Veloce Power App

POWER ANALYSIS

VALUE

Motivations for Good Vectors & Emulation

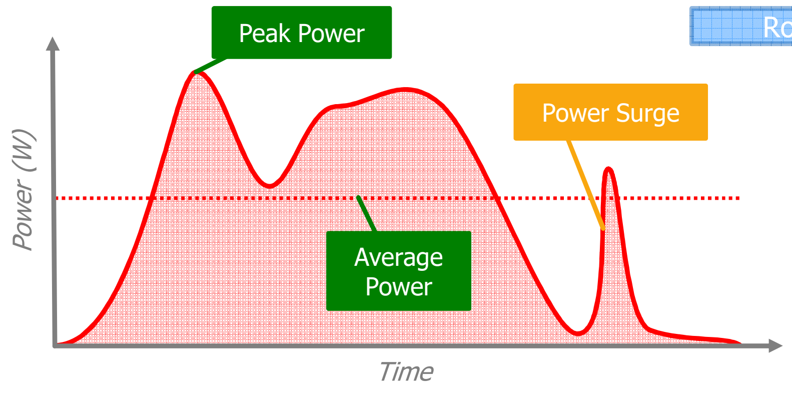
- Representative Power Estimation
  - **Average power** (battery life, cooling requirements, cost of energy,…)
  - **Power peaks**
    - **Large peaks** (~1us) -> Supply integrity,…
    - **Narrow peaks** (~1ns) -> IR-Drop,…
  - **Hot spots** (Local IR-Drop…)
  - **Power domains partition** verification 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

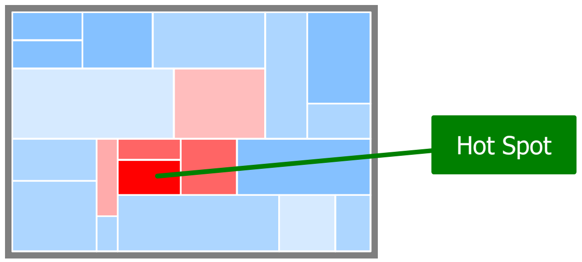
BUT ALSO…

- 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)

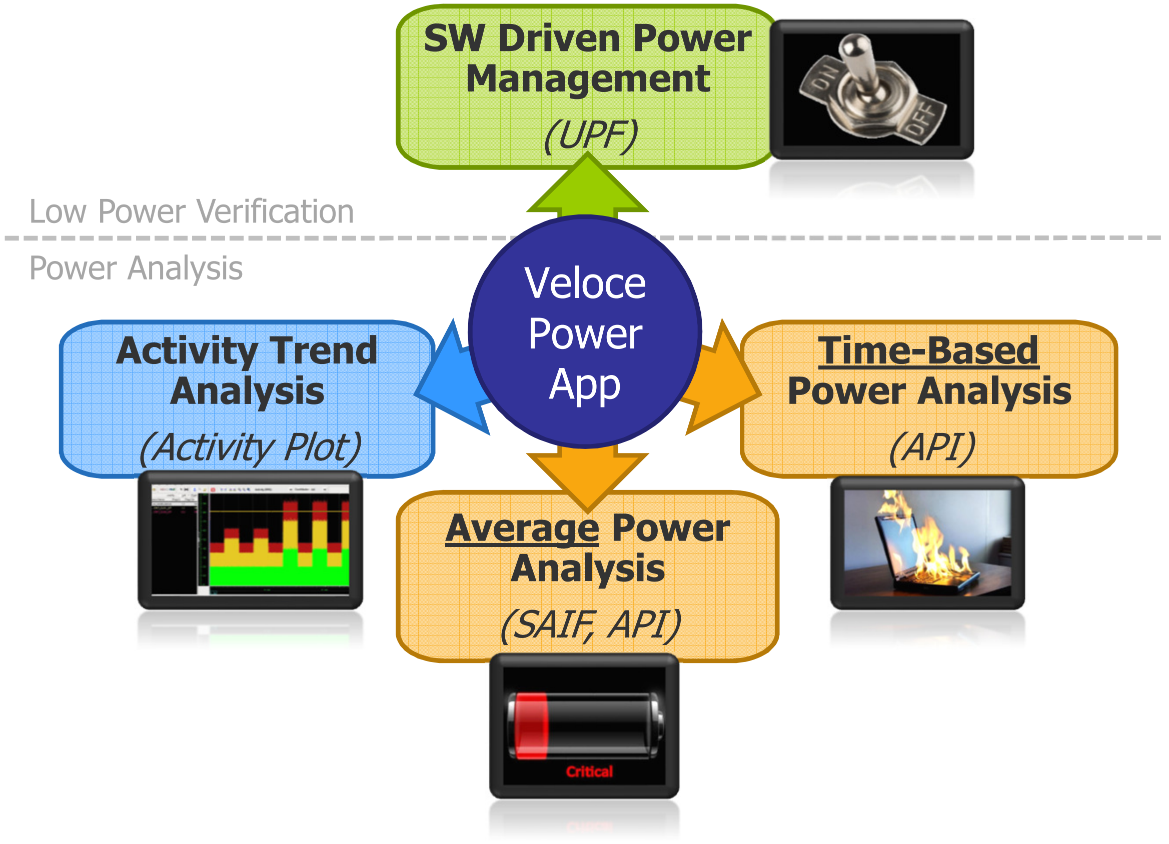


$$VDD_{PMOS} = VDD_{supply} - IR_{grid} - L_{interface} dI/dt$$

Voltage Drop



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

OVERVIEW

Overview

UPF

UPF (Power Aware) session in near future

Forward/Backward SAIF

Activity Plot

#CPU/16: Consume 1 VelocePower license for every 16 distributed processes; one for UPF

Dynamic Read Waveform API

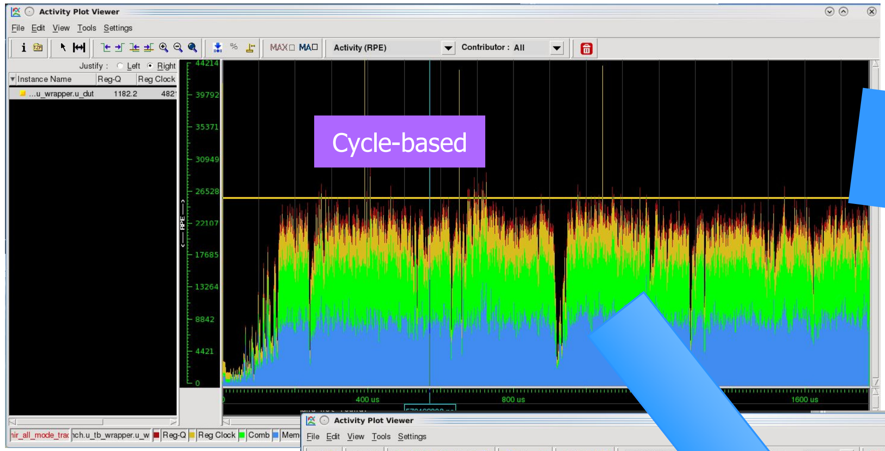
Description	Sales Part Number	Licenses	Comments
Veloce Power App SW	259467	VelocePower (1)	Power management (UPF) and power Analysis
Veloce OS App SW	259469	VeloceOS (1) visualizer_c (2)	Prerequisite for Power application

