

Panel: It is All About Power Analysis, Exploration and Trade-offs

ISLPED 2009

August 20, 2009





- Moderator
 - Brian Fuller

- Panelists
 - Sandeep Mirchandani, Broadcom
 - Ran Avinun, Cadence
 - Sanjeev Das, NXP
 - Jon McDonald, Mentor
 - Camille Kokozaki, IDT

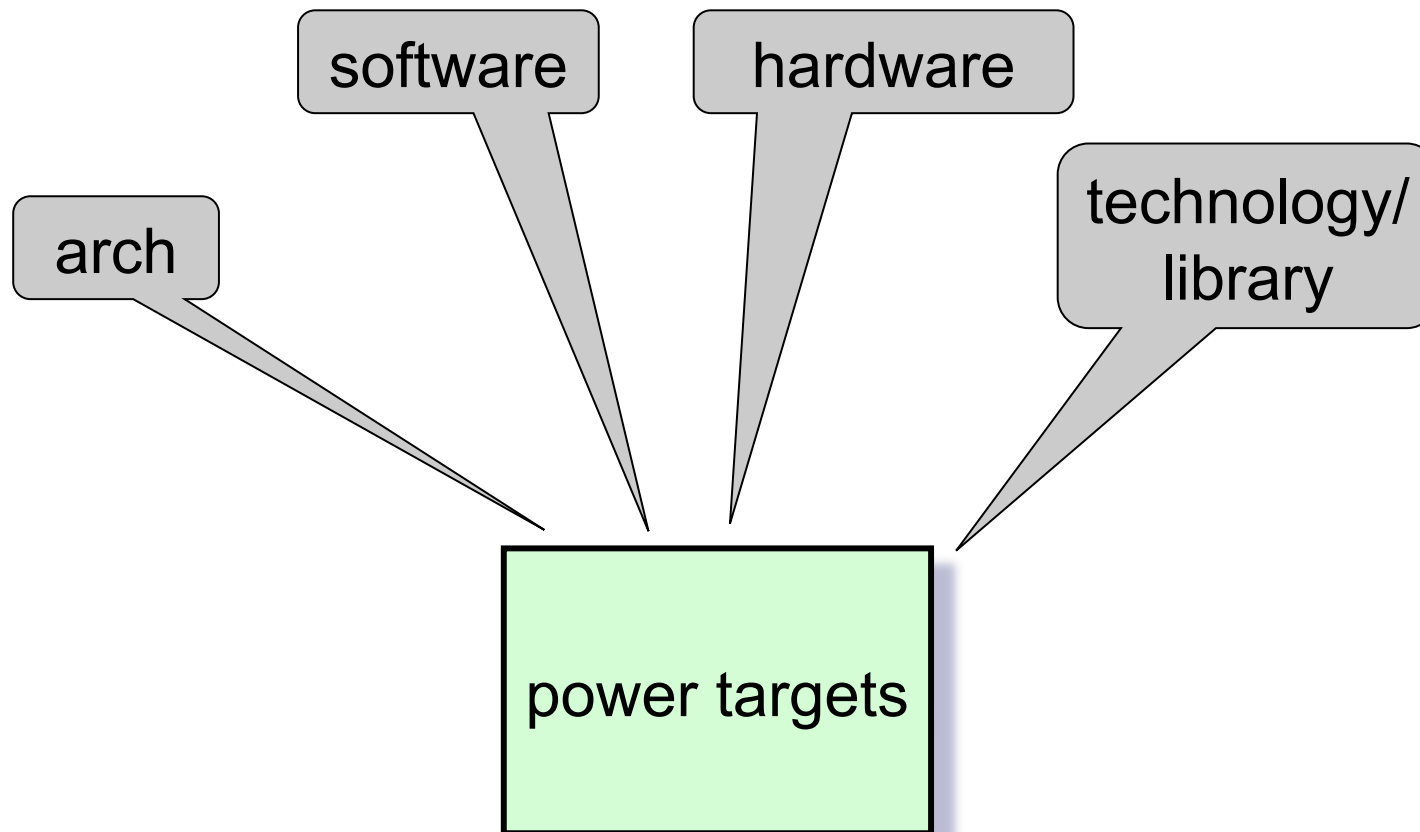




Power estimation and reuse

Sandeep Mirchandani
Associate Technical Director
sandeep@broadcom.com

power estimation for arch/sw/hw optimization

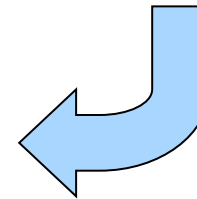


power estimation by scaling

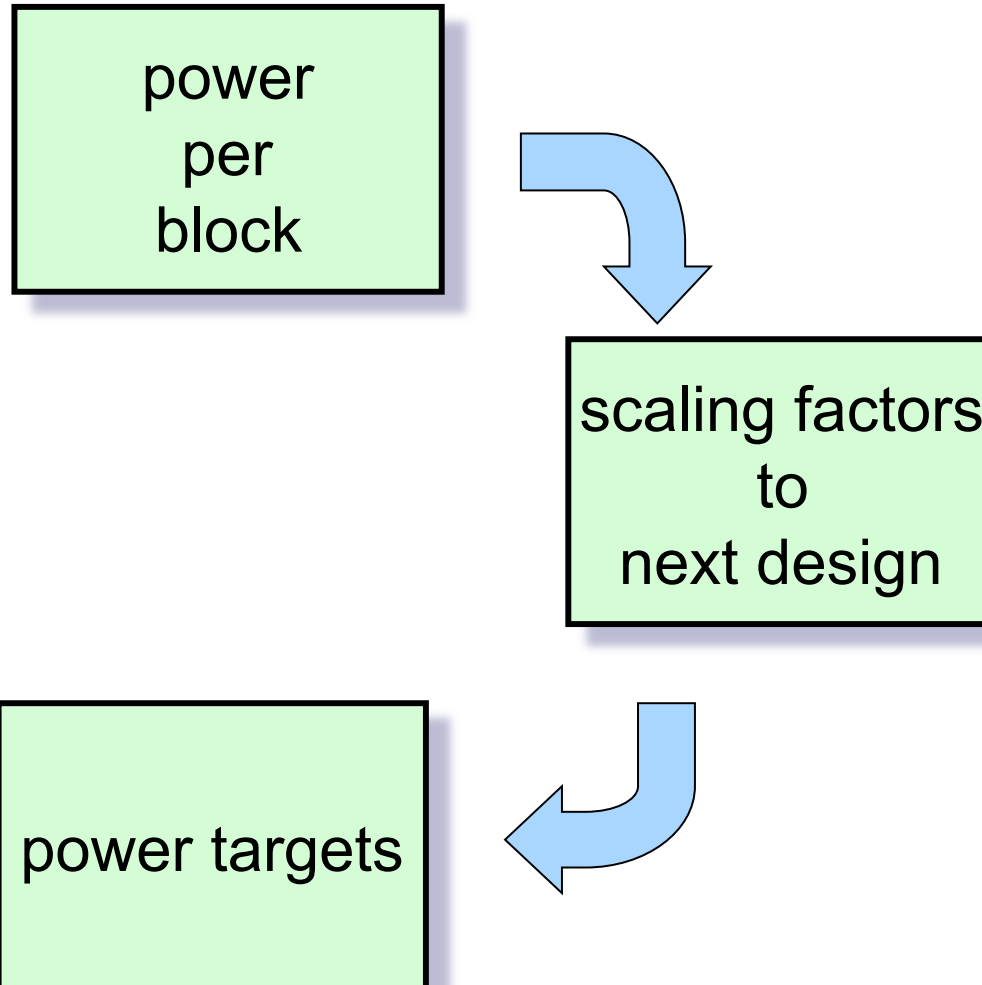


power targets

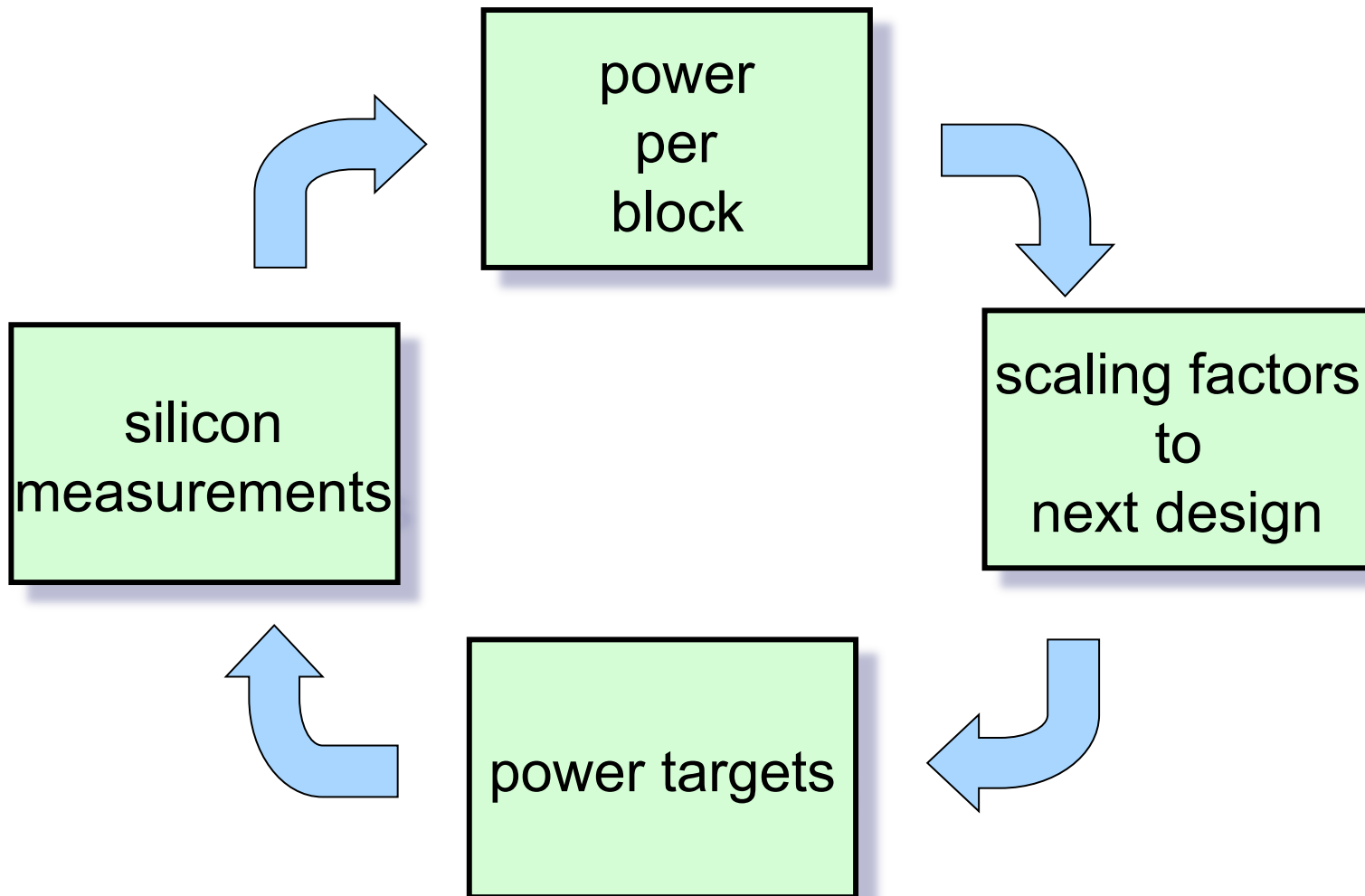
scaling factors
from
last design



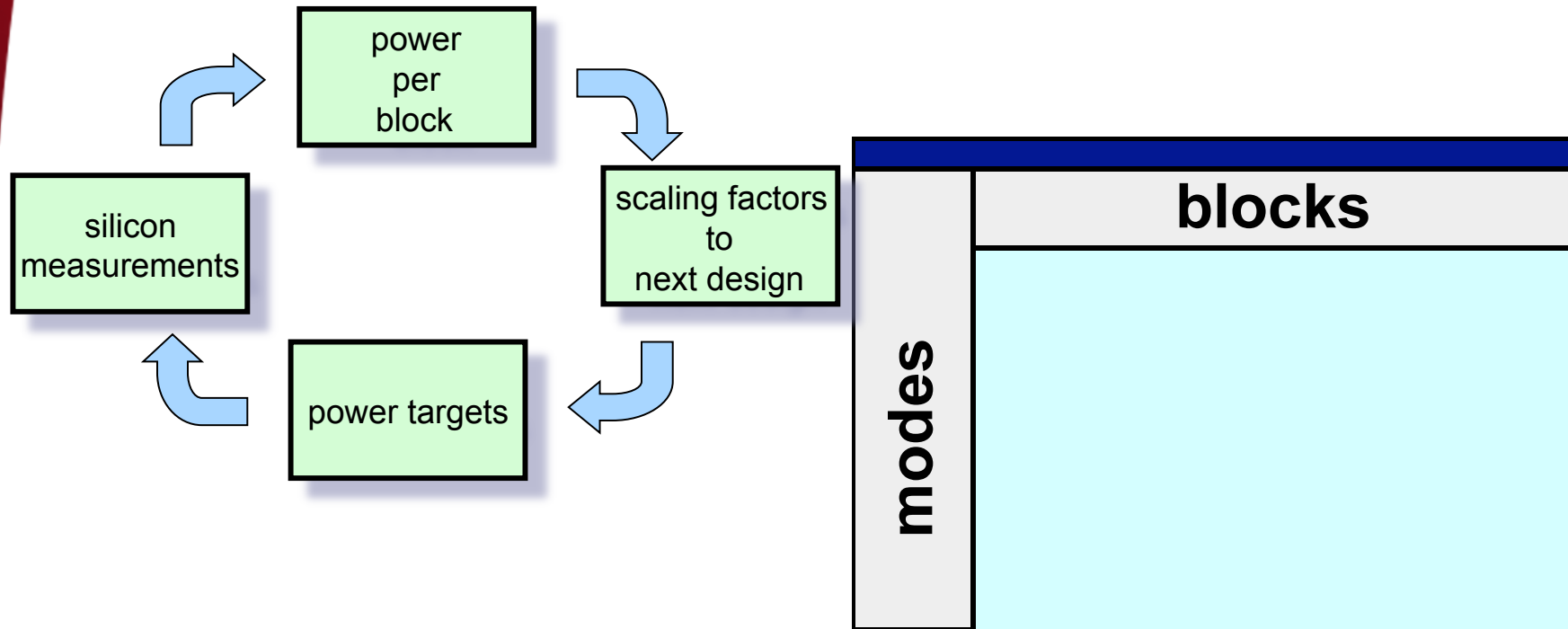
power estimation and reuse



power estimation and reuse



