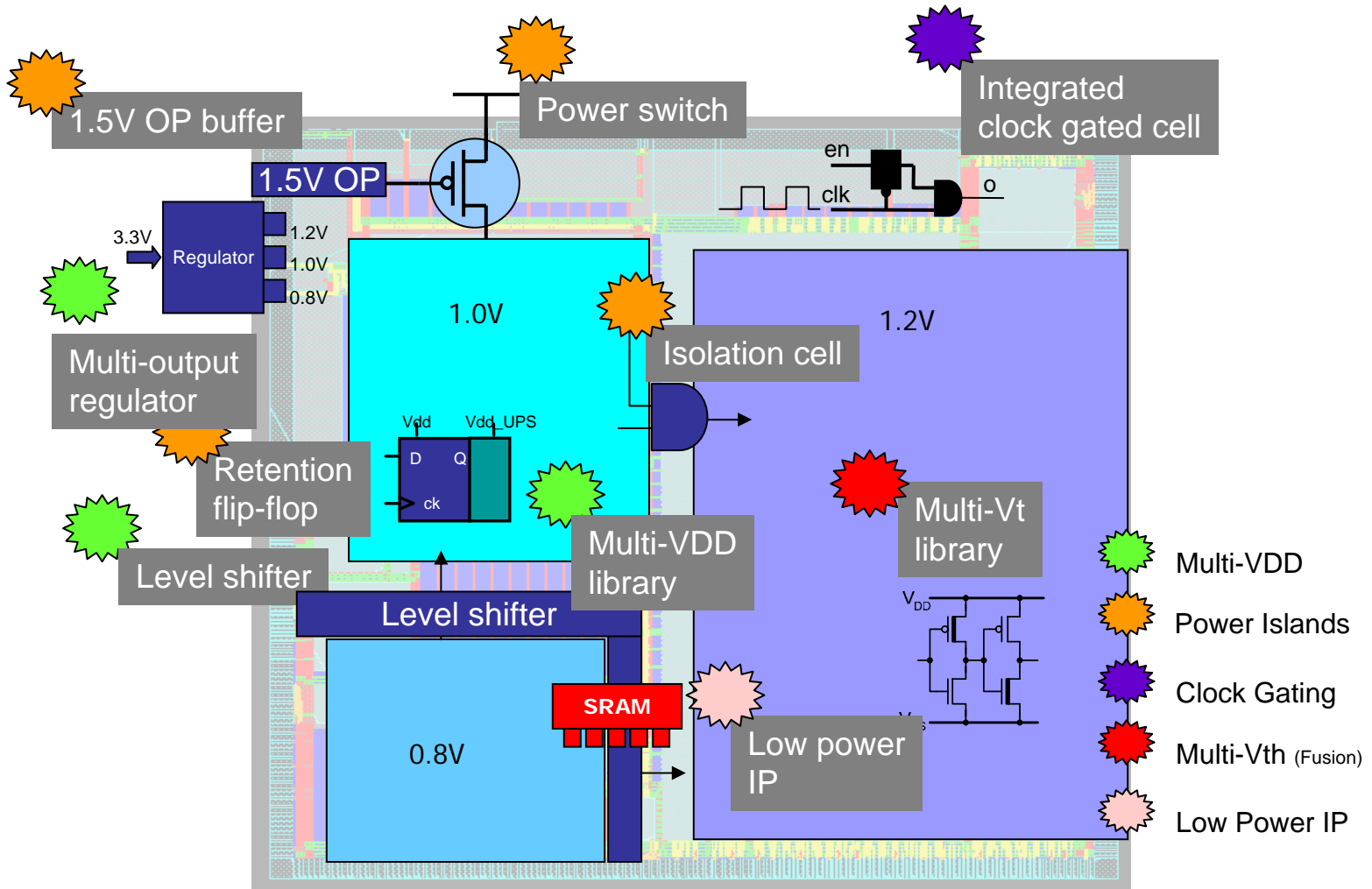
# So What Changes? ... Everything ...



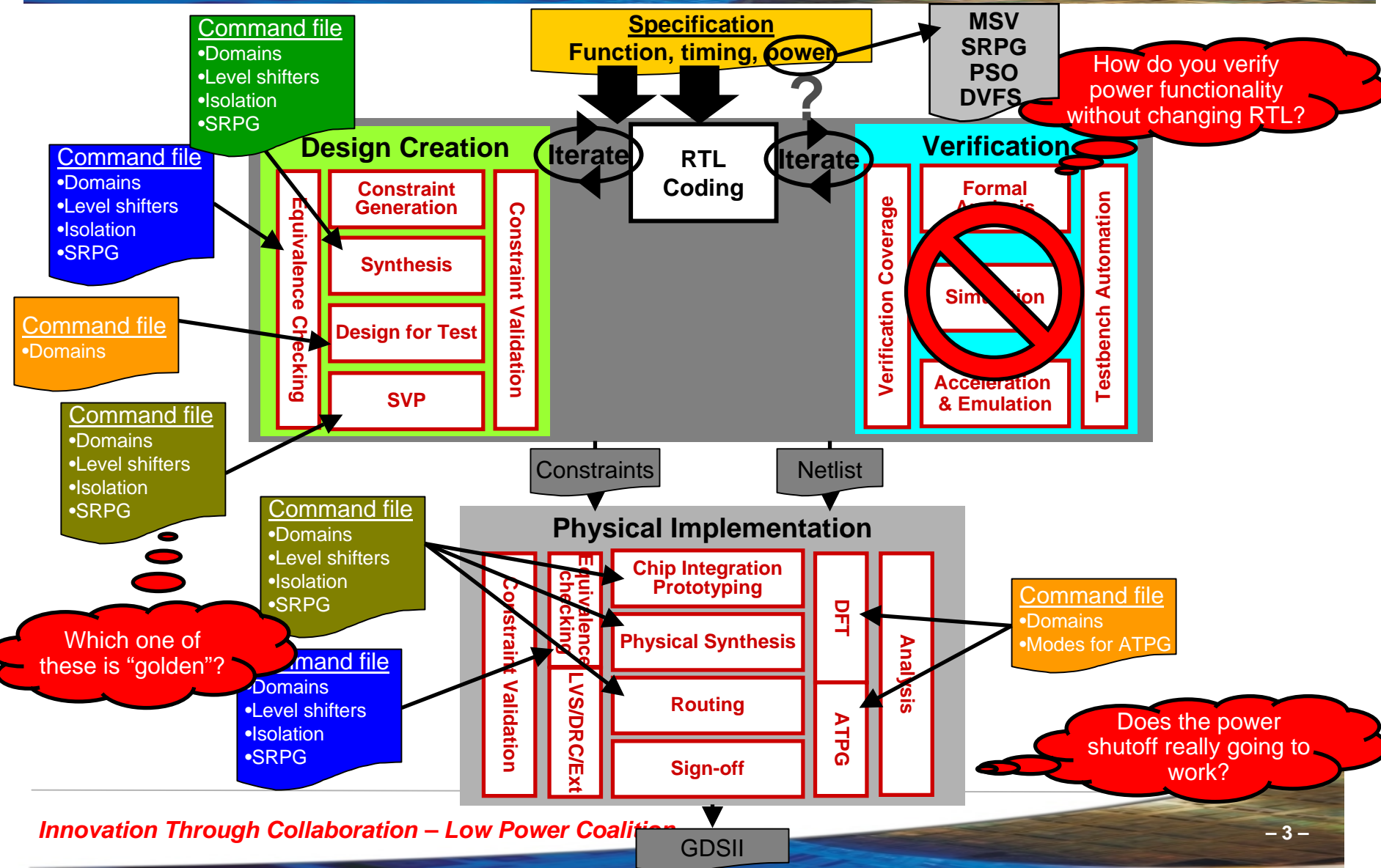
# So What Changes? ... Everything ...