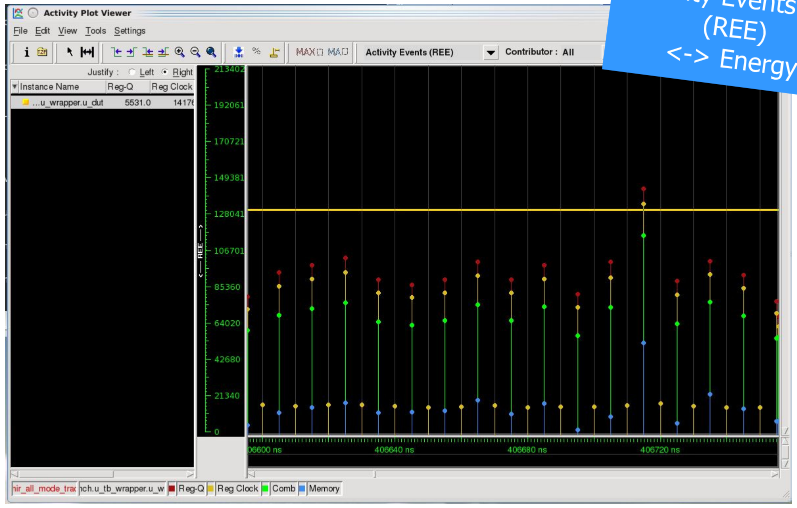
SAIF

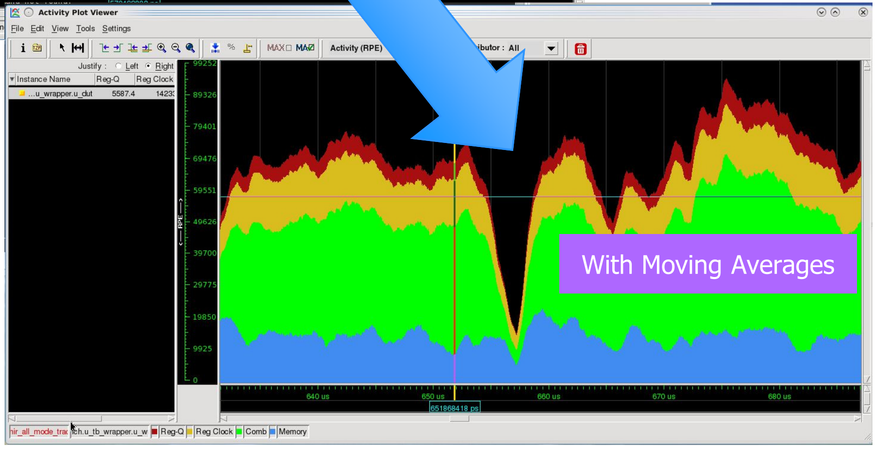
ACTIVITY PLOT

Activity Plot

- All contributors broken down (Reg-Q, Reg-Clk, Comb, Mem)
- Multiple views (RPE, REE, RPE/RAE…)
- Responsive environment to analyze data



