2019 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES

Utilizing Technology Implementation Data in blended hardware/software power optimization.

Theodore Wilson, Wilson Consulting, Vancouver, Canada BC
Frank Schirrmeister, Cadence Design Systems, San Jose, CA

***Abstract*- This presentation will illustrate methodologies for dynamic power analysis and UPF verification, linking software executing on the processors of a SoC to silicon technology. We will illustrate the productivity of emulation in power analysis, and note critical aspects of handling design activity files. Careful production and archiving of design activity files enables feeding proven vectors to logic synthesis, blended firmware/hardware resolution of power problems and sign off on dynamic power consumption under key work loads. Special attention will be given to validating design activity provided by emulation and optimizing the application of .lib power coefficients to design activity via parallelization.**

I. Introduction

Software development continues to cause profound changes for design of electronic components. Hardware-software optimizations for power, performance and area have been known trade-offs in the past, now firmware-initiated power-noise and thermal effects are forcing teams to confirm pre-silicon that complex software/hardware interaction does not compromise device operation with excessive noise, voltage droop or power draw either in end use or under any conceivable use. These pre-silicon confirmations are not amenable to statistical toggle estimates or similar past techniques but require end-use or corner case firmware and hardware work loads.

Practical pre-silicon sign-off that a SOC product meets power requirements requires all of emulation capture of design activity under key workloads, and the power consequence defined in liberty files. Emulation capture is required because firmware practically precludes simulation. Design activity under key work loads is required to confirm power under end-use does not exceed design envelope nor introduce noise or dI/dt problems. Liberty file data is required to map design activity captured on an emulator to power consequence of that activity. Critical thermal effects will become visible only under real software workloads on the design, with for example AnTuTu benchmarks dropping when adding more processors that eventually have to be throttled, sets of blocks coming out of low power states causing voltage drop and timing failures or low-latency blocks causing voltage drops and timing failures as the blocks track with rapid increases in workload. None of these critical real world cases can be identified without captures of hardware/software interaction, nor can the firmware resolution of these problems be confirmed pre-silicon without showing the corrected system behavior under revised firmware.

II. Low Power challenges in Chip and System Design

Several challenges exist in low power optimization for chip and system design, based on the long known challenge of balancing between the intent to do power analysis and optimization as early as possible and the availability of accurate enough power information to actually perform power analysis. As described in [2], the earlier power optimization can be performed during the design flow, the bigger the impact on overall power consumption will be.