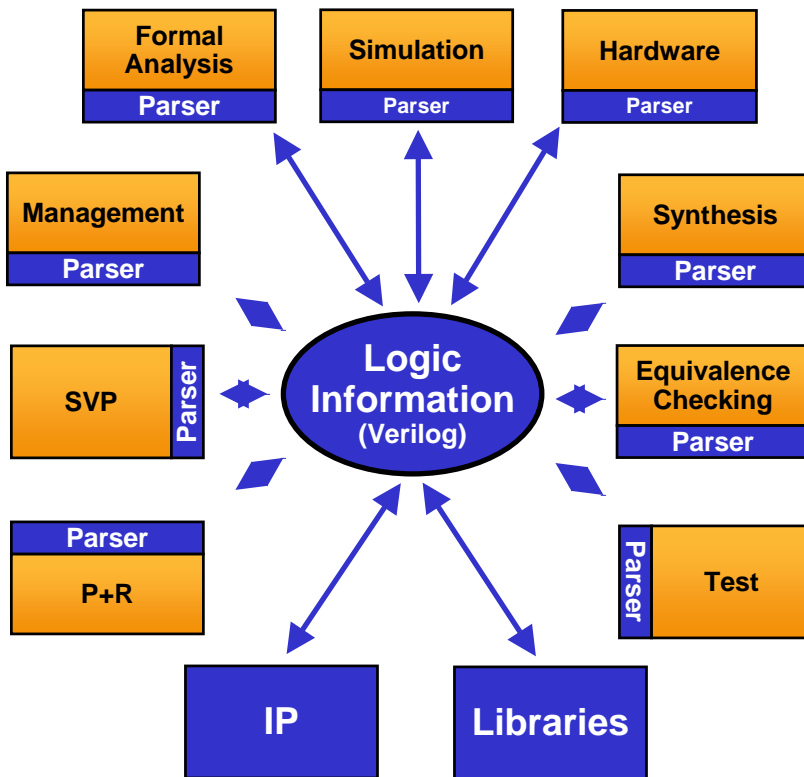
# PowerSmart™ -- Low Power Design Methodology



# Low Power Design Without A Power Format

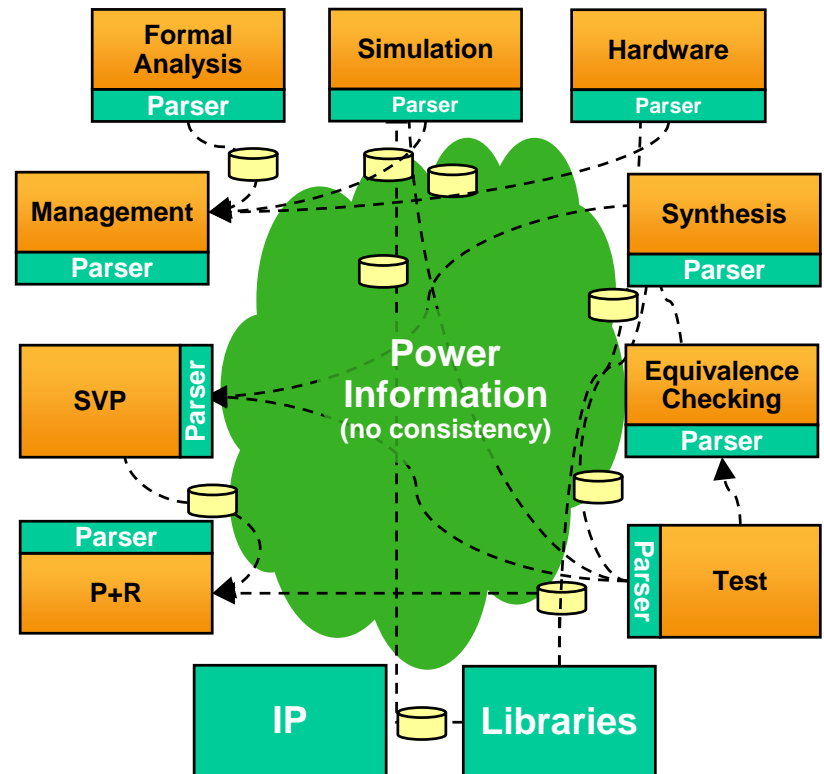


## Logic is “Connected”



Can be Automated

## Power is Not “Connected”



Very Difficult to Automate



- Dec 4, 2006
  - ◆ **Cadence contributed CPF v1.0 to Si2**
- January 12, 2007
  - ◆ **LPC members unanimously voted and approved CPF v1.0 as Si2 Specification for low power standard**
- January 17, 2007
  - ◆ **Cadence contributed CPF v1.0 parser source code to Si2**
- March 5, 2007
  - ◆ **CPF 1.0 available to everyone at no cost as a Si2 standard**