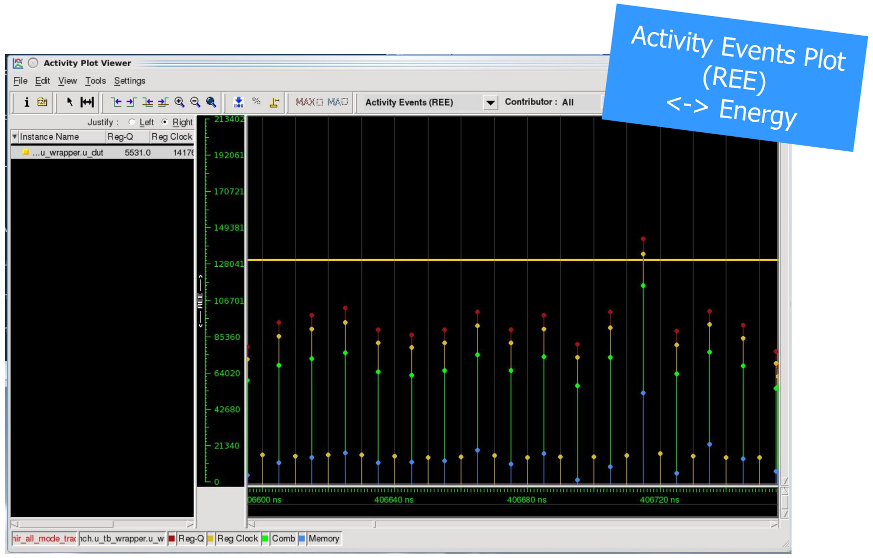


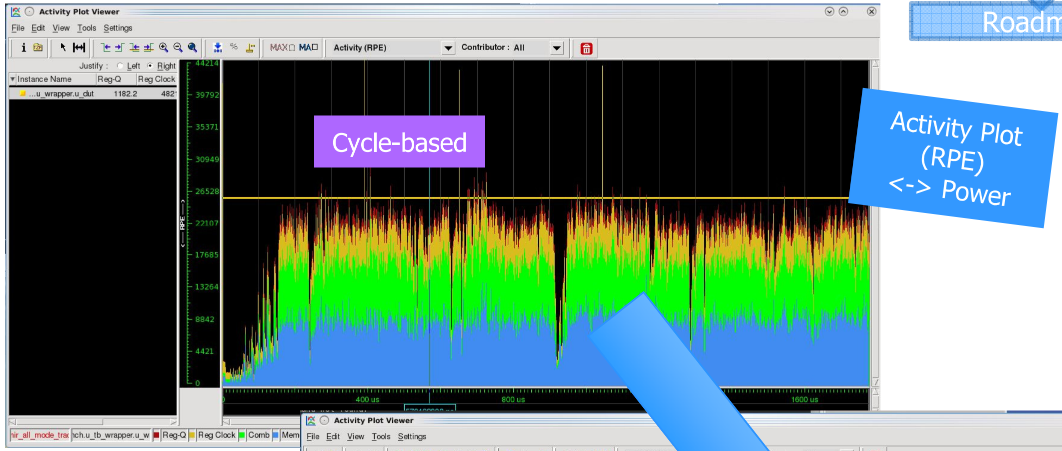


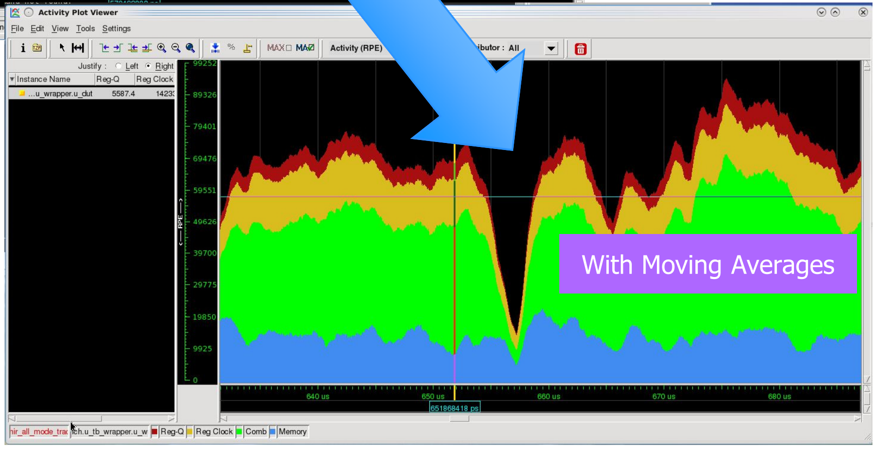
Activity Plot Enhancements

3.16.1.8
(Beta)

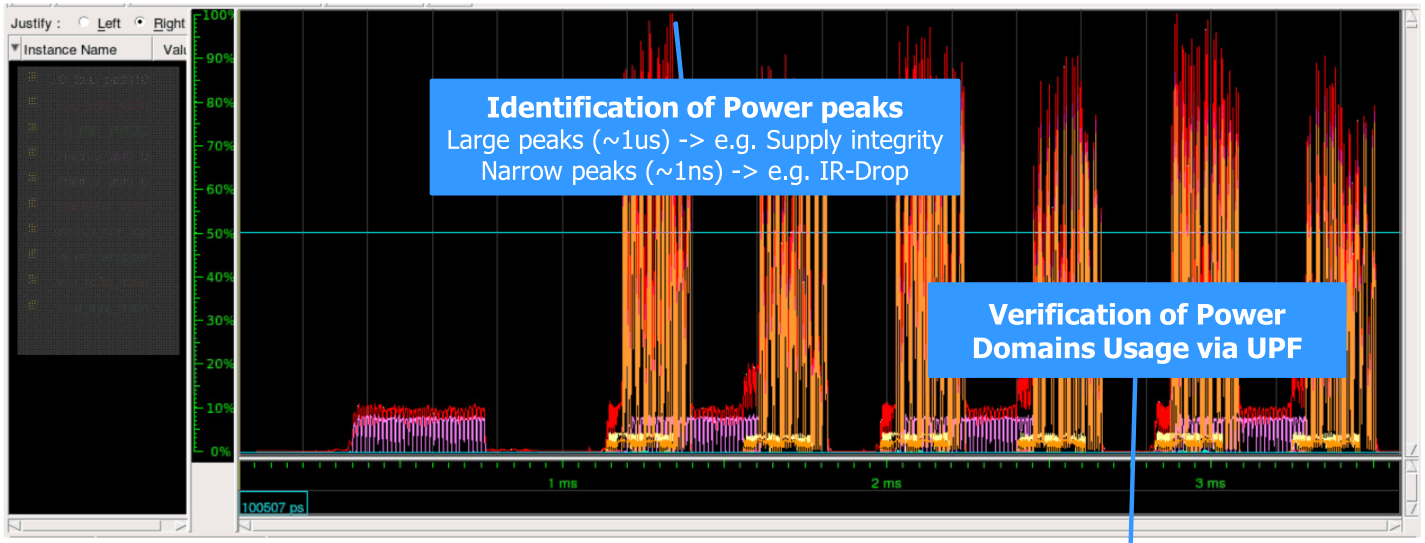
- All contributors broken down (Reg-Q, Reg-Clk, Comb, Mem)
- Multiple views (RPE, REE, RPE/RAE…)
- Responsive environment to analyze data







Activity Plot Applications

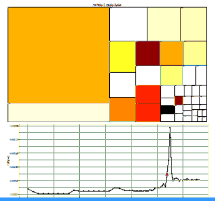


Identification of Power peaks
Large peaks (~1us) -> e.g. Supply integrity
Narrow peaks (~1ns) -> e.g. IR-Drop

Verification of Power Domains Usage via UPF

ON	OFF	ON	OFF	ON	OFF

Should be OFF



Hot Spots Identification
Optimization targets, Local IR-Drop,…

Power Trends
Compare activity plots across RTL drops

100X+ faster than with traditional Power Analysis tool

Power Surges Identification
dI/dt voltage drop

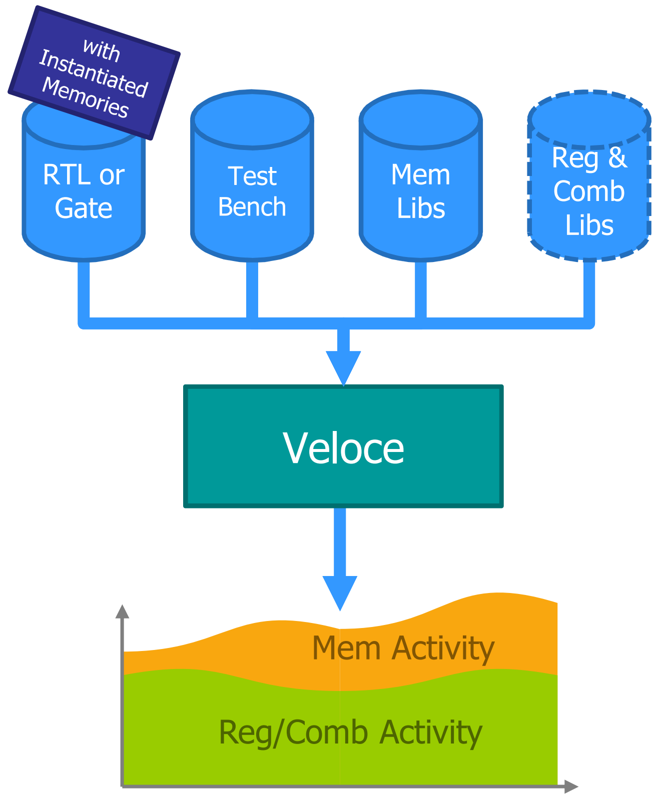
Electro-migration
High power on very long periods