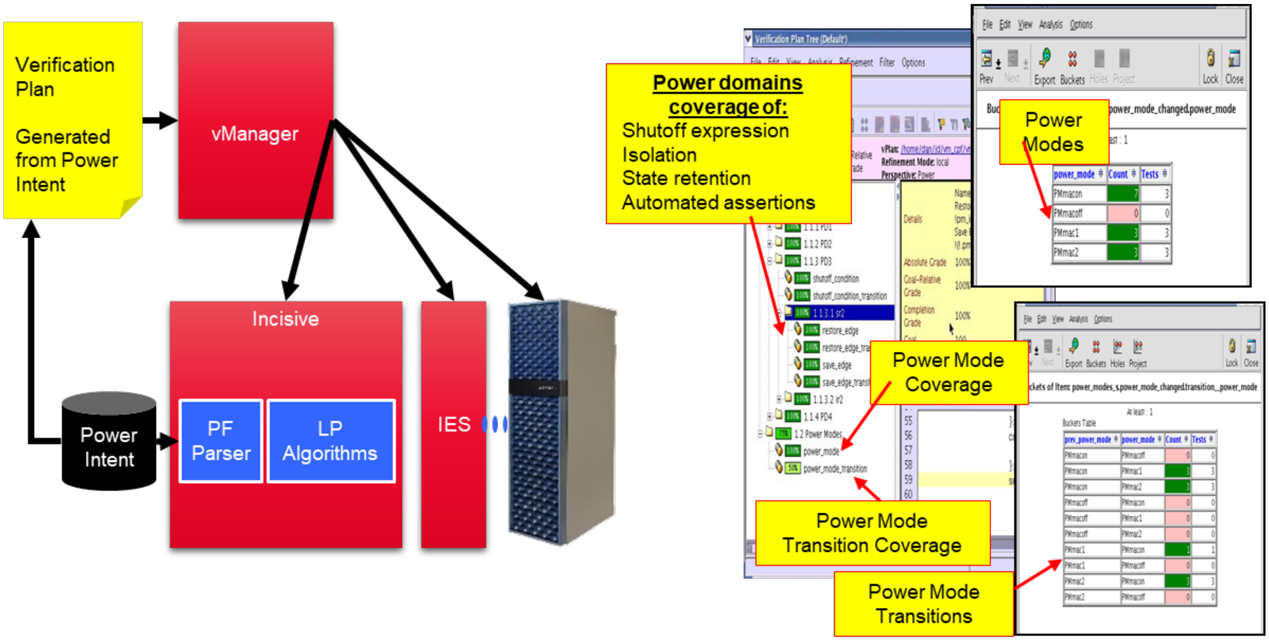


Figure 1 - Low Power Verification using IEEE 1601 UPF/CPF

For low power verification, the industry has successfully standardized on IEEE 1801 low power formats to capture the low power intent. Figure 1 shows a typical intent based power verification flow. Based on the power intent captures using IEEE 1801 compatible information, a verification plan is created that is executed in simulation and emulation. Today's complex chip designs can contain 10s if no 100s of power domains, each of which with different power states and transitions amongst them.

Today's verification solutions for emulation and simulation – like Cadence Xcelium™ and Palladium® - contain the ability to automatically model based on the power intent defined in IEEE 1801 compliant descriptions the effects of switching between different power modes of domains. Application of UPF gives users accurate estimations of the functional impact of switching off power domains – often controlled by software algorithms – and allows users to verify the effects of falling into or resuming from deep sleep states, in which for example memories may have to be re-initialized. SAIF or similar files flowing from this work to power analysis tools can directly show power consequence of utilizing power domains under critical or real world use cases.

Consequently, support for UPF in emulation lets users confirm that production firmware control of power states meets design specifications and does not exceed capabilities. UPF+RTL+firmware in emulation enables early discovery and resolution of potential power problems and is a good candidate for portable stimulus as well.

Aside from power state changes, other equally critical impacts on power consumption can be caused by the dynamic power consequential of changing workloads. As shown in [2], adding/correcting logic (such as additional multipliers or corrected clock gates) can be shown to be beneficial for power consumption if the data path can be optimized to reduce the total toggling and in specific unnecessary toggling. Today these problems are identified and resolved iterating through RTL, high-level synthesis, real-world data capture, power analysis tools and RTL updates.. Figure 2 shows an example flow to iteratively analyze and optimize dynamic power consumption.

