Low Power EDA on the Bleeding Edge

April 2015

Agenda

- **Power Planning in the Real World**
- **Managing Plan Misses**
  - Figuring out where you're at
  - Techniques to reduce power
- **Longer-Term: System Power Modeling @RTL**

Power Planning in the Real World

SPYGLASS® FROM ATRENTA

In Theory

- Instrumented TLM, high-level power-aware synthesis, ..
- Mostly research labs and very high volume uPs

In Practice

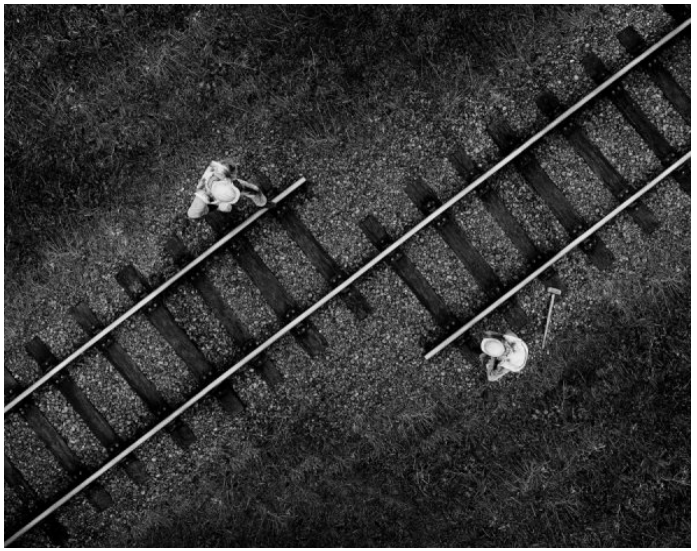
- 80% of designs derivative, lots of IP content with no HL models and limited characterization above the gate-level
- Excel spreadsheets – detail IP power params versus use-cases – estimate time spent in each mode to get power contribution / IP / case
- Low tech but still the dominant planning technique in the industry

IP	Idle power	Low perf power	Hi perf power	Use-case 1	Use case 2	…
A57						
PCIX						
DDR4						
SRAM						
MAC						
…						

The Plan and the Implementation..

SPYGLASS® FROM ATRENTA

- **The Best Laid plans…**
  - Plans are not always perfect
  - IP power models are not always completely accurate
  - Block designers trade off performance and power => some blocks overshoot budget
  - You find that logic you thought could be power-switched in fact has to be always-on => higher energy drain
  - So now you have to figure out what you've got
  - And how you might make up the shortfall