Memory Modeling in Activity Plot

- Inputs
  - Requires the memory IPs to be instantiated in design (RTL or Gate)
  - Requires the Liberty files for Memories
  - Liberty files for Registers & Comb cells recommended
- Implementation
  - Veloce computes power for the memory instances by capturing its inputs in the trace uploads
  - Power number converted into Register Energy Equivalent (REE) and Register Power Equivalent (RPE)
  - Contributions of register, combinational and memory activity are compatible

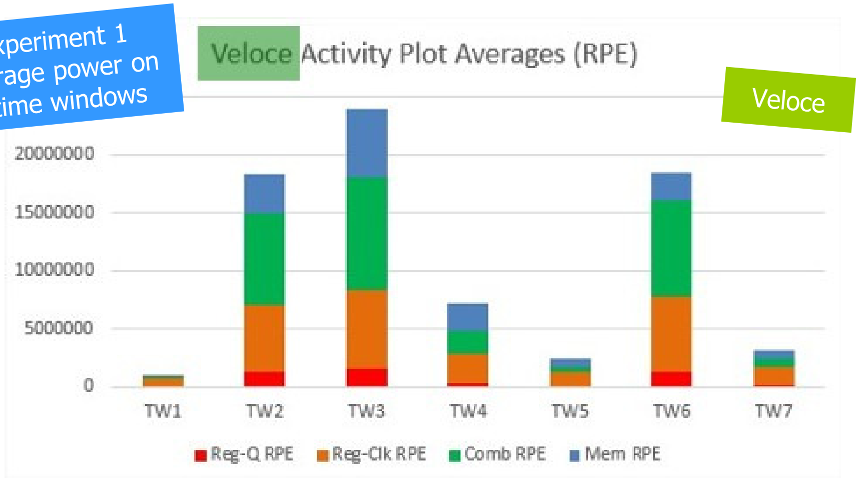
➢ **Total activity plot correlates better to power plot**

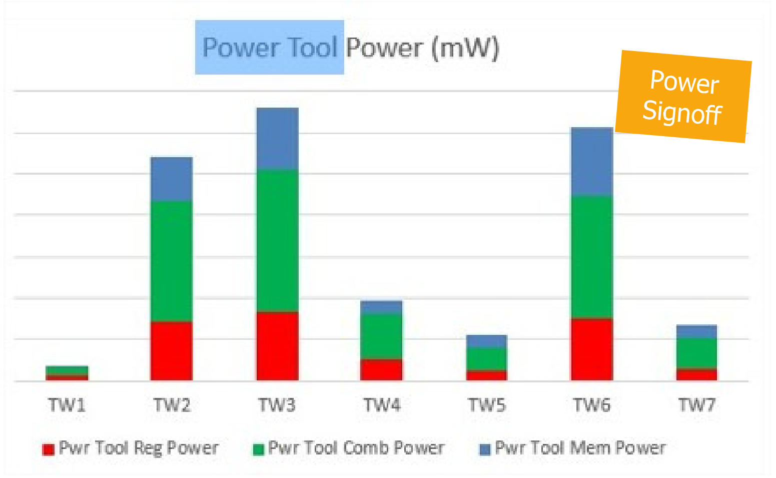
➢ **Enhanced ability to identify power peaks**



Enhanced Accuracy (Results on customer GPU)

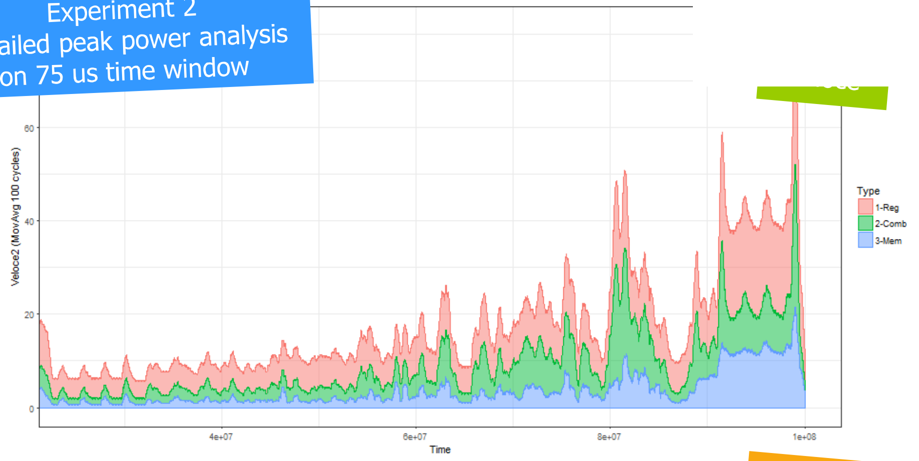
Experiment 1
Average power on
7 time windows