- **Design intent and constraints**

- ◆ Power domain
  - Logical: instances as domain members
  - Physical: power/ground nets and connectivity
  - Analysis view: timing library sets for power domains
- ◆ Power Logic
  - Level Shifter Logic
  - Isolation Logic
  - State-Retention logic
  - Switch Logic & Control Signals
- ◆ Power mode
  - Mode definitions
  - Mode transition definitions

- **Technology information**

- ◆ Level Shifter Cells, Isolation Cells, State-Retention Cells, Switch Cells, Always On Cells

- **CPF is TCL-based.**
- **CPF Language = TCL commands + CPF objects + Design objects**
  - Power domain
  - Analysis view
  - Delay corner
  - Library set
  - Operating condition
- **Design objects: objects that already exist in the RTL/gate netlist**
  - Module, Instance, Net, Pin, Port
- **Commands – 42 commands**
  - set\_\* commands [version, scope, and general commands]
  - define\_\*\_cell commands [library cell description]
  - create\_\*\_rule commands [design intent]
  - update\_\*\_rules commands [implementation directives]



# Minimal Command Set For Different Design Stages

create\_power\_domain  
create\_nominal\_condition  
create\_power\_mode  
create\_state\_retention\_rule  
create\_isolation\_rule  
create\_level\_shifter\_rule

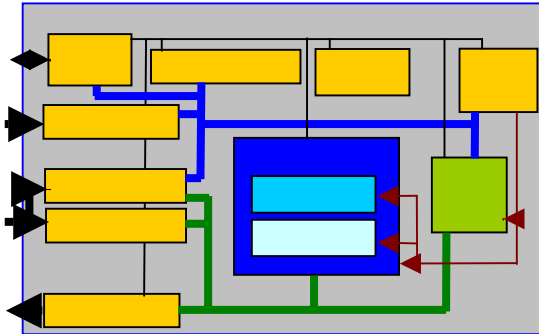
define\_library\_set  
update\_nominal\_condition  
update\_power\_mode

create\_ground\_nets  
create\_power\_nets  
update\_power\_domain  
create\_power\_switch\_rule  
create\_analysis\_view  
create\_operating\_corner

Specify power intents  
verification and simulation  
design exploration  
early power estimation

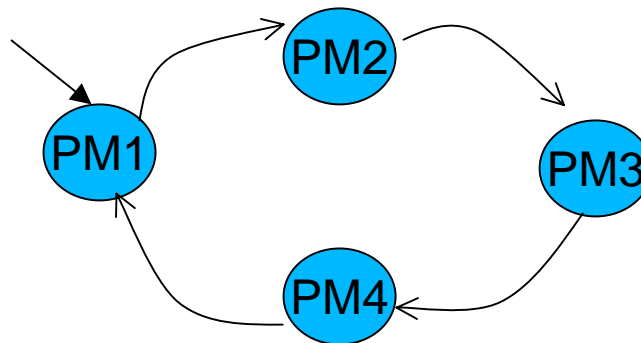
More implementation details  
synthesis  
formal verification  
DFT, ATPG,  
gate level power estimation

Complete physical implementation  
details  
silicon virtual prototyping  
power planning  
physical synthesis  
structural verification  
sign-off power analysis



	PDcore	PDau	PDlu	PDalu	PDrf
PM1	1.2v	1.2v	1.2v	1.2v	1.2v
PM2	0.8v	off	1.2v	1.2v	1.2v
PM3	0.8v	off	off	off	1.2
PM4	0.8v	1.2v	1.2v	1.2v	off

	PDcore	PDau	PDlu	PDalu	PDrf
PM1	1.2v	1.2v	1.2v	1.2v	1.2v
PM2	0.8v	off	1.2v	1.2v	1.2v
PM3	0.8v	off	off	off	1.2
PM4	0.8v	1.2v	1.2v	1.2v	off



```

create_mode_transition -name PM1toPM2 --from_mode PM1 --to_mode PM2 \
  -start_condition { pcu_inst/ctrl[0] & pcu_inst/ctrl[1] }
  -clock_pin { pcu_inst/clk } --cycles 100
create_mode_transition -name PM2toPM3 --from_mode PM2 --to_mode PM3 \
  -start_condition { pcu_inst/ctrl[0] & !pcu_inst/ctrl[1] }
  -clock_pin { pcu_inst/clk } --cycles 1000
create_mode_transition -name PM3toPM4 --from_mode PM2 --to_mode PM3 \
  -start_condition { !pcu_inst/ctrl[0] & pcu_inst/ctrl[1] }
  -clock_pin { pcu_inst/clk } --cycles 1000
create_mode_transition -name PM4toPM1 --from_mode PM2 --to_mode PM3 \
  -start_condition { !pcu_inst/ctrl[0] & !pcu_inst/ctrl[1] }
  -clock_pin { pcu_inst/clk } --cycles 200
  