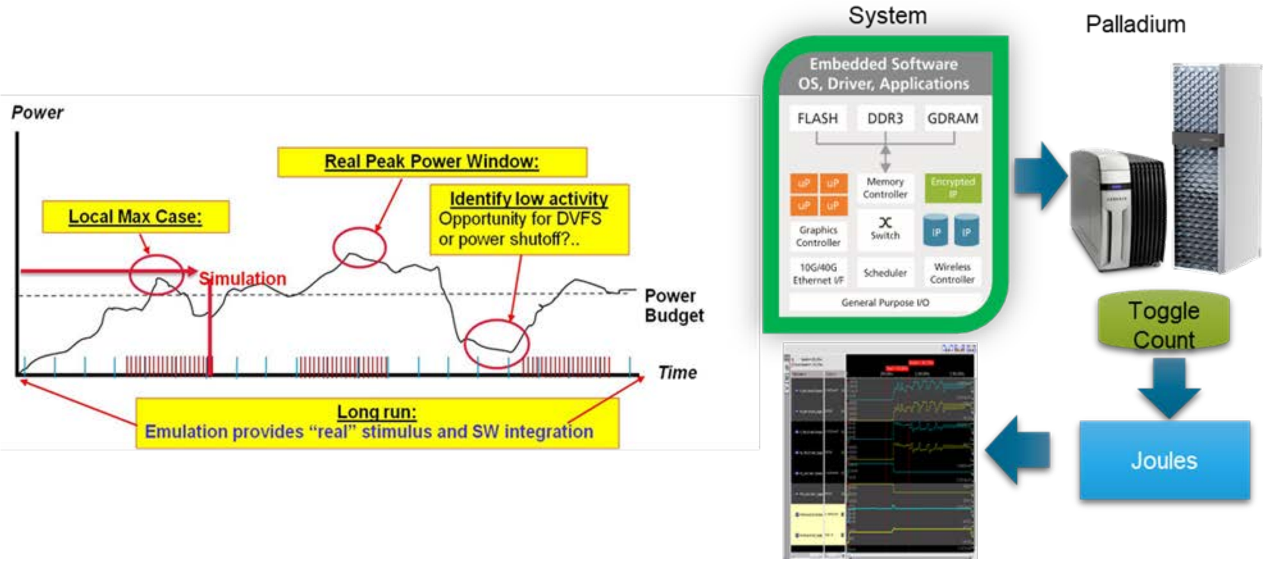


Figure 2 – Dynamic Low Power Analysis using emulation and power estimation based on .lib information

As illustrated in Figure 2, it is critical for dynamic power analysis to see long traces of end-use test cases such that the team can adequately focus effort on the simplest resolutions and most critical issues.

Figure 2 shows a local maxima problem not detected in simulation but resolved by emulation. This figure is however is really illustrative of a more fundamental problem-- that power analysis needs to see sufficient full, real world cases to even discover or rank what is critical or possible to redress or correct.

The Palladium/Proteum/Joules product family to some extent uniquely combine elements critical to rapid completion/resolution of power sign-off pre and post-silicon: rapid firmware iteration on Proteum, immediate re-execution and massive data capture and parallel decompression on Palladium, quick recompile under small RTL changes, easy swaps between RTL and gate images on Palladium without emulator resource impact, clocking of RTL or gates design images at arbitrary clock rates without regard to timing closure, and offline power analysis of design activity via Cadence Joules™. Joules can play two roles in this ecosystem-- internal-to-Joules synthesis and mapping of RTL captures to gate level activity and power, as well as the more direct mapping of gate level captures to power. The two modes of Joules let teams flexibly manage two risks: availability of product netlists for emulation vs RTL to gates activity mapping and power inaccuracies. The type of risk management facilitated by Joules' two modes of operation can be critical to completing power analysis under schedule pressure.

Real world cases are instrumental to weighting possible design and firmware changes given comparative cost to implement and days-of-use for any particular case when the product is in the customer's hands. Without real-world data captured under executing firmware it is not possible to find nor redress critical cases nor prioritize between non-critical cases nor confirm resolution of any such cases by updated firmware and hardware. Emulation systems designed for rapid compile, rapid design upload, sophisticated triggering, rapid data offload and parallel trace decompression appear to be irreplaceable to the power signoff of SOCs.

III. A Design Flow for Dynamic Power Analysis

The effectiveness of a power analysis flow can be assessed via the following questions--