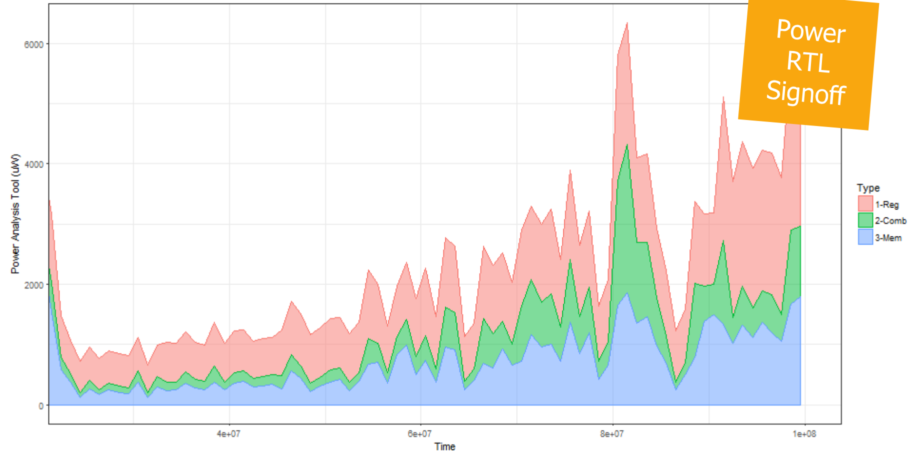




Reliable
Trend
Detection

Experiment 2
Detailed peak power analysis
on 75 us time window



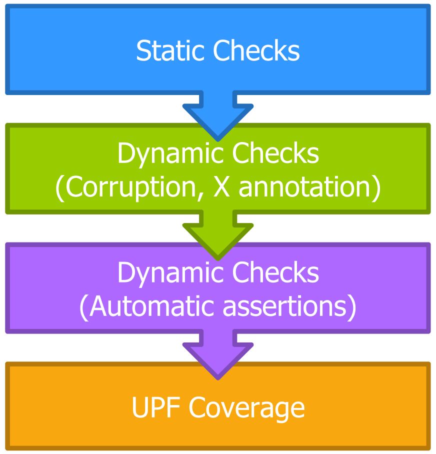


Activity Plot

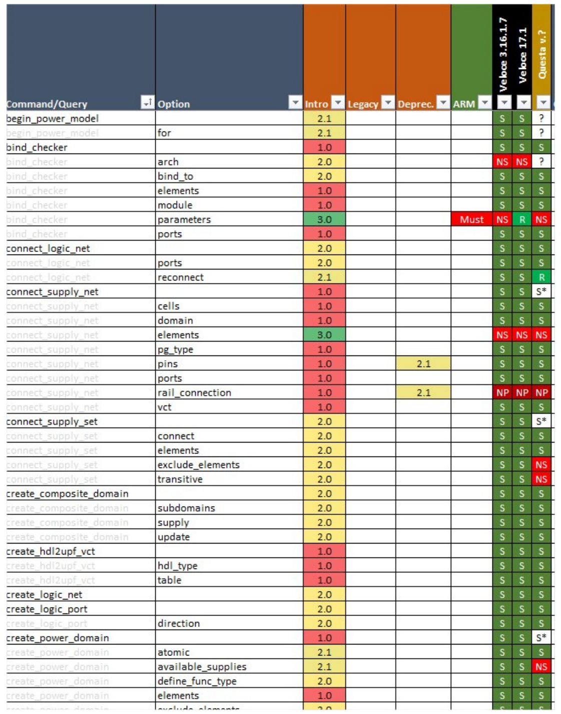
- Shows high activity spots in time over long emulation
  - Can use instance list to find power distribution in design hierarchy
- Generate during emulation run
  - hwtrace autoupload on –tracedir {name} -gen activityplot –captureratio <default is 96> -activityplot_options "...."
  - Captureratio 96 means 1k cycles captured every 96k cycles run
  - Other valid values are 1, 8, 16, 32, 256, 512, 1024
  - Smaller ratio would take longer and consume more space
- Generate after emulation run
  - velpc -tracedir <trace_dir> -instlist <instance_list_file>
- To view activity plot created as above
  - velview -powerdir <trace_dir>
- Can select horizontal (hwtrace settzf <..>) and vertical range

LOW POWER CHECKS

Value Proposition


Static Checks
Dynamic Checks (Corruption, X annotation)
Dynamic Checks (Automatic assertions)
UPF Coverage

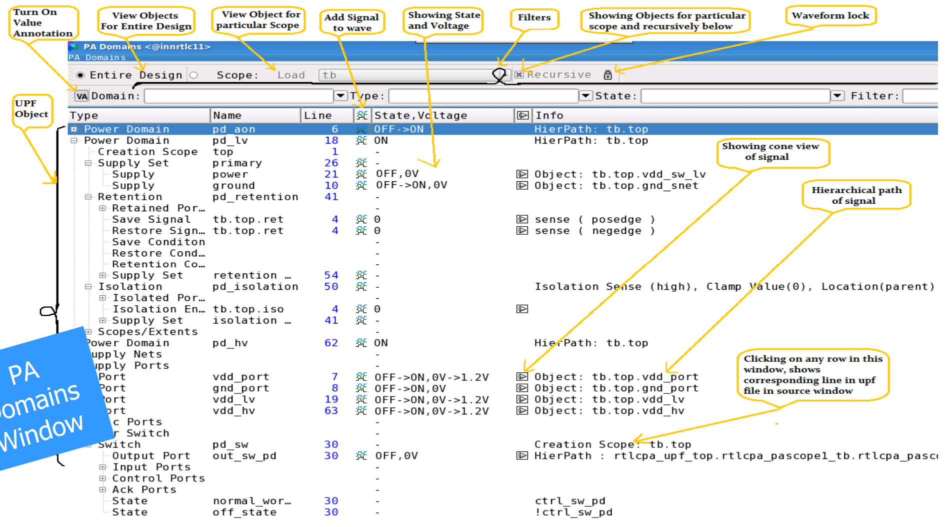
- Breadth of checks
  - Static, dynamic and coverage checks for realistic scenarios
  - Can address common case where power controller is implemented in SW
- UPF Support
  - Very good UPF 2.x support
  - UPF 3.0 support planned for 17.1 release
  - Alignment with Questa



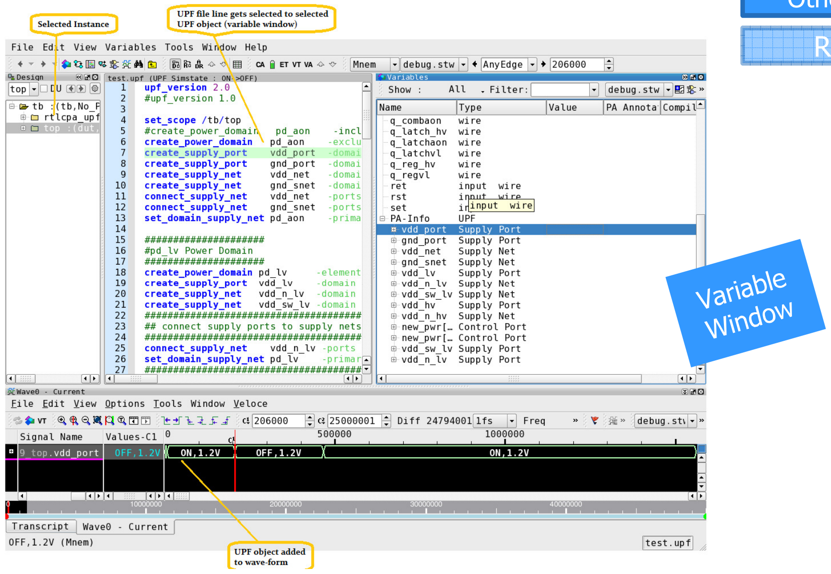
Command/Query	Option	Intro	Legacy	Deprec.	ARM	Veloce 3.16.1.7	Veloce 17.1	Questa v.?
begin_power_model		2.1				S	S	?
begin_power_model	for	2.1				S	S	?
bind_checker		1.0				S	S	S
bind_checker	arch	2.0				NS	NS	?
bind_checker	bind_to	2.0				S	S	S
bind_checker	elements	1.0				S	S	S
bind_checker	module	1.0				S	S	S
bind_checker	parameters	3.0			Must	NS	R	NS
bind_checker	ports	1.0				S	S	S
connect_logic_net		2.0				S	S	S
connect_logic_net	ports	2.0				S	S	S
connect_logic_net	reconnect	2.1				S	S	R
connect_supply_net		1.0				S	S	S*
connect_supply_net	cells	1.0				S	S	S
connect_supply_net	domain	1.0				S	S	S
connect_supply_net	elements	3.0				NS	NS	NS
connect_supply_net	pg_type	1.0				S	S	S
connect_supply_net	pins	1.0		2.1		S	S	S
connect_supply_net	ports	1.0				S	S	S
connect_supply_net	rail_connection	1.0		2.1		NP	NP	NP
connect_supply_net	vct	1.0				S	S	S
connect_supply_set		2.0				S	S	S*
connect_supply_set	connect	2.0				S	S	S
connect_supply_set	elements	2.0				S	S	S
connect_supply_set	exclude_elements	2.0				S	S	NS
connect_supply_set	transitive	2.0				S	S	NS
create_composite_domain		2.0				S	S	S
create_composite_domain	subdomains	2.0				S	S	S
create_composite_domain	supply	2.0				S	S	S
create_composite_domain	update	2.0				S	S	S
create_hdl2upf_vct		1.0				S	S	S
create_hdl2upf_vct	hdl_type	1.0				S	S	S
create_hdl2upf_vct	table	1.0				S	S	S
create_logic_net		2.0				S	S	S
create_logic_port		2.0				S	S	S
create_logic_port	direction	2.0				S	S	S
create_power_domain		1.0				S	S	S*
create_power_domain	atomic	2.1				S	S	S
create_power_domain	available_supplies	2.1				S	S	NS
create_power_domain	define_func_type	2.0				S	S	S
create_power_domain	elements	1.0				S	S	S