```



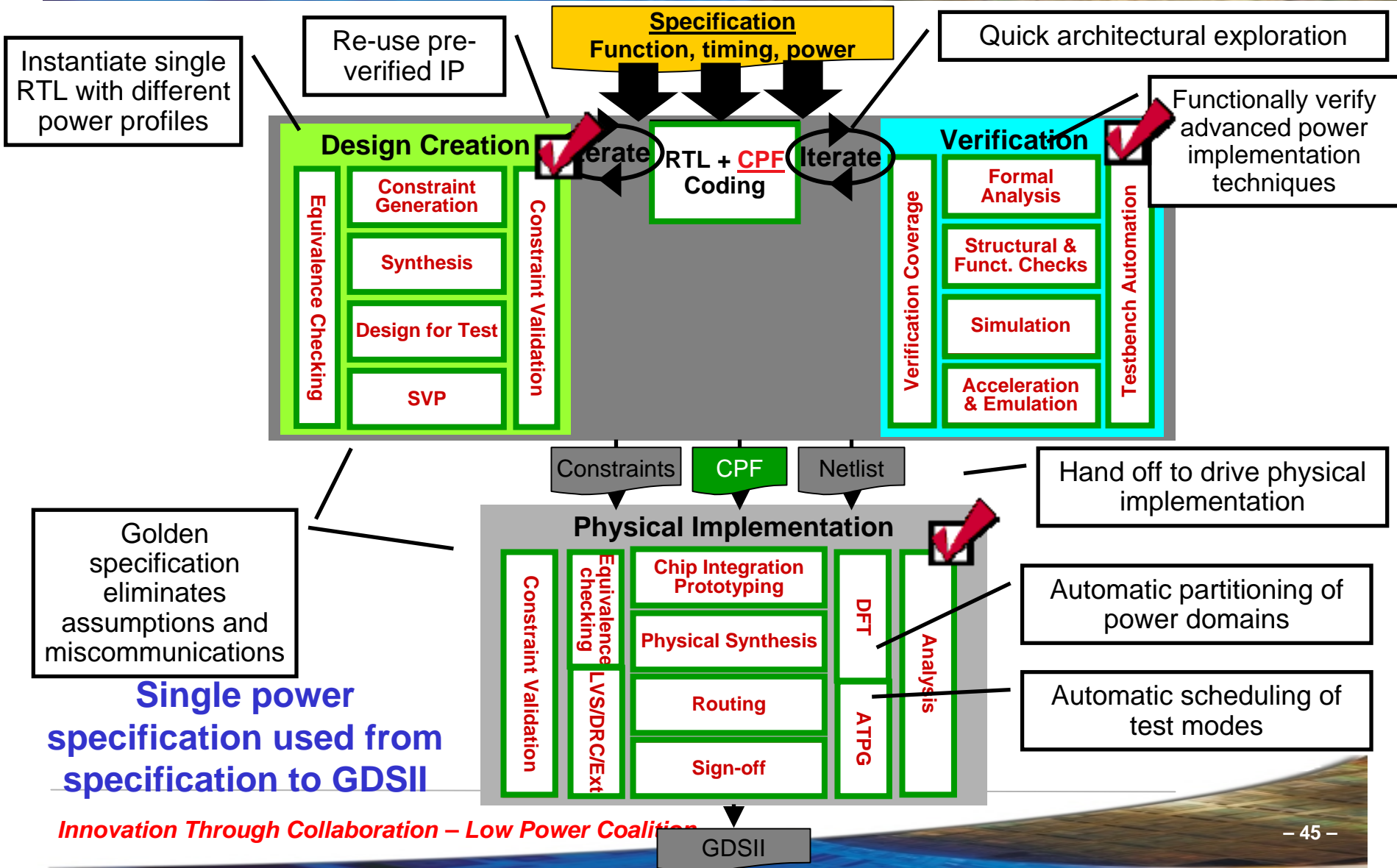
# Low Power Design Verification Using CPF

- **No need to specify power or ground nets at RTL stage**
- **No need to specify implementation related constraints at this stage such as library, timing constraints etc**
- **Minimal set of CPF commands for front-end designers to use**
  - ◆ Simulation tools
    - to simulation power domain on and off
    - to simulate power mode transitions for DVFS
  - ◆ Coverage tools
    - to check power mode coverage
    - to check power mode transition coverage
  - ◆ Assertion tools
    - to generate power domain and mode aware assertions
  - ◆ Verification tools
    - to check for the correctness and completeness of CPF

- **Still, no need to specify power or ground nets at this design stage**
- **Minimal set of CPF commands for designers to use**
  - ◆ Logic synthesis tools
    - to synthesize isolation, level shifter and state retention logic
    - to perform power domain aware logic synthesis
    - to perform power mode aware (DVFS) synthesis
  - ◆ Test synthesis tools
    - to perform power domain and power mode aware DFT synthesis
    - to generate power domain aware test control logic
  - ◆ Formal Verification tools
    - to check the correctness of low power structural implemented by synthesis tools
    - to perform low power equivalency checking (RTL+CPF vs Netlist)
  - ◆ Simulation tools
    - to perform power aware gate level simulation
    - to generate additional assertions for gate level simulation
  - ◆ Analysis tools
    - to perform power domain aware and power mode aware power analysis



# CPF Enabled Low Power Design Flow



1Q2007

2Q2007

2H2007



- CPF becomes Si2 standard
- Cadence Low Power Solution production released V 1.0