power estimation and reuse



common problems in creating this table:

- isolating each block's power per mode
- differentiating between leakage and dyn
- accounting for PVT effects
- ...

what's the big problem with power estimation?



- **measurement-based power estimation:**
 - depends on high reuse factor
 - use model has major impact (activity factor, temp,...)
 - need sims to get full value from measurements
- **simulation-based power estimation:**
 - easy to feed erroneous assumptions into sophisticated tools and produce garbage results
 - complex and time consuming to run real-world apps through gate-sims
 - need measurement to get full value from sims

bottom line: sims and meas need each other

what's the big problem with power optimization?



- Power is reduced in places where it doesn't count
- Reduction in one type of power increases another
- Design schedule and resources are consumed by implementation
- Technique doesn't produce expected power reduction
- Technique causes yield loss or functional failure
- Application software never uses low power modes
- Change in design or use model invalidates early analysis



thank you

It is All About Power Analysis, Exploration and Trade-offs

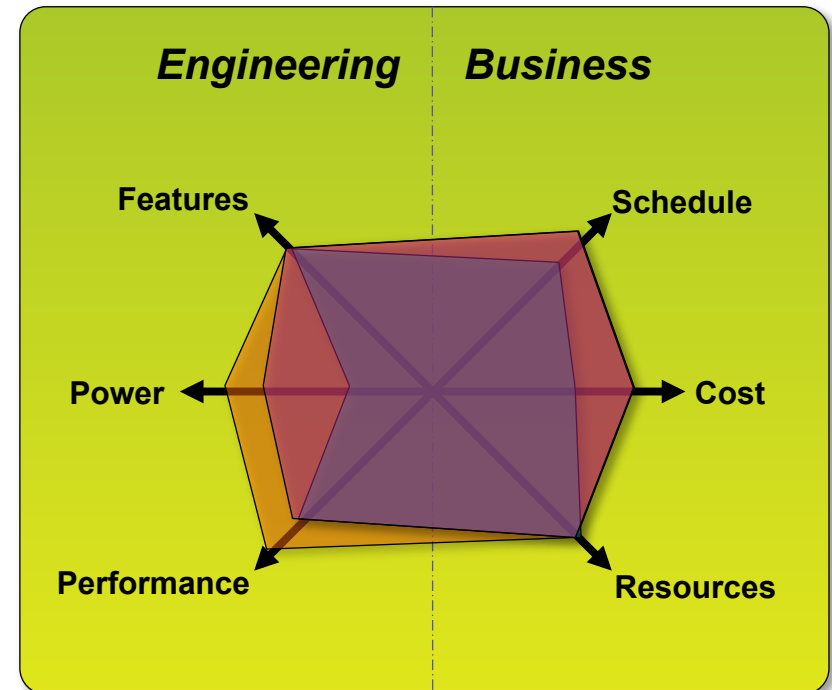
Ran Avinun ran@cadence.com
Marketing Group Director
System Design and Verification
Cadence Design Systems



Maximizing potential of power-aware design

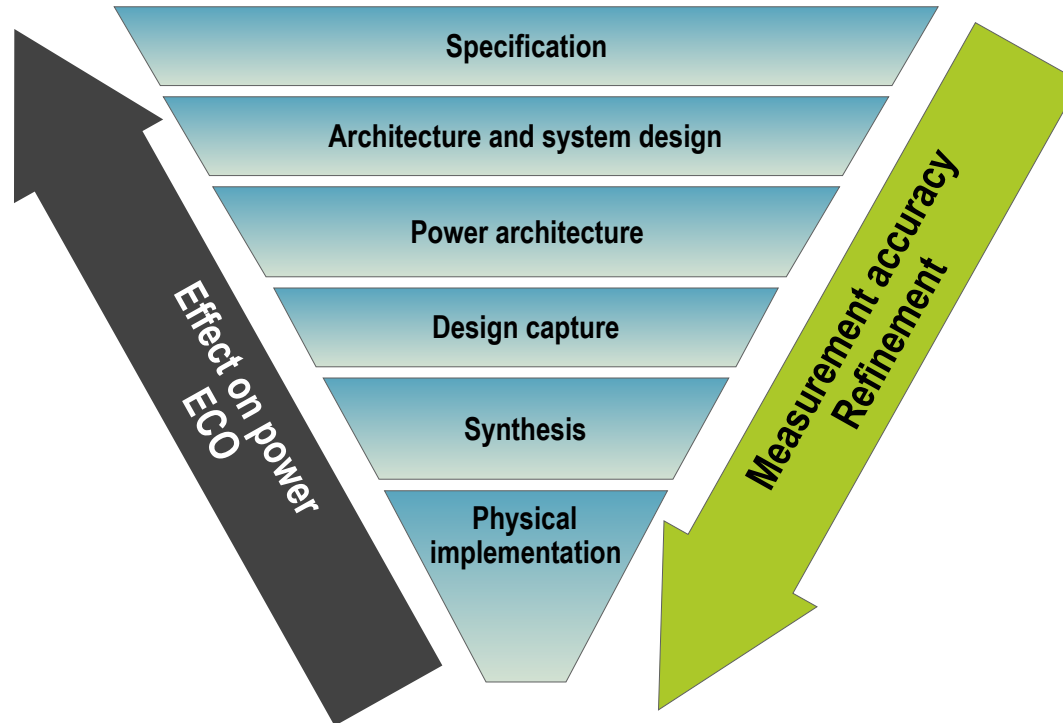
The need for true multi-objective optimization

