DUMP, CONVERT AND REPLAY:
A TARGETED METHODOLOGY TO MITIGATING POWER SIMULATIONS EFFORT

Ioannis Savvidis, SoC Architect ASIC, ioannis.savvidis@ericsson.com

DAC Designer Track, June 2019

This poster presents an advanced VCD based input stimuli replay methodology that enables ASIC developers to:

- **Mitigate the effort required to bring up a power GLS by leveraging existing RTL simulation data**
- **Run targeted, early gate-level power analysis at any scope standalone and only for the critical time windows**
- **Increase productivity by facilitating better allocation of scarce verification resources**

Gate-level power analysis: Costly but necessary

- A typical functional verification environment is built around RTL model simulations (UVM/TLM based)
  - A netlist DUT cannot easily plug-and-play in such an environment
    - Scarce verification resources for GLS bring-up
  - No easy way to run subscopes standalone to save runtime and disk space
- Need a major reduction in effort and an increase in designer productivity for GL power analysis!
  - *Earlier, Easier, Targeted GLS* ✔

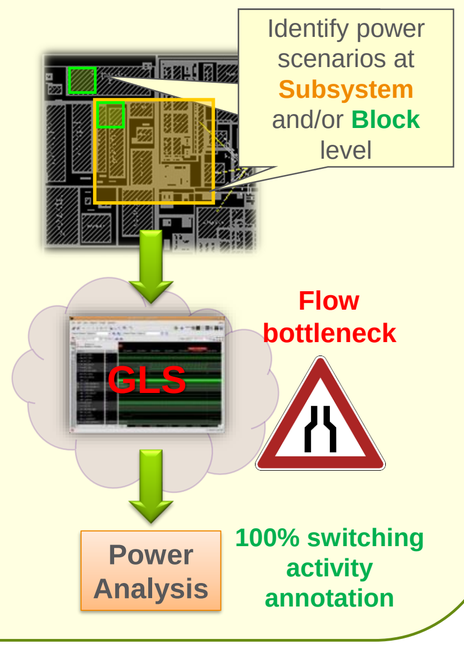
- Leverage the existing RTL simulation data at any scope
  - No special GLS environment bring-up effort
  - Decouple verifiers from power GLS to save resources
  - Facilitate reasonable runtimes and VCD dump sizes

This work

Prerequisites for an accurate analysis

- Acquire (realistic) simulation vectors
- Work with gate-level models
- Avoid relying on statistical activity propagation

Most of the effort and complexity in the power analysis flow is associated to the corresponding power simulations, especially for signoff accuracy analysis which necessitates generating a gate-level simulation (GLS) environment.