Reference Flow 8.0



PRIDE Flow



Common Platform Flow



UMC Joins PFI

CALYPTO PowerPro CG

denali DDR PHY



EnergyPro Technology



Joins PFI

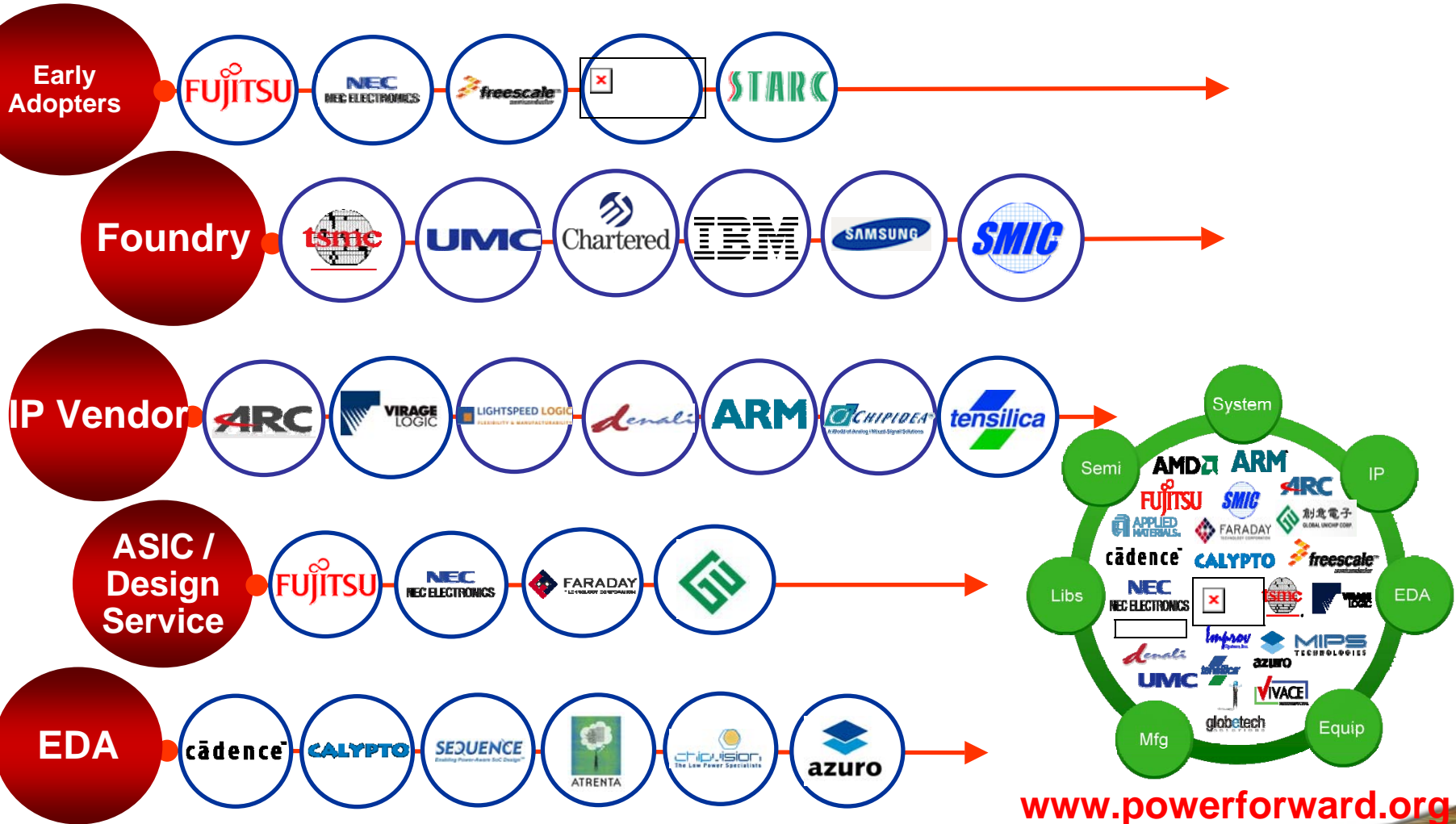


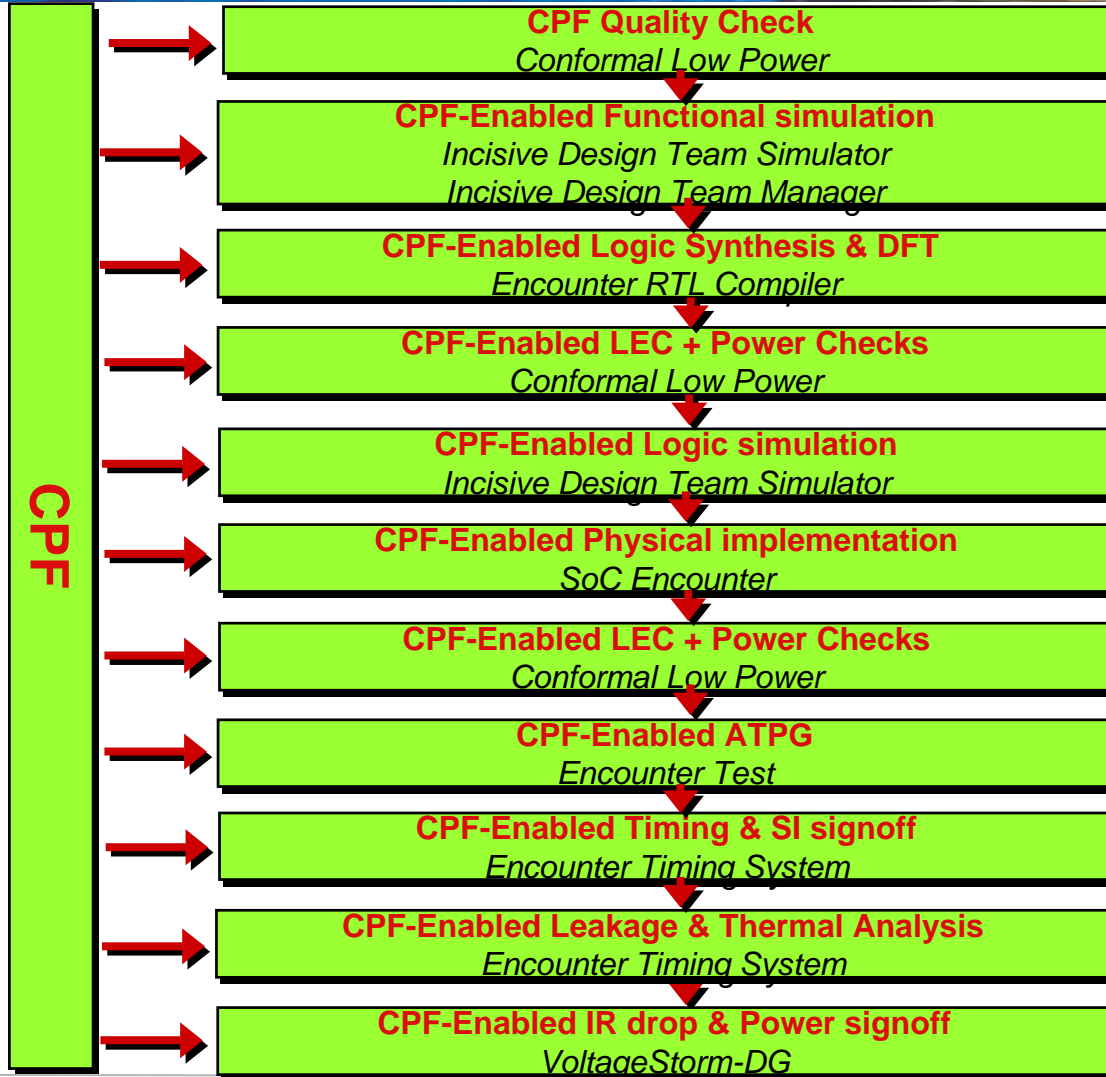
Joins PFI



Joins PFI

- > 100 customer adopting CPF-based advanced low power solution
- ~ 50 tapeouts  
Freescale, Fujitsu, NEC, NXP..





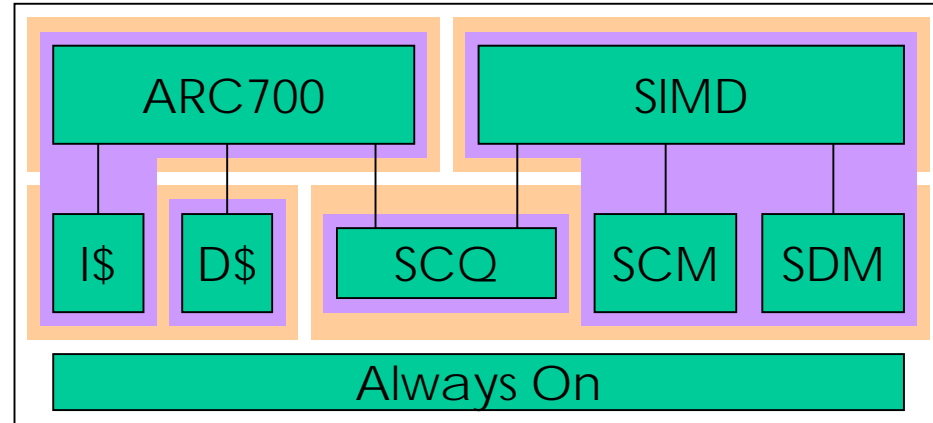


## Power Forward

### low-power implementation & verification project results

- Simulation with CPF identifies problems that you will not otherwise identify
- CPF aids communication of power intent across team boundaries, ensuring accurate implementation at all flow stages
- Significant power savings results using these techniques

## ARC700 with SIMD Co-Processor

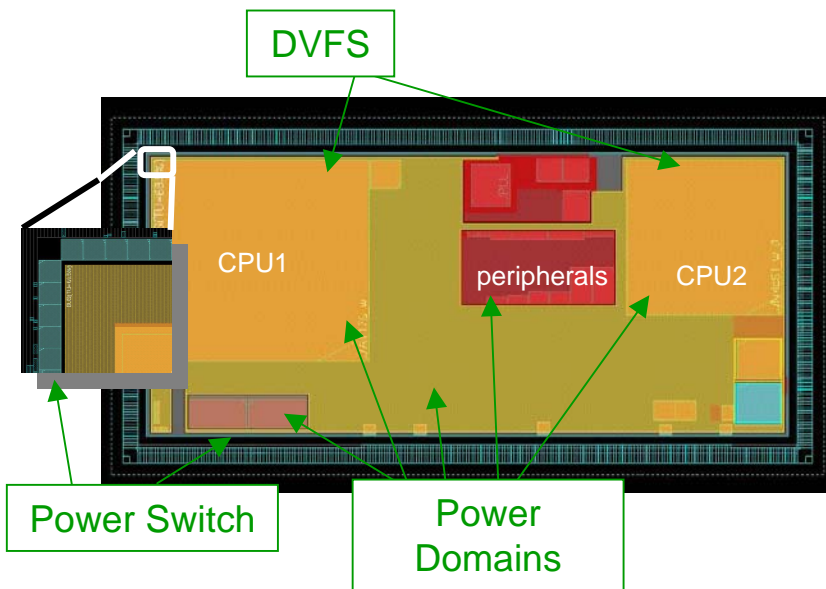


