

# Comprehensive Metrics-Based Methodology to Achieve Low-Power System-on-Chips

Ellie Burns – Director of Marketing for Calypto Systems Division

Gabriel Chidolue – Verification Technologist

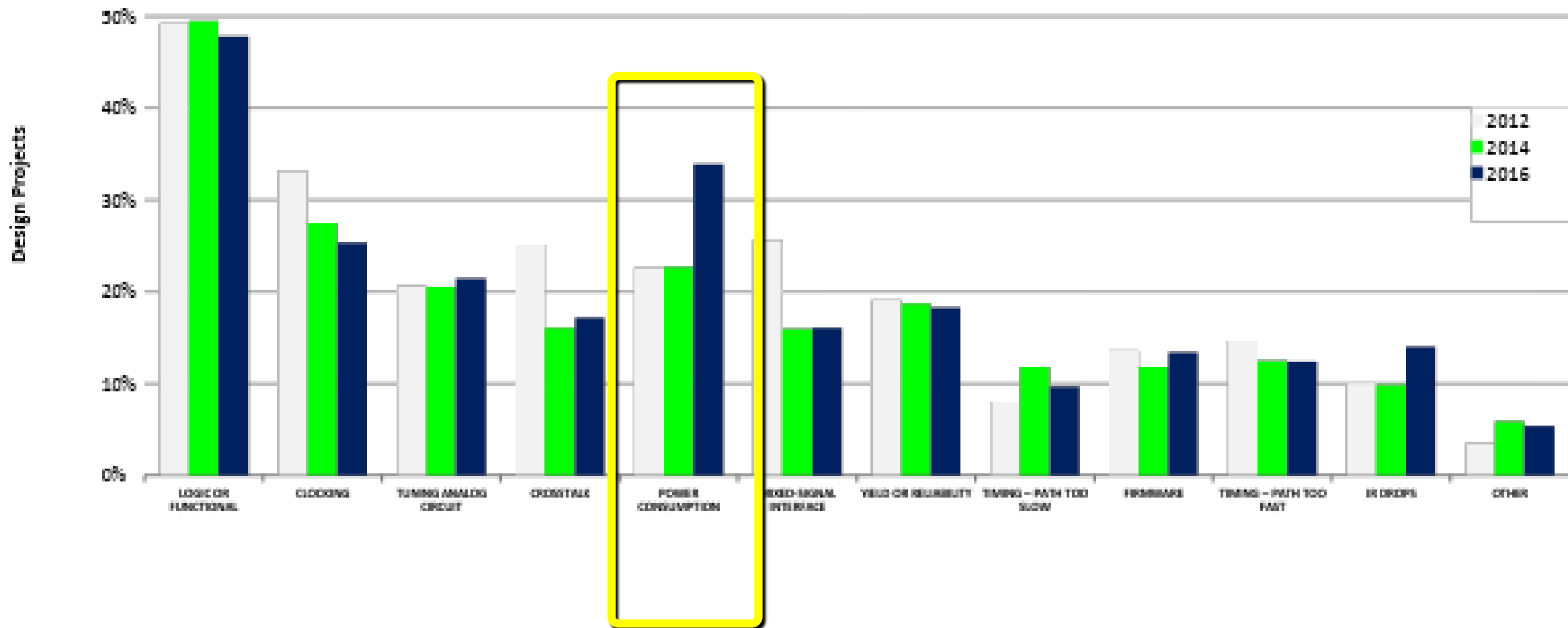
Guillaume Boillet – Power Product Specialist

**Mentor**®

A Siemens Business

# Welcome and Introduction

# Power is Now the #2 Driver of Respins!



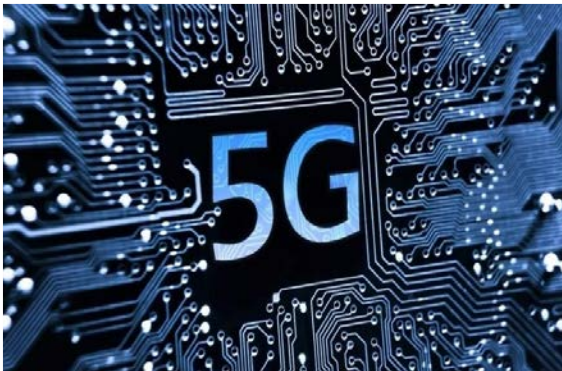
Trends in Types of Flaws Resulting in Respins

\* Multiple answers possible

Source: Wilson Research Group and Mentor Graphics, 2016 Functional Verification Study

# Markets Like 5G, Smart IoT, Automotive Low-Power Critical

- 5G both ultra low-power and high-performance mobile



- IoT expanding digital content and changing landscape

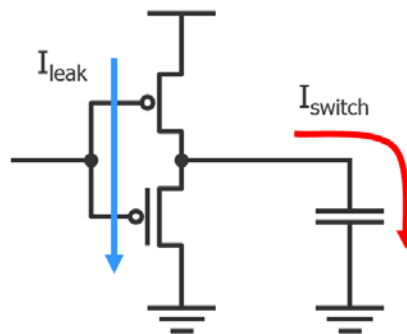
- Path to Autonomous vehicles requires huge computations and power



# Dimensions of Low-Power Design

## Leakage Power

Power consumed by current flow through transistors even when turned off

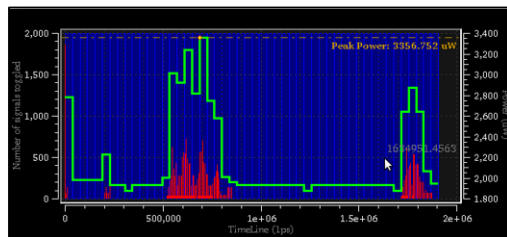


## Dynamic Power

Power dissipated when circuit is active, charging and discharging the loads

## Power Estimation (RTL/Gate)

Measuring power consumption of a design. Typically used to know if the design is within specified power budget

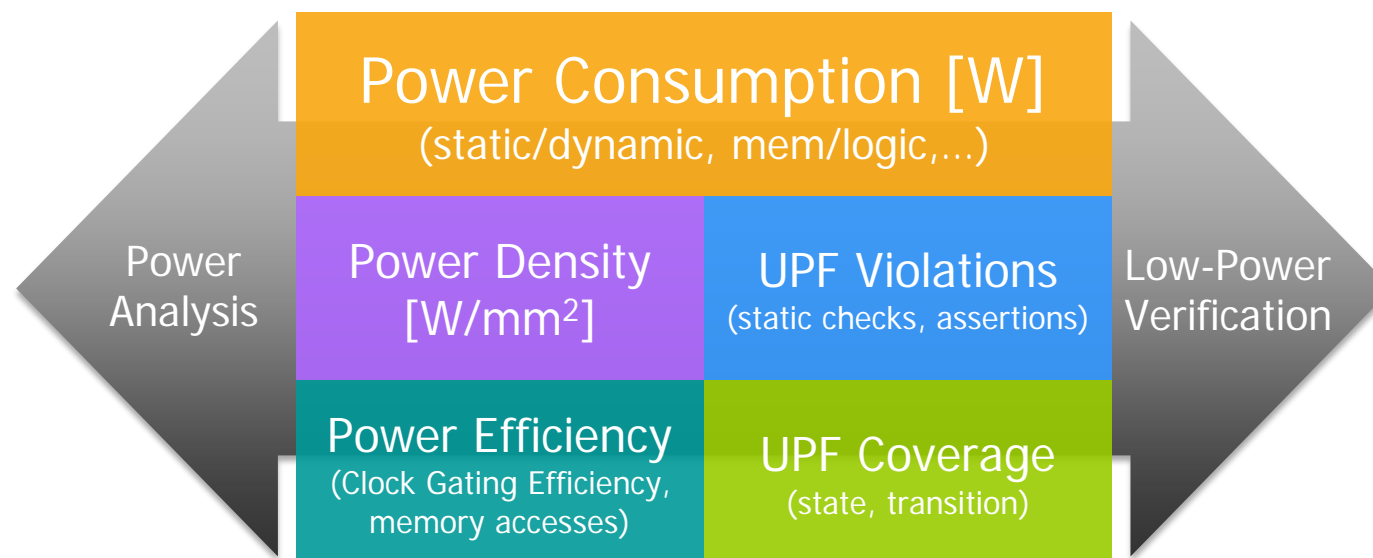


## Power Reduction

Process of optimizing a design to reduce its power consumption. Necessary when the design is over budget or when competing on power



# Important Metrics for Successful Low-Power Design and Verification



- There are many kinds of metrics for power that should be measured at various times in the design cycle

# Lifecycle of Low-Power Design and Verification

Early RTL Power Metrics with no vectors

RTL/GL Metrics with vectors and power vectors

Power Management coverage metrics

SoC Power Management coverage metrics

Power Metrics for full SoC with real power scenarios

Measure SoC power and benchmark/SoC scenarios

Design/Test/Measure Power management

Measure power block/sub-system tests scenarios

RTL Designers Use low-power design techniques

Top Level

Sub-System

Block Level

RTL Design and Verification Timeline

# Tutorial Agenda

- Introduction
- Introduction of Low-Power Coverage & Debug metrics in simulation
- Update on UPF 3.0 and 3.1
- Break
- Analysis and Reduction: Metrics for Designing low-power IP
- Emulation-Based Power Analysis and Verification
- Wrap up



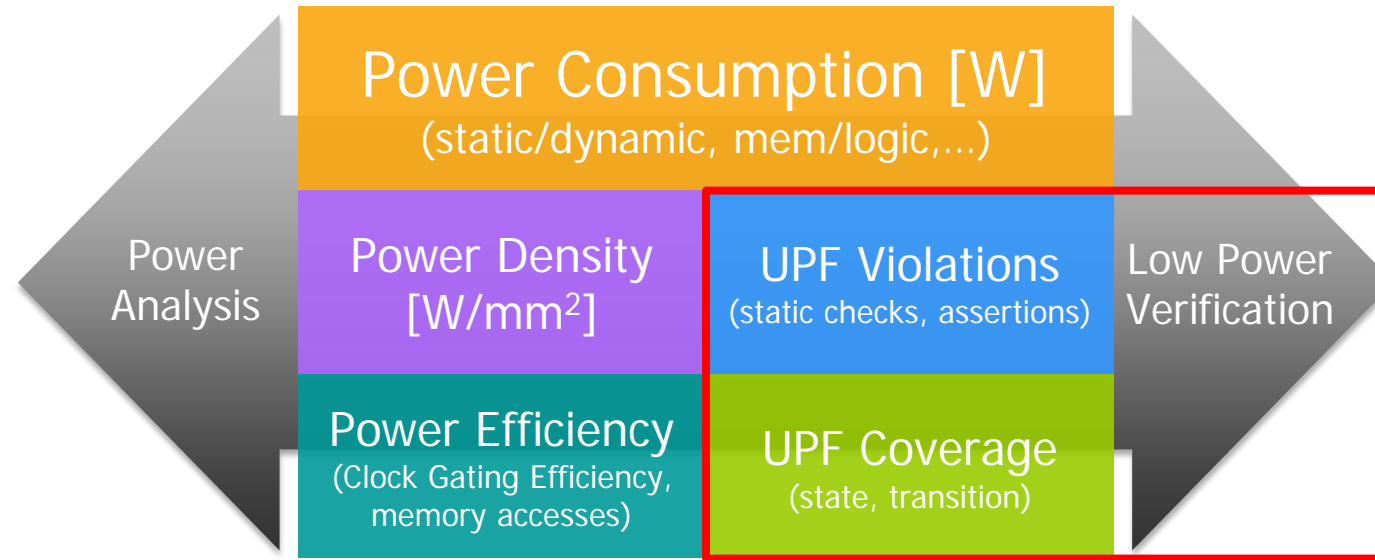
# Low-Power Verification Metrics (UPF Power Architecture)

Gabriel Chidolue, Verification Technologist



A Siemens Business

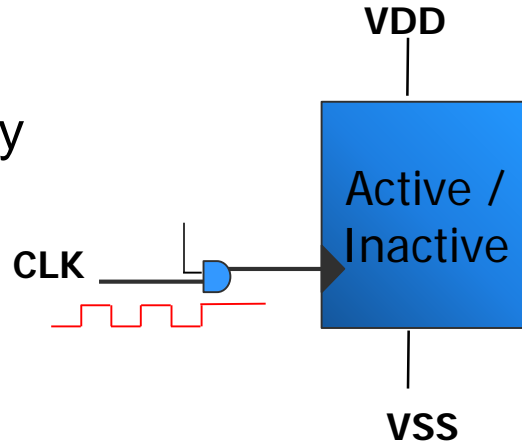
# Metrics for UPF Based Designs



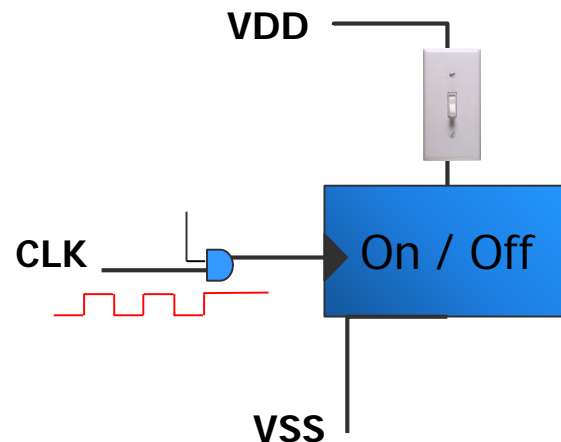
- Leakage power management requires active power reduction techniques

# Power Reduction Techniques

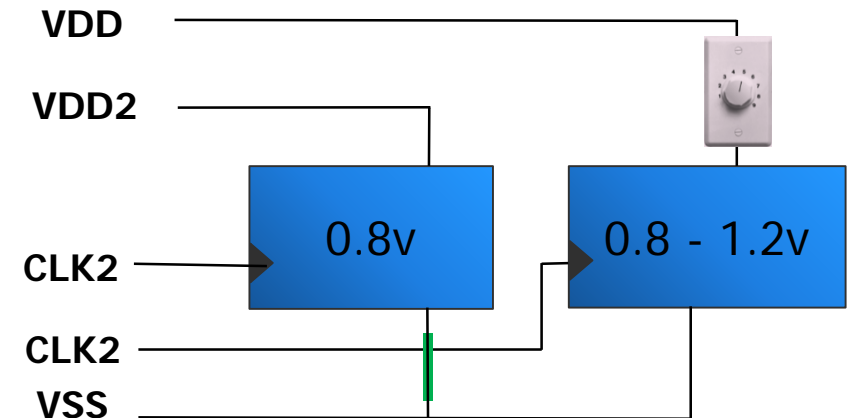
- Clock Gating
  - Avoid unnecessary switching activity



- Power Gating
  - Shut off when logic is not in use

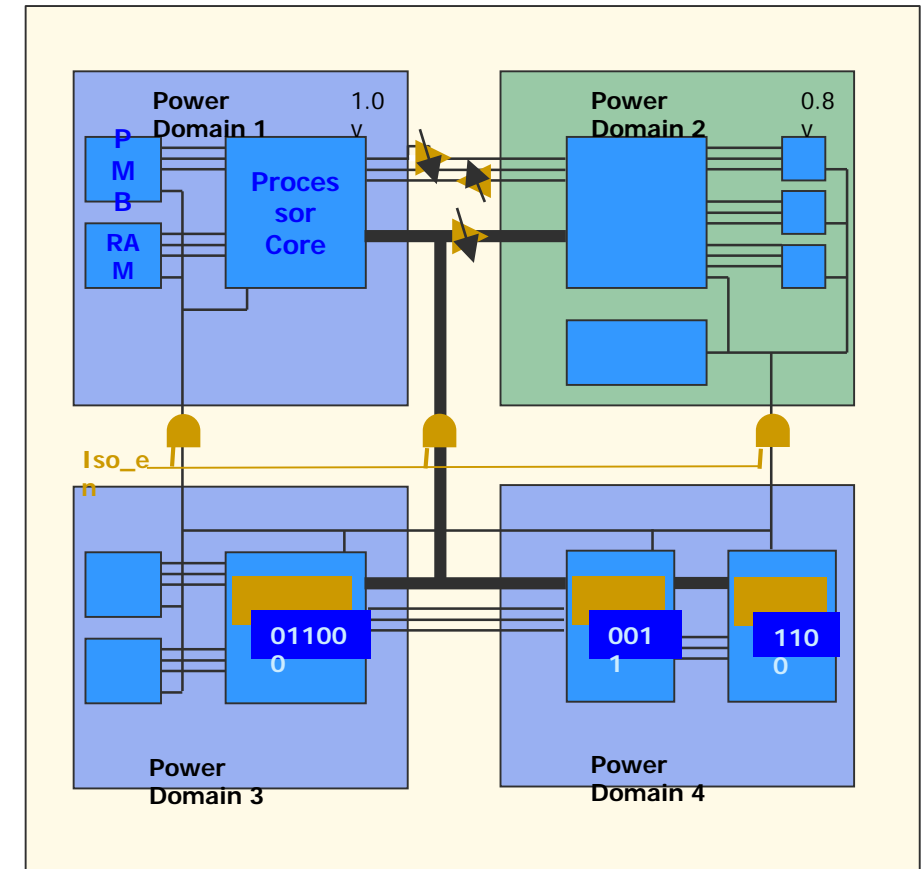


- Various Multi-Voltage Techniques
  - Optimize power used for required performance
  - Dynamically adjust voltage/frequency
  - Non-operational back biasing to reduce leakage



# Power Management Concepts

- Power Domains
  - Independently powered regions
  - Apply different power reduction technique in each region
- State Retention
  - Save essential data when power is off
  - To enable quick resumption after power up
- Isolation
  - To ensure correct electrical and logical interactions between domains in different power states
- Level shifting
  - To ensure correct communication between domains operating at different voltage levels



# Introducing IEEE1801 (Unified Power Format)

- **IEEE 1801 UPF**

- Standard format for defining power management
- Extension of “Tcl” language
- Defined separately from the HDL
- Enables early verification of power intent
- Drives verification and implementation from RTL to Layout
- Contains commands for power intent and driving testbenches

- **An Evolving Standard**

- Accellera UPF in 2007 (1.0)
- IEEE 1801-2009 UPF (2.0)
- IEEE 1801-2013 UPF (2.1)
- IEEE 1801a-2014 UPF (2.2)
- IEEE 1801-2015 UPF (3.0)

- **For Power Intent**

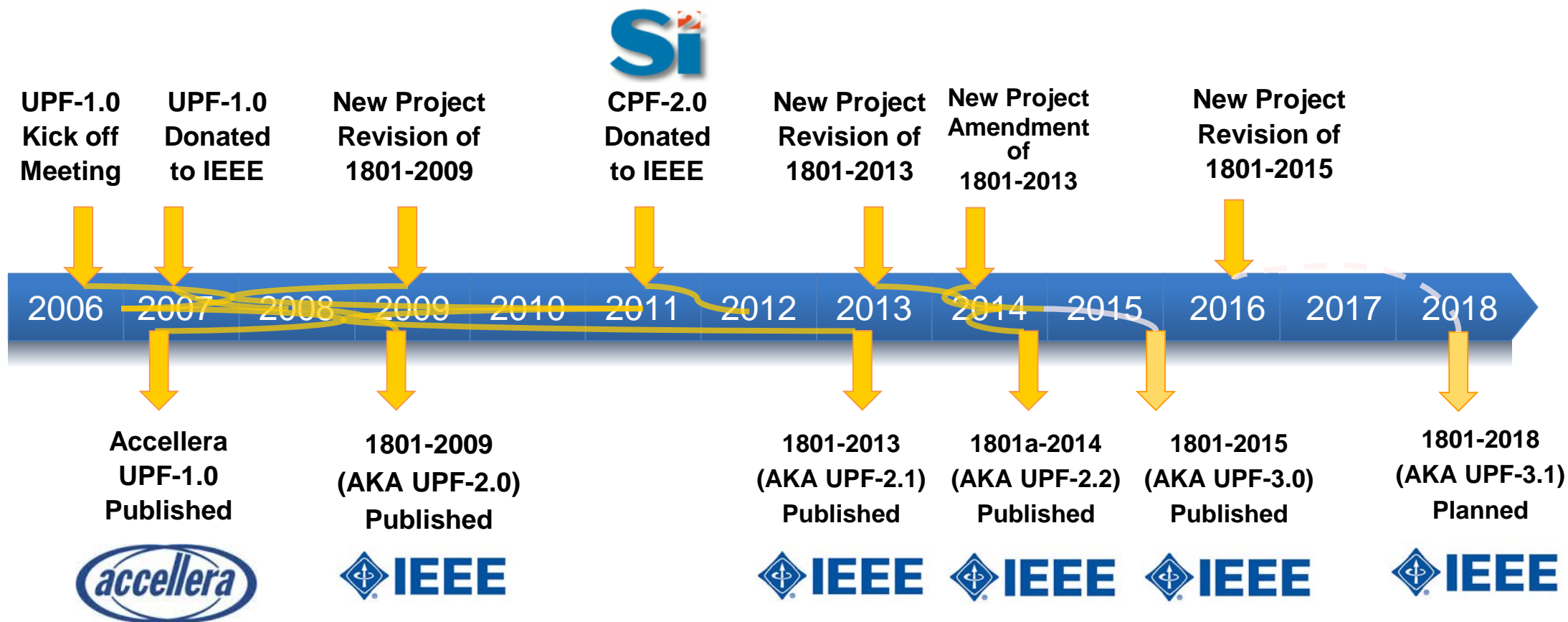
- To define power management
- To optimize power consumption

- **For Power Analysis (in 3.0)**

- Component Power Modeling

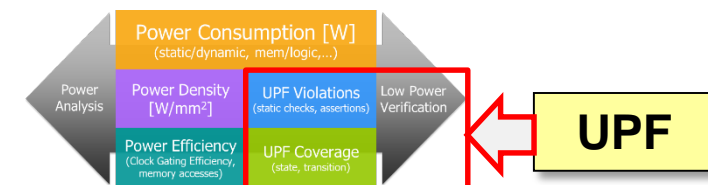


# IEEE1801 is an Evolving Standard

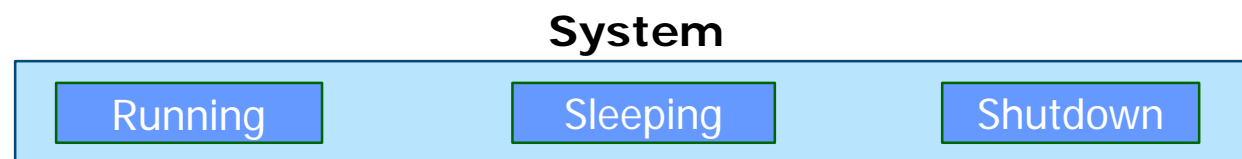


# Power State Definition

## UPF 3.0



- A key part of the power architecture specification of a design
- Representing
  - Operational Modes of IP within system as well as System
  - Ability for Supply states of supply\_sets
- Specified on Power domains, supply sets, groups and other objects
- Specified using `add_power_state` command
  - `-logic_expr`: functional mode (control signals, power states)
    - `Interval()`: clock frequency
  - `-supply_expr`: supply set function state and voltage and simstate
- With state space controlled using
  - `-complete` to indicate that specified states are the only legal power states
    - All unspecified power states are illegal
  - `-illegal` | `-legal` to mark specific states as illegal states
    - Default is Legal