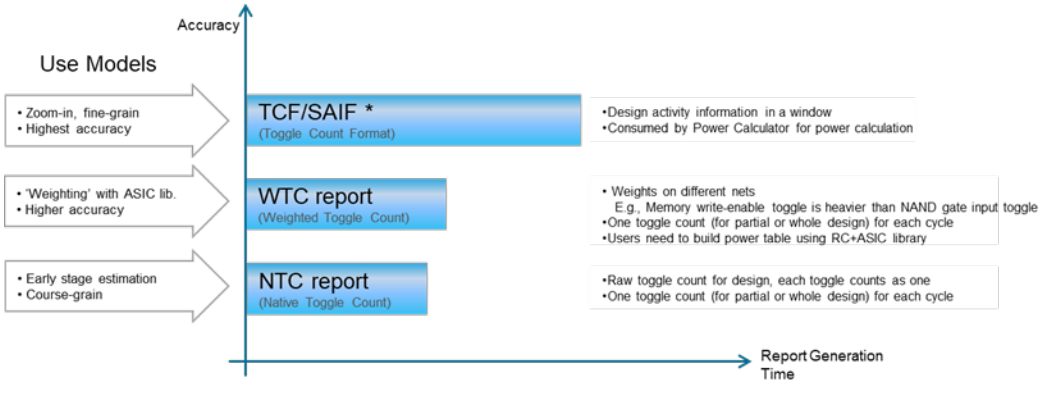
- Are any of the results actionable?
- *Do results correlate to lab, and are product deficiencies and their possible resolution clearly enumerated?*
- What is the minimum cost?
- *What do you need in place to obtain results? How many teams & deliverables and at what points in the program?*
- What is minimum time?
- *What are the rate limiting steps of the flow? Are any of these rate-limiting steps intrinsic or just implementation?*

The approach of using Palladium & Proteum with end-product firmware executing on the emulation system is a solid path to obtaining reproducible & representative cases for power analysis. Palladium offers Full Vision, Deep Trace capture of design activity with sophisticated triggers, and this in turn ensures the entire design over reproducable windows of time an be completely and efficiently captured for offline analysis.

Palladium also enables flexible swapping between RTL and gates images of the design at no net change of emulator resources, and Palladium offers the ability to take early synthesis results at low clock rates to full clock rates. Joules offers an ability to perform power estimation with either gate level or RTL data captures which allows the emulation and power analysis team to trade high accuracy against netlist availability on a weekly basis. Proteum provides an offload capability critical to developing and confirming end-use cases relevant to power analysis, ensuring Palladium session time is fruitful.

Consequently, this set of product features enables the blend of Proteum, Palladium and Joules to together answer the questions of actionable results provided at controlled incremental cost.

An effective power analysis flow lets teams trade between time it takes to create the data sets, the size of the data sets and data set accuracy. Related to this, an effective flow lets users carefully produce and store expensive data sets at low compression/high accuracy (e.g. SST2 format) while specific power analysis tasks flexibly reduce size, accuracy, and scope to gain iteration speed and answer specific questions. Whether taking data from emulation or simulation, the primary captures of design activity should be full; with subsets in time and hierarchy, coarse or narrow time windows taken from these full data sets as an offline exercise that avoids repeated simulation or emulation sessions. Figure 3 illustrates relevant data formats, their accuracy and role.


Accuracy
Use Models
• Zoom-in, fine-grain
• Highest accuracy
• 'Weighting' with ASIC lib.
• Higher accuracy
• Early stage estimation
• Course-grain
TCF/SAIF *
(Toggle Count Format)
WTC report
(Weighted Toggle Count)
NTC report
(Native Toggle Count)
•Design activity information in a window
•Consumed by Power Calculator for power calculation
• Weights on different nets
E.g., Memory write-enable toggle is heavier than NAND gate input toggle
•One toggle count (for partial or whole design) for each cycle
•Users need to build power table using RC+ASIC library
•Raw toggle count for design, each toggle counts as one
•One toggle count (for partial or whole design) for each cycle
Report Generation Time

* Shown here is the relative time to generate TCF file compared to WTC or NTC. Additional analysis time required to run through power calculator

Figure 3 – Options for Toggle format generation

For early assessment to check at which point the highest and lowest design activity exists, Palladium offers the creation of Native Toggle Count (NTC) and Weighted Toggle Count reports. Native toggle reports automatically selects the N largest instances in the design hierarchy and checks all signals in an instance in design hierarchy. More toggles indicate more activity and with that more dynamic power. NTC reports can be created quickly and can enable users to identify peak activity for further analysis.