- Functional Blocks
- Power Domains
- Clock Gating Domains

- For high bit-rate data streams, both the ARC and the SIMD run flat out
- For lower bit-rate data stream, the subsystem can be run at a lower frequency
- For generic processing, the SIMD can be inactive



90nm  
940K instances  
11 Power Domains  
19 Power Modes



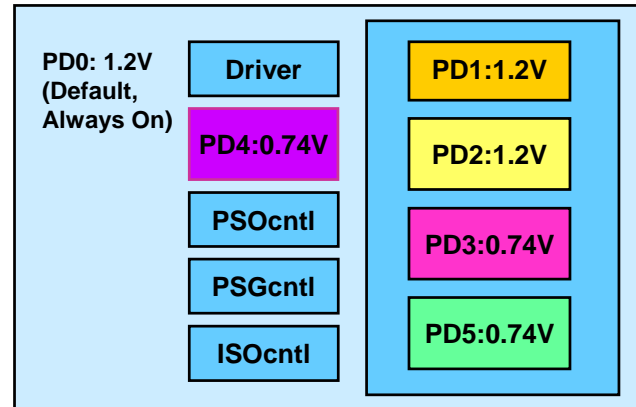
Silicon Proven September '07

- **Verified with test design**
  - ◆ PSO functional verification with simulation
  - ◆ Low power structural and physical check (Shifters/Isolators/Power switches)
  - ◆ Domain aware place and route
- **Conclusion**
  - ◆ Functional verification is necessary for complex PSO design for design bugs
  - ◆ Structural check with CPF could verify LP design
  - ◆ Fujitsu will support CPF-based ASIC flow for their customers