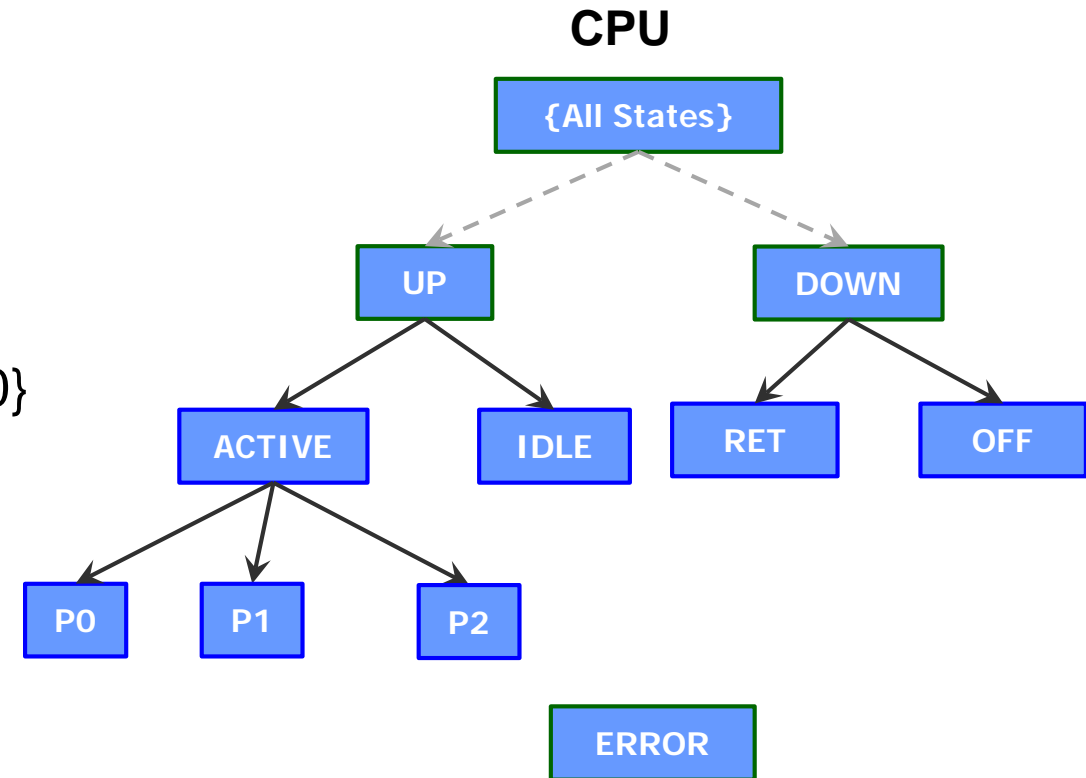


```
add_power_state -domain System \
  -state {Running -logic_expr {...}}
  -state {Sleeping -logic_expr {...}}
  -state {Shutdown -logic_expr {...}}
```

`-logic_expr` defines condition in which object enters the state

# Abstract/Refined Power States

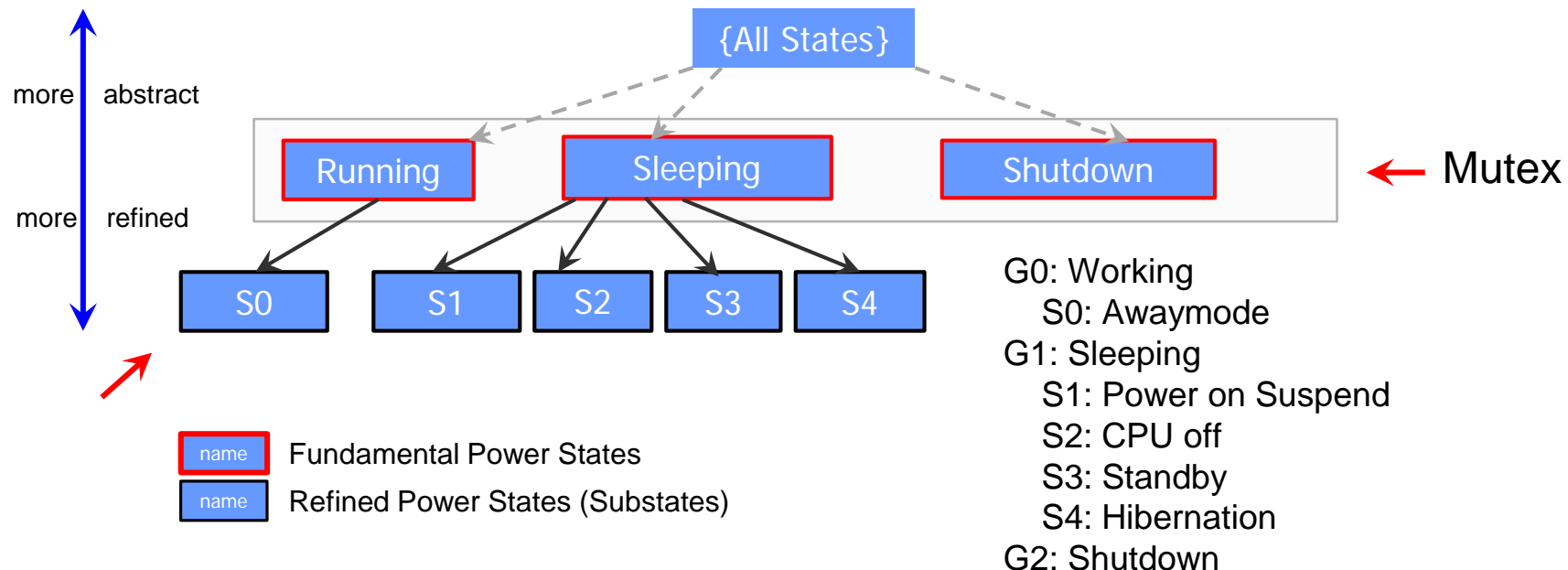
- Create a more refined state of an object by referring to another state of the **same object**
  - UP : -logic\_expr {pwron}  
is a fundamental state
  - UP.ACTIVE : {CPU == **UP** && running}  
is a refinement of UP
  - UP.ACTIVE.P0 : {CPU == **ACTIVE** && perflvl==2'b0}  
is a refinement of UP.ACTIVE
- The “{All States}” state shown represents all possible states of the CPU





# Mutual Exclusivity of Power States

- States at same level of refinement are mutually exclusive
- Abstract states contain (overlap) sub-states – non Mutex
- Error if more than one mutually exclusive state becomes active



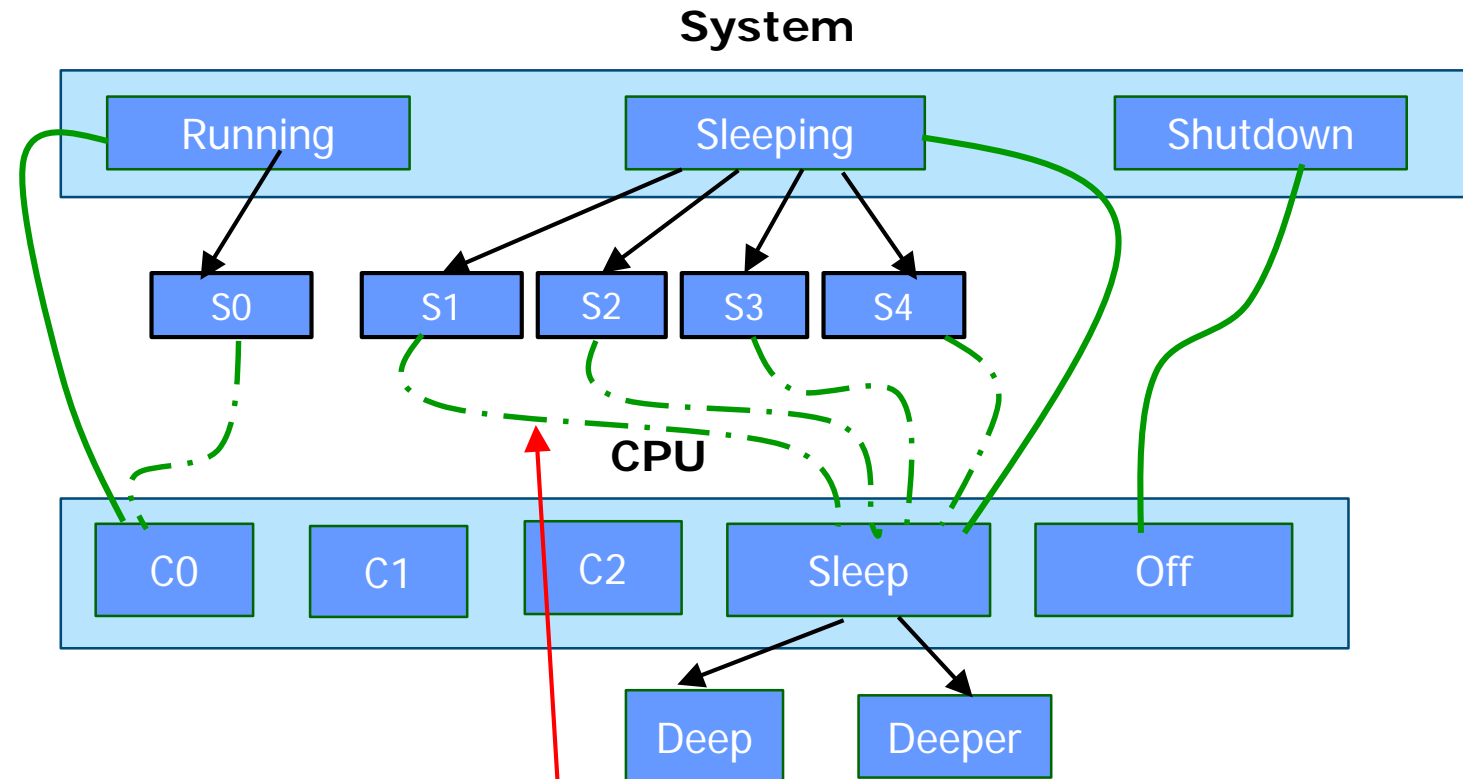
# Hierarchical Composition of Power states

## System States

```
add_power_state system
-state {Running -logic_expr {CPU == C0}}
```

```
add_power_state system
-state {Sleeping -logic_expr {CPU == Sleep}}
```

```
add_power_state system
-state {Shutdown -logic_expr {CPU == Off}}
```

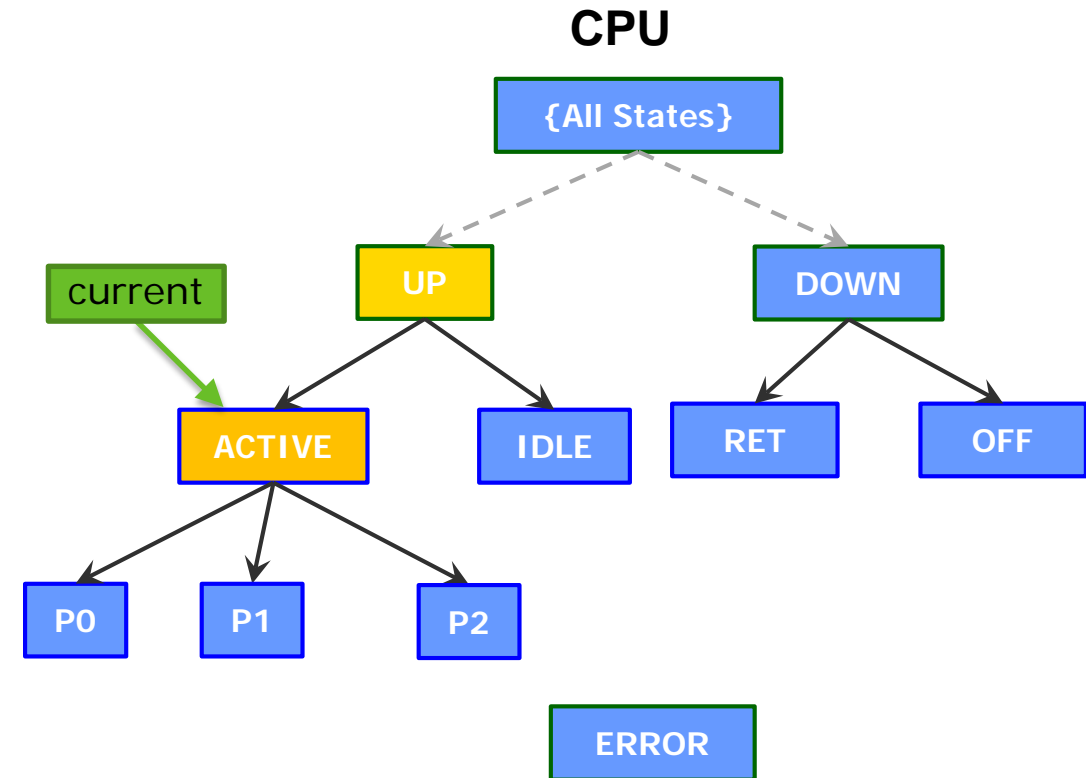


System Power States are defined in terms of the states of the dependent Power domain states.

A Substate inherits dependencies of the State it refines

# Current vs Active State Semantics

- Active vs Current State semantics
  - A state is *active* if its logic expression is satisfied
  - A state is the *current* state if it is the most refined *active* state
    - ACTIVE and UP are *active* states
    - ACTIVE is the *current* state (since it is the most refined state that is activated)
- Predefined state ERROR is activated if 2 or more mutually exclusive states are activated at the same time



# Power State Transitions

- Defined using
  - `add_state_transition` command
- States can be group and/or paired to aid in specification
- Identify legal and illegal transitions between power states using
  - `-illegal`, `-legal(default)`
- Control the possible transition space using
  - `-complete`
- Essential for transition coverage

```
add_state_transition -domain System \  
-from Sleeping -to {Running Shutdown}  
  
add_state_transition -domain System \  
-from Running -to Sleeping  
  
add_state_transition -domain System \  
-from Running -to Shutdown -illegal
```

# UPF Power State Definition in Verification

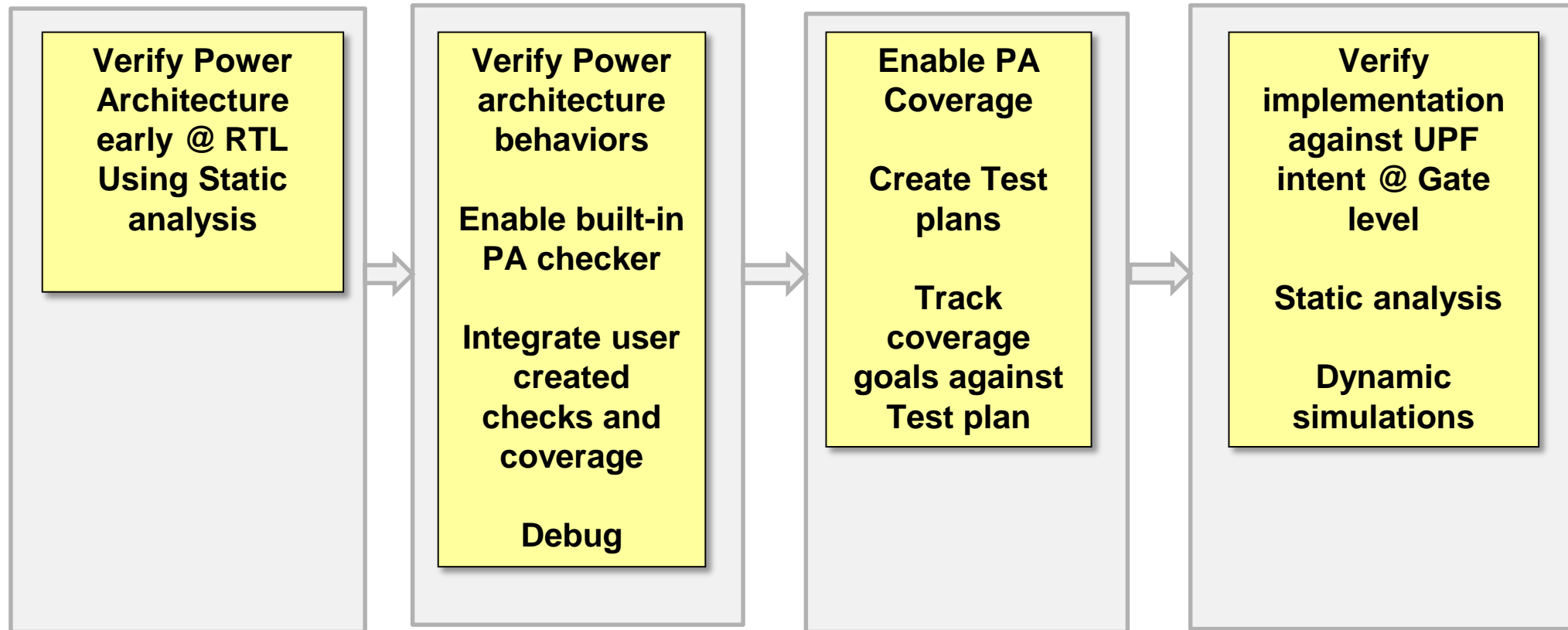
- Enables Static analysis tools
  - To determine isolation and level shifting requirements at Power Domain and Macro boundaries
    - Aided by hierarchically composed power states in UPF
- Enables Dynamic verification tools
  - To create test plans and
  - To create comprehensive coverage models of all legal system and subsystem power states and transitions
    - Aided by all power state specifications in UPF
  - Helps answer the question:
    - Have we verified correct operation of all legal power states of the system ?
    - Have we verified correct operation while system transitions between all legal power state transition combinations?

- 
- The diagram illustrates a design flow for power management. At the top, two boxes labeled 'UPF' (orange) and 'HDL' (green) are connected by a plus sign. A large blue arrow points down from this combination to a central orange cloud shape labeled 'Design + Power management'. Below the cloud, another large blue arrow points down to a blue cylinder labeled 'Information Model'. To the left of the cylinder is a blue rounded rectangle labeled 'API'. Two horizontal blue arrows point from the 'API' box to the 'Information Model' cylinder.

# What's Coming in UPF 3.1

- New simulation control commands
  - Suspend non-PA user assertions dynamically at power down
- Ease of use enhancements to Power states
  - Create power state groups of supply sets
  - Use named states on supply objects created using `add_supply_set` in `-supply_expr` of `add_power_state`

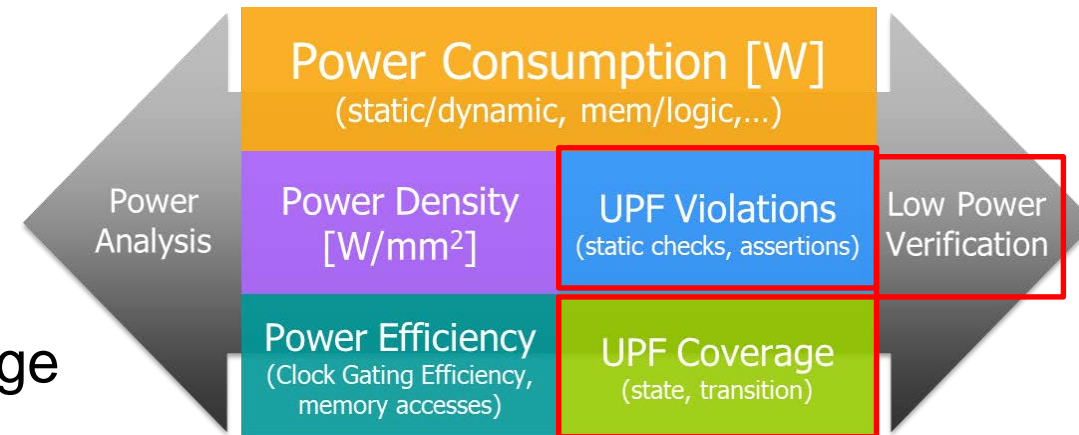
# Power Architecture Verification Flow



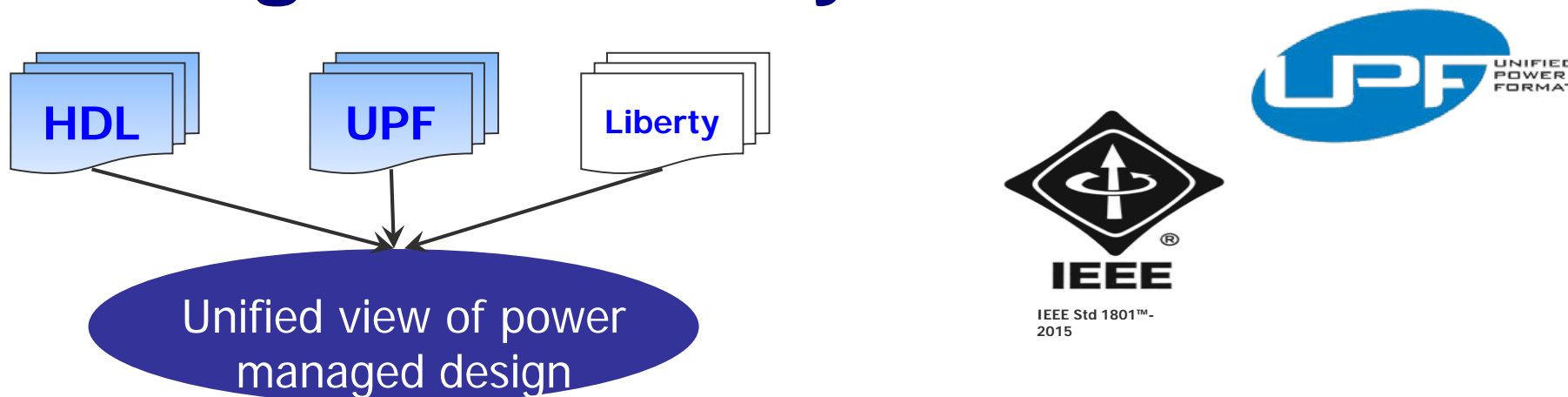


# Verification Strategy for Power Managed Designs

- Static Analysis\Checking
  - Power Architecture Consistency at RTL
  - Structural Checks and implementation consistency checks post synthesis
- Automated Dynamic Power Aware checkers
  - To find common power control sequence protocol bugs
- User Created Power aware assertions and coverage
  - Build power aware coverage and checkers using UPF Information Model and TCL and HDL API functions
  - Bind to DUT (UPF + RTL) using UPF bind\_checker
- Leverage Unified Power Aware Debug Environment
  - Visibility
  - Root cause Analysis
- Verification Completeness Metrics using Power Aware (PA) Coverage