In contrast, Weighted Toggle Count (WTC) reports take longer to create. Each signal is assigned a different weight that is derived from *.lib definitions for ASIC cells. This is a capability of the emulation system, Palladium. The toggle number is modified by the weight accordingly. While the absolute value has no real meaning, WTC reports allow to compare relative power consumption across time and instances, resulting in a better representation of real power consumption.

The most accurate toggle information can be achieved using Toggle Count Format (TCF) reports using the SAIF switching format. The TCF represents design activity for a time window and for each signal in the design, including signal names, the number of toggles and the percentages of signals being in active state.

Balancing the different toggle techniques allows to implement a refinement methodology in which less accurate information is created quickly to identify hot spots, which then can be looked at in more detail using flows for Dynamic Power Analysis (DPA). Accurate flows for DPA split a time window into multiple segments to create one TCF for each segment, and use a Power Calculator such as Cadence Joules to compute the power for that segment and merge the results from individual segments to create an accurate power profile.

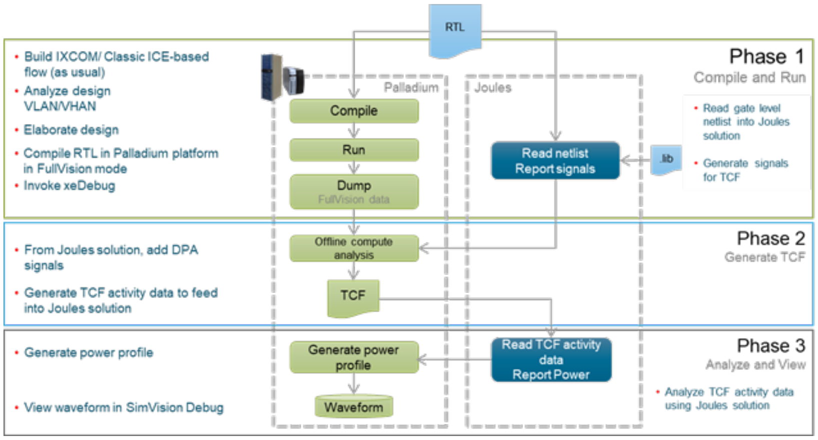

RTL
Phase 1
Compile and Run
• Build IXCOM/ Classic ICE-based flow (as usual)
• Analyze design VLAN/VHAN
• Elaborate design
• Compile RTL in Palladium platform in FullVision mode
• Invoke xeDebug
Palladium
Joules
Compile
Run
Dump
FullVision data
Read netlist
Report signals
lib
• Read gate level netlist into Joules solution
• Generate signals for TCF
Phase 2
Generate TCF
• From Joules solution, add DPA signals
• Generate TCF activity data to feed into Joules solution
Offline compute analysis
TCF
Phase 3
Analyze and View
• Generate power profile
• View waveform in SimVision Debug
Generate power profile
Waveform
Read TCF activity data
Report Power
• Analyze TCF activity data using Joules solution

Figure 4 – Palladium Dynamic Power Analysis Design Flow Details

Figure 4 illustrates key details of a proven, accurate, and flexible DPA flow.

In the first phase, the incoming RTL is built and elaborated for use in emulation. Parallel to this, early product synthesis and layout runs produce product netlists that can flexibly be placed on the emulator in place of RTL and teams develop first pass firmware and data capture triggers relevant to power analysis and sign off. Also in stage 1, replay simulation of captured traces is used to confirm critical design/system behavior relevant to power analysis such as DMA/s for power/performance scatter plots spanning the domain of design end-use.