- Power must be considered with other design goals
- Simultaneous multi-objective optimization is required
- Traditional sequential optimization approach will fail to converge



The need for power predictability

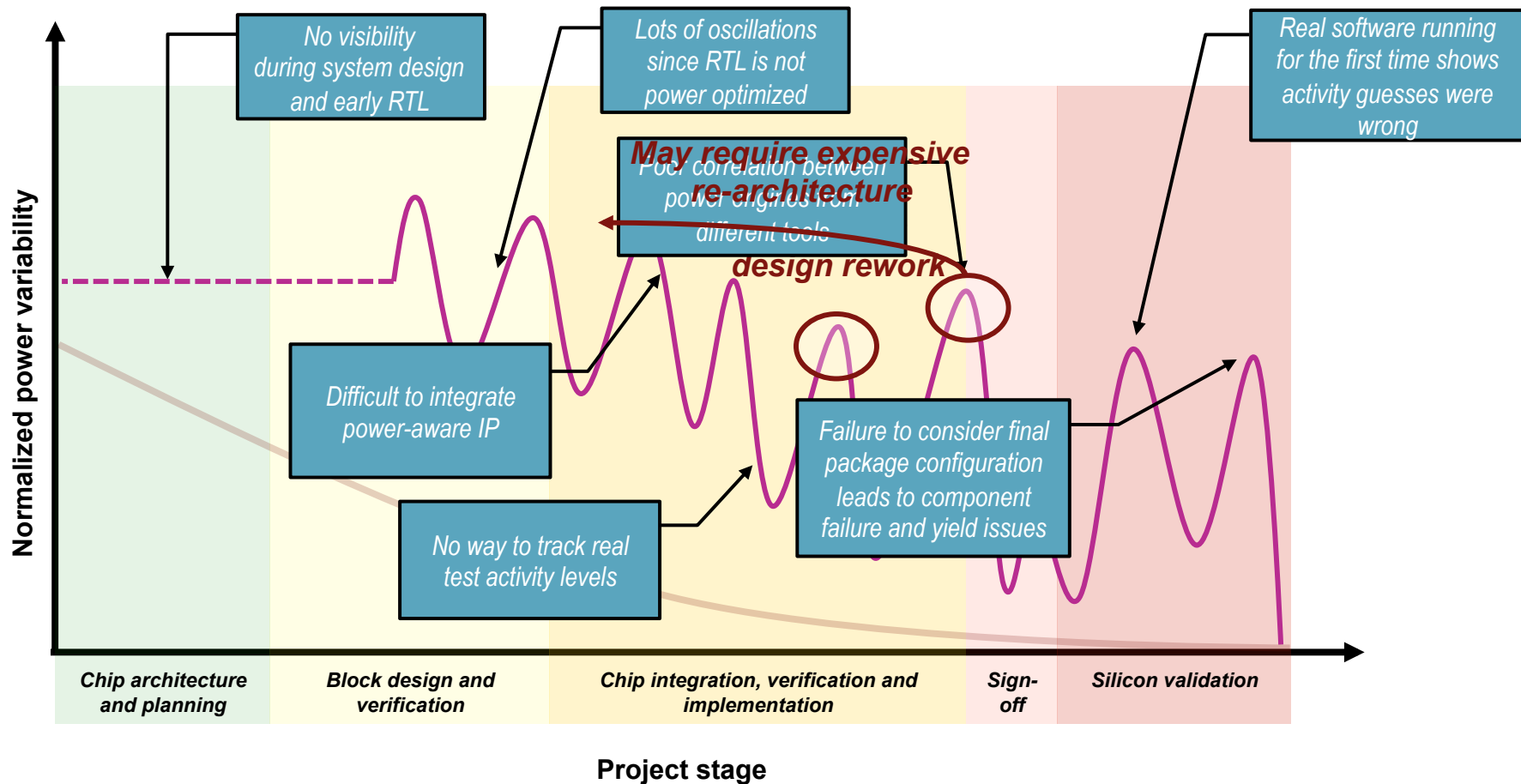
Providing you visibility into the unknown...