# Verify Power Management Architecture Using Static Analysis



- Static Check Analysis with Questa PA Sim
  - No Testbench required
  - RTL and Gate Level HDL support
  - Global analysis of all Power States
- Extended Static Checking Questa CDC (Power Aware)

# Static Checking

## *Essential Checks at Different Design Abstraction Levels*

### RTL Checks

- Power architecture checks
  - ELS/LS /ISO are inserted where required
  - NO missing, redundant or back to back ELS/LS/ISO cells
- Information from add\_power\_state is checked against strategies in UPF during this checking
  - UPF (1.0) PST also supported
- Does not require test vectors

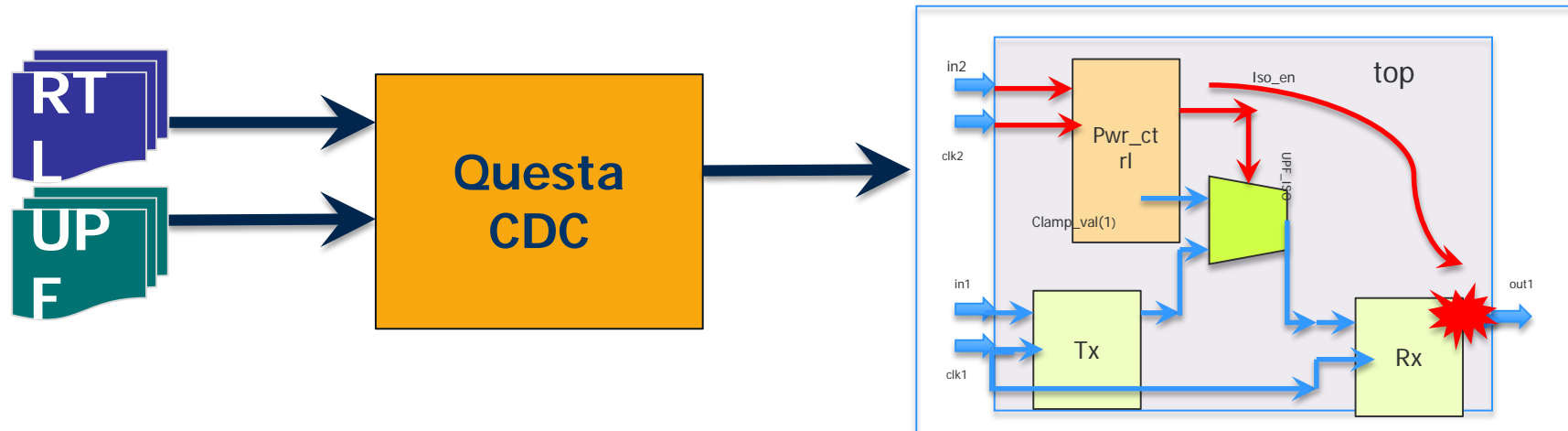
### Gate Level Checks

- Micro architectural checks
  - Ensure control signals are driven from relatively always\_on domain
    - Control signals are determined from UPF strategies and switch specifications
- Implementation checks
  - Check implemented ISO/LS/ELS/RFF/PSW and AON buffer cells against UPF specifications (Strategies and power states)
    - Requires Liberty
    - E.g Correct polarity of clamp value, wrong cell type used , no back to back cells, etc

# Extended Static Checks

*with Power Aware CDC*

- Power control signals unconnected in RTL
- Unified Power Format (UPF) defines power intent
- Power management structures from UPF could introduce CDC paths
- Verify these at the RTL level



# Verify Dynamic Power Aware Behaviors

- Verify power up/down Sequencing for each block/power Domain
  - nonoperation biasing, save operation on power down
  - reset/restore on power up
- Verify that interfaces are correctly isolated and level shifted
- Verify that appropriate retention protocols are followed
  - 0/1 pin retention FF (master slave alive style retention)
  - 1/2 pin edge sensitive Ballon latch style Retention FF
  - Non retention FFs may require resetting on power up
- Leverage Questa PA Dynamic checkers (assertions)
  - To check for common power management protocol violations
- Verify that all power states and transitions are covered
  - Enable automatic PA coverage collection

# Visualize UPF objects in Design Hierarchy

## Synchronized with Source

Visualizer - DEV-main 2968275 <2>

File Edit View Source Tools Window Help

Design

interleave\_tester.interleave1

interleave\_tester : (interleave\_tester)

- interleave1 : (interleave1, PD\_intl)
  - bypass\_ram\_UPF\_ISO : (mspa\_iso\_chk)
  - intl\_glue : (glue\_logic, PD\_glue\_logic)
  - intl\_in\_acpt\_UPF\_ISO : (mspa\_iso\_chk)
  - intl\_io : (intl\_io, PD\_intl)
    - in2wire : (rdyacpt, PD\_intl)
    - out2wire : (rdyacpt, PD\_intl)
    - intl\_out\_rdy\_UPF\_ISO : (mspa\_iso\_chk)
  - ram\_we\_UPF\_ISO : (mspa\_iso\_chk\_cell)
  - shift\_ram : (hm\_shift\_ram, PD\_shift\_ram)
  - dout\_UPF\_ISO : (mspa\_iso\_chk\_cell)
  - hm\_sram : (hm\_sram2p\_2kx8, PD\_sram)
  - raddr\_UPF\_ISO : (mspa\_iso\_chk\_cell)
  - re\_UPF\_ISO : (mspa\_iso\_chk\_cell)
  - shift\_ram\_perif : (periph, PD\_shift\_ram)
  - waddr\_UPF\_ISO : (mspa\_iso\_chk\_cell)
- qpa\_top : (qpa\_top, Undefined)
  - PA\_HIGHLIGHT\_INST\_4 : (PA\_EVENT\_PROCESSOR)
  - PA\_HIGHLIGHT\_INST\_8 : (PA\_EVENT\_PROCESSOR)
  - inst0\_0\_upf\_iso\_cell\_model\_pwr : (upf\_iso\_cell\_model\_pwr)
  - inst1\_0\_upf\_iso\_cell\_model\_pwr : (upf\_iso\_cell\_model\_pwr)
  - inst2\_0\_upf\_iso\_cell\_model\_pwr : (upf\_iso\_cell\_model\_pwr)
  - inst3\_0\_PD\_ACTVT\_INFO : (PD\_ACTVT\_INFO)
  - inst4\_0\_PD\_ACTVT\_INFO : (PD\_ACTVT\_INFO)

hm\_interleaver.v (UPF Simstate : NORMAL->CORRUPT)

```

1  module interleaver_v (
2      input clk, reset_n, di_rdy, do_acpt, enable,
3      input [7:0] di_data,
4      output do_rdy, di_acpt,
5      output [7:0] do_data
6  );
7
8
9  wire in_hs;
10 wire out;
11 wire intl_in_acpt;
12 wire intl_out_rdy;
13 wire ram_re;
14 wire bypass_ram;
15 wire [7:0] ram_data;
  
```

Transcript

```

# Reading file /in/inndt55/Manu/MTI/MyTest/Axiom/PA/UPF_Hier_PA_DEMO/RTL_SRC/glue_logic.v
# Reading file /in/inndt55/Manu/MTI/MyTest/Axiom/PA/UPF_Hier_PA_DEMO/PA_upf/interleaver.upf
# Reading file /in/inndt55/Manu/MTI/MyTest/Axiom/PA/UPF_Hier_PA_DEMO/RTL_SRC/rdyacpt.v
# Reading file /in/inndt55/Manu/MTI/MyTest/Axiom/PA/UPF_Hier_PA_DEMO/RTL_SRC/hm_periph.v
Visualizer>
Visualizer>
Visualizer>
Visualizer>
  
```

hm\_interleaver.v

Current status of power

Power Domain information

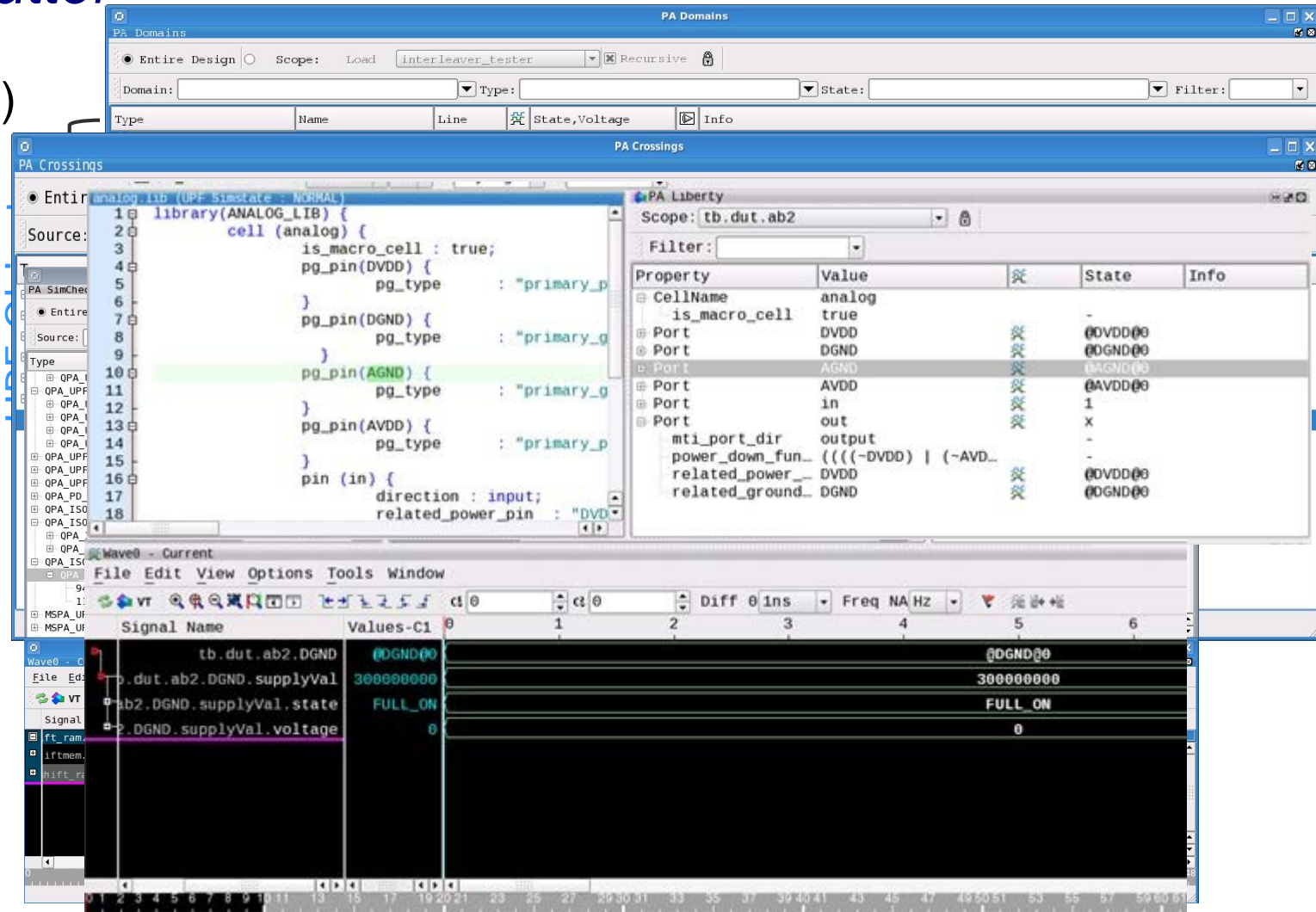
Color coded to display PA violation



# Visualize UPF in Context

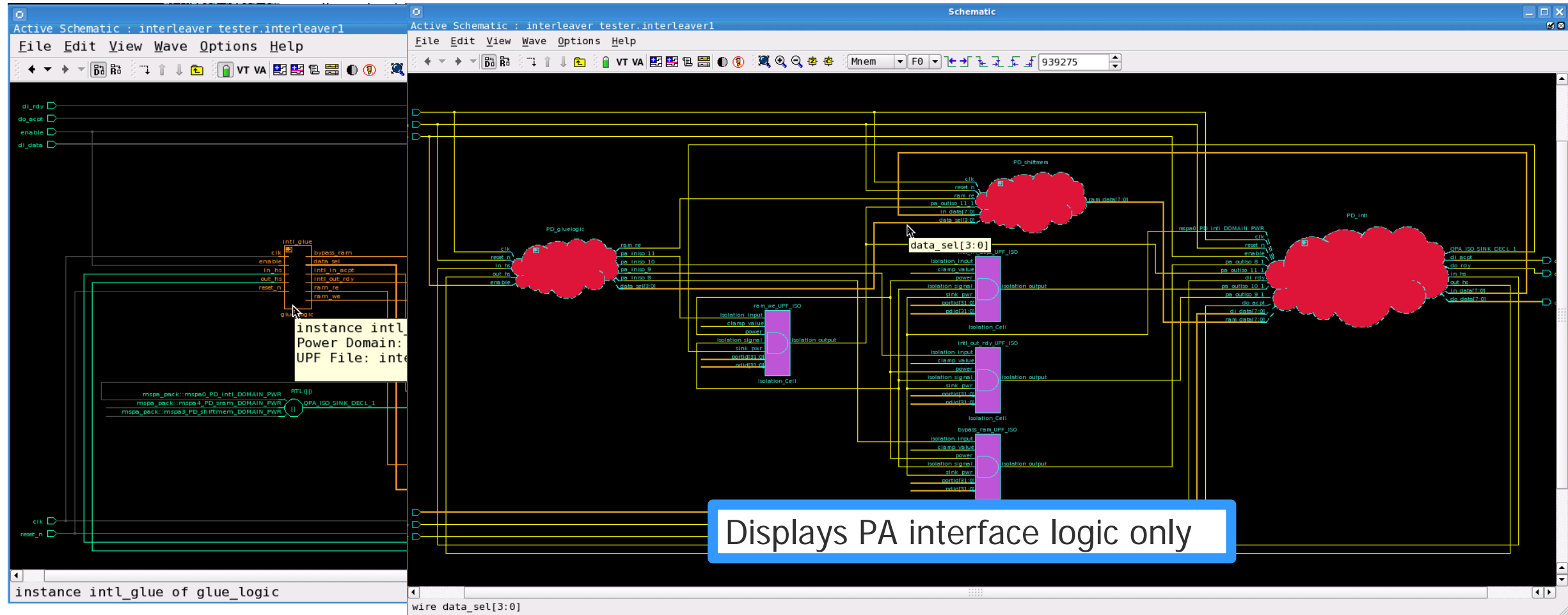
## *Without HDL Clutter*

- PA View of the Design (PA Domains)
  - Power Domains
  - Supply Sets
  - Power States
  - Supply Nets/Ports
  - Logic Nets/Ports
  - Power Switch
- PA Crossing Window
  - Debug Static check results
  - View source and sink domain crossings
- PA SimChecks Window
  - Sort and Debug Dynamic Checker Violations
- PA Liberty Window
  - View Liberty PA attributes of Macros



# Power Aware Schematic

## *View Power Annotation and Interface PA Logic*







# Coverage Closure in Power Aware Verification

- Start with a plan ..
- Collect relevant coverage metrics – List all of Questa's automated PA coverage metrics
- User can augment leveraging
  - PA Information Model to access the appropriate UPF object and HDL objects
  - Bind\_checker UPF command to bind in the coverage collector at the appropriate design scope
- Bring it all together using automation that allows plan to be tracked against coverage metric

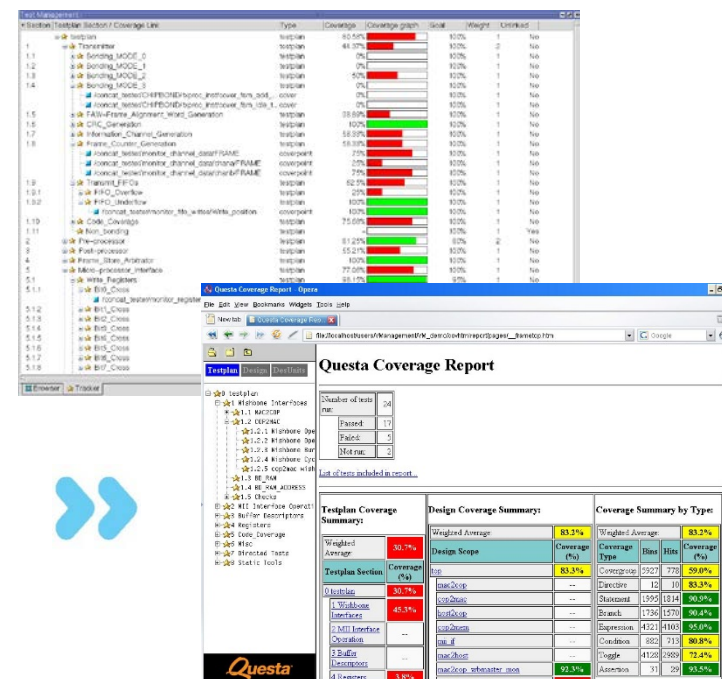
# Essential Power Aware Coverage

- Power states and transitions from `add_power_state`, `add_port_state`, `add_pst_state`
  - To capture states of groups, power domains, supply sets, PSTs, supply ports
    - Including supply ports/nets of supply set functions used in `-supply_expr` of `add_power_state`
    - and supply set simstate and transition coverage for primary, isolation\_supply and retention\_supply of each domain
- Power Domain state cross coverage based on hierarchically composed power states of `add_power_state`
- Retention `save event` and `restore event` coverage
- Built-in **PA Checks** : when checker has activated and passed with no firings in simulation
- State and transition of isolation, retention and Power switch **logic control** and ack signals
- Power switch coverage
  - State and transition of **power switch output supply port**
  - State and transition of user states `-on_states`, `-off_states`, `-error_states`

# Unified Coverage

- All power aware coverage saved in UCIS compatible database UCDB
- Supports Merging coverage across PA regressions
- Unified Coverage waiving mechanism
  - coverage exclude –pstate default\_off my\_supply\_ss –scope dut
- Extensive Reporting
  - coverage report –pa -html
  - Supported formats include Text, HTML, XML
- Flows for Merging PA and Non PA coverage
- Track coverage metrics against Test Plan

- 
- The collage shows several overlapping windows from different applications:
- A document titled "Testplan for the Cuscut Device" with sections like "Transmitter", "Receiving\_Misc\_0", and "Receiving\_Misc1".
  - A window titled "Specification and Verification of Multi-agent Systems Interaction Protocols Using a Combination of AUMI and Event B".
  - A large red box with the word "PDF" in white.
  - A spreadsheet showing a list of items with columns for ID, Name, Description, Status, and Date.
  - A code editor window displaying XML or JSON-like data structures.
  - A database management interface showing a table with columns like "ID", "Name", "Description", "Status", and "Date".
  - A document titled "DOC, PDF, Excel, XML, DOORS, Calc, API, SQL ...".