In the second phase the design is loaded into a full data capture emulator (Palladium versus Proteum or FPGA in stage 1) with the previously developed firmware and wave capture triggers and consequently the required design activity is reliably captured and offloaded releasing the emulator for other tasks. Critical to note this image of the design can be a flexible blend of product gates netlists and RTL. Gates images will have higher power accuracy but may not be consistently available, Joules enables utilization of either. Palladium flexibly supports replacement of RTL and gates on a subsystem basis as suits the team. Regardless of gates/RTL design image or chosen final data formats, emulator traces are decompressed to SST2/SAIF to complete the second phase because raw emulator traces can only be converted while the specific emulator design image is also available, which typically cannot be ensured over long periods of time. SST2 is also good destination format for the end of stage 2 as this full data set can be further compressed to SAIF, VCD.gz or similar in parallel make driven flows at any time afterwards.

Replay simulation against SST2 is a critical technique to minimize emulator session time and to manage the fact that the set of critical design behavior statistics related to power cannot be known at the time of waveform capture--the set of key statistics and monitors needed to capture these is iteratively discovered after initial emulator sessions are complete. Key design behavior statistics must be used to identify or certify trace windows for power analysis. Either replay simulation or repeated emulator sessions must be used together with NTC and WTC reports to identify and confirm power windows. Replay of captured traces is uniquely capable of indicating if device power is correlated/dominated with (for example) processor instruction processing rates, DMA engine loading, or cache hit/miss rates. On a practical basis, replay is required to assess design power against actual required work--DMA/s, instructions per second and similar.

For the third stage, RTL power analysis can require waveforms of signals such as clock gate enable logic and this data is taken by the power tool, eg Joules, to accurately determine power. Aside from the example case of clock enables in RTL power analysis, power tools require only toggle counts per window to compute power per window. Earlier work with NTC and WTC reports refined the chosen windows for power analysis by the power tool, which annotates power information back onto the waveforms of the design.

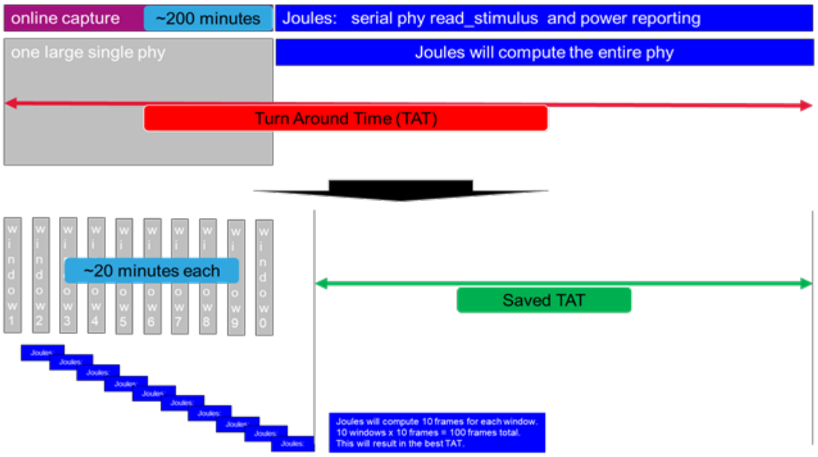

online capture
~200 minutes
Joules: serial phy read_stimulus and power reporting
one large single phy
Joules will compute the entire phy
Turn Around Time (TAT)
window1
window2
window3
window4
window5
window6
window7
window8
window9
window0
~20 minutes each
Saved TAT
Joules:
Joules will compute 10 frames for each window.
10 windows x 10 frames = 100 frames total.
This will result in the best TAT.

Figure 5 – Streaming Interface between Toggle creation and Power Estimation

IV. Optimizing the Interface between Collection of Toggle Data and Power Analysis

The design flow shown in Figure 4 illustrates the steps in each power analysis task, implying the project waterfall for this work. However, for specific longer runs, and the project as a whole, step-wise waterfall processes imply schedule risk and missed windows for timely feedback to RTL and firmware teams. To optimize for time to results, giving users insight as early as possible, the waterfall flow can be heavily optimized using Make, LSF and even Jenkins to ensure new results are delivered quickly and compute and emulator time related to any specific objective is rational. Power analysis sign-off in particular is a driver for new best-practice in compute management, and would benefit from reliable water-marks in trace data of specific design and firmware revisions in any trace thereby identifying and avoiding repeated work and stale results both. Even first-pass makefile automation of data file preparation, synthesis and power analysis runs presented in [4] dropped the time to a next incremental result from hours to minutes by avoiding repeated compute/reusing previous results.