Advanced Debug with Visualizer

- Visualizer debug environment (common with Questa)
  - 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
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
PA Domains Window
PA Domains <@innrtlc11>
PA Domains
Entire Design
Scope:
Load
tb
Recursive
Domain:
Type:
State:
Filter:

```
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 Por…                          -
      Save Signal     tb.top.ret       4     0                      sense ( posedge )
      Restore Sign…   tb.top.ret       4     0                      sense ( negedge )
      Save Conditon                          -
      Restore Cond…                          -
      Retention Co…                          -
    ⊞ Supply Set      retention …     54     -
  ⊟ Isolation         pd_isolation    50     -                      Isolation Sense (high), Clamp Value(0), Location(parent)
    ⊞ Isolated Por…                          -
      Isolation En…   tb.top.iso       4     0
    ⊞ Supply Set      isolation …     41     -
  ⊞ Scopes/Extents                           -
ower Domain           pd_hv           62     ON                     HierPath: tb.top
upply Nets                                   -
upply Ports                                  -
 Port                 vdd_port         7     OFF->ON,0V->1.2V       Object: tb.top.vdd_port
 Port                 gnd_port         8     OFF->ON,0V             Object: tb.top.gnd_port
 ort                  vdd_lv          19     OFF->ON,0V->1.2V       Object: tb.top.vdd_lv
 ort                  vdd_hv          63     OFF->ON,0V->1.2V       Object: tb.top.vdd_hv
  c Ports                                    -
  r Switch                                   -
  Switch              pd_sw           30     -                      Creation Scope: tb.top
    Output Port       out_sw_pd       30     OFF,0V                 HierPath : rtlcpa_upf_top.rtlcpa_pascope1_tb.rtlcpa_pasc
  ⊞ Input Ports                              -
  ⊞ Control Ports                            -
  ⊞ Ack Ports                                -
    State             normal_wor…     30     -                      ctrl_sw_pd
    State             off_state       30     -                      !ctrl_sw_pd
```


Selected Instance
UPF file line gets selected to selected UPF object (variable window)
Variable Window
UPF object added to wave-form
File Edit View Variables Tools Window Help
Mnem
debug.stw
AnyEdge
206000
Design
top
Variables
Show : All
Filter:
Wave0 - Current
File Edit View Options Tools Window Veloce
Transcript
Wave0 - Current
test.upf

```
tb :(tb,No_P
  rtlcpa_upf
  top :(dut,
```

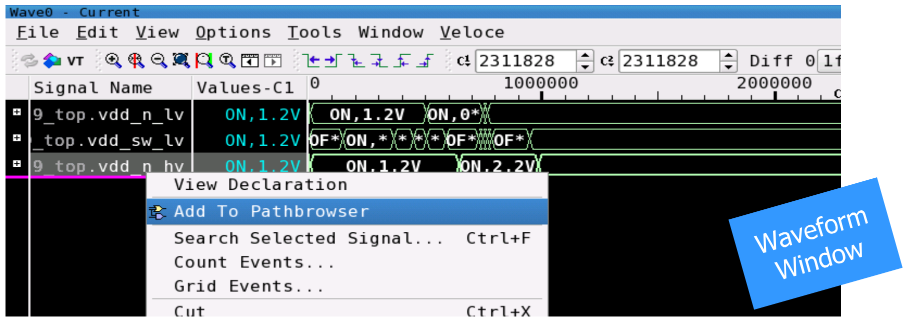
```
test.upf (UPF Simstate : ON->OFF)
upf_version 2.0
#upf_version 1.0

set_scope /tb/top
#create_power_domain   pd_aon     -incl
create_power_domain    pd_aon     -exclu
create_supply_port     vdd_port   -domai
create_supply_port     gnd_port   -domai
create_supply_net      vdd_net    -domai
create_supply_net      gnd_snet   -domai
connect_supply_net     vdd_net    -ports
connect_supply_net     gnd_snet   -ports
set_domain_supply_net  pd_aon     -prima

####################
#pd_lv Power Domain
####################
create_power_domain pd_lv      -element
create_supply_port  vdd_lv     -domain
create_supply_net   vdd_n_lv   -domain
create_supply_net   vdd_sw_lv  -domain
####################################
## connect supply ports to supply nets
####################################
connect_supply_net     vdd_n_lv -ports
set_domain_supply_net pd_lv     -primar
####################################
```

```
Name          Type          Value   PA Annota  Compil
 q_combaon    wire
 q_latch_hv   wire
 q_latchaon   wire
 q_latchvl    wire
 q_reg_hv     wire
 q_regvl      wire
 ret          input  wire
 rst          input  wire
 set          input  wire
⊟ PA-Info     UPF
 ⊞ vdd_port   Supply Port
 ⊞ gnd_port   Supply Port
 ⊞ vdd_net    Supply Net
 ⊞ gnd_snet   Supply Net
 ⊞ vdd_lv     Supply Port
 ⊞ vdd_n_lv   Supply Net
 ⊞ vdd_sw_lv  Supply Net
 ⊞ vdd_hv     Supply Port
 ⊞ vdd_n_hv   Supply Net
 ⊞ new_pwr[…  Control Port
 ⊞ new_pwr[…  Control Port
 ⊞ vdd_sw_lv  Supply Port
 ⊞ vdd_n_lv   Supply Port
```

```
C1 206000   C2 25000001   Diff 24794001 1fs   Freq   debug.stv
Signal Name      Values-C1   0            500000                1000000
9_top.vdd_port   OFF,1.2V    ON,1.2V | OFF,1.2V |          ON,1.2V
                             10000000   20000000   30000000   40000000
OFF,1.2V (Mnem)
```