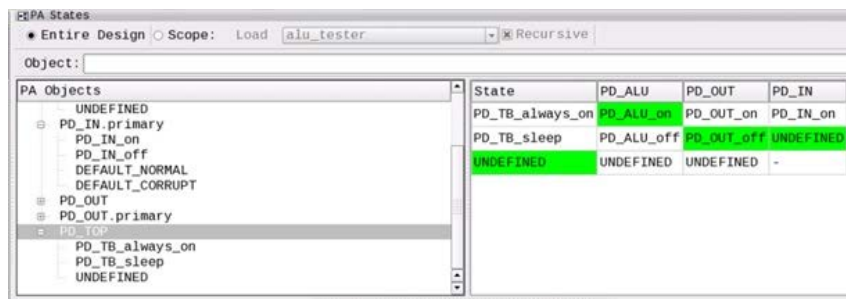
- Supports process of tracking to any metric
  - Project specific attributes/metrics
- Questa PA Sim generates Test plan based on UPF specification



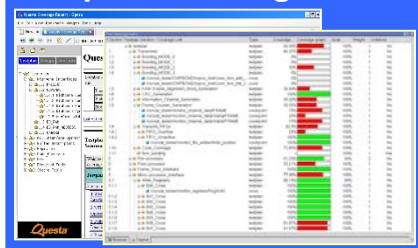
# Combining PowerAware Simulation

Coverage, Assertions and VM

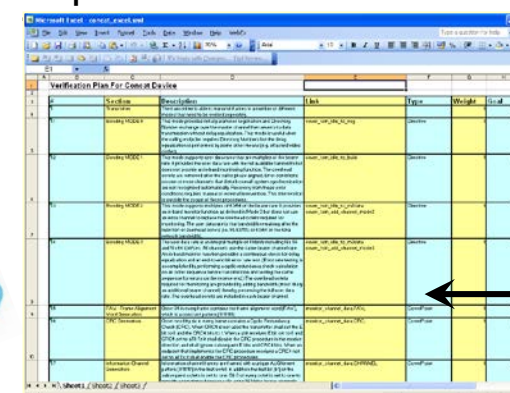
## Analysis



## Testplan Tracking



## Top Level

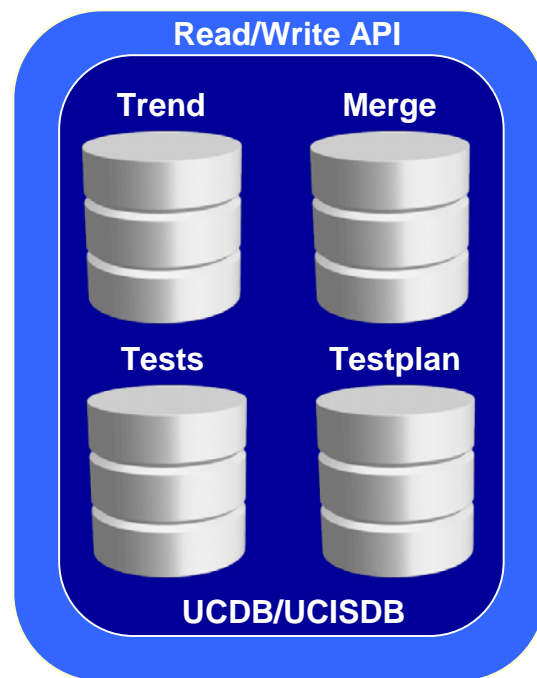
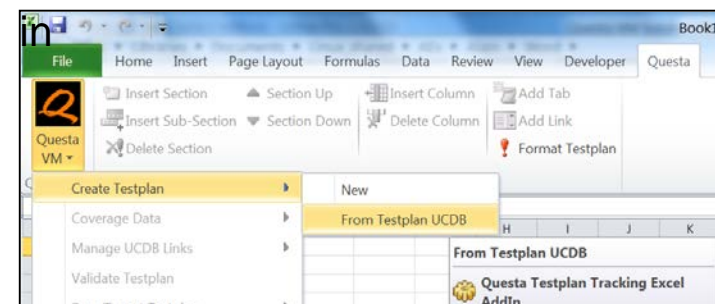


## Sub-plan

## PowerAware Testplan

Section	Title	Description	Link	Type	Weight	Goal
1	Power Aware State Coverage	Coverage plan for PA State machines	/t/bpa_coverageinfo/MyPowerState	FSM	1	100
1.1	Power State Table	State coverage data for PST PASM	Table			
1.2	Supply Port PD_ALU_PORT	State coverage data for supply port PASM	/t/bpa_coverageinfo/PD_ALU_sw/o	FSM	1	100
1.3	Supply Port PD_RAM_PORT	State coverage data for supply port PASM	ut_sw_PD_ALU	FSM	1	100

## Excel Add-In



Power State Coverage,  
PowerAware  
Assertions,  
Other PA Coverage

PowerAware Testplan

PowerAware  
Simulation

## Sub-plan

ID	Description	Link	Type	Weight	Level
1	Windows XP Features				
2	Windows XP Features				
3	Windows XP Features				
4	Windows XP Features				
5	Windows XP Features				
6	Windows XP Features				
7	Windows XP Features				
8	Windows XP Features				
9	Windows XP Features				
10	Windows XP Features				
11	Windows XP Features				
12	Windows XP Features				
13	Windows XP Features				
14	Windows XP Features				
15	Windows XP Features				
16	Windows XP Features				
17	Windows XP Features				
18	Windows XP Features				
19	Windows XP Features				
20	Windows XP Features				
21	Windows XP Features				
22	Windows XP Features				
23	Windows XP Features				
24	Windows XP Features				
25	Windows XP Features				
26	Windows XP Features				
27	Windows XP Features				
28	Windows XP Features				
29	Windows XP Features				
30	Windows XP Features				
31	Windows XP Features				
32	Windows XP Features				
33	Windows XP Features				
34	Windows XP Features				
35	Windows XP Features				
36	Windows XP Features				
37	Windows XP Features				
38	Windows XP Features				
39	Windows XP Features				
40	Windows XP Features				
41	Windows XP Features				
42	Windows XP Features				
43	Windows XP Features				
44	Windows XP Features				
45	Windows XP Features				
46	Windows XP Features				
47	Windows XP Features				
48	Windows XP Features				
49	Windows XP Features				
50	Windows XP Features				
51	Windows XP Features				
52	Windows XP Features				
53	Windows XP Features				
54	Windows XP Features				
55	Windows XP Features				
56	Windows XP Features				
57	Windows XP Features				
58	Windows XP Features				
59	Windows XP Features				
60	Windows XP Features				
61	Windows XP Features				
62	Windows XP Features				
63	Windows XP Features				
64	Windows XP Features				
65	Windows XP Features				
66	Windows XP Features				
67	Windows XP Features				
68	Windows XP Features				
69	Windows XP Features				
70	Windows XP Features				
71	Windows XP Features				
72	Windows XP Features				
73	Windows XP Features				
74	Windows XP Features				
75	Windows XP Features				
76	Windows XP Features				
77	Windows XP Features				
78	Windows XP Features				
79	Windows XP Features				
80	Windows XP Features				
81	Windows XP Features				
82	Windows XP Features				
83	Windows XP Features				
84	Windows XP Features				
85	Windows XP Features				
86	Windows XP Features				
87	Windows XP Features				
88	Windows XP Features				
89	Windows XP Features				
90	Windows XP Features				
91	Windows XP Features				
92	Windows XP Features				
93	Windows XP Features				
94	Windows XP Features				
95	Windows XP Features				
96	Windows XP Features				
97	Windows XP Features				
98	Windows XP Features				
99	Windows XP Features				
100	Windows XP Features				

**VELOCE®**  
**Emulation**

## Power Aware Coverage, Assertions



## Read/Write API

## Merge

## Testplan

UCDB/UCISDB

# Tracking Verification Against Plan

User Interface, Filter, Query, Sort

The screenshot displays the Verification Management Tracker (VMT) interface. The main window shows a tree view of testplan sections on the left and a table of verification results on the right. The table columns include Sec#, Testplan Section / Coverage Link, Type, Coverage, Coverage graph, Goal, Weight, ENGINEER, MANAGER, METHOD, MILESTONE, PRIORITY, and TEAM. A context menu is open over the table, showing options like View Coverage Details, Configure Selected..., Test Analysis, Find Unlinked, Filter, Summary..., Reimport Testplan and Refresh..., Set Precision..., and Configure Colorization... The 'Filter' option is highlighted. A yellow box labeled 'Filtering' points to the 'Filter' option in the menu. Another yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A third yellow box labeled 'Testplan sections as original document' points to the tree view on the left. A fourth yellow box labeled 'Linked Coverage' points to the 'Coverage' column in the table. A fifth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A sixth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A seventh yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A eighth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A ninth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A tenth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A eleventh yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A twelfth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A thirteenth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A fourteenth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A fifteenth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A sixteenth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A seventeenth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. An eighteenth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A nineteenth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A twentieth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A twenty-first yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A twenty-second yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A twenty-third yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A twenty-fourth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A twenty-fifth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A twenty-sixth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A twenty-seventh yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A twenty-eighth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A twenty-ninth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A thirtieth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A thirty-first yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A thirty-second yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A thirty-third yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A thirty-fourth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A thirty-fifth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A thirty-sixth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A thirty-seventh yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A thirty-eighth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A thirty-ninth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A fortieth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A forty-first yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A forty-second yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A forty-third yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A forty-fourth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A forty-fifth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A forty-sixth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A forty-seventh yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A forty-eighth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A forty-ninth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A fiftieth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A fifty-first yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A fifty-second yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A fifty-third yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A fifty-fourth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A fifty-fifth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A fifty-sixth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A fifty-seventh yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A fifty-eighth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A fifty-ninth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A sixtieth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A sixty-first yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A sixty-second yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A sixty-third yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A sixty-fourth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A sixty-fifth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A sixty-sixth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A sixty-seventh yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A sixty-eighth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A sixty-ninth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A seventieth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A seventy-first yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A seventy-second yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A seventy-third yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A seventy-fourth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A seventy-fifth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A seventy-sixth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A seventy-seventh yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A seventy-eighth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A seventy-ninth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. An eightieth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. An eighty-first yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. An eighty-second yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. An eighty-third yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. An eighty-fourth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. An eighty-fifth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. An eighty-sixth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. An eighty-seventh yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. An eighty-eighth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. An eighty-ninth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A ninetieth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A ninety-first yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A ninety-second yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A ninety-third yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A ninety-fourth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A ninety-fifth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A ninety-sixth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A ninety-seventh yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A ninety-eighth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A ninety-ninth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu. A hundredth yellow box labeled 'Test Association' points to the 'Test Association' option in the menu.

**Testplan sections as original document**

**Linked Coverage**

**Filtering**

**Test Association**

**Verification Test Analysis - Non Zero Coverage for track/testplan/Pre-processor**

Non Zero Coverage Tests	Coverage
datatests~TxDataTest	87.5%
Randtest~1	87.5%
Randtest~2	87.5%
Randtest~6	87.5%
datatests~DataTest	4.16%
datatests~InitialTest	4.16%
Randtest~3	4.16%
Randtest~8	4.16%
Randtest~23	4.16%
Randtest~25	4.16%
Randtest~26	4.16%



The screenshot displays the 'Verification Management Browser' interface. The table lists various test cases with columns for File Name, Test Name, Test Status, Tool, Total Coverage, Total Coverage InCr, CPU Time, Compulsory, User Name, Host Name, and Host OS. Annotations include:

- Test Redundancy:** A green box highlights the first 30 rows of the table, which show multiple instances of 'Randtest' tests.
- Sort on Status:** A green box highlights rows 31-40, which are sorted by status, showing 'Confirm' and 'Prove' tests.
- Verification Method:** A green box highlights rows 41-50, which show tests with different verification methods like 'Simulation' and 'Emulation'.
- CPU:** A green box highlights rows 51-60, which show tests with CPU time measurements. A magnifying glass is placed over the 'FastTest-2.ucdb' row, highlighting the CPU time of 03:47:18.

## Rank Details

# Coverage Results Aligned by Power Domain

Power Domain  
view of Coverage  
results

Sec#	Testplan Section / Coverage Link	Type	Coverage	Goal	% of Goal	Status	Weight	Li
0	testplan	Testplan	32.46%	-	32.46%		1	
1	User Defined Low Power Coverage	Testplan	45.05%	100%	45.05%		1	
1.1	Vhm_top_tb/hm_top_inst/VDD_MX\	Testplan	0%	100%	0%		1	
1.1.1	Misc Covergroups	Testplan	0%	100%	0%		1	
1.1.1	/hm_top_tb/hm_top_inst/cov_pwr_states_sequence_VDD_MX/cg_cov_pwr_states_seq	CoverGroup	0%	100%	0%		1	
1.2	Vhm_top_tb/hm_top_inst/VDD_HM\	Testplan	0%	100%	0%		1	
1.2.1	Misc Covergroups	Testplan	0%	100%	0%		1	
1.3	Vhm_top_tb/hm_top_inst/u_ret_block_2/CX_INT\	Testplan	37.5%	100%	37.5%		1	
1.3.1	CX2INT_iso_keep	Testplan	50%	100%	50%		1	
1.3.1.1	Isolation bind_checker Coverage	Testplan	50%	100%	50%		1	
1.3.1.1	/hm_top_tb/hm_top_inst/u_ret_block_2/check_cov_isolation_UPF_GENERIC_OUTPUT_din_bus_1/cg_cov_keep_iso_op	CoverGroup	50%	100%	50%		1	
1.3.1.1	/hm_top_tb/hm_top_inst/u_ret_block_2/check_cov_isolation_UPF_GENERIC_OUTPUT_din_bit_0/cg_cov_keep_iso_op	CoverGroup	50%	100%	50%		1	
1.3.2	Misc Covergroups	Testplan	25%	100%	25%		1	
1.3.2	/hm_top_tb/hm_top_inst/u_ret_block_2/cov_pwr_states_sequence_CX_INT/cg_cov_pwr_states_seq	CoverGroup	25%	100%	25%		1	
1.4	Vhm_top_tb/hm_top_inst/u_ret_block_1/CX_INT\	Testplan	37.5%	100%	37.5%		1	
1.5	Vhm_top_tb/hm_top_inst/u_ret_block_0/CX_INT\	Testplan	41.66%	100%	41.66%		1	
1.6	Vhm_top_tb/hm_top_inst/u_hm_lp/VDD_INT\	Testplan	87.5%	100%	87.5%		1	
1.7	Vhm_top_tb/hm_top_inst/u_level_1/VDD_SUB\	Testplan	81.25%	100%	81.25%		1	
1.8	Vhm_top_tb/hm_top_inst/u_level_1/VDD_HM_INT\	Testplan	75%	100%	75%		1	
1.8.1	Misc Covergroups	Testplan	75%	100%	75%		1	
2	Tool Low Power Coverage	Testplan	19.87%	100%	19.87%		1	
2.1	Vhm_top_tb/hm_top_inst/VDD_MX\	Testplan	25%	100%	25%		1	
2.1.1	primary	Testplan	25%	100%	25%		1	
2.1.1.1	Power State Coverage	Testplan	50%	100%	50%		1	
2.1.1.1	/hm_top_tb/hm_top_inst/pa_coverageinfo/VDD_MX/primary/primary_STATE_COVERAGE/hm_top_tb/hm_top_inst/pa_coverageinfo/VDD_MX/primary/PS_primary	CoverInstance	50%	100%	50%		1	
2.1.1.2	Power State Transition Coverage	Testplan	0%	100%	0%		1	
2.2	Vhm_top_tb/hm_top_inst/VDD_HM\	Testplan	33.33%	100%	33.33%		1	
2.3	Vhm_top_tb/hm_top_inst/u_level_1/VDD_SUB\	Testplan	27.77%	100%	27.77%		1	
2.4	Vhm_top_tb/hm_top_inst/u_level_1/VDD_HM_INT\	Testplan	25%	100%	25%		1	

# Summary

- UPF 3.0 eases specification of design power intent
  - Abstract/refinement of power states
  - Hierarchical composition of power states from IP to System
  - Information Model and API
- Comprehensive PA Coverage and Test plan
  - Extracted from UPF power states and UPF strategies
- Questa PA enables users deploy metrics based verification of low power designs

Let's take a break

# Analysis and Reduction: Metrics for Designing Low-Power IP

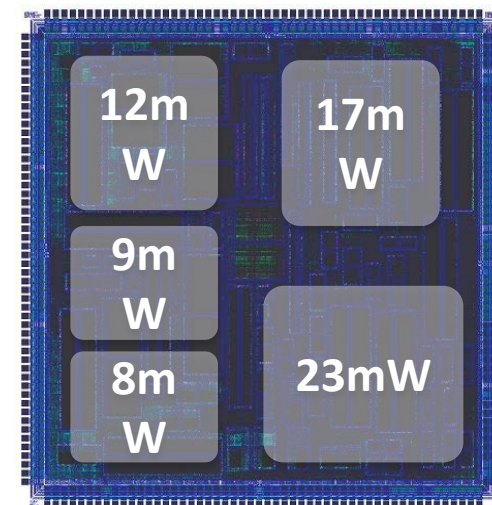
Ellie Burns, Director of Marketing for Calypto Systems Division

**Mentor**®

A Siemens Business

# To Prevent “Power Surprises” Need IP Power Audit Methodology

- RTL IP Qualification
  - Customers require IP RTL Qualification Metrics for Power
  - Metrics required to define Power Signoff Criteria
  - Based on the criteria, IPs may be “ranked”
- RTL IPs may require “re-spin” if they do not meet power qualification criteria
  - Coding styles must not be power hungry
  - Wasted power must be reduced
  - RTL designers need to work from a “Low-Power” mindset





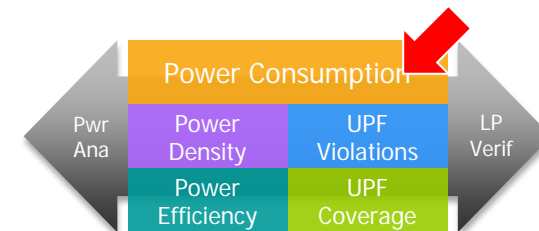
# But Why Not Just Use Power Consumption Numbers?

- Looking at power consumption numbers through Power Regression
  - Micro-architectural Exploration – Evaluating how different micro-architectures impact power.
  - What is the power trend as RTL progresses?
  - Is the power budget being met?
- Power Optimization may need to be looked at “Objectively”
  - How many flops in the design are gated?
  - How many flops are highly efficient?
  - How effective are the enables?
  - Are there any redundant toggles in the design?

Power Group	Count	Leakage Power (uW)	Internal Power (uW)	Total Power (uW)	Percentage (%)
memory	2	0.388697	719.494	719.882	20.41%
black_box	1	0.314219	87.93	88.2442	2.5%
register	1171	7.50225	2124.29	2131.8	60.45%
combinational	3048	4.50639	540.253	544.759	15.45%
sequential	11	0.0879686	42.002	42.09	1.19%
total	4233	12.7995	3513.97	3526.77	100%
clock_network	NA	2.63006	18.8737	21.5038	NA

Number of flops	:	26 (100.00%)
Number of conditionally enabled flops	:	8 (30.77%)
Number of always enabled flops	:	18 (69.23%)
Number of CGIC enabled flops	:	0 (0.00%)
Clock Gating Efficiency	:	7.692%

# Power Consumption Metrics



- Leakage power
  - Power consumed by current flow through transistors even when turned off. Concern in lower geometries and ultra-low-power devices
- Dynamic power = internal+switching power
- Internal power
  - Power consumed by the combination/sequential logic during the switching of the logic states. Depends upon the technology node, switching activity (SA), Load capacitance and input slew
- Switching power
  - Consumed by the nets and is a function of the load capacitances of the cells connected and the switching activity (SA)



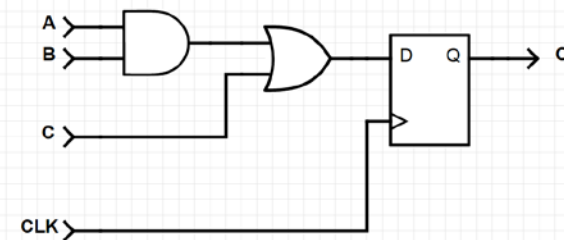
# Power Consumption Metrics

- Clock tree power
  - Summation of all the power groups (leakage, internal and switching) of the clock networks in the design mostly comprised of buffers/inverters and CGIC cells
- Memory power
  - Summation of all the power groups (leakage, internal and switching) of the memories in the design
- Peak power
  - Peak power is defined as the total power of time window in the entire simulation duration where the maximum switching occurs

# Measuring Power Consumption: RTL vs Gate-Level

- RTL Power Estimation
  - Fast turnaround time
  - Target within 15% accuracy of gate-level
  - Best suited to track power during RTL development
- Gate-Level Power Estimation
  - Most accurate flow, close to silicon power consumption
  - Uses either synthesis or post P&R netlist
  - Can use SDF and find glitch power problems
  - Long turnaround time
    - Need to run synthesis and gate-level simulation
    - Good compromise is to be able to use RTL simulation vectors

```
always @(posedge CLK)
  O <= (A & B) | C;
```



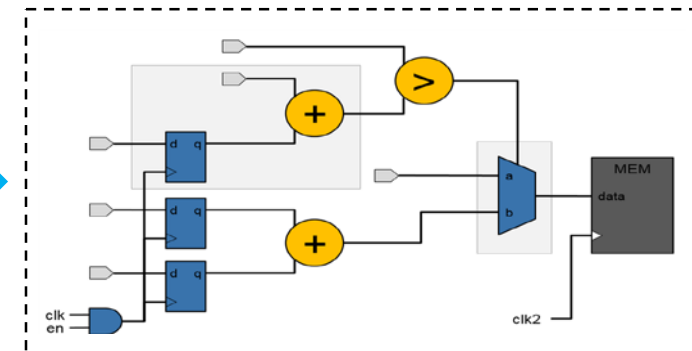
# Accurate Power Consumption at RTL Need to Account for Physical

```
always @ (posedge clk or negedge rst)
begin
  if (~rst)
    cnt <= 0;
  else
    if (~en)
      if (~load)
        cnt <= val;
      else
        cnt[3:0] <= cnt[3:0] + 1;
    end
end
```

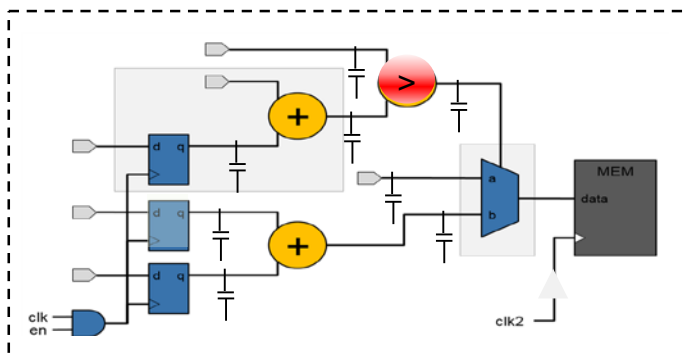
RTL

- Word level infrastructure
- Technology aware mapping
- Buffering
- Clock tree synthesis

Generate quick prototype



Gate level prototype



Physical prototype

- Bind power critical cells – Flop, latch, CGIC
- Vth distribution
- Clock tree topology
- Fine-grain wire cap model

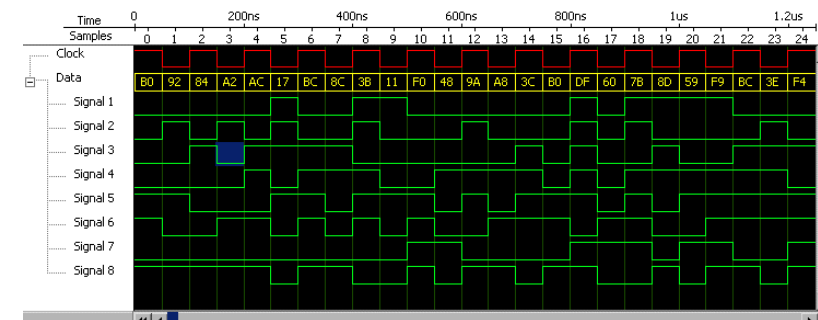
Physical char. ← SPEF

```
*SPEF "IEEE 1481-1998"
*DESIGN "top"
*DATE "Wed Oct 1 2014"
*VENDOR "Mentor Graphics Corp."
*PROGRAM "Olympus"
*VERSION "1.0"
*DESIGN_FLOW "PIN_CAP NONE" "NAME_SCOPE LOCAL"
*DIVIDER /
*DELIMITER :
*BUS_DELIMITER []
*T_UNIT 1.0000 PS
*C_UNIT 1.0000 FF
*R_UNIT 1.0000 KOHM
*L_UNIT 1.0000 HENRY
*143620 n20
*169 clk_buf
*73901 mips/dp/rf/auto_clk_c10525
...
```

SPEF

# Accurate Power Consumption Requires Good Switching Activity Input

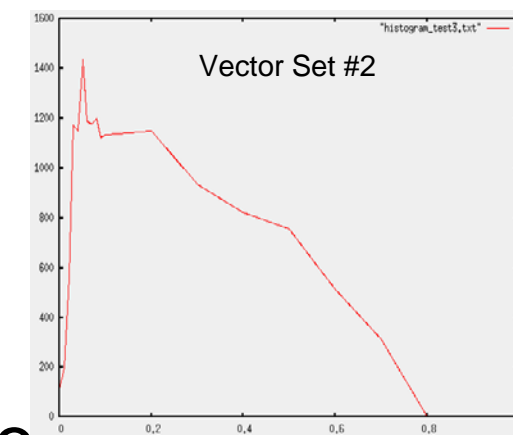
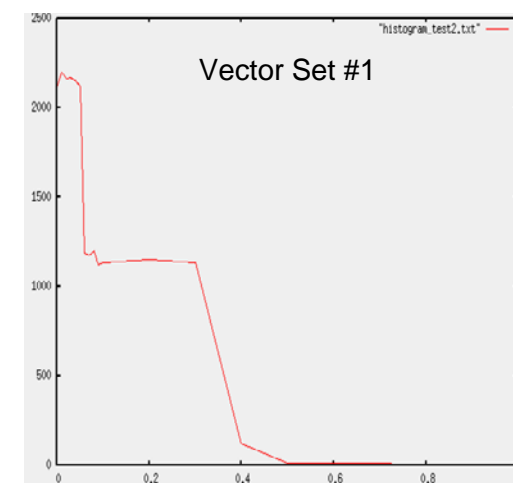
- Understanding switching activity scenarios are important for estimation
- User assumptions about vector sets are often incorrect
  - Could cause a power problem to be missed
- Format choices for switching activity
  - SAIF – Sums of toggles, smaller file size, only average power can be calculated
  - SDPD SAIF – More accurate
  - Waveform data – FSDB, VCD, other native formats – large file size, can be used for peak power and optimization
  - Emulation – Native waveform, Direct API



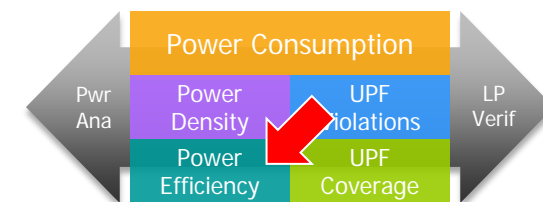
# Need to Have a Way to Measure the Quality of Switching Activity Data

- Need to audit switching activity data
  - Helps Categorize: Idle, Sustained or Traffic
  - Generates toggle count histograms
  - Detailed reports for all critical signals
  - Flags problematic vectors
  - Flags dead/blocked clocks
- Saves valuable debug time
- Better productivity through better quality vector sets – Don't waste time!