## NEC Electronics Corporation



65nm  
6 Power Domains  
5 Power Modes  
2 Supply Voltage



**Validated CPF and CPF-based flow for major low power methodologies in NEC Electronics**

- ✓ 386 checkpoints evaluated successfully
- ✓ CPF describe-ability
- ✓ Multi-Supply-Voltage (MSV)
- ✓ Power Shut Off (PSO)
- ✓ State Retention Logic (SRL)
- ✓ Variable Voltage Library (VVL)
- ✓ Clock Tree Gating (CTG)

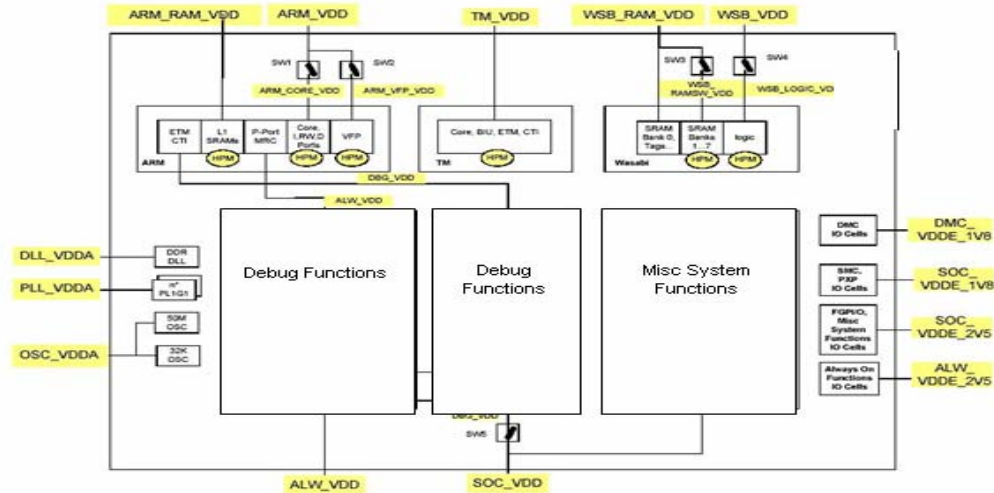
CPF based flow will be in use from Q3/2007

Power Mode	Power Domain					
	PD0	PD1	PD2	PD3	PD4	PD5
PM1	1.2V	1.2V	1.2V	0.74V	0.74V	0.74V
PM2	1.2V	PSO	1.2V	0.74V	0.74V	0.74V
PM3	1.2V	1.2V	PSO	0.74V	0.74V	0.74V
PM4	1.2V	1.2V	1.2V	PSO	0.74V	0.74V
PM5	1.2V	PSO	PSO	PSO	0.74V	PSO



## Power Forward low-power platform SoC results

- CPF-based functional verification (using simulation) catches system level power issues early in the flow
- Use of CPF ensured what implementation built was what was verified



- SoC consists of 11 islands
- 3 major power consumers -RISC CPU, VLIW DSP & L2 System Cache are controlled using DVFS
- High bandwidth expansion ports enable extension, with graphics or cellular modem subsystems



- Three Working Groups
  - ◆ Data API
    - Common Glossary
  - ◆ Design Flow
    - Low Power Design Flow Document
  - ◆ Format Requirement
- Format Requirement Working Group
  - ◆ Clarification on CPF 1.0 semantics
  - ◆ Collect new requirements for format improvements
    - Custom macro modeling
    - More flexibility on IP reuse
    - Complete hierarchical flow
    - ...



# What Is The Low-Power Coalition?

---

- Flow-based solutions
    - ◆ Standards to promote integration of open technologies into cohesive flows
      - *CPF contributed to LPC 4Q'06, approved as new Si2 standard in Mar'07*
    - ◆ Analyze / develop semantic consistency across data exchanges
  - User-centric and comprehensive
    - ◆ Focused on user needs for faster adoption into production chip design flows
    - ◆ Owns the industry's low-power roadmap of requirements
    - ◆ Comprehensive: enabling software, training & educational materials, articles, books, conferences, press coverage, etc.
-



# LPC Member Companies

---

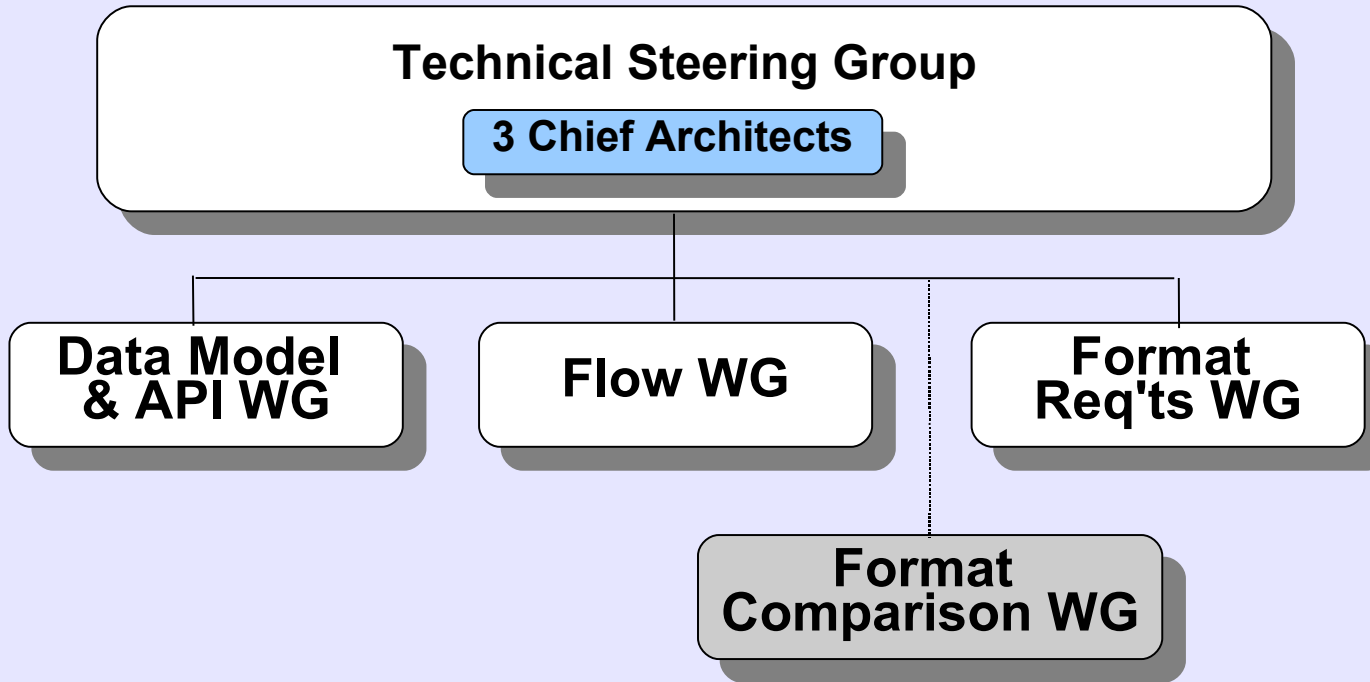
- **Advanced Micro Devices**
- **Apache Design Solutions**
- **ARM**
- **Atrenta**
- **Azuro**
- **Cadence Design Systems**
- **Calypto Design Systems**
- **Chipvision Design Systems AG**
- **Entasys Design**
- **Freescale Semiconductor**
- **Golden Gate Technology**
- **IBM Corporation**
- **Intel Corporation**
- **LSI**
- **NXP Semiconductors**
- **Sequence Design**
- **ST Microelectronics**
- **Virage Logic**