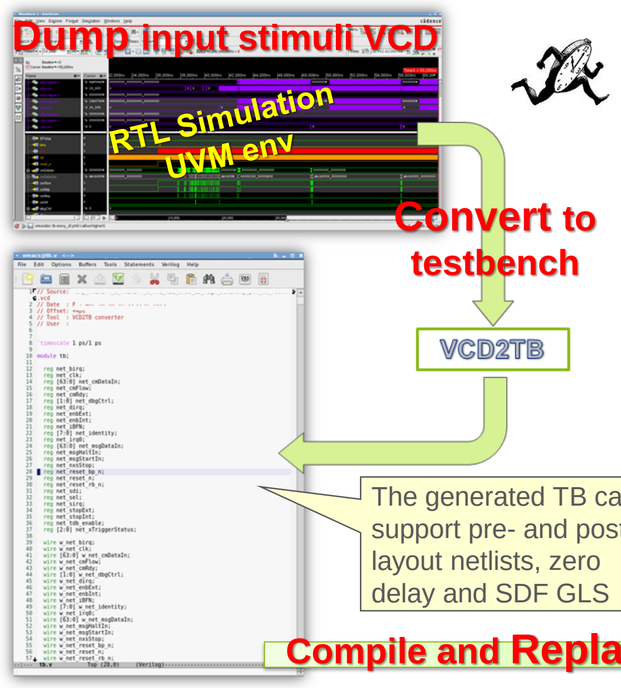
Main idea

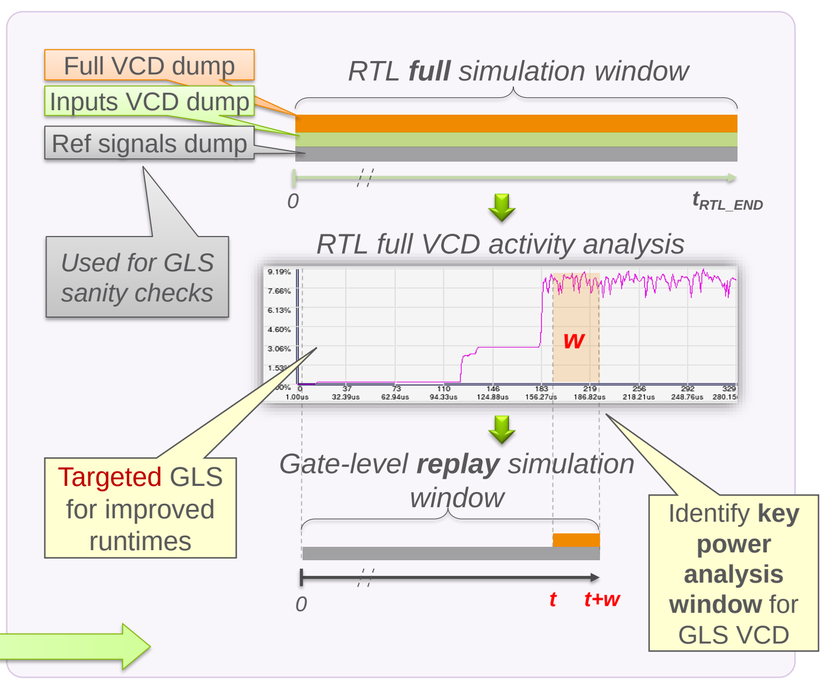
- Avoid incurring the expense of migrating a UVM environment to the GL by auto-generating a testbench that can recreate/replay the readily available RTL simulation data on partial netlists

About VCD2TB and future work

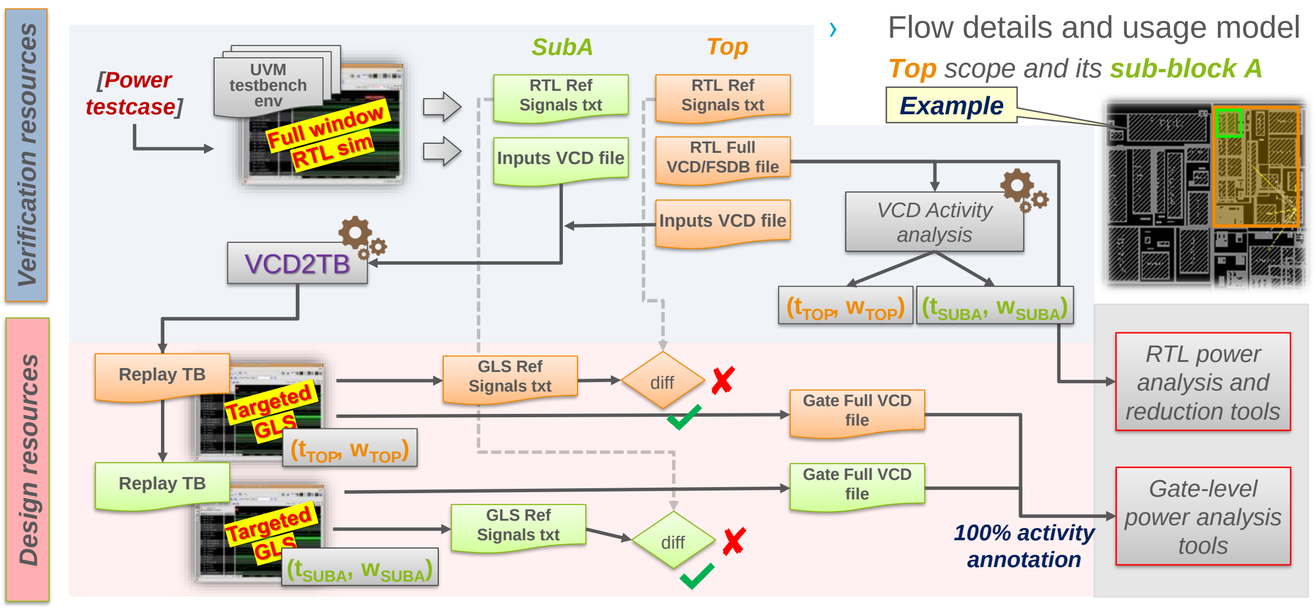
- VCD2TB is an in-house tool that converts an i/f VCD file into a Verilog testbench
- Post-GLS validation is done by comparing the RTL and GL reference signal values on every clock negedge
- ***Future work:*** C++ implementation of the VCD parser for improved VCD2TB conversion times