Technology is only adopted when risks and results can be predicted

Providing “power closure” for your designs

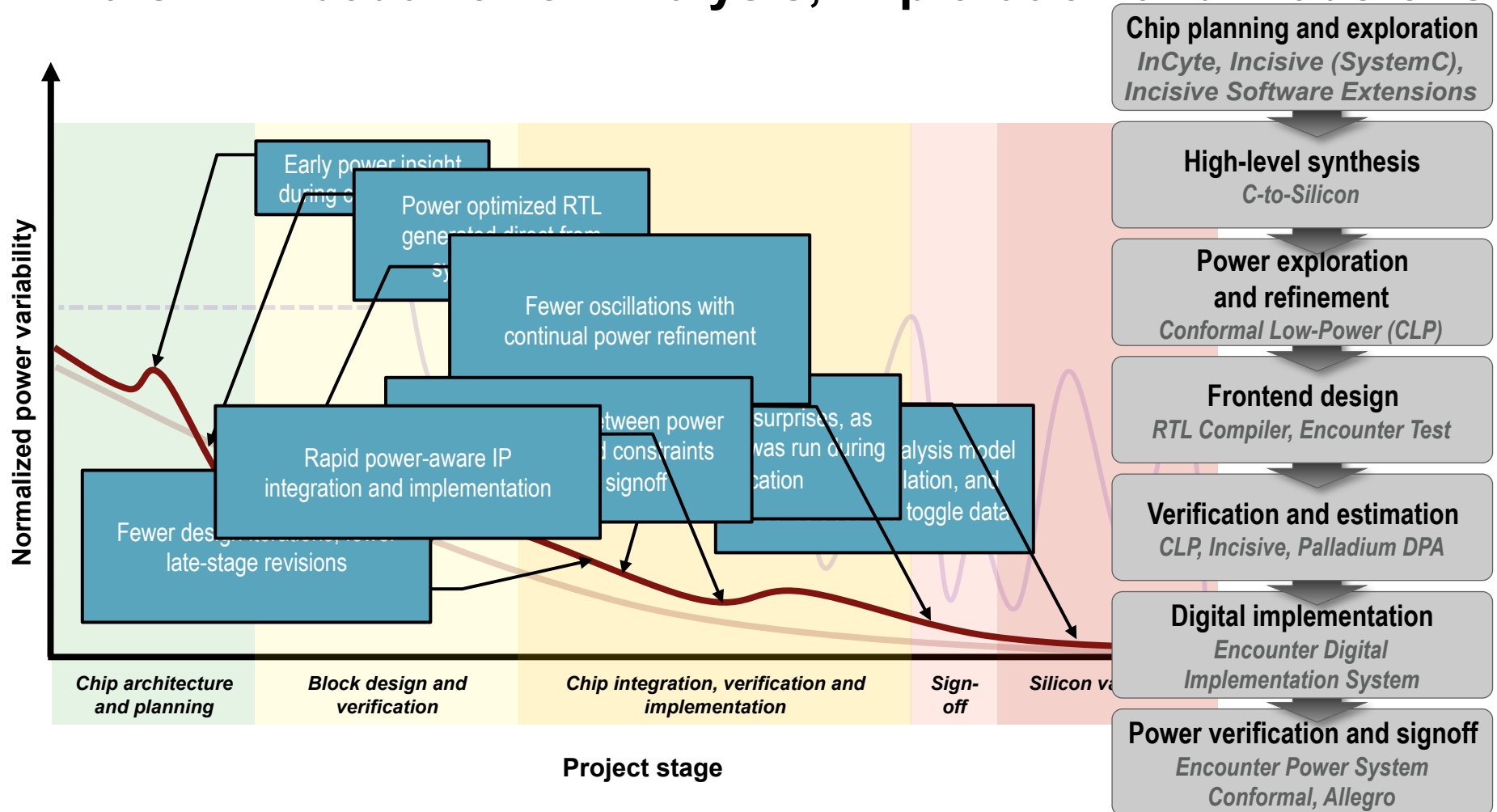
Comprehensive power-aware design solution