Toggle rate histograms



# Examples of Power Efficiency Metrics



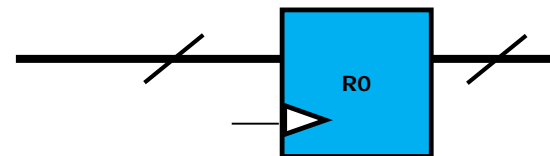
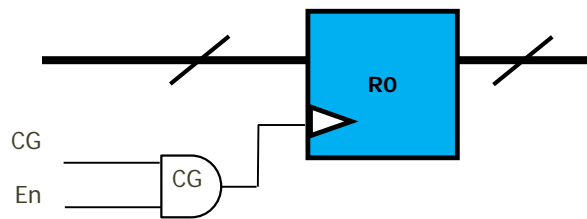
- Clock-gating Ratio
  - Metric used to measure the percentage of un-gated flops vs. gated flops
- Constant Enable/Clock for ICGs
  - Metric to measure if any ICG's are driven by a constant and can be removed
- Redundant Resets
  - Metric to measure how many flops have resets which are not required and therefore a lower power cell could be used

# Examples of Power Efficiency Metrics

- Clock-gating efficiency
  - Metric used to measure the effectiveness of clock gating (ie., reduction of switching activity due to CGIC insertion) of registers in the design
- Memory-gating efficiency
  - efficiency of the memory enables of the memory and the effectiveness of the signals controlling the sleep modes available in the memory.
- Redundant Toggles/Wasted Toggles – “Power Leak”
  - Any toggle-activity in the design which is not required for correct functionality

# Ungated vs Gated Flops

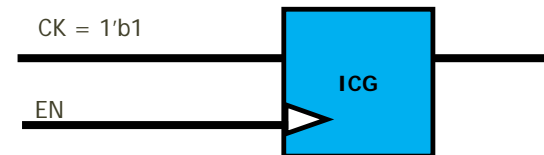
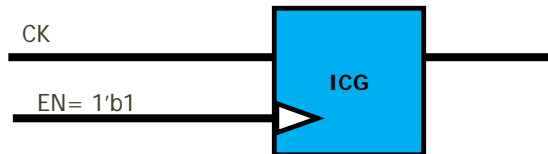
- Can be determined statically
- Just a number (ratio)
- Tells how many flops in the design are gated vs how many flops in the design are ungated
  - Excludes Synchronizer flops
- Ideally, all flops must have an enable
  - Indicates that RTL designer can be more power sensitive in coding





# Constant Enable/Clock for ICGs

- Can be determined statically
- Check to see if ICG enable or clock is driven by a constant
- Based on this, user can decide to remove the ICG
- Can question validity of the enable (will consume leakage power)



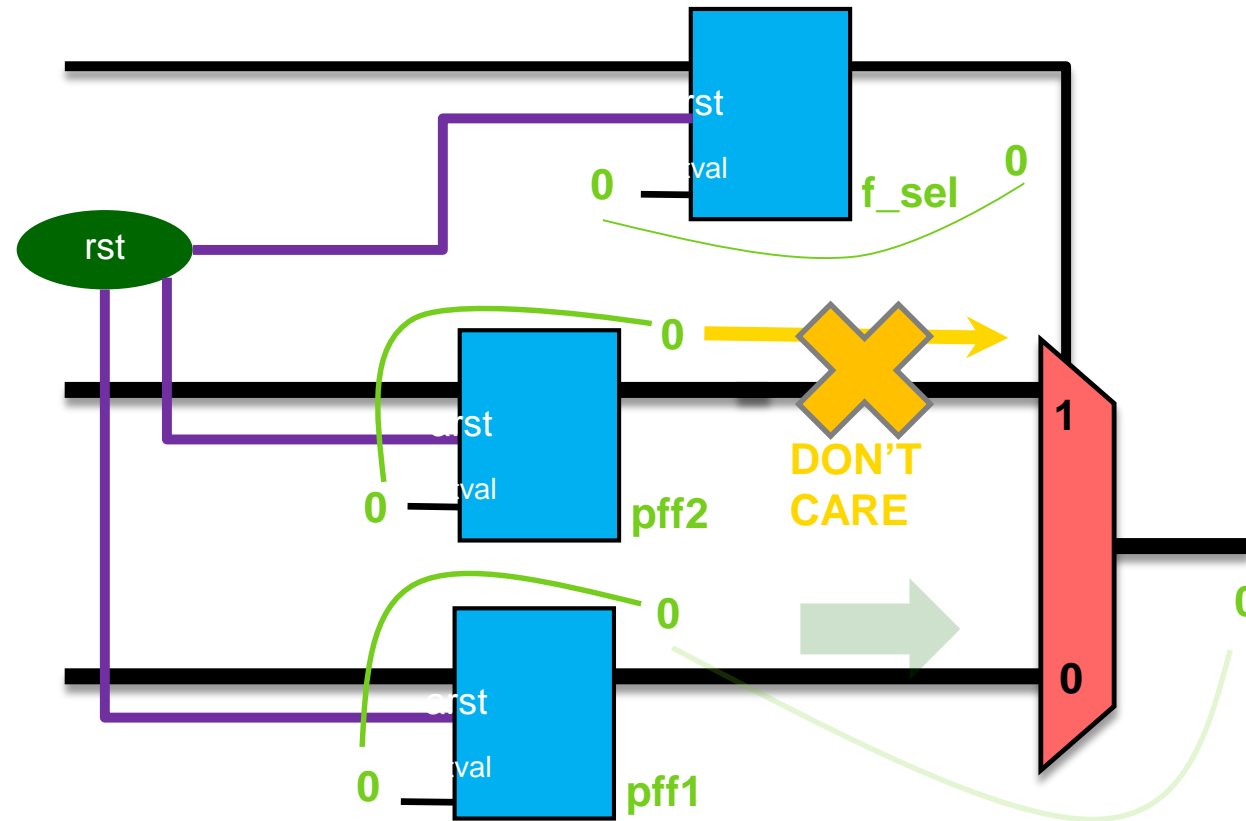
# Redundant Resets

Whenever reset  
goes HIGH...

Output of pff2 would be  
DON'T CARE

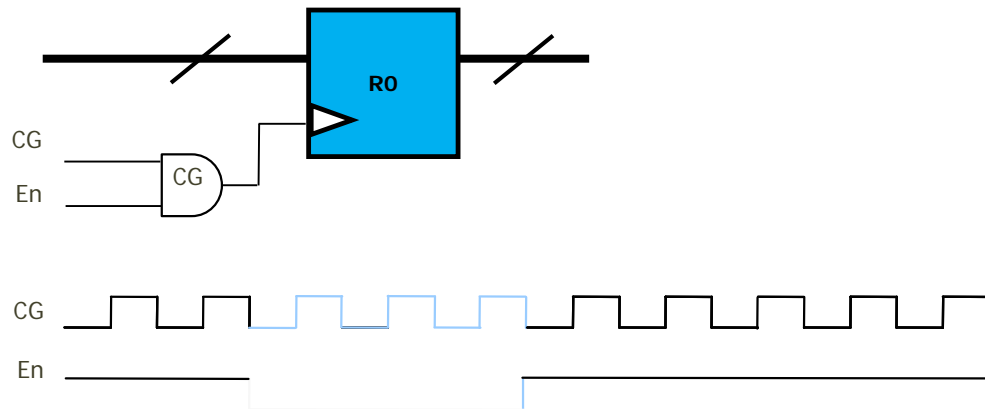
ResetVal of pff2  
is  
never observable

**Reset of pff2  
is redundant**



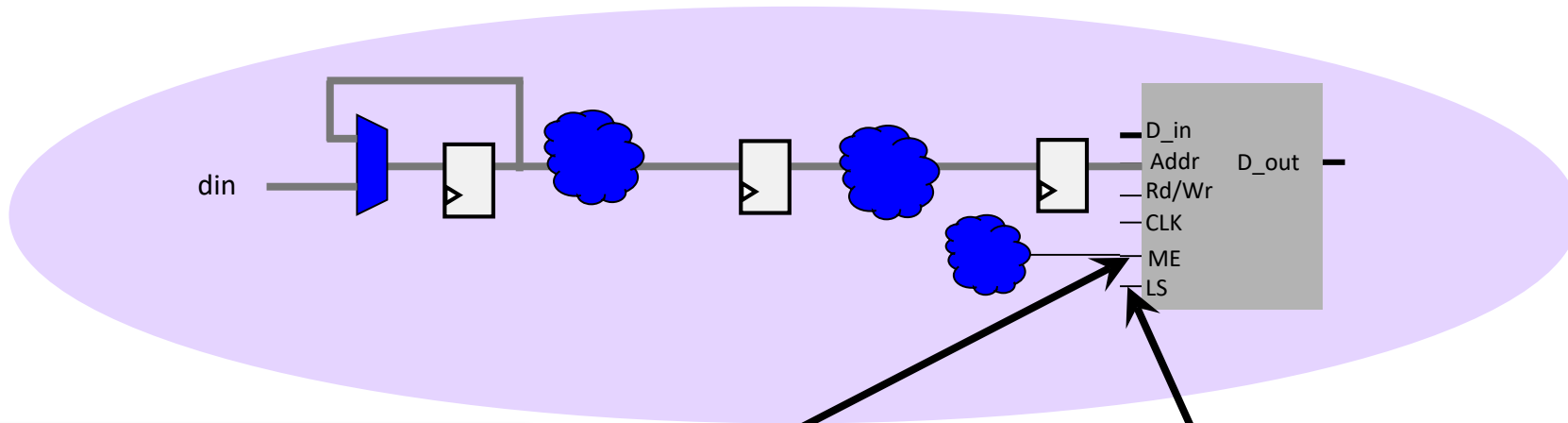
# Clock Gating Efficiency

- Percentage of simulation cycles that the clock is gated off
- Higher CG Eff means lower dynamic switching power consumption
- 100% efficiency => Flop doesn't switch at all
  - Can be indicative of an inactive use mode
- More gating does not always save power
  - Gating logic for the enable can cost more than register power saved



Total Clock Cycles	=	10
Clock gated for	=	3
Cycles		
Clock-gating Efficiency	=	
		30%

# Memory Gating Efficiency

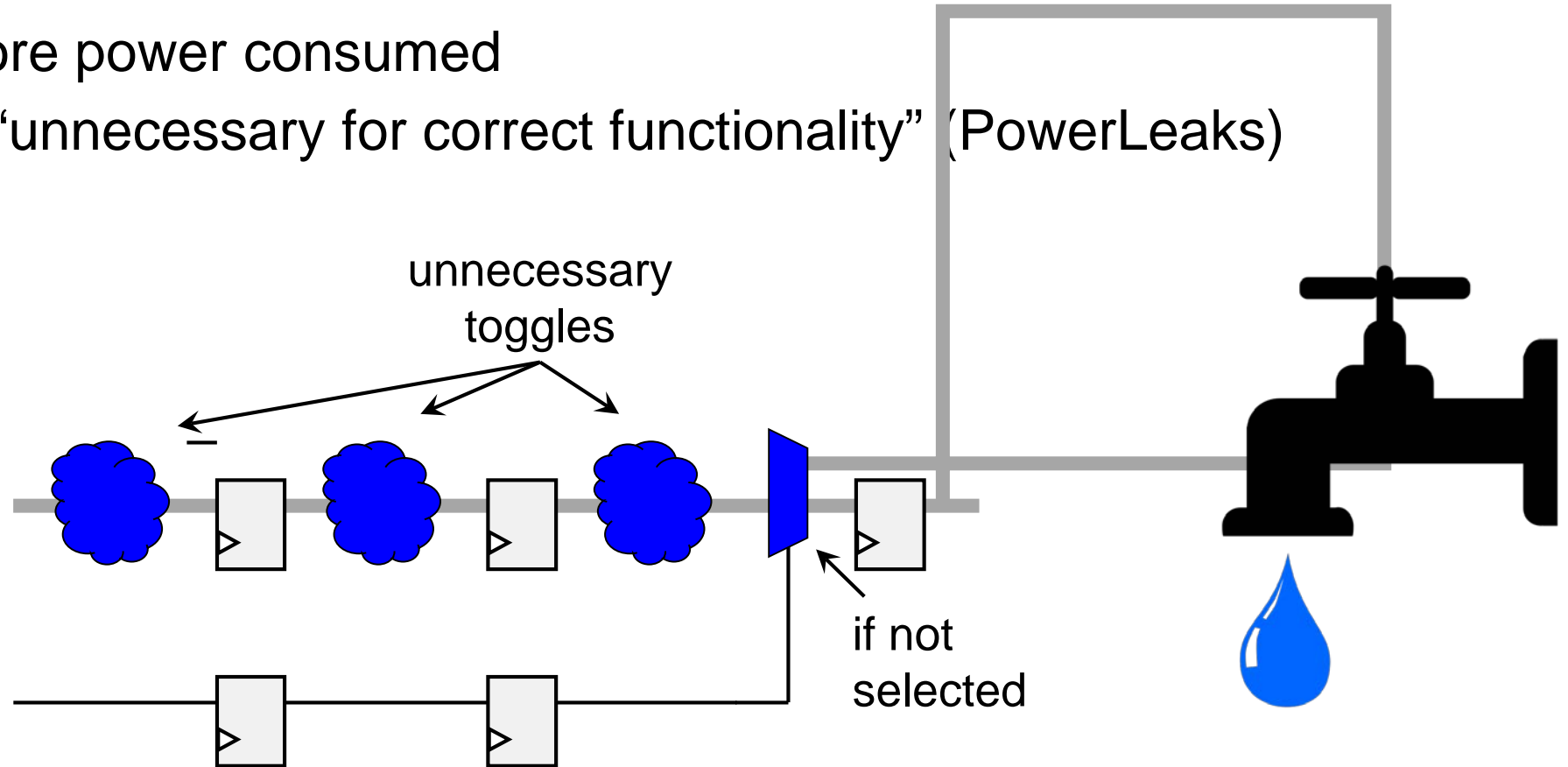
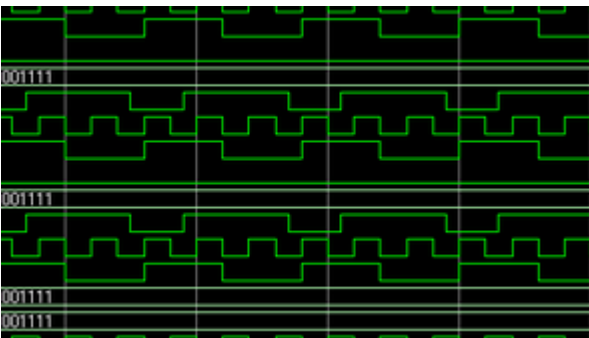


Probability-of-being-0 of "ME/CS" signal is a good indicator of potential dynamic power-saving due to memory-gating and can serve as proxy for measuring power savings

Probability-of-being-1 of "LS" signal is a good indicator of potential leakage power-saving due to sleep-gating and can serve as proxy for measuring power savings

# Redundant Toggle Metric – Power Leak

- More toggles = More power consumed
- Many toggles are “unnecessary for correct functionality” (PowerLeaks)



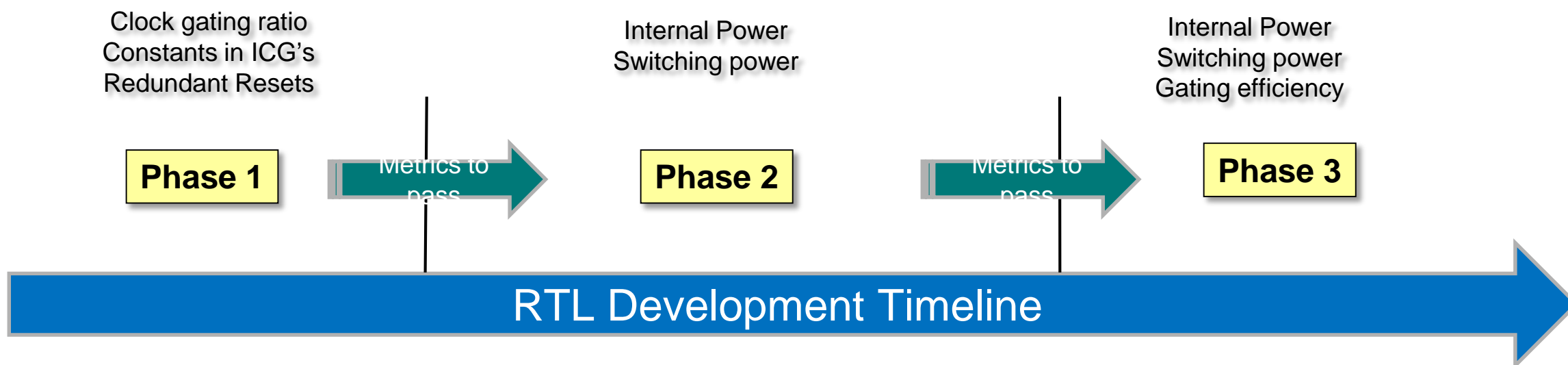
*Any toggle which is not needed for correct functionality is a power leak*

# Types of Redundant Toggles and Events that Waste Power

- DATA GATING:
  - Datapath operators (adders, multipliers etc.) whose inputs can be gated to reduce redundant toggles.
- REDUNDANT MUX TOGGLES:
  - Multiplexers whose unselected inputs are consuming a lot of power.
- CLOCK-TOGGLE DATA-STABLE GATING POTENTIAL:
  - Reports potential power savings and efficiency improvement that can be achieved by applying Clock-Toggle Data-Stable based gating on stable flops.
- REDUNDANT MEMORY ADDRESS TOGGLES:
  - Memory address ports which are very active despite memory not being enabled for that duration.
- REDUNDANT MEMORY DATA TOGGLES:
  - Memory data ports which are very active despite memory not being enabled for that duration

# Qualifying your IP for Power Throughout the RTL Development Flow

- Use metrics that don't require "power simulations" very early in RTL to gauge whether RTL designers have done adequate work for power
- Track metrics in regression as RTL blocks mature and working towards power budget
- Use metrics for efficiency to "scrub" for power and judge RTL readiness



# PowerPro in the RTL Design Cycle

## RTL Development Cycle/Timeline

- Just need design

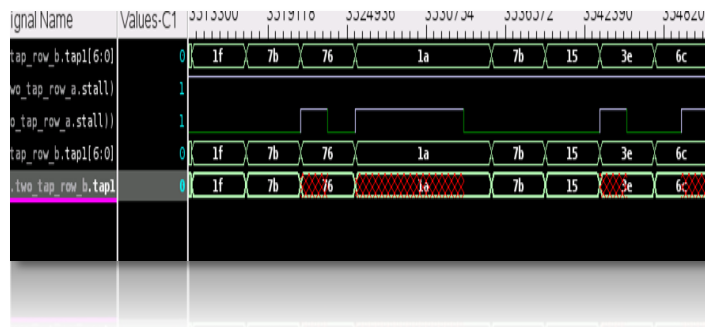
```
1 module adder
2   #(length = 4)
3   (input wire [length-1:0] a, b,
4    output logic [length-1:0] sum,
5    output logic overflow);
6
7   logic [length-1:1] carry;
8
9   always @(a or b)
10    begin
11      carry[1] = 0;
12      for (int i=0; i<length; i = i+1)
13        begin :addloop
14          sum[i] = a[i] ^ b[i] ^ carry[i-1];
15          carry[i] = ((a[i] & b[i]) & carry[i-1]) | (a[i] & b[i]);
16        end :addloop
17      overflow = carry[length-1] ^ carry[length-2];
18    end
19 endmodule :adder
```

```
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
```