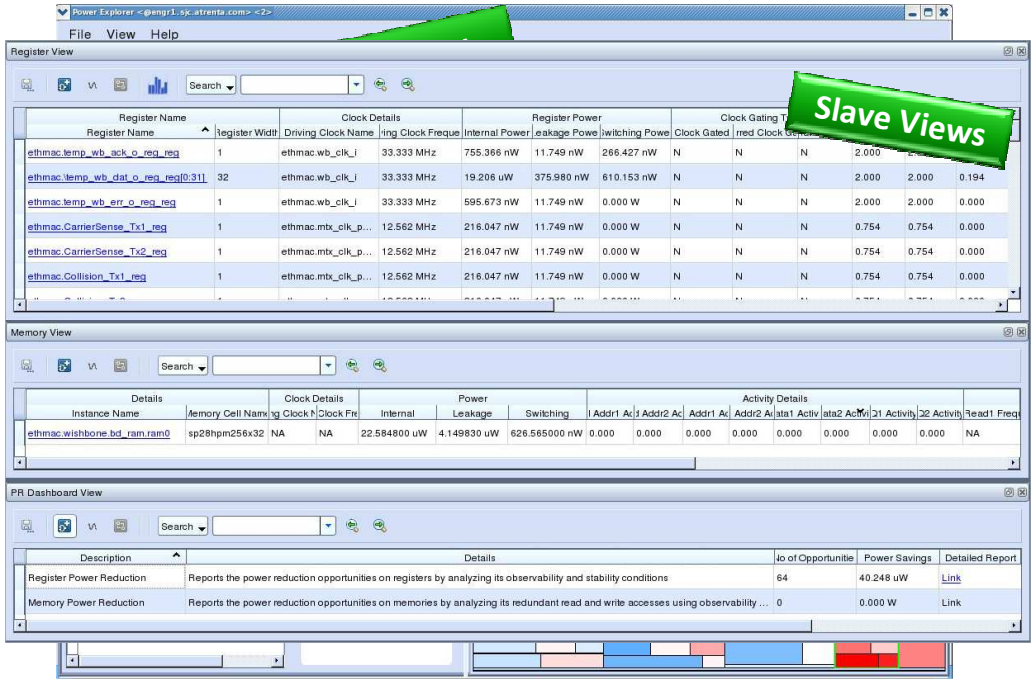


Agenda

- **Power Planning in the Real World**
- **Managing Plan Misses**
  - Figuring out where you're at
  - Techniques to reduce power
- **Longer-Term: System Power Modeling @RTL**

What Have I Got and What Can I Tweak?

- **Power Explorer used as Central Cockpit for Efficient Analysis**
  - Complete picture of SoC power
  - Re-analyze with same activity files, vary power parameters
  - Efficient what-if analysis across multiple scenarios
  - Experiment with reduction on other blocks: Vt mix, clock gating, lower voltage, power switch, …
  - What-if on reducing memory power, register power, …



Power Exploration Accuracy

Factors in Accuracy

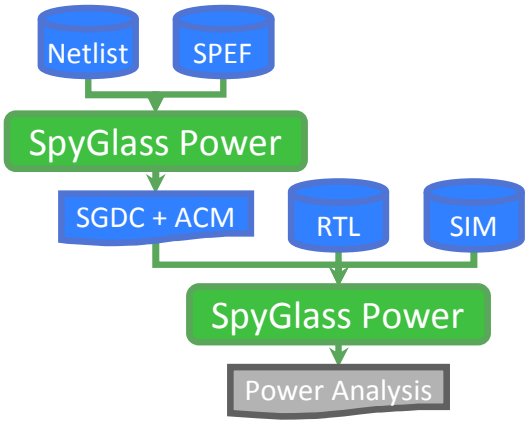
- Want to work with pre-implementation RTL, where can still optimize
- But need to model with post-implementation accuracy
- Requires calibration of multiple estimates against similar production RTL: Vt mixes, clock tree, capacitances, drives, etc

Correlation

- Intuitive set of structural models (not scaling factors!)
  - Advanced Capacitance Model (ACM), clock tree, drive distribution, Vth mix
- Models automatically set from netlist of same design class (same technology node, similar timing characteristics)
- Multi-variate regression analysis

Typical Deviation

- vs. reference power @ gate <15%



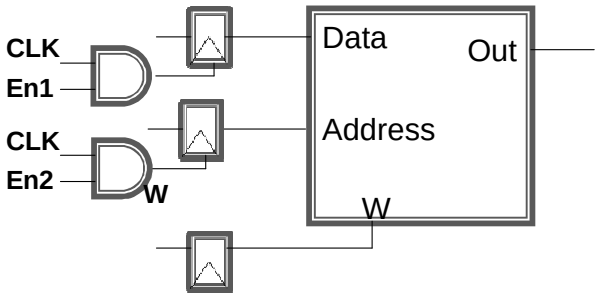
Agenda

- **Power Planning in the Real World**
- **Managing Plan Misses**
  - Figuring out where you are at
  - Techniques to reduce power
- **Longer-Term: System Power Modeling @RTL**