Providing “power closure” for your designs

Comprehensive power-aware design solution

It is All About Power Analysis, Exploration and Trade-offs





Power Methodologies

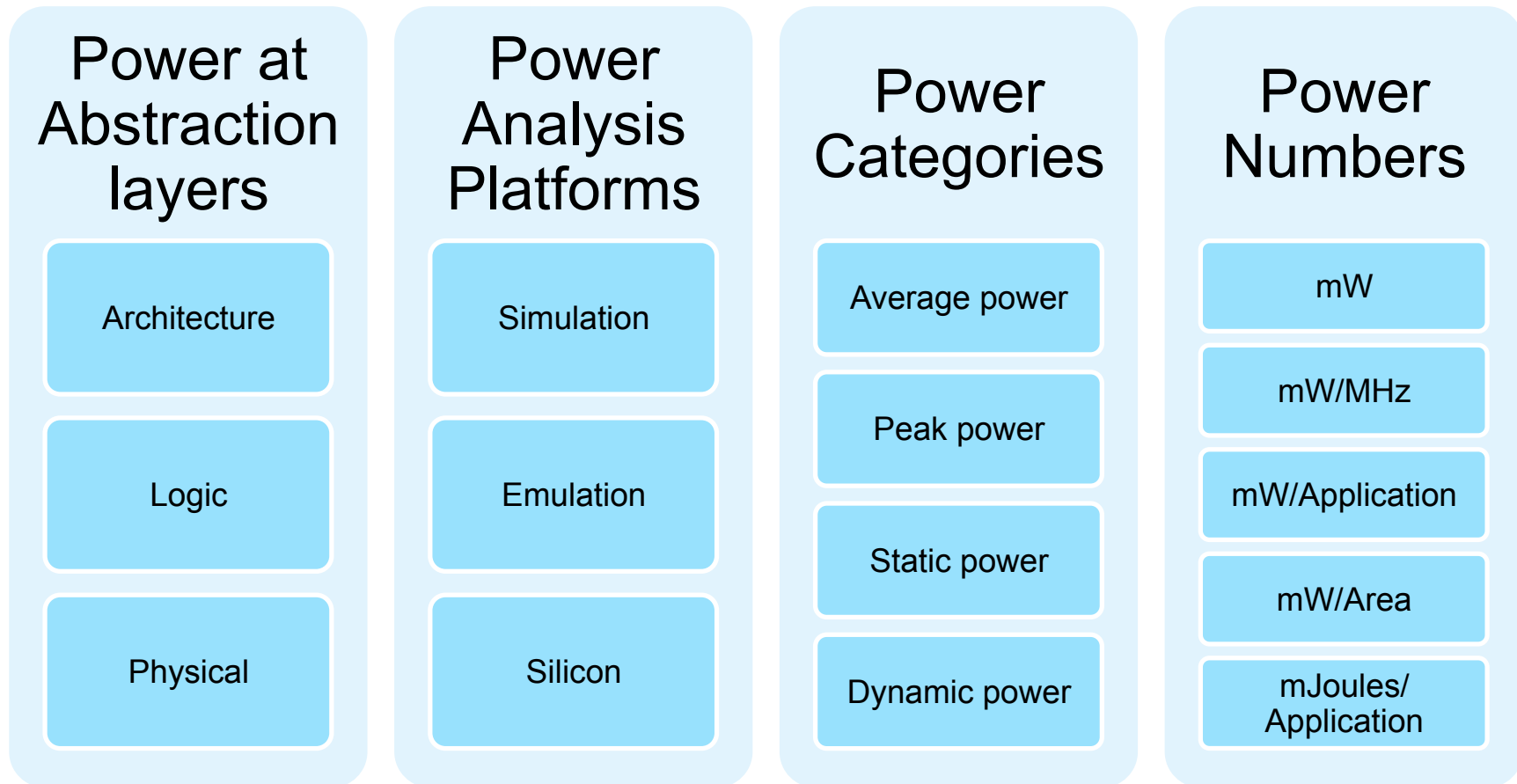
Sanjeev Das

Media Processing Center of Excellence, San Jose

20 August 2009

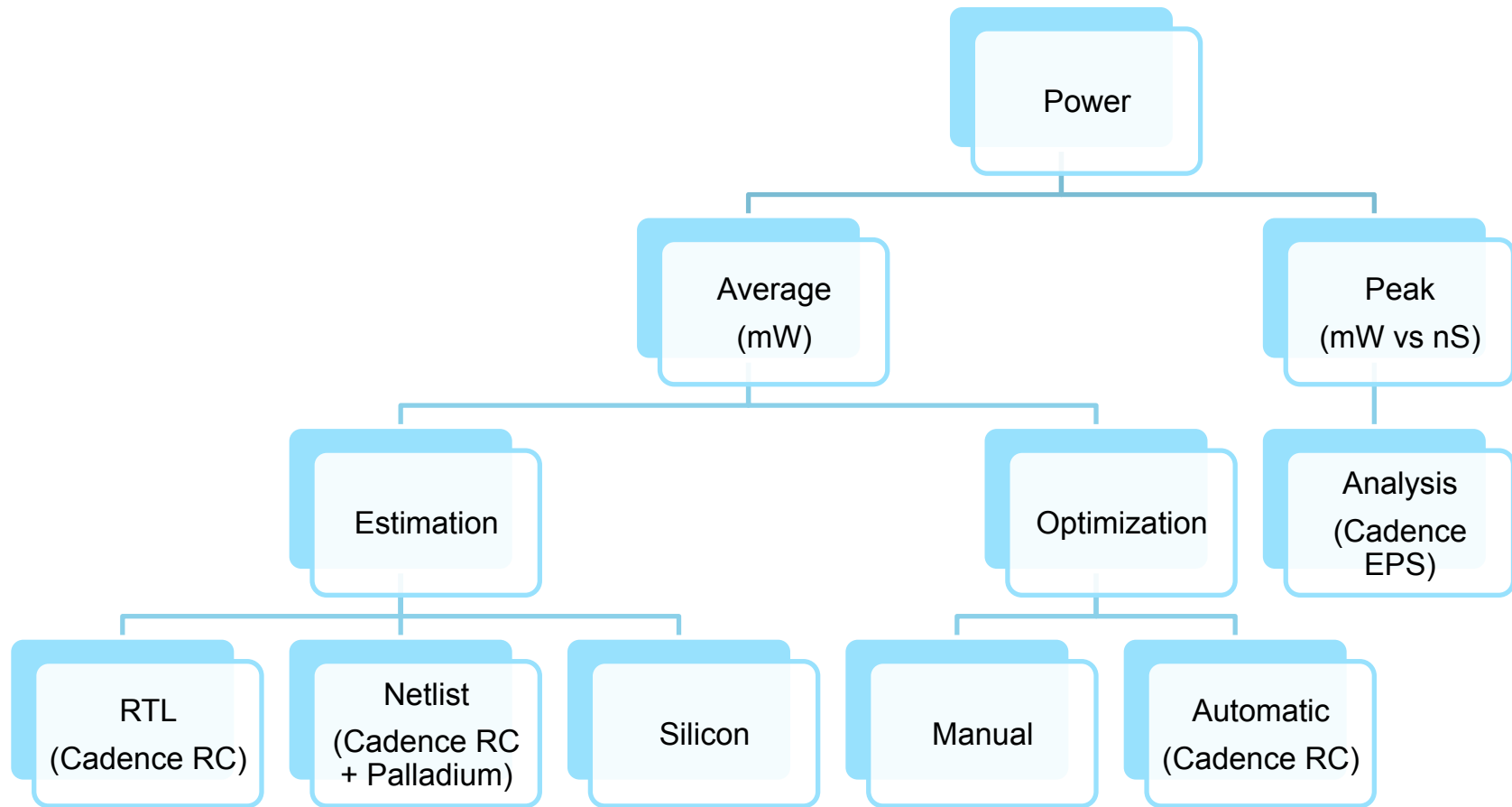


The Power Methodology Challenge!



TriMedia Processor Power Methodology Flow

(High-performance energy-efficient programmable media processor)



TriMedia Power Optimization Techniques

SW Architecture

- Performance-centric applications => Less MHz
- Less MHz load => Better mW/Application!

HW Architecture

- Sleep modes
- Better performance => lower mW/application
- DVFS
- Power savvy Instruction and Data cache

Logic

- Manual and automatic clock gating
- Manual and automatic logic gating

Physical

- Leakage optimization
- Clock gating saves area (mux replacement) and helps routing!

Summary of Viewpoints

Power methodologies addressed at all levels:
Software to Physical

mW/MHz is the biggest misindicator in the
power industry! It's the use-case mW or
mJoules that matter!!

No engineer or tool alone is smart enough to
do the most comprehensive power
optimization. We need both!!

Architectural Power and Performance Optimization

Jon McDonald

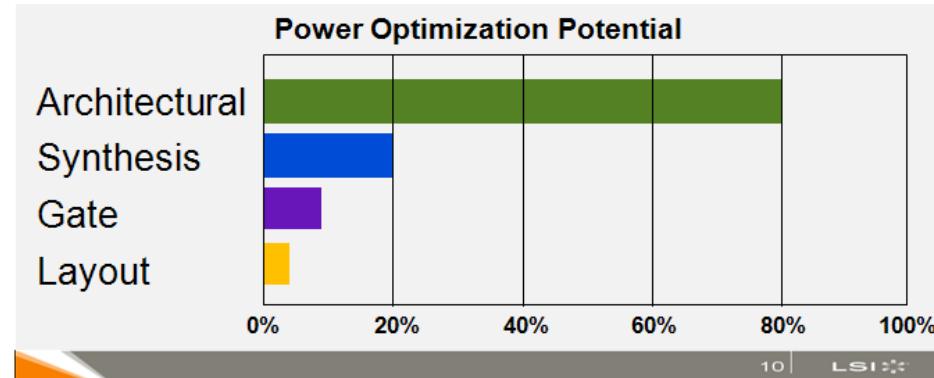
jon_mcdonald@mentor.com

**Mentor
Graphics®**

Why Optimize the Architecture for Power?