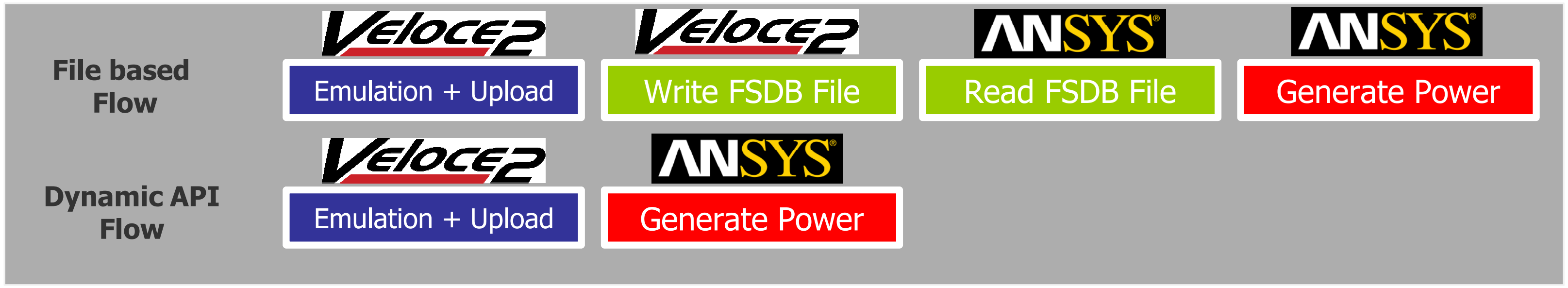
```
C1 2311828   C2 2311828   Diff 0 1f
Signal Name        Values-C1   0           1000000        2000000
9_top.vdd_n_lv     ON,1.2V     ON,1.2V   ON,0*
_top.vdd_sw_lv     ON,1.2V     OF*  ON,*  *  *  *  OF*  OF*
9_top.vdd_n_hv     ON,1.2V     ON,1.2V   ON,2.2V
```

DYNAMIC READ WAVEFORM API

Customer Motivation

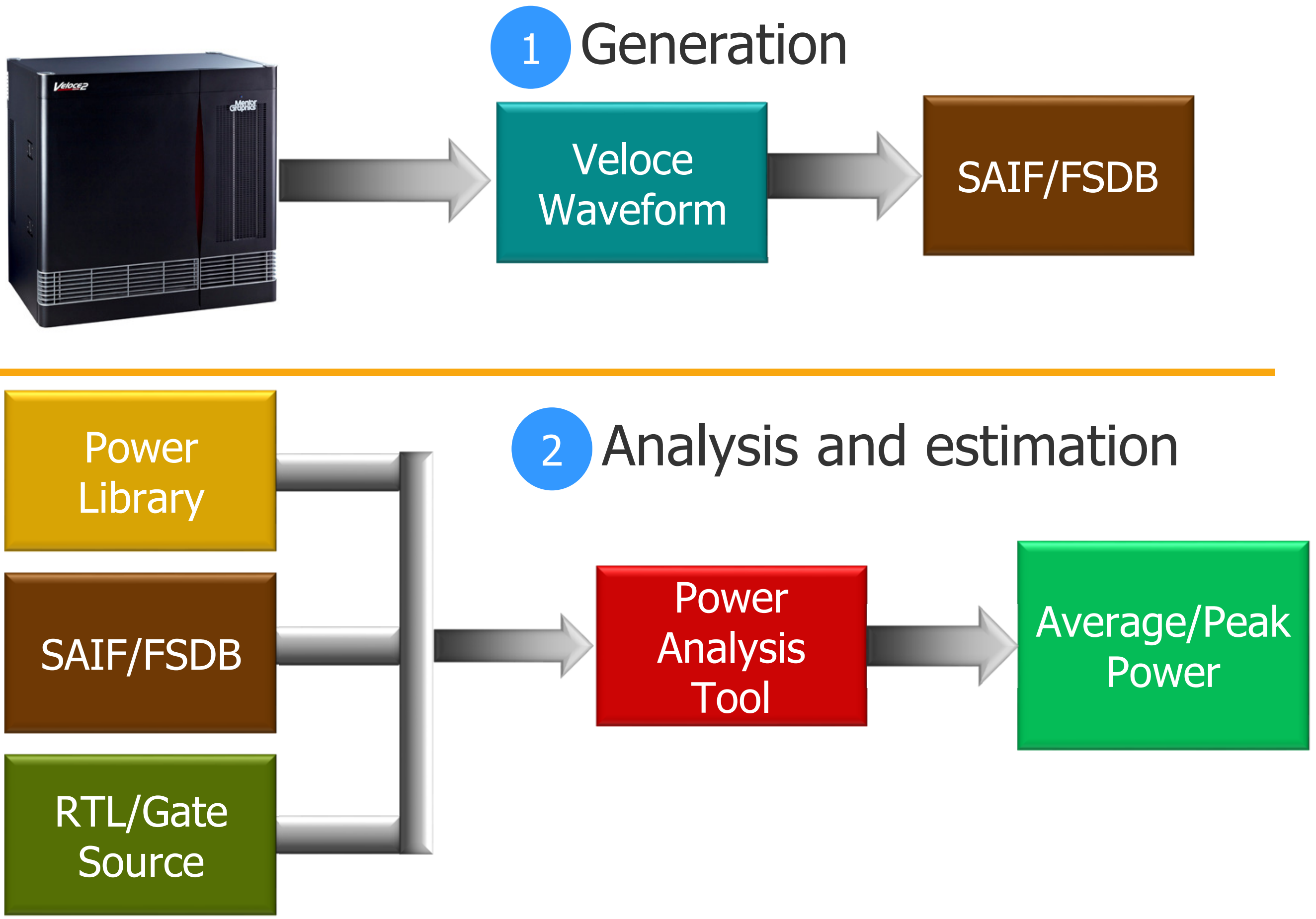
- Want to use full SOC design RTL/Gate and run real life stimulus to generate accurate power numbers
  - Power tools use info from fsdb ; SAIF doesn't have temporal information
  - Billions of nets for millions of cycles = Large fsdb
  - Compromise with limited nets and functional tests = Low accuracy
- Get a faster implementation independent of FSDB
  - FSDB writing/reading takes a long time
- Desire an end to end flow, where simulation is accelerated as well as wave data can be consumed effectively by Power tool