Early RTL

PowerPro Static Checks

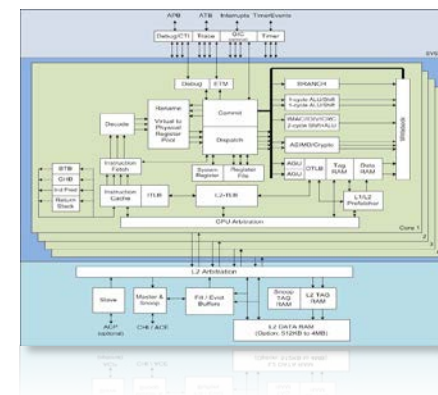
- Need design and “ok” toggle activity



Functional RTL

PowerPro Estimation and  
Guided Optimization

- Design both block and SoC
- Good activity and power scenarios



Stable RTL

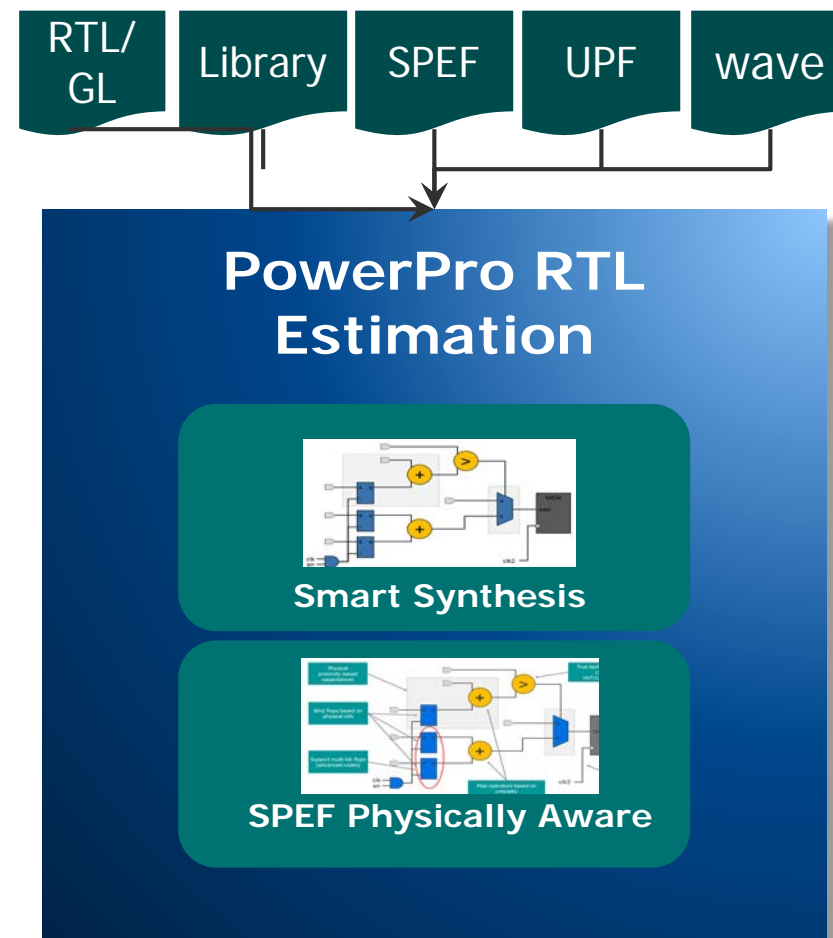
Block and SoC Verification

PowerPro RTL and Gate  
Estimation and Automatic  
Optimization



# PowerPro Fast and Accurate Power Estimation

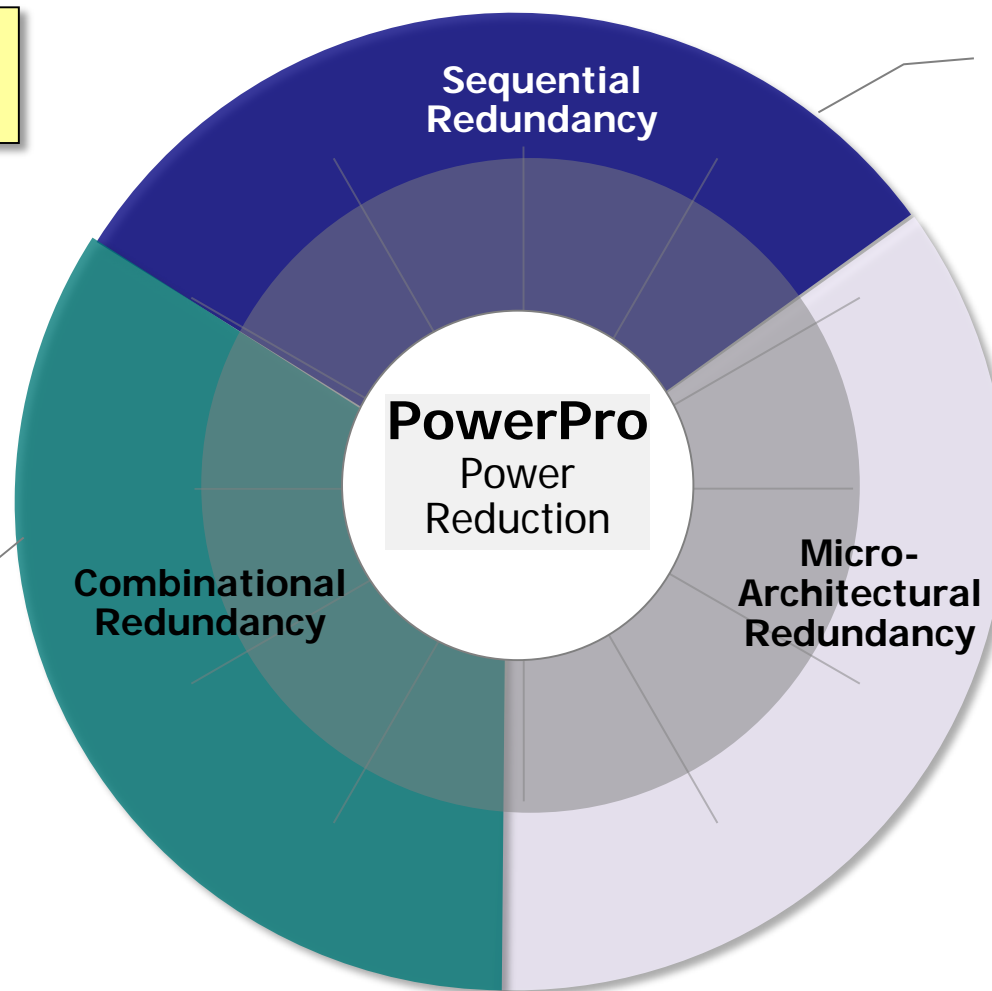
- “Physical Aware” flow
  - RTL accuracy: within 15% of layout
  - GL accuracy: within 2-5% of layout
- High Performance and Capacity
- Supports both RTL and Gate-level estimation
- Comprehensive power reports
  - Average and Peak
  - Dynamic and Leakage
  - Logic, memory, register, clock-tree
  - Hierarchical reporting



# PowerPro Power Optimization Opportunities and Metrics

*PowerPro Finds More Kinds of Opportunities than other Tools*

- Identification of wasted power...  
... in combinational logic...  
... on a single timing boundary



- Identification of wasted power...  
... in sequential logic...  
... on one or more timed sequential boundaries

- Identification of wasted power...  
... that requires more complex micro-architectural transformations

# Visualizer Debugging for PowerPro

The screenshot displays the Visualizer software interface with several key components highlighted by red boxes and labels:

- Design Tree:** Located on the left, showing a hierarchical view of the design components, including 'testbench', 'correlator\_top', and various memory and shifter instances.
- RTL Browser:** Located in the center-left, displaying the RTL code for the 'data' module, showing assignments and logic blocks.
- Dashboard:** Located in the center, providing a summary of design statistics and micro-architectural redundancy metrics.
- RTL Schematic Viewer:** Located on the right, showing a schematic diagram of the 'InferredFF' block, illustrating the internal logic and connections.
- Reports:** Located at the bottom left, displaying a table of efficiency and power metrics for different components.
- Waveform Viewer:** Located at the bottom right, showing a timing diagram for the 'XoR Gating Potential-2' signal, with a time scale from 0 to 500,000,000.

# Arm® – Low-Power Mindset with PowerPro

- Used across Arm IP: CPUs, GPUs, interconnect sub-systems, and display cores
- Arm deploys PowerPro from the very start of a project cycle
- All block IP's run PowerPro frequently before RTL check-in
  - Power benchmarks run in weekly regressions
- RTL estimation within 4-14% accuracy across projects
- Power savings of 10-15% on benchmarks across project schedule

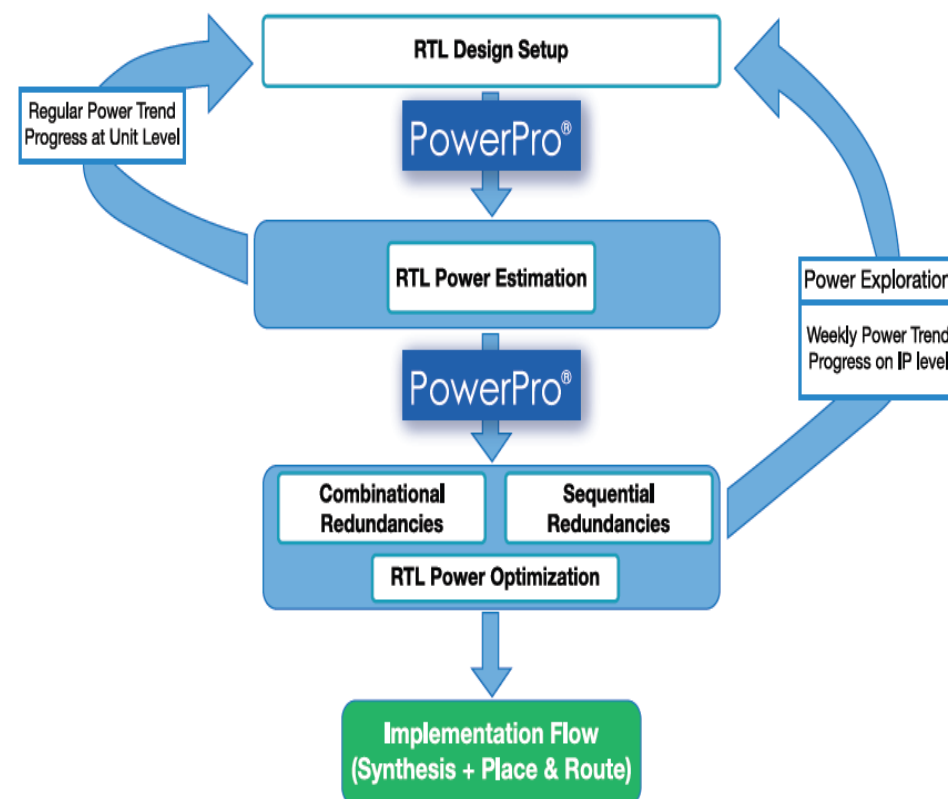


Figure 3: Low-power IP design flow

<http://go.mentor.com/4PmgB>

# Summary

- Today's ultra low-power designs will demand a predictable RTL design methodology to achieve power
- Meaningful metrics will be required to qualify IP for power
- Mentor's PowerPro is a complete Low-Power Solution providing metrics, debugging and optimization

# Using Emulation for Meaningful Metrics-Based Power Analysis and Verification

Guillaume Boillet, Power Product Specialist





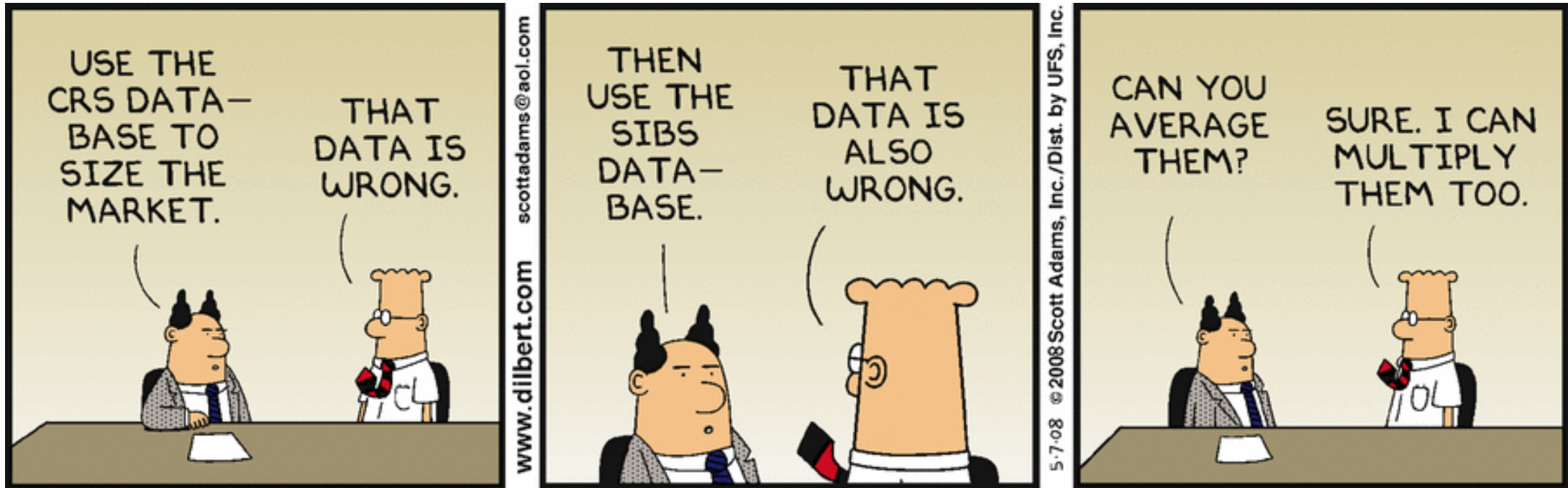
# Agenda

- Why is Emulation needed for Power Analysis?
- What emulation can help with
  - SAIF Generation
  - Dynamic Read Waveform API
  - Activity Plot
    - Real life example: System activity and Arm AXI protocol debug
  - Low-Power Verification with UPF





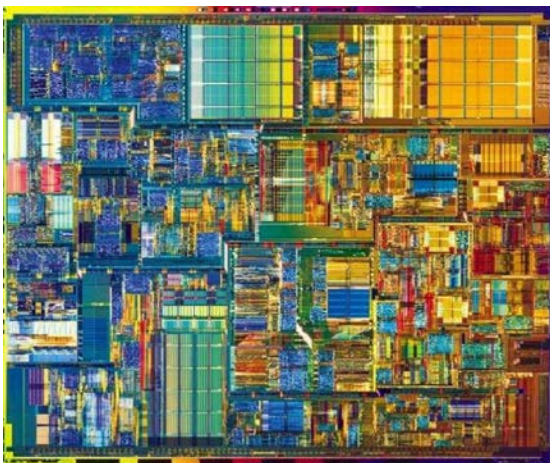
# Garbage In Gospel Out



From <http://dilbert.com/strip/2008-05-07>

- How good is your simulation dataset for the power analysis task at hands?

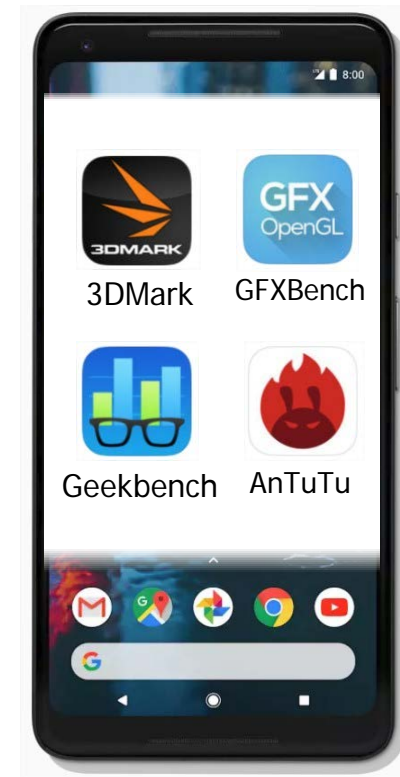
# Why Emulation for Power?



Handle Large SoC  
(RTL/Gate)

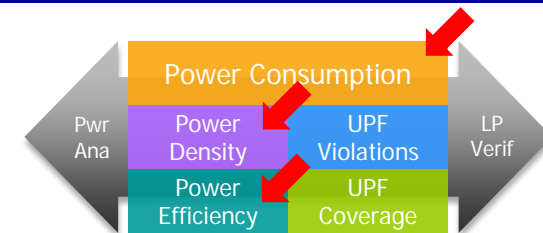


Performance for Complete  
Verification (e.g. OS Boot)  
[100s Millions of Cycles !]



Accurate Power Numbers  
Based on **Real Switching  
Activity**

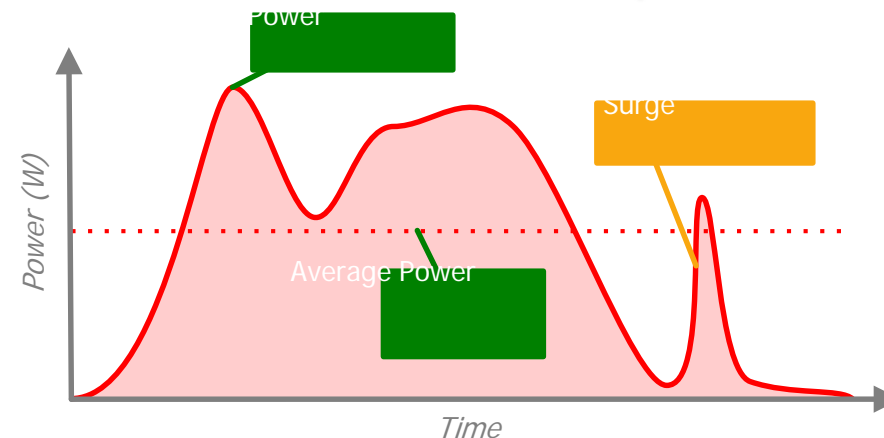
# Motivations for Good Vectors & Emulation



- Representative Power Estimation
  - **Average power** (battery life, cooling requirements, cost of energy,...)
  - **Power peaks**
    - **Large peaks** (~1us) -> Supply integrity,...
    - **Narrow peaks** (~1ns) -> IR-Drop,...
  - **Hot spots** (Local IR-Drop...)
  - **Power domains partition** verification via UPF [Emerging trend – Users increasingly Ready]
  - **Power surges** (dI/dt voltage drop)
  - **High power on very long periods** (Electro-migration)

Primary Concerns

Secondary Concerns

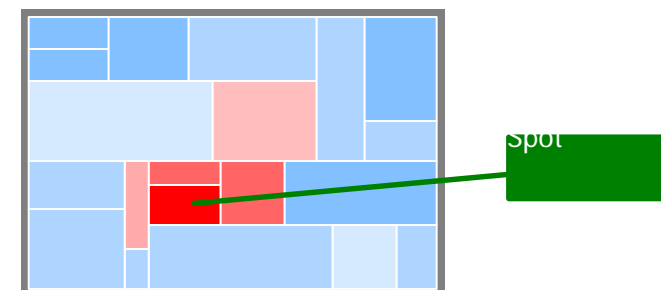


$$VDD_{PMOS} = VDD_{supply} - IR_{grid} - L_{interface} dI/dt$$

Voltage Drop

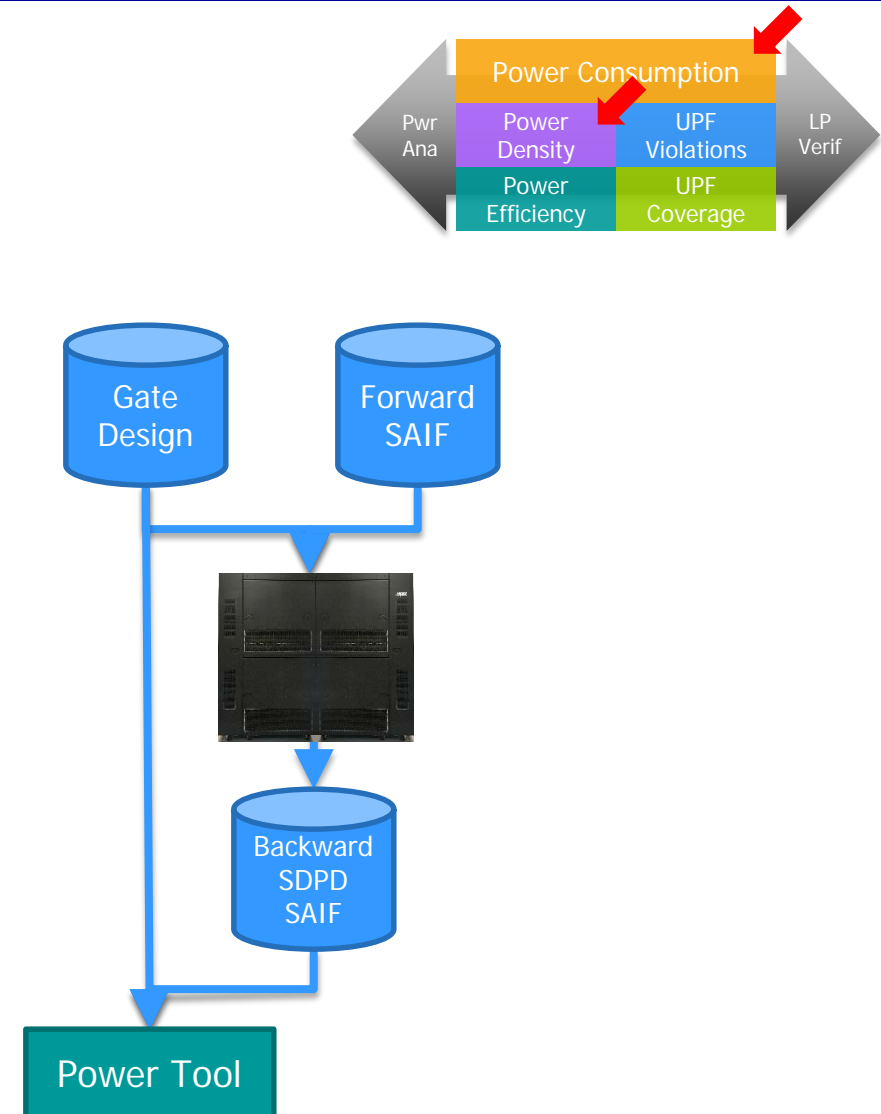
BUT ALSO...