Power GLS from the get-go via RTL input stimuli replay





Added value to accelerated power analysis flows



› Flow details and usage model

Top** scope and its **sub-block A

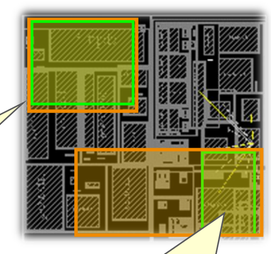
Example

Results and summary

- *Example A*: Memory controller subsystem IP
  - Design size is ~35 Million Gates, 1000+ memory instances
  - **RTL** and **GL** simulations are at the same scope
  - Pre-layout netlist, zero-delay GLS

runtime (min)	Compilation	Simulation	VCD Activity Analysis	VCD2TB Conversion
RTL(full)	14	45	34	72
GL(t,w)	48	39	*n/a*	*n/a*

Size (MB)	Inputs VCD	Full VCD
RTL(full)	70	12050
GL(t,w)	*n/a*	5060



Same scope simulations

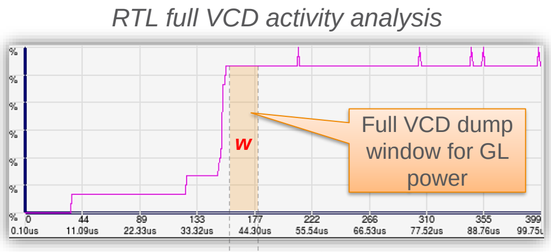
Different scope simulations

- *Example B*: Filtering and digital upconversion subsystem IP
  - Design size is ~20 Million Gates at netlist scope, ~145 Million Gates at RTL scope
  - **RTL** simulation at top scope and **GLS** at subsystem scope (standalone)
  - Pre-layout netlist, zero-delay GLS

runtime (min)	Compilation	Simulation	VCD Activity Analysis	VCD2TB Conversion
RTL(full)	57	50	4	3
GL(t,w)	21	12	*n/a*	*n/a*

Size (MB)	Inputs VCD	Full VCD
RTL(full)	2.6	1066
GL(t,w)	*n/a*	3099

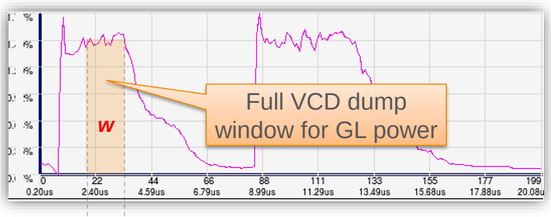
RTL full VCD activity analysis



(t, w)

Targeted GLS with **~54%** simulation window reduction

RTL full VCD activity analysis



Targeted GLS with **~82%** simulation window reduction

Summary

- Estimated **3x** productivity improvement compared to using the UVM environment to drive the GLS

ERICSSON