Speed Improvements: File Based vs. Dynamic Read Waveform API Flow



Design Description	Design Size	Number of Cycles	File Based Flow	Dynamic Read Waveform API Flow	Speed Improvements
GPU	13M	24 M	20 hours	7 hours	2.85X
CPU Core	45M	3 M	27 hours	12 hours	2.25X
Processor	65M	15 M	51 hours	12 hours	4.25X
PCI Subsystem	42M	11M	40 hours	10 hours	4x
Video Enc-Dec block	25M	0.3	2.1 hours	0.9 hour	2.3x
Video Enc-Dec block	25M	72 M	61 hours	12 hours	5x

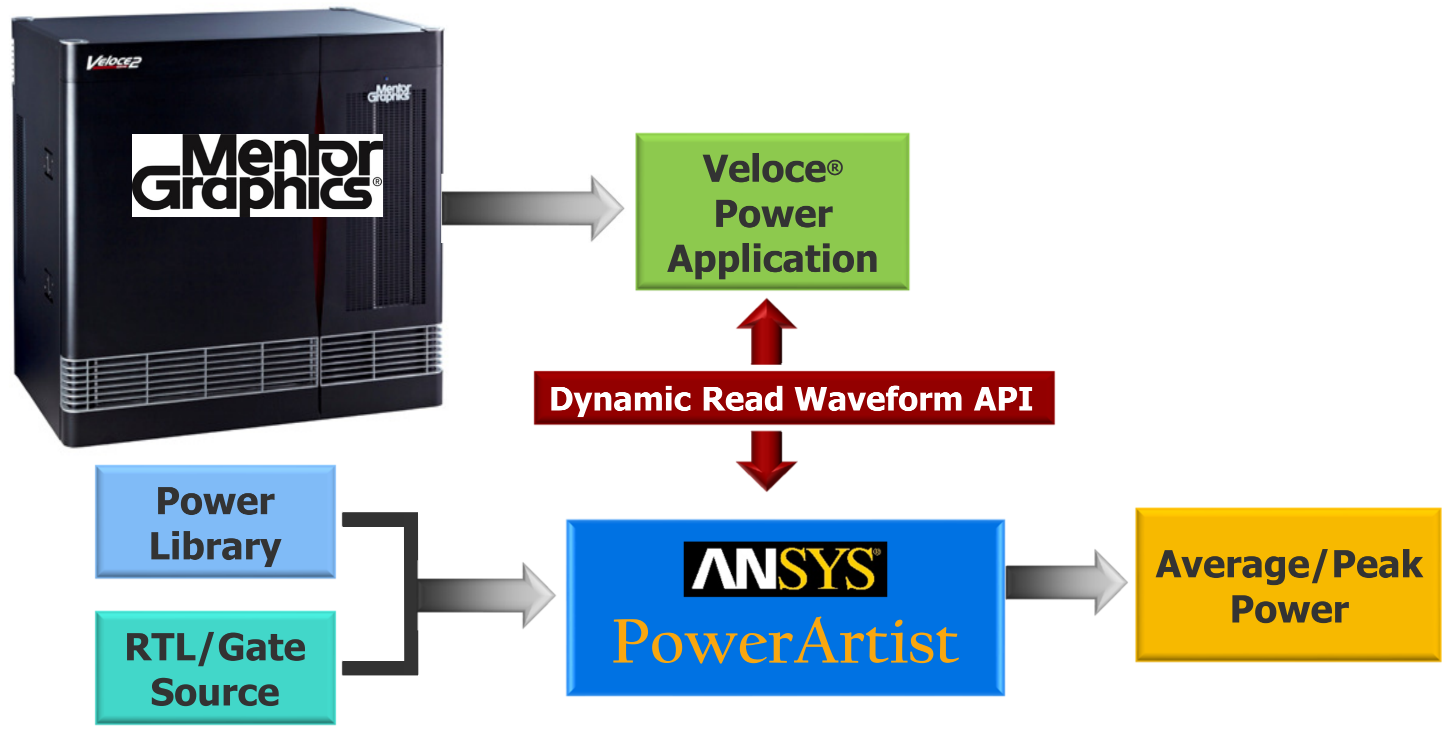
Existing Use Model : Average/Peak power



Existing Use Model : Before Integration

- Veloce compile and run to generate .stw (Or –gen switch for fsdb/saif)
- In Parallel, analyze the design in PowerArtist using RTL/Gate source list with **Elaborate** command to generate .scn file (binary scenario file)
- **ecf2wave** to convert .stw to .fsdb (**velsaif** for .saif)
- Invoke PowerArtist and specify the FSDB/SAIF/VCD files to generate GAF (Global Activity File)
- **CalculatePower** command in PowerArtist to generate power report

Veloce Power Application Flow



Use Model : Post Integration

- Start Veloce compile and in Parallel, user can invoke PowerArtist and **Elaborate,** same as before
- Run Veloce with additional switch for hwtrace autoupload (-gen wave_stream –instance <..>)
- Invoke PowerArtist with updated tcl file
  - In online mode, it will wait till dataset is generated by Veloce and attach to it during Emulation. Data stream starts as soon as some uploads are available
  - In offline mode, it start immediately as .stw is already available
  - Generates GAF
- **CalculatePower** same as before

Integration Guidelines

- Make sure that
  - Power-critical signals are included in Veloce waveforms
      - Example: Small arrays shouldn't be inferred as memories in RTLC
    - Ask user/AE for PowerArtist list of power-critical signals missing activity
    - Re-run Veloce ensuring the missing signals are dumped
  - Same design setup provided to both, Veloce and PowerArtist
    - Same source files and defines are used for Veloce and PA
      - Example: Different scan mode setting between Veloce and PA setup
    - Black boxes in Veloce are not different from PA
- Inactive edge optimization might affect power numbers
  - Duty cycle is important for generating power numbers

Guidelines

- Run job should consist of:
  - Velrun on comodel host queue & PowerArtist job to non host queue
  - Since PA run job is not a forked process from comodel host queue , it wont block comodel host after Velrun completes, and will run to completion on its own
- Helpful logs
  - Velwavegen and ecf2wave logs inside <trace dir>.stw/logs/

Mentor Graphics®

www.mentor.com