- **7 End-Users**
  - **9 EDA Companies**
  - **2 IP Providers**
-



# LPC Structure

## Full LPC Membership





# LPC Structure, Working Groups

---

- Full LPC membership (AMD, Chair)
    - ◆ Business/policy & standards approvals
  - Technical Steering Group (TSG): charters working groups, owns the low-power technology roadmap
    - ◆ Includes 3 Chief Architects (Cadence, IBM, LSI)
  - Active and completed working groups:
    - ◆ ***Format Comparison WG*** – report on technical comparison of CPF and UPF (Done, results widely shared)
    - ◆ ***Flow WG*** – align on low-power reference design flow and design techniques to drive clarity for enhancements
    - ◆ ***Data Model and API WG*** – map clear semantics and data relationships in CPF, add API interface support to CPF
    - ◆ ***Format Requirements WG*** – define priorities and detailed requirements for upcoming revision of CPF
-



# LPC Working Groups

---

- Flow Working Group
    - ◆ Definition of complete reference flow *from ESL to GDSII*
      - Target completion date: 1Q08
    - ◆ Analysis of power stimuli for *SoC power estimation*
      - Target completion date: 1Q08
    - ◆ Compilation of *all known low power design techniques*
      - Target completion date: 2Q08
    - ◆ Communication plan under early discussions
      - How do we publish each of these? How to propagate broadly across entire industry for better alignment?
      - Target completion date: 06/08
-



# LPC Working Groups

---

- Data Model Working Group status:
    - ◆ LP Glossary v1.0 completed
      - 60-day exclusion period (EP) ends 12/31/2007
      - Will be posted for general availability after EP
      - Target date: (not later than) 01/08
    - ◆ Developing UML-based models to support enhanced power-aware design
      - Will be based on OpenAccess data model
      - Target date: 1H08
-



# LPC Working Groups

---

- Format Working Group status:
    - ◆ Developing format extensions requirements document
      - Target date: 01/08
    - ◆ Open RFT... contributions expected by 01/08
    - ◆ CPF 1.1 standardization target: 06/08 (DAC)
    - ◆ CPF roadmap discussions underway with both Flows and data model WGs
-



# CPF Enhancements Roadmap

---

- Immediate
    - ◆ Expanded hierarchical flow support
    - ◆ Memory modeling styles and support
    - ◆ Gate level verification Flow CPF support
    - ◆ Power estimation support
    - ◆ Clarifications on CPF 1.0
  - Medium Term
    - ◆ Clocking and related updates required to drive power optimization
    - ◆ Pre-Si and post-Si power modeling and power budgeting
    - ◆ Test power definitions not already represented in formats
  - Long term
    - ◆ IO modeling and representation
    - ◆ Formats need to drive power-related silicon debug
    - ◆ Format based system level definition
-



# Recent Achievements

---

- Data Model and API WG
    - ◆ v1.0 Low-Power Glossary completed
  - Flow WG
    - ◆ Version 1.0 *Power Aware Reference Design Flow* completed
    - ◆ Document summarizing all known low power design techniques completed
  - Format Requirements WG
    - ◆ Technical work began in September with 7 active participants
    - ◆ CPF 1.0 semantic clarification process developed
    - ◆ CPF 1.x / 2.x requirements document nearing completion
  - Supplemental
    - ◆ CPF parser software released for general public downloads
    - ◆ Defined clear member and (non-member) contribution process
-



# CPF Adoption And Market Assessment

- CPF momentum (continued)
    - ♦ Over 50 CPF design flow engagements now active
    - ♦ Completed tape-outs include: **NEC, Fujitsu, Freescale, NXP, TSMC...**
    - ♦ 11 EDA tool suppliers now committed to CPF (with more in work)
      - **ARC, Apache, ArchPro, Atrenta, Azuro, Cadence, Calypto, Chipvision, Entasys Design, Golden Gate, Sequence Design**
-



# CPF Education / Training

---

- Free CPF 1.0 tutorial course, hosted by Si2 on Dec 6<sup>th</sup>
    - ◆ Presented using LiveMeeting streaming video / audio and 100 dial-in phone lines
    - ◆ 108 advance registrations for class
    - ◆ >750 downloads of CPF tutorial slides in 48 hours
    - ◆ Recorded for additional replays at student's convenience
  - Free CPF Pocket Guide
    - ◆ Includes CPF information model, command syntax, CPF example, and information on LPC
    - ◆ Free download from Si2 website
-