- Representative Power Efficiency analysis
  - Clock-Gating Efficiency (instantiated or inferred)
  - Memory accesses,...
- Relevant Power Reduction suggestions RTL
  - Some techniques are highly dependent on quality of local activity information (i.e. Sequential Clock Gating)

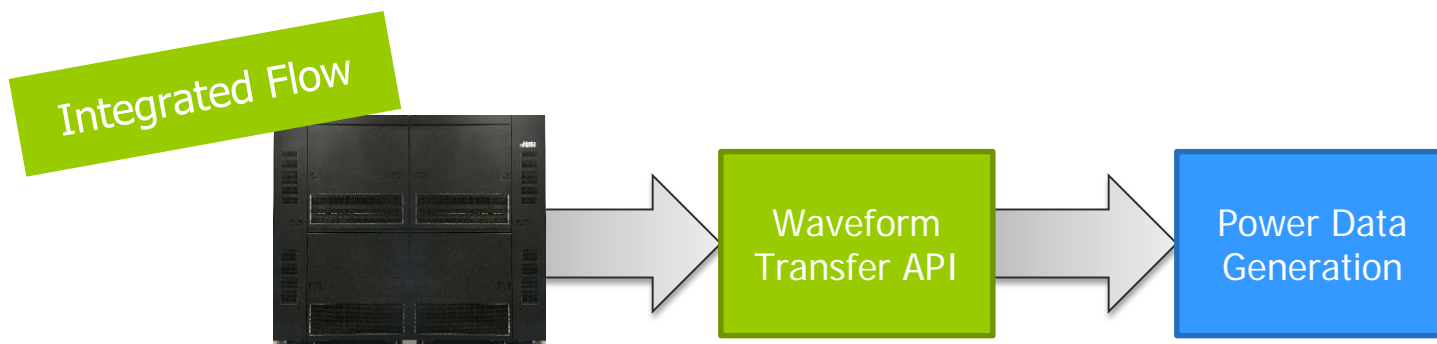
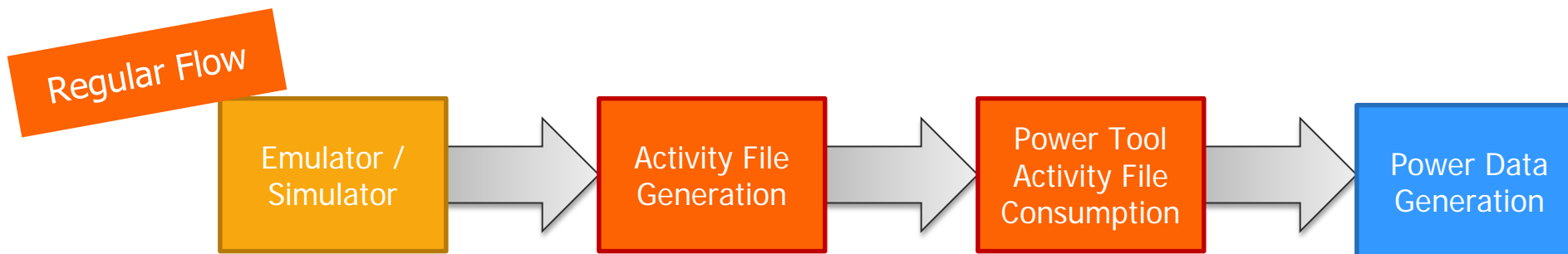
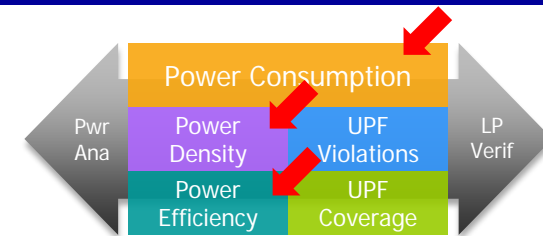


# SAIF Generation

- For RTL or netlist
- Online (as emulator is running) or Offline generation
- State-Dependent Path Dependent at gate-level
  - E.g. Toggles of Net A caused when Net B has a rising edge
  - These condition/directives are stored in a Forward SAIF file



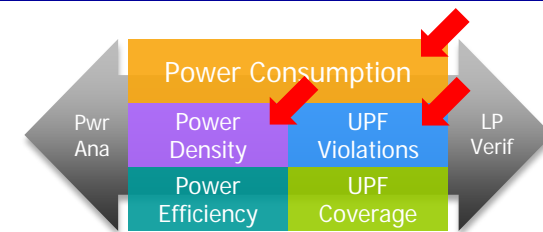
# Regular vs. Integrated Power Analysis Flow (API)



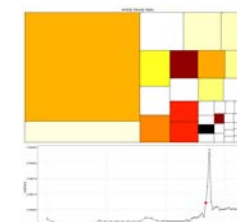
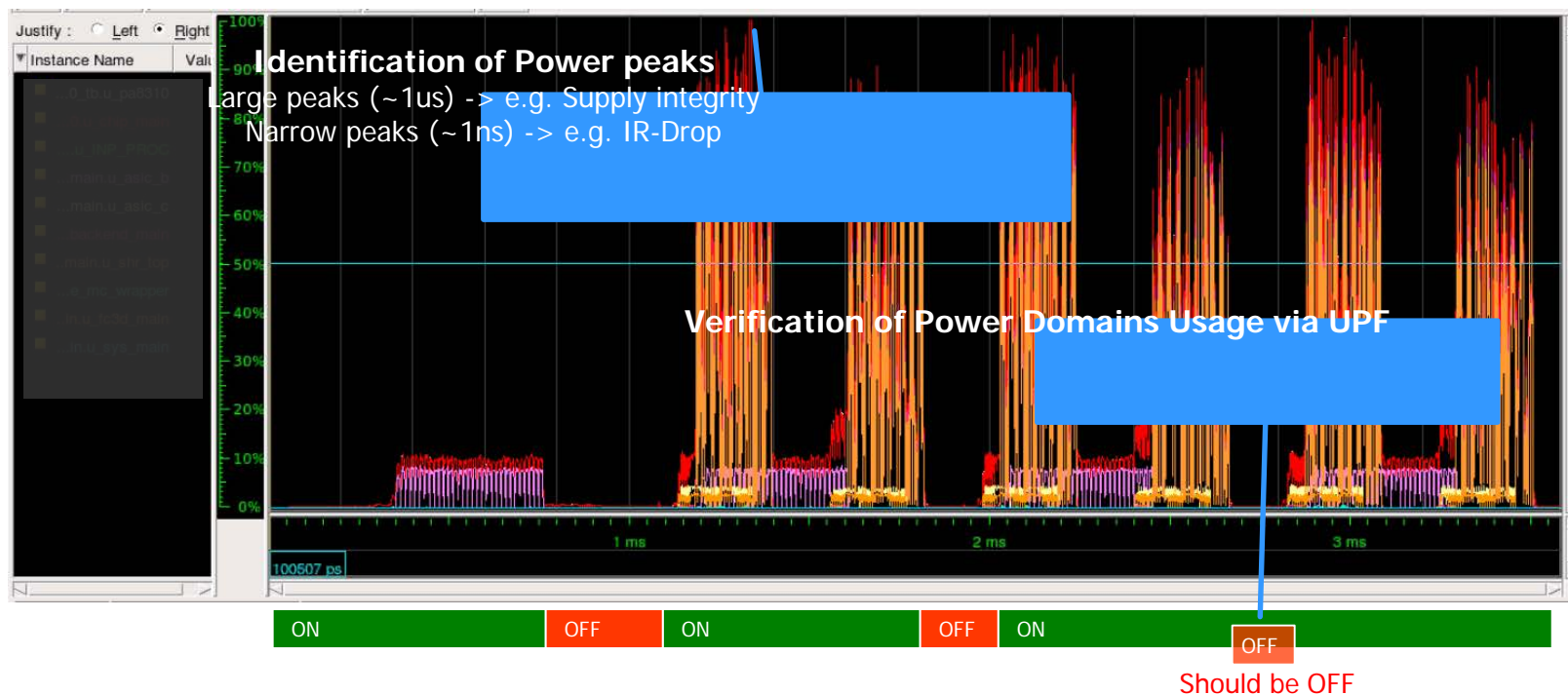
**Up to 10X faster  
Reduced disk space  
requirements**



# Activity Plot Applications



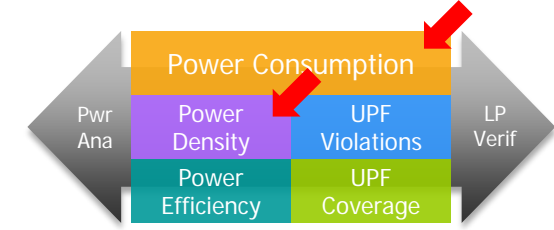
100X+ faster than  
with traditional  
Power Analysis tool



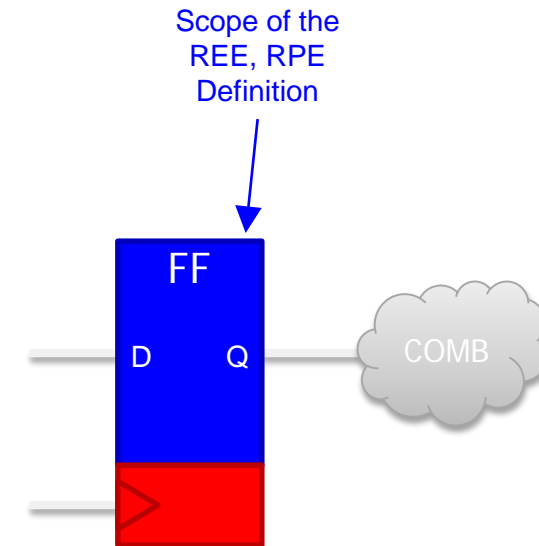
**Hot Spots Identification**  
Optimization targets, Local IR-Drop,...

**Power Trends**  
Compare activity plots across RTL drops

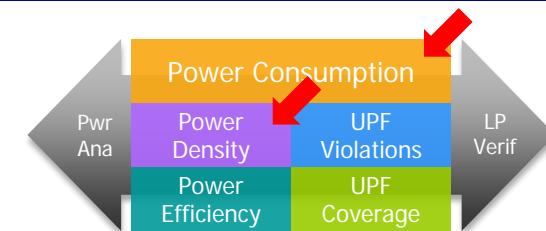
# Activity Plot Concepts & Definitions



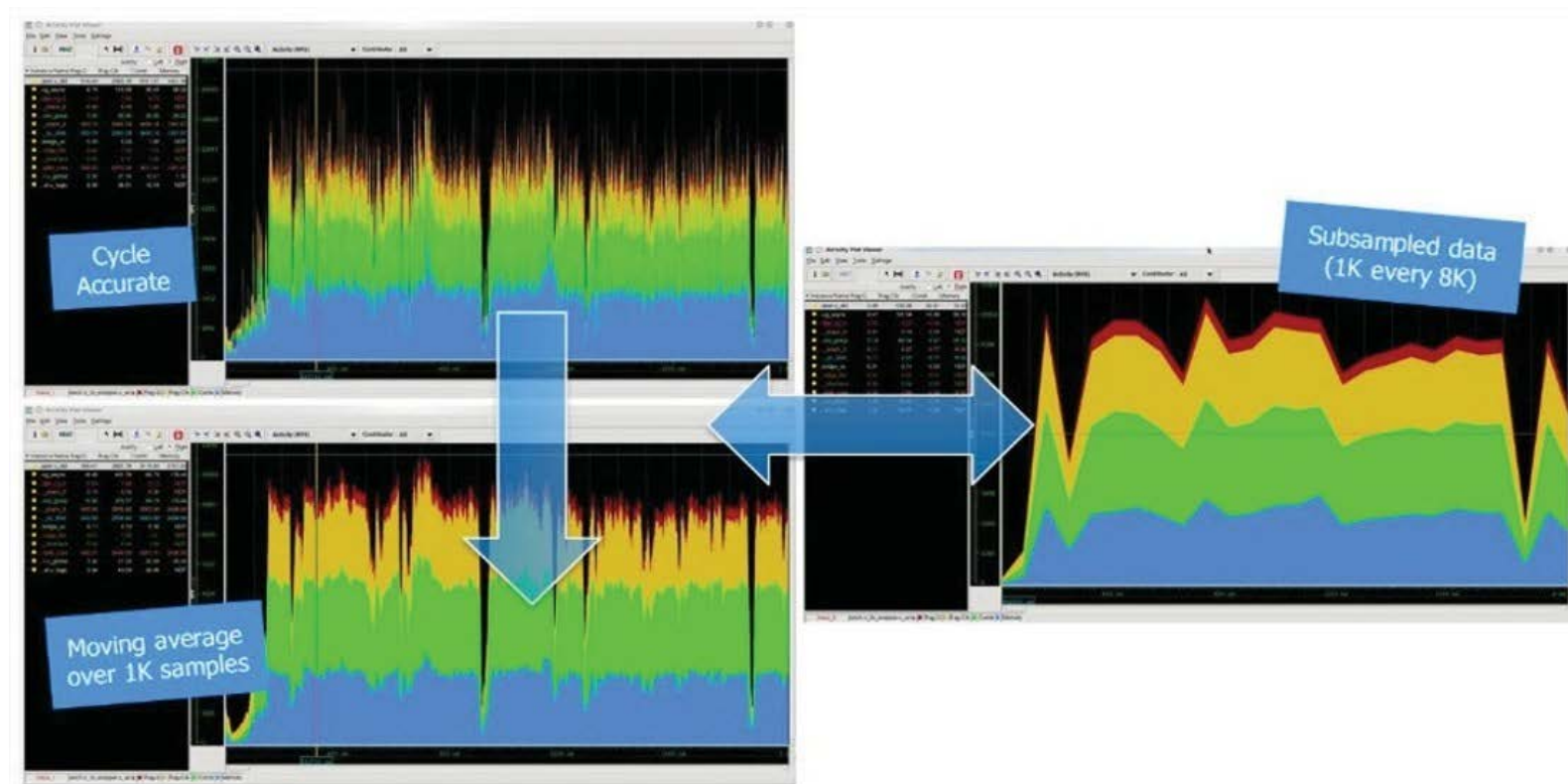
- Activity Plot is not a power estimation but a power analysis tool (using modeling approach)
  - No attempt at generating Watt numbers
- All metrics expressed with regard to a reference that is chosen to be a FF
  - Register Energy Equivalent (REE)
    - Typical **energy** consumed by a register data toggle
  - Register Power Equivalent (RPE)
    - Typical **power** consumed by a register whose data toggles every *ns*
  - Register Area Equivalent (RAE)
    - Typical register **area**
- Solution to provide viable approximation of
  - Power estimation metrics
    - Energy <-> Activity Events Plot (REE)
    - Power <-> Activity Plot (RPE)
    - Power Density <-> Activity Density Plot (RPE/RAE)



# Activity Plot Use Model

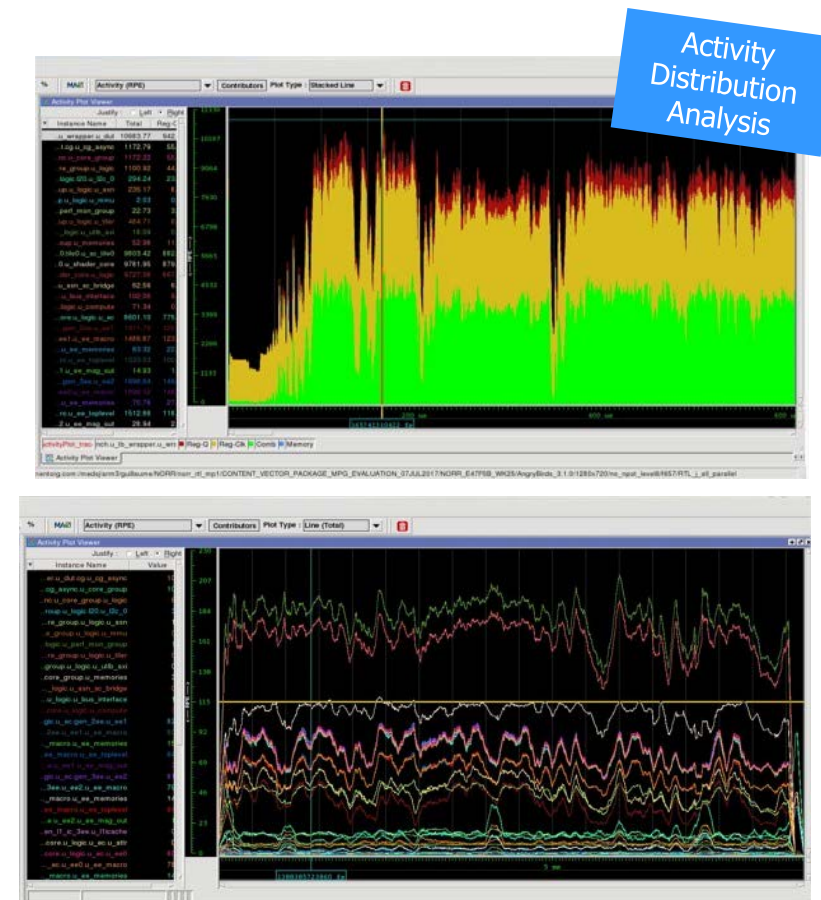
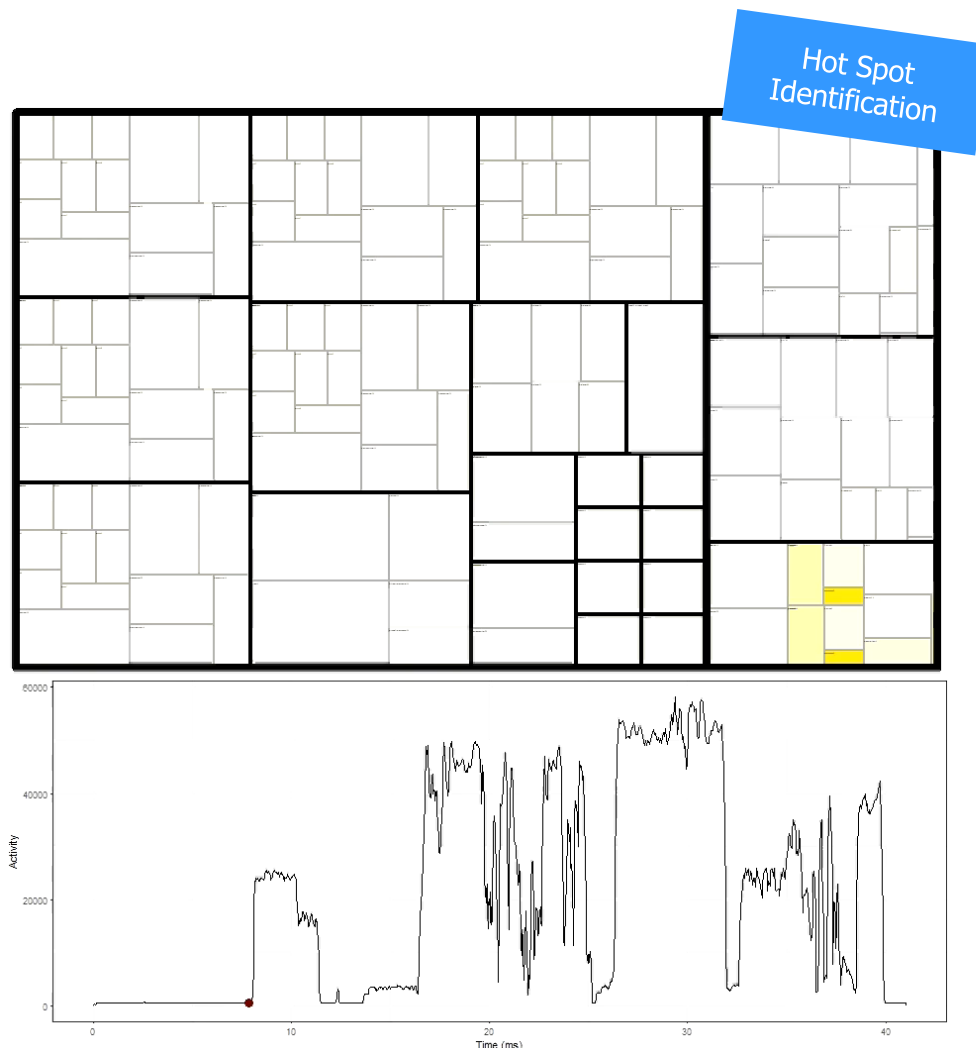
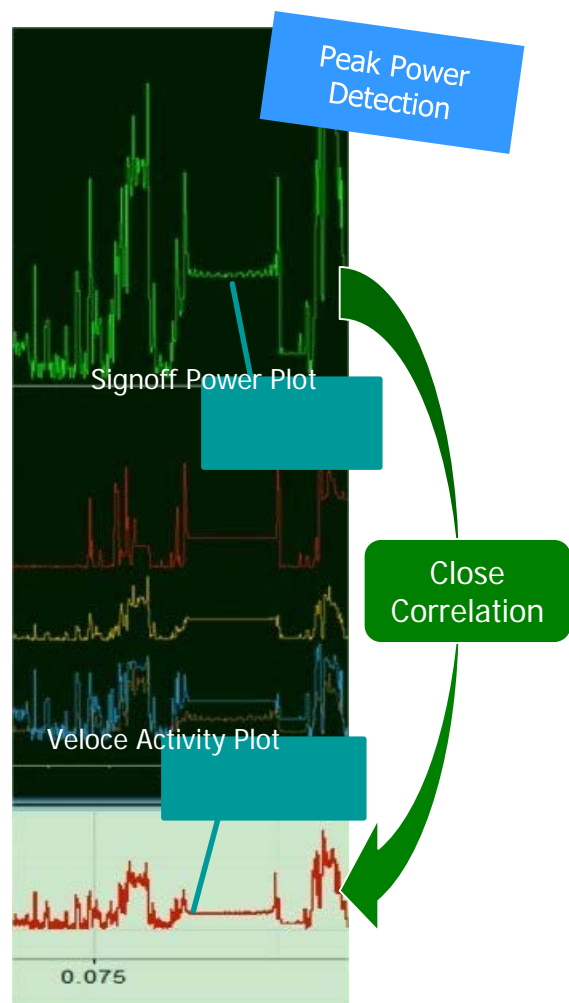
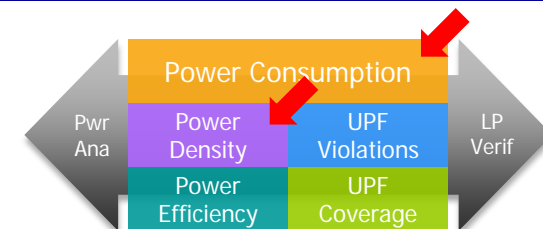


- All contributors broken down (Reg-Q, Reg-Clk, Comb, Mem)
- Multiple views (Power, Energy, Power Density,...)
- Responsive environment to analyze data

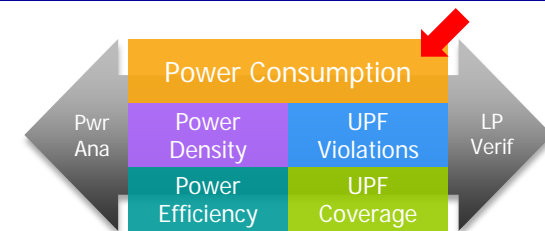




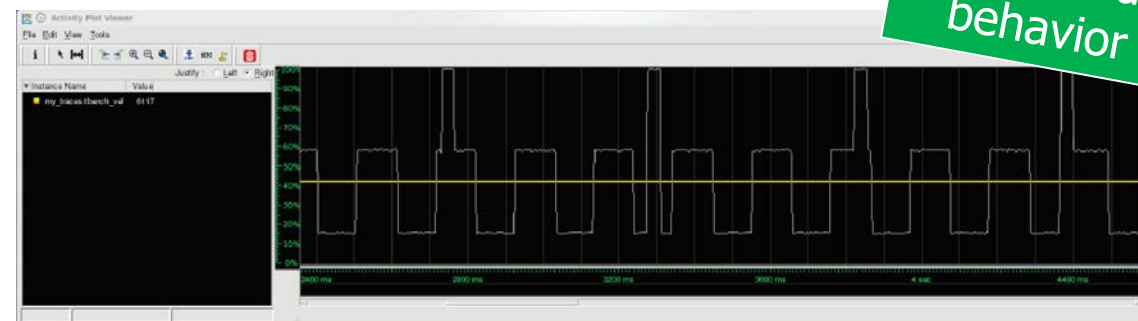
# Some Applications...