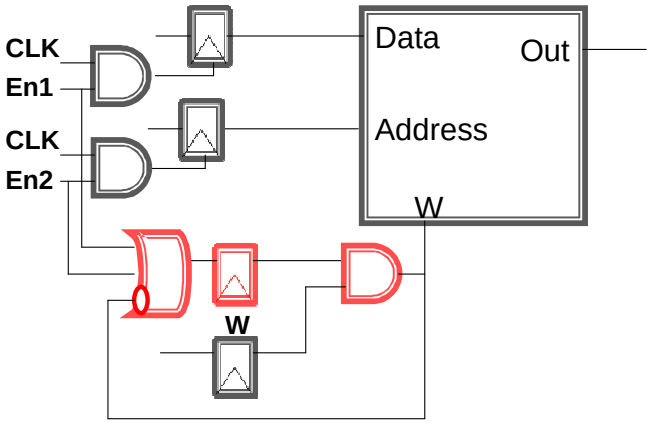
Memory Power Reduction

- **Redundant write (one example): If the Data and Write address are stable, then every write after the first one is redundant and can be removed**

Original implementation



Modified implementation

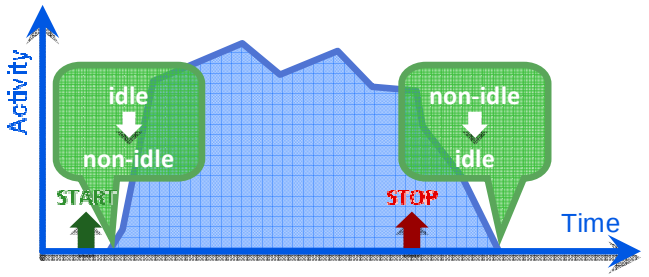


- **Memory power reduction typically has a more significant impact on power than register optimizations**

Additional Clock Gating

SPYGLASS® FROM ATRENTA

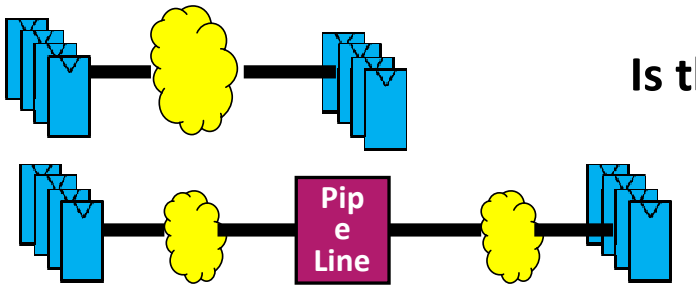
- **May be Additional Gating Opportunities Above Register Level**
  - Especially in legacy IP
  - But not always clear when you can and cannot gate
  - Empirical analysis a practical starting point, not requiring detailed understanding of IP architecture
- **Activity Trigger Detection**
  - Automated analysis of activity, RTL analysis and formal proving of derived triggers
  - Use to determine when can gate the clock (in idle)
  - Demonstrated to save ~30% on a video processor
  - Can also highlight potential power bugs (spikes)



Physically-Dependent Optimization

SPYGLASS FROM ATRENTA

- **Really Early-Stage but Maybe Late if you are Desperate..**
  - Normally optimize RTL for power, lose some of gain in timing closure
  - To not lose gains, need to manage power/timing tradeoffs
  - Below, with one stage for combo logic need big drivers to meet timing
  - Split the logic with a pipeline, drivers smaller but pipeline adds power
  - Which is better requires physically-aware power analysis



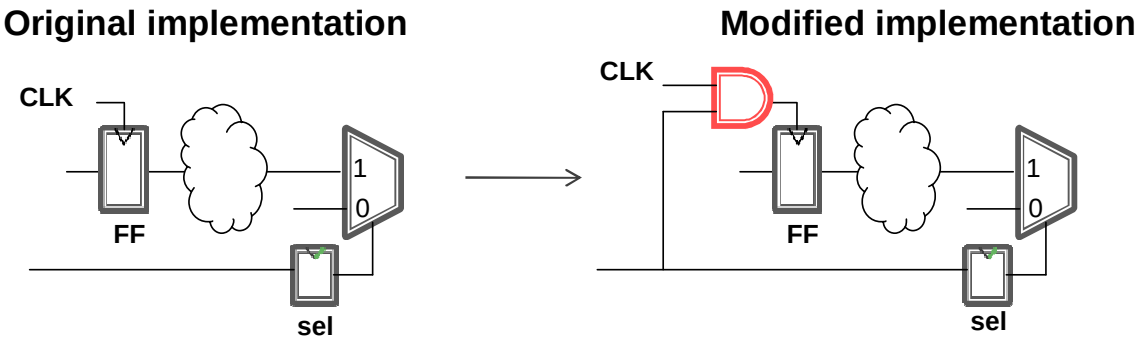
Is this lower power?