80% Power Savings at the Architectural Level by

- Optimizing architectural trade-offs between area/performance/power
- Addressing Hardware/Software architectural tradeoffs
- Tuning the application software
- Correlating power with system workload



Source: LSI Logic

Design and Optimization at the ESL Domain has the Biggest Impact on Power Consumption

System Architectural Power Optimization

- Optimizing **system architectural attributes** significantly impacts area, timing and power
 - **10X power variation** resulted from system architecture tradeoffs in the following example

Transmitter Design (IFFT Block)	Min. Freq. to Achieve Req. Rate	Average Power (mW)
Combinational	1.0 MHz	3.99
Pipelined	1.0 MHz	92
Folded (16 bfy4s)	1.0 MHz	~X10
Folded (8 bfy4s)	1.5 MHz	14.40
Folded (4 bfy4s)	3.0 MHz	21.10
Folded (2 bfy4s)	6.0 MHz	34.6

Source: Chip Design Magazine ESL Synthesis + Power Analysis By Holly Stump and George Harper

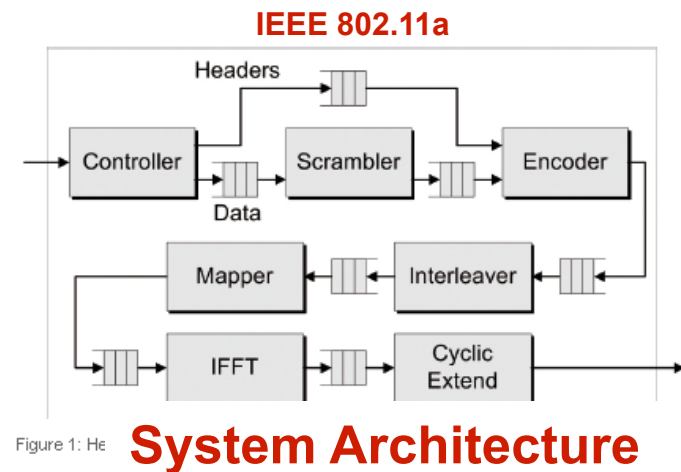


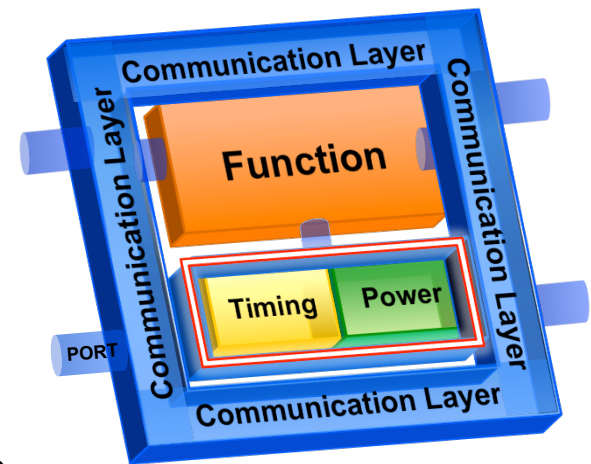
Figure 1: He

Only Transaction Level Platform Allows Users to Quickly Modify and Evaluate Various System Architectures

Modeling Power at the ESL

Accomplished using scalable TLM

- Modeling the core **Function**
- Providing the **Communication Layer**
- Adding a separate **Timing/Power** model

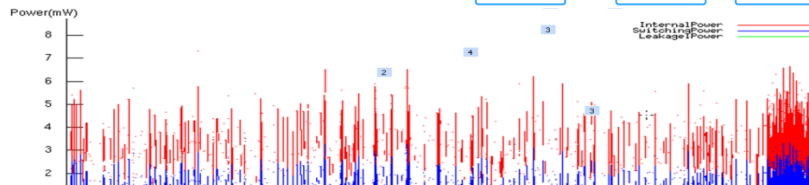
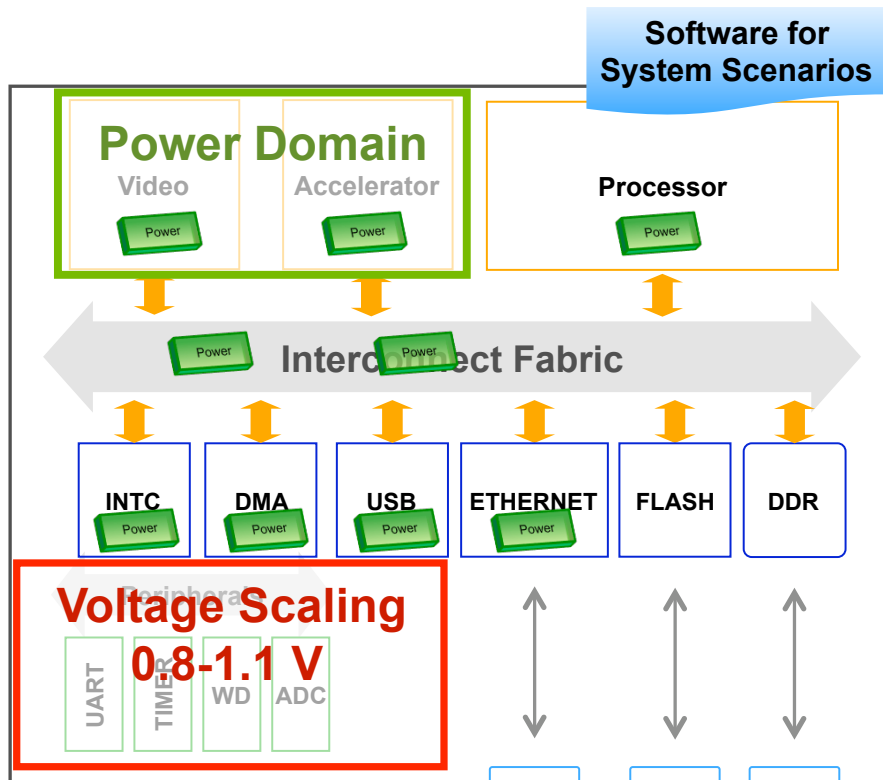


Models of all power types using policies

- Static (leakage) power
- Clock tree power
- Dynamic power (per transaction)

Power Policies		
Dynamic Power		
Transaction	Power	Time Interval
USB.READ	0.16mw	200
USB.WRITE	0.44mw	20
Clock Tree Power		
0.5mw		
Static (Leakage) Power		
0.2mw		

Architectural Power Optimization Philosophy



- ✓ Use estimates to model power of new IP
- ✓ Use accurate power models for legacy IP
- ✓ Analyze power profiles under
 - ✓ typical system scenarios
 - ✓ running software application
- ✓ Explore various power domain management strategies and voltage/frequency scaling techniques