It is critical to note that timely power analysis introduces an early continuous-integration internal customer for high level synthesis and layout teams which in the long run can improve their own processes but Joules again plays a critical role by managing delivery dates and reproducible results between emulation, synthesis and layout teams.

In [4] the author identifies throughput as a key metric in power analysis as the time from activity trace to actionable power reports. It is described how a design team has focused on using Palladium emulation and Protium prototyping to capture real world workloads and assess a SOC and firmware for fitness-to-end-use. Emulation became irreplaceable for fitness-to-end-use power analysis with emulation clocks at about 1 MHz, natural support for end-use SOC workloads and massive data capture combined with high speed offline decompression.

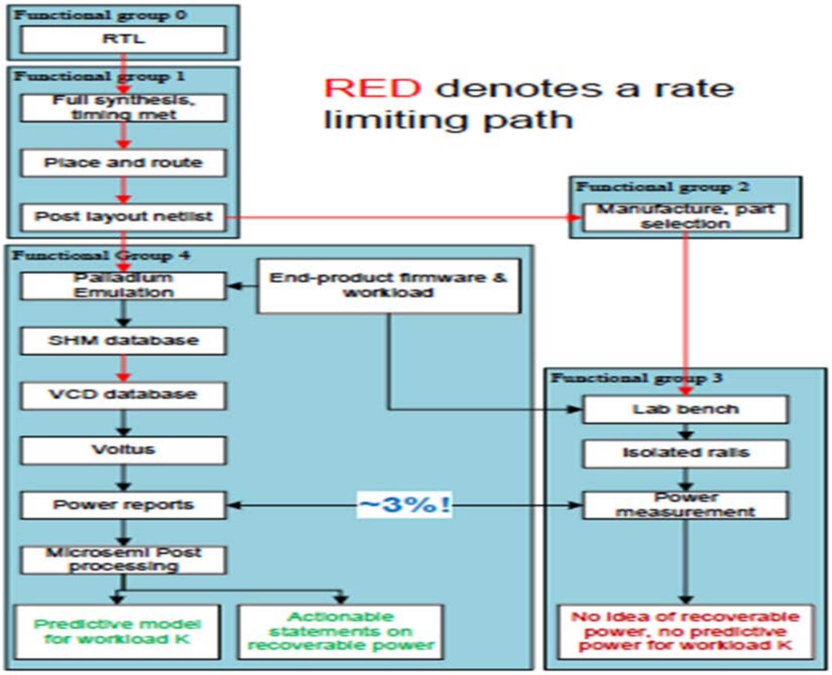

Functional group 0
RTL
Functional group 1
Full synthesis, timing met
Place and route
Post layout netlist
RED denotes a rate limiting path
Functional group 2
Manufacture, part selection
Functional Group 4
Palladium Emulation
End-product firmware & workload
SHM database
VCD database
Voltus
Power reports
~3%!
Microsemi Post processing
Predictive model for workload K
Actionable statements on recoverable power
Functional group 3
Lab bench
Isolated rails
Power measurement
No idea of recoverable power, no predictive power for workload K

Figure 6 – Microsemi Trial Emulation Power Analysis Flow

V. Results and Examples

Dynamic Power Analysis (DPA) flows can be very accurate as reported in [1], showing how early power optimization can be applied in emulation to application processor development, getting as close as 1.8% in accuracy compared to the actual silicon. [3] describes a case in which the full refinement methodology as described above has been applied and 1,000x performance improvement over RTL simulation was achieved, emulation enabled early system-level integration and software validation (Android/Linux) with an emulated application processor and correlated power consumption using realistic runtime environments and applications before silicon became available, greatly improving overall verification efficiency. Finally, the impact of using a streaming interface between toggle collection and power calculation, and key design considerations are outlined in [4].