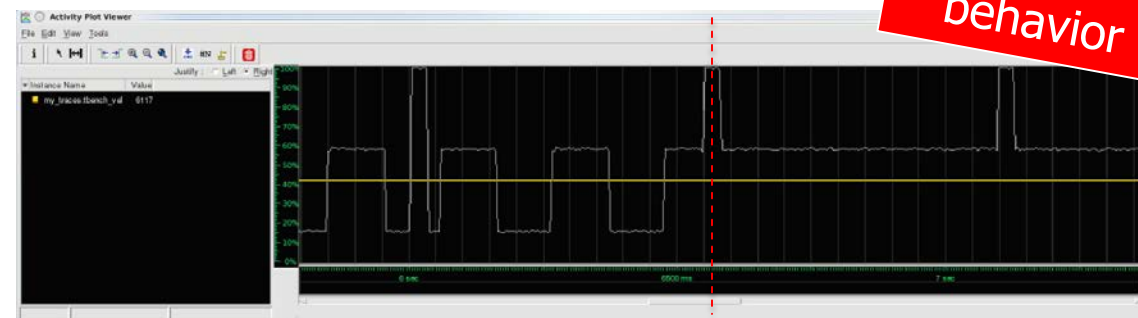
# System Activity Validation



- Arm-based system physical prototype showing excessive power
- Problem reproduced with Activity Plot
  - Periodicity of tasks accelerated to increase probability of failure
  - 2 processes using AXI peripheral A and A&B alternatively
  - Process using peripheral A remains **active** after a while
  - But why?

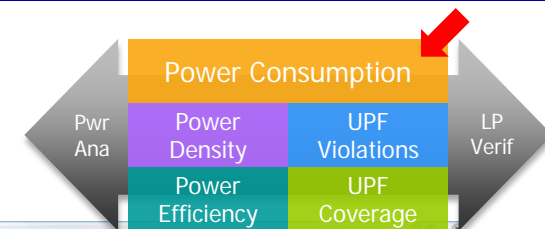


Expected behavior

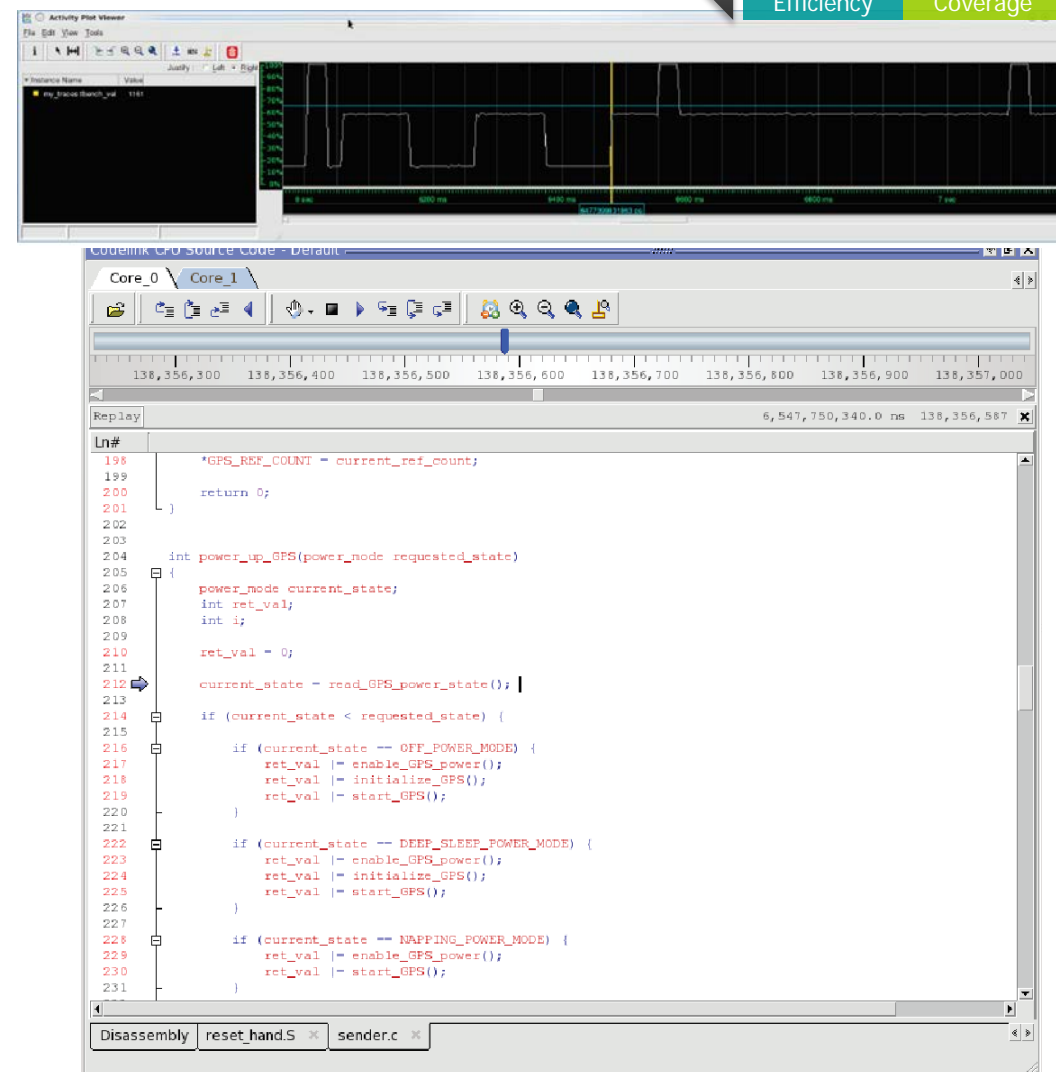


Faulty behavior

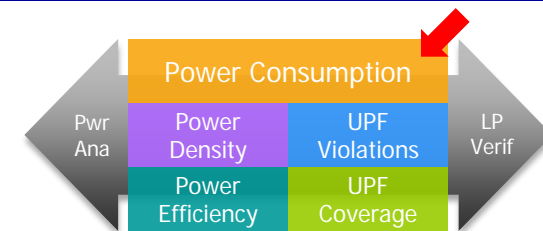
# Correlation of HW events with SW



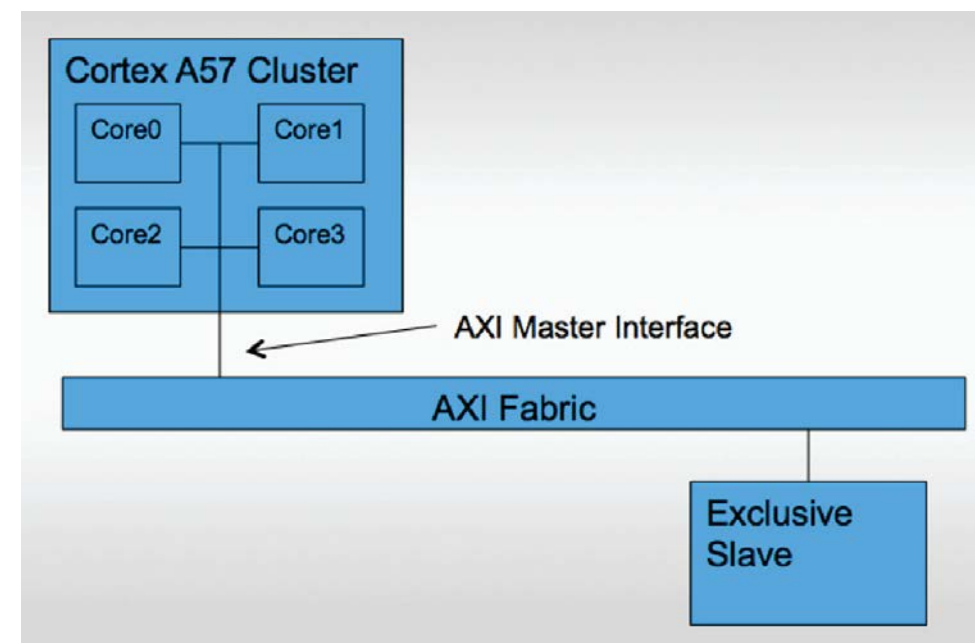
- Non-intrusive debug methodology using code source analyzer
- Traces the activity of the processors as they execute code
- Code source analyzer cursor was set to where the system should have switched off peripheral A
- Root cause of issue traced back to power controller



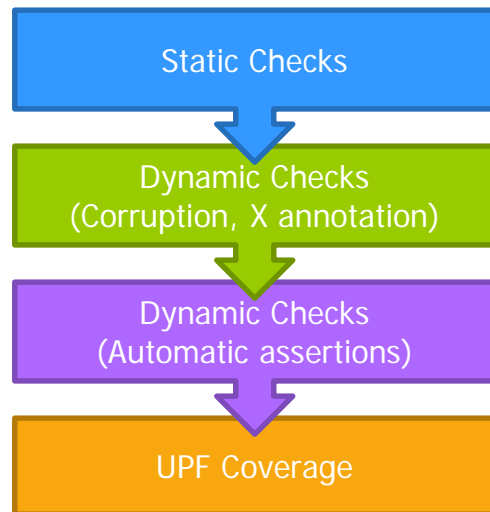
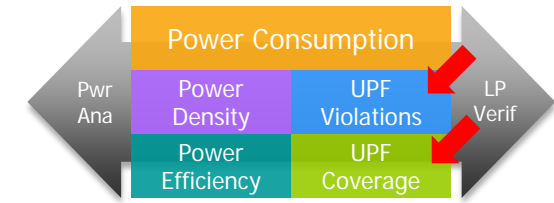
# Explanation of Issue



- 2 processes on two different cores trying to simultaneously turn off peripheral A
- Implementation was reliant on simple counter of active processes
  - Both processes read counter (answer = 2)
  - Both decrement to 1 and leave peripheral running
- Not just a simple race condition
  - SW actually uses Arm AXI exclusive access instructions
  - Pb is that all cores were identified with same AXI master bus port
- Fix consisted in added part of the transaction ID to the master identifier to truly differentiate originators



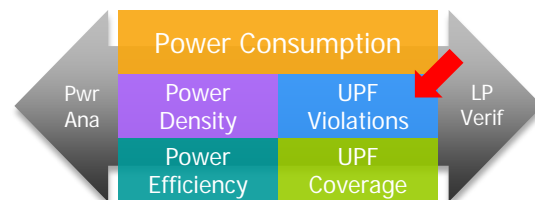
# Emulation-based Low-Power Verification



- Types of checks
  - Static, dynamic and coverage checks for realistic scenarios
  - Can address common case where power controller is implemented in SW
- UPF Support
  - UPF 2.x / UPF 3.0 support



# Need For Advanced Debug Capabilities



- Debug environment
  - UPF objects are available in the variable window
  - PA Domains to summarize UPF data
  - Waveform window displays supplies and power states, link to PathBrowser

Turn On Value Annotation  
View Objects For Entire Design  
View Object for particular Scope  
Add Signal to wave  
Showing State and Voltage  
Filters  
Showing Objects for particular scope and recursively below  
Waveform lock

UPF Object

Type	Name	Line	State, Voltage	Info
Power Domain	pd_aon	6	OFF->ON	HierPath: tb.top
Power Domain	pd_lv	18	ON	HierPath: tb.top
Creation Scope	top	1	-	
Supply Set	primary	26	-	
Supply	power	21	OFF, 0V	Object: tb.top.vdd_sw_lv
Supply	ground	10	OFF->ON, 0V	Object: tb.top.gnd_snet
Retention	pd_retention	41	-	
Retained Port	tb.top.ret	4	0	
Save Signal	tb.top.ret	4	0	
Save Condition	tb.top.ret	4	0	
Restore Cond	tb.top.ret	4	0	
Retention Co	tb.top.ret	4	0	
Supply Set	retention _	54	-	
pd_isolation	pd_isolation	50	-	
Isolated Port	tb.top.iso	4	0	
Isolation En	tb.top.iso	4	0	
Supply Set	isolation _	41	-	
Scopes/Extents	pd_hv	62	ON	
Power Domain	vdd_port	7	OFF->ON, 0V->1.2V	Object: tb.top.vdd_port
Supply Nets	gnd_port	8	OFF->ON, 0V	Object: tb.top.gnd_port
Port	vdd_lv	19	OFF->ON, 0V->1.2V	Object: tb.top.vdd_lv
Port	vdd_hv	63	OFF->ON, 0V->1.2V	Object: tb.top.vdd_hv
Port	pd_sw	30	OFF, 0V	Object: tb.top.pd_sw
Output Port	out_sw_pd	30	OFF, 0V	Object: tb.top.out_sw_pd
Input Ports	normal_wor...	30	-	ctrl_sw_pd
Control Ports	off_state	30	-	ctrl_sw_pd

Showing cone view of signal  
Hierarchical path of signal  
Clicking on any row in this window, shows corresponding line in upf file in source window

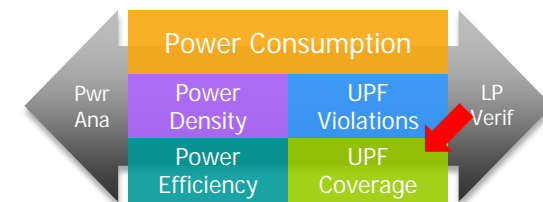
Selected Instance  
UPF file line gets selected to selected UPF object (variable window)

Variable Window

Waveform Window

View Declaration  
Add To Pathbrowser  
Search Selected Signal... Ctrl+F  
Count Events...  
Grid Events...  
Cut Ctrl+X

# UPF Coverage

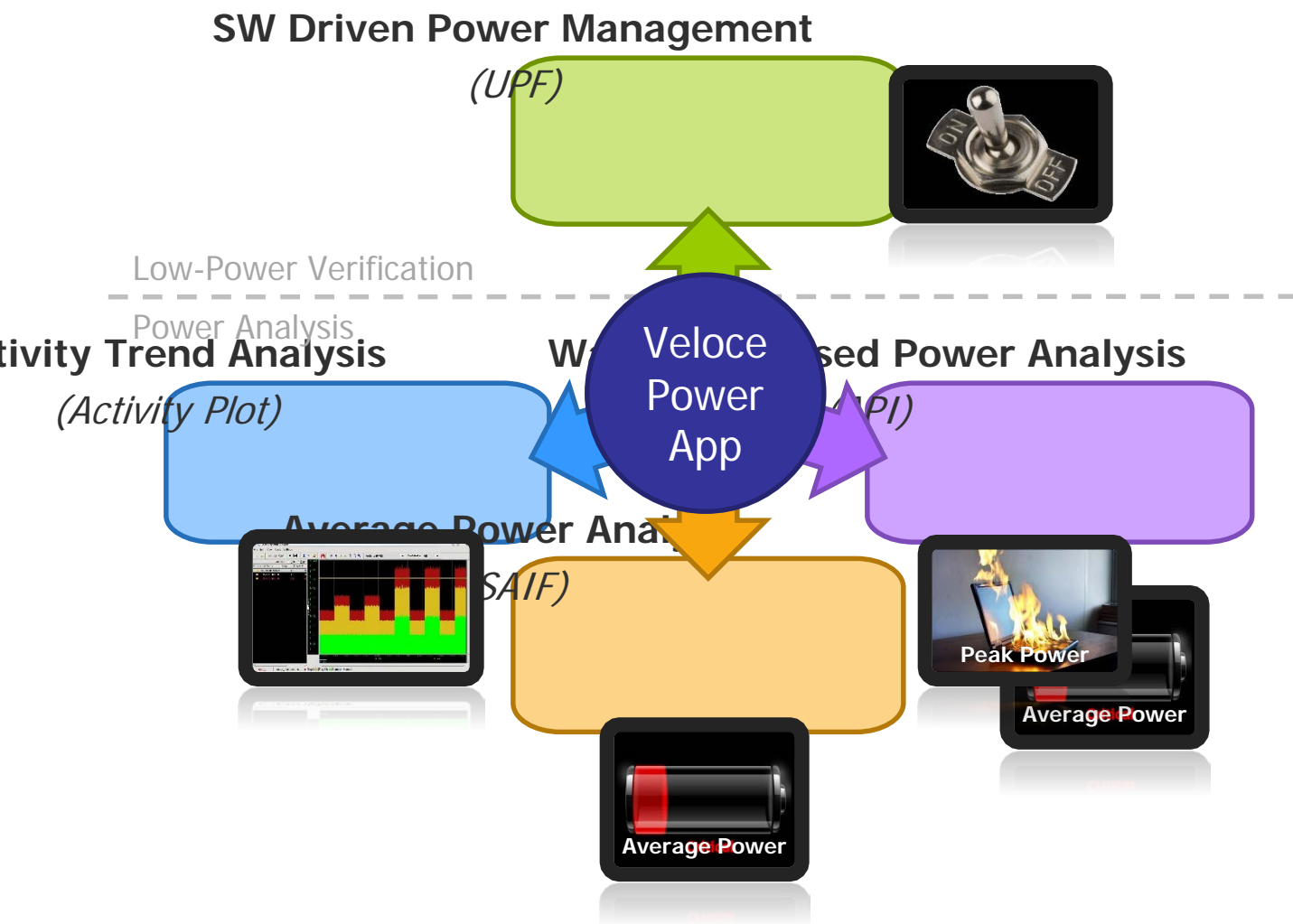


- Emulators Allow For Exhaustive Exploration of Power Transition Scenarios
  - Power State Coverage
    - Which states are actually reached and how often
  - Power State Transition Coverage
    - Which of the allowed transitions are actually covered

Coverage Stats

UPF OBJECT	Metric	Goal	Status
TYPE : SUPPLY SET /TESTBENCH/tb/TOP/top_PD.primary			
SUPPLY SET coverage instance cov6	50.0%	100	Uncovered
Power State ON	50.0%	100	Uncovered
bin ACTIVE	100.0%	100	Covered
Power State OFF	2	1	Covered
bin ACTIVE	0.0%	100	ZERO
Power State DEFAULT_NORMAL	0	1	ZERO
bin ACTIVE	100.0%	100	Covered
Power State DEFAULT_CORRUPT	2	1	Covered
bin ACTIVE	0.0%	100	ZERO
	0	1	ZERO
TYPE : SUPPLY SET /TESTBENCH/tb/TOP/top_PD.primary			
SUPPLY SET coverage instance cov7	16.6%	100	Uncovered
Power State ON_to_OFF	16.6%	100	Uncovered
bin ACTIVE	0.0%	100	ZERO
Power State ON_to_DEFAULT_NORMAL	0	1	ZERO
bin ACTIVE	100.0%	100	Covered
Power State ON_to_DEFAULT_CORRUPT	31	1	Covered
bin ACTIVE	0.0%	100	ZERO
Power State OFF_to_ON	0	1	ZERO
bin ACTIVE	0.0%	100	ZERO
Power State OFF_to_DEFAULT	0	1	ZERO
bin ACTIVE	0.0%	100	ZERO
Power State OFF_to_DEFAULT_CORRUPT	0	1	ZERO
bin ACTIVE	0.0%	100	ZERO
Power State DEFAULT_NORMAL_to_ON	0	1	ZERO
bin ACTIVE	100.0%	100	Covered
Power State DEFAULT_NORMAL_to_OFF	31	1	Covered
bin ACTIVE	0.0%	100	ZERO
Power State DEFAULT_NORMAL_to_DEFAULT_CORRUPT	0	1	ZERO
bin ACTIVE	0.0%	100	ZERO
Power State DEFAULT_CORRUPT_to_ON	0	1	ZERO
bin ACTIVE	0.0%	100	ZERO
Power State DEFAULT_CORRUPT_to_OFF	0	1	ZERO
bin ACTIVE	0.0%	100	ZERO
Power State DEFAULT_CORRUPT_to_DEFAULT_NORMAL	0	1	ZERO
bin ACTIVE	0	1	ZERO

# Veloce Power App



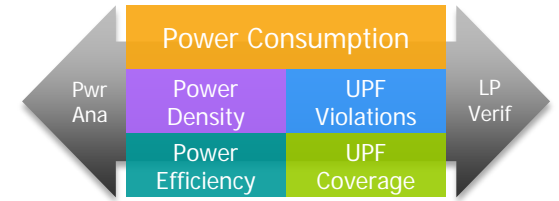
## Value Proposition

- Capacity & Performance
  - Large SoC handling (RTL/Gate), real applications
  - Orders of magnitude faster than simulation & power tools
- Low-Power Verification at SoC level
  - HW or even SW-based power controller
- Activity Trend Analysis
  - Over time and across scenarios
  - Identification of realistic peaks and hotspots
- Realistic Power Vectors Generation
  - For estimation and reduction



# Conclusion

- Emulation to generate **representative system payloads**
- Quick estimation of **power profile** with activity plots
- Ability to **dump power vectors** for power analysis tools
- **Vector streaming** without need for intermediary files
- Correlation of **hardware events with software**
- For more info download whitepapers on Mentor website ([www.mentor.com](http://www.mentor.com))
  - “Using Emulation for Meaningful Power Analysis”
  - “System Activity Validation”



**VELOCE**  
STRATO

# Thank you