Or this?

Register-Level Optimization

Sequential Clock Gating

- Observability Don't-Care: State change blocked downstream, therefore can gate upstream
- In the same vein: Stability condition check, enable strengthening

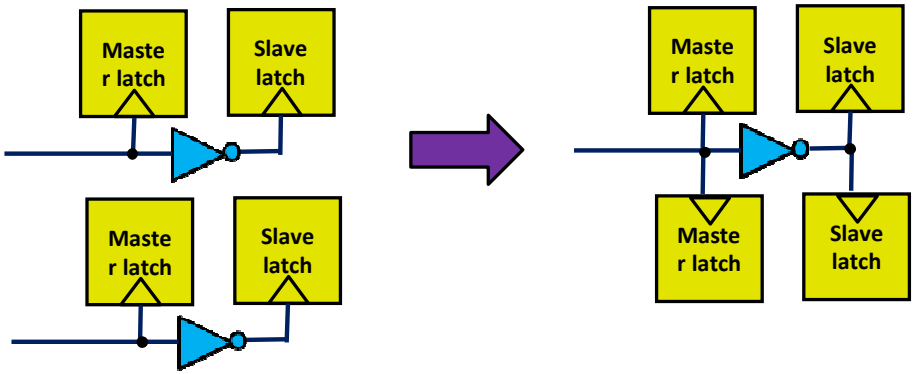


- Low-level savings – unlikely to select more than a few big hitters
- Logic proven by sequential equiv check, but also changes timing
- Can also mess up CDC – need to couple closely with CDC analysis

Fine-Tuning Flops

Multi-Bit Flops

- Combining single flops into dual, quad, … flop macros which use a common inverter between master and slave stages
- Saves one inverter for a dual flop, 3 for a quad flop, …
- Only makes sense for flops which will be physically close
  - On busses – possible to implement in RTL – dominant power saving
  - Otherwise close by chance – opportunistic saving based on placement



Agenda

SPYGLASS FROM ATRENTA

- **Power Planning in the Real World**
- **Managing Plan Misses**
  - Figuring out where you are at
  - Techniques to reduce power
- **Longer-Term: System Power Modeling @RTL**

System Power Modeling @RTL

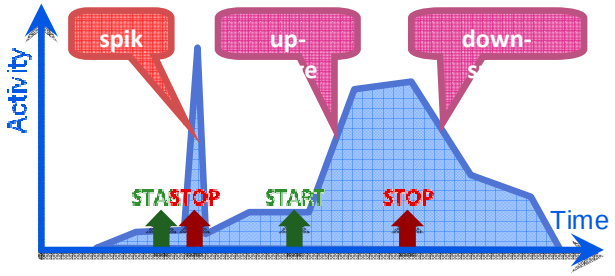
SPYGLASS® FROM ATRENTA

- **The Best Way to Model SW Loads is in Emulation**
  - Typically very fast, but becomes impossibly slow if dumping activity for power estimation
  - Today average power estimation based on software simulation
  - But software simulators can't model realistic loads to capture potential peak power problems
  - Solution has to be emulation-based, but requiring less dumped nodes for estimation
- **Power Model Abstraction**
  - Active standard development in (IEEE) P1801 SLP
  - An extension of UPF for abstracted power modeling
  - Defines how the models can be represented, but not of course how the models should be created

System Power Modeling @RTL

SPYGLASS® FROM ATRENTA

- **Default Would be to Create the Models Manually**
  - Reasonable approach for IP vendors
  - May be more challenging for internal IP – legacy, original developer long gone, many tweaks, …
- **An Alternative – Empirical Model Development per IP/Block**
  - Based again on activity trigger analysis
  - Here use to find triggers for and averaged power in major modes of operation
  - Still an R&D activity, but looks promising



Thank You!