Figure 6 above taken from [4] highlights the market value of an RTL base power estimation flow. Very long delays between functional groups dominated the productivity of the power analysis team during early trials. However the critical outcome was achieved-- very tight correlation between emulator and lab results using identical firmware.

A second take away from the trial exercise was that replay simulation and post processing spanning sweeps of voltage, clock frequency and traffic patterns are required to draw actionable conclusions on design imperfection under real world work loads. Replay simulation was central in this and critical.

The final take away was that the power analysis team was going to have to work more closely with the vendor to rapidly achieve adequate productivity. This engagement spanning several product teams at Cadence was exemplary from the point of view of the Microsemi team but also a critical learning exercise in collaboratively finding the simplest/next path forward. This engagement included a pivot from Voltus to Joules, refactoring of the Genus workflow, and rapid development of Joules workflow.

Figure 7 taken from [4] Illustrates the time consumed by the original Palladium power analysis flow and final Palladium power analysis flow employed at Microsemi. Critical to note is the implied make processing of data once an emulation session completes. Further gains are expected by helping placing more of the work flow under continuous integration including Genus and Innovus (synthesis and place-and-route) such that

1. *the incremental support cost of power analysis is as low as possible*
2. *any specific power investigation is working from pre-prepared netlists, firmware, palladium compiled designs, trigger setup files and similar inputs reducing time to data-capture*
3. *as often as possible data captures and power results are working from signed-off design and firmware revisions.*

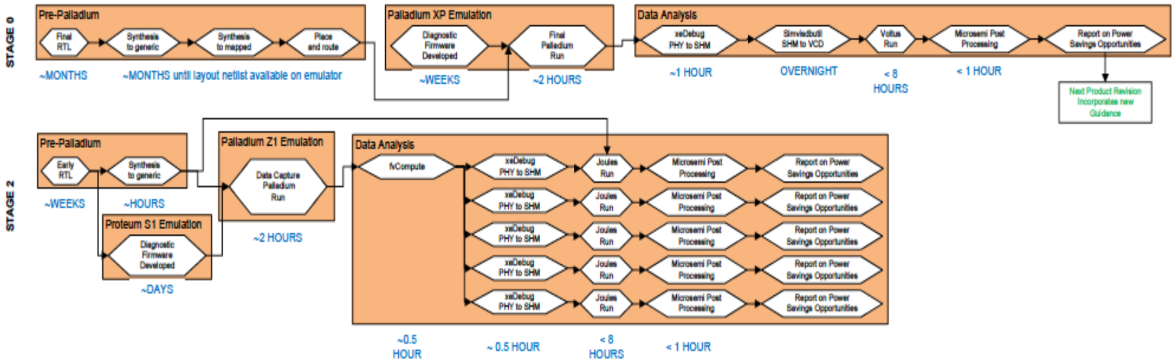


Figure 7 – Net Turn Rate of a Streaming Production Flow

VI. References

[1] Emulation Based Full Chip Level Low Power Validation at Pre-Silicon Stage, Woojoo Kim, Samsung, DVCON 2017

[2] Design for Low-Power at the Electronic System Level, Frank Schirrmeister, ChipVision Design Systems, http://bit.ly/2DSLrL3, downloaded 11/1/2018

[3] Optimizing ARM Based Designs for Low Power using Emulation, Frank Schirrmeister, Cadence System Design & Verification Community, http://bit.ly/2DGMYmH, downloaded 11/1/2018

[4] Rapid Turns with Palladium and Joules, Theodore Wilson, MicroSemi, CDNLive 2017 Proceedings, http://bit.ly/2J0Cspy, downloaded 11/1/2018