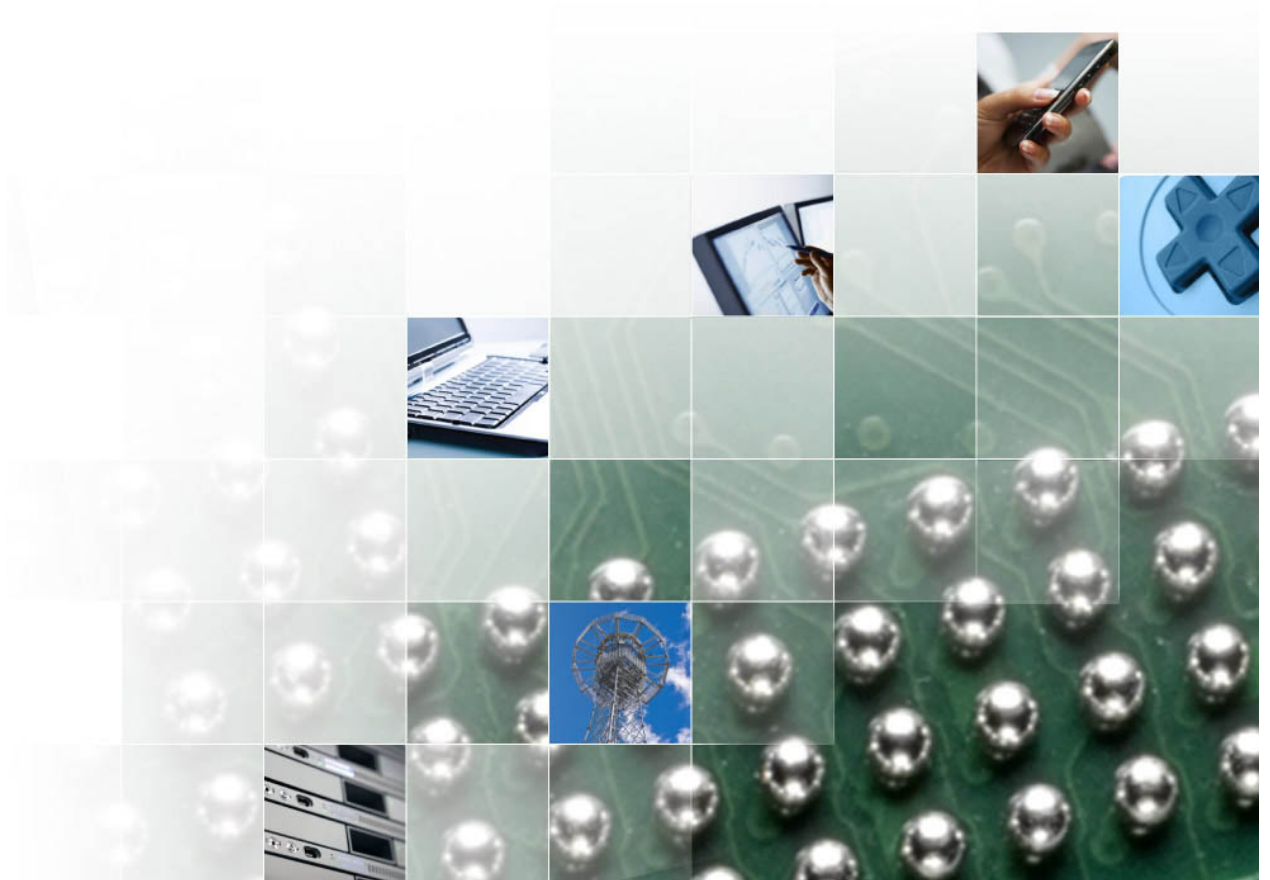
Only Transaction Level Platform Allows Users to Quickly Modify and Evaluate Various System Architectures



ISLPED Panel

Camille Kokozaki

August 20, 2009



- When exploring architectural tradeoffs, power can impact the choice of the solution adopted or even the functionality sought.
- In the early design stages, one of the missing elements for an accurate power analysis is the lack of top level verification capability to exercise the target design in a realistic manner.
- Once design architecture has been settled on, the mechanisms for power optimization are understood and techniques are in place to address them (clock gating, power aware clock tree synthesis, multi Vt, multi VDD, power shutoff).
- Designing for power reduction adds a new degree of complexity
 - the architecture has to comprehend and gracefully handle the entry and exit out of sleep modes or standby modes.
 - Adding circuitry to save power adds power if not properly done
- When it comes to budgeting power on re-used elements, hard IP is usually easier to account for in that soft IP may have interconnect and placement dependencies.
- Either way the application's use of the IP impacts dynamic power.

- Maximum power can be budgeted for but an accurate power estimate to say within 15% may not be possible
 - A switching activity estimate can be provided early on.
 - May not be enough, need a way to realistically exercise the design
 - If an unrealistic use scenario is analyzed for power, power could be under or overestimated
 - The accuracy is needed as design decisions may depend on this estimate
 - What package to use
 - What functionality goes in
 - Critical standby specs

- RTL changes to reduce or optimize power have the potential to impact power savings more dramatically
 - After an RTL has been converted to gates, the laws of physics will pretty much dictate the power number and the techniques to address are known
 - ESL to RTL inroads have been made to help in the creation of RTL optimized for power, area, or performance from a high level design abstraction.
 - Automatic and intelligent vector generation early in the design when no verification suite is ready can provide guidance on power
 - Shutting off the clock networks of logic functions during periods when logic functions are not producing any useful work can result in substantial power savings
 - Challenges remain in going beyond knowing early on what you relatively saved to calculating what you actually saved

- **Less Is More**
 - Less complexity: choose features wisely
 - Less computation: Smarter algorithms, activated on demand
 - Less redundancy through integration
- **More Is More:**
 - More realistic simulation scenarios
 - More power down intelligence
 - Longer realistic simulation sets
 - More context: Saving a little on an active big block may be better than a lot saved on an infrequently used small block
- **Better Early than Late:**
 - Early stage architecture level optimizations pay off big time
- **Better Late than Never:**
 - Optimize even if diminishing returns occur late in the design stage
- **Better Never than Always:**
 - No power management scheme if overhead adds to power negating the savings
 - If power sequencing is not understood, design may not function
- **Better (Weighted) Average than Peak:**
 - Peak could be intermittent



Q&A