System Level Functional Verification and Power Analysis for Low Power Design

T.C. Lin
Cadence Design Systems, Inc.
April 21, 2016
Agenda

- Reasons low power design becomes so critical
- Low power design issues from system level’s perspective
- What are the main concerns in low power design
- Power reduction techniques and key design questions
- Low power functional verification power analysis at system level
  - Challenges
  - Things you need to do
  - Tools you need to leverage
Low Power: Everyone’s Concern
Different drivers in different verticals

Mobile/Handheld
Battery life (energy)
Unit cost (chip package)

Consumer/Digital Home
Unit cost (package, fans)
Energy certification
Reliability

Network/Data Center
Power efficiency
Total cost of ownership
Reliability
Green

Low-power requirements drive different design decision
• Design architecture
• IP make versus buy
• Manufacturing process
• Specific low-power design techniques
It’s a System Level Issue - SoC running With Power Control and Application Software
Main “Power Problems” in SoC and Low Power Design Techniques

- Key power design concerns
  - Peak power consumption
  - Maximum instantaneous difference in power
  - Maximum average power consumption
  - Average power consumption
  - Stand-by power consumption

- Power reduction techniques
  - Goals - reduce leakage power, internal power, dynamic power
  - Multi-supply voltage, power gating
  - Logic sharing, clock gating, DVFS, using low power ASIC cells, …
  - Using CPF or IEEE1801/UPF for low power control – power domain concept

- Key design questions
  - Does my design work with all power control HW/SW kick in?
  - Is power consumption under budget? Meeting requirements under different applications? Will my circuitry safe under different working condition?
IEEE1801/UPF Example

```gherkin
define design top tb/dut

create_supply_port VDD
create_supply_port VSS
create_supply_port VDD_sw -direction out

create_supply_set SS -function {power VDD} -function {ground VSS}
create_supply_set SS_sw -function {power VDD_sw} -function {ground VSS}

create_power_domain PD_def -include_scope -power_up_states high
create_power_domain PD_r0 -elements r0 -power_up_states low
set_domain_supply_net PD_r0 -primary_power_net VDD_sw -primary_ground_net VSS

create_power_switch sw -domain PD_def -output_supply_port { Pout VDD_sw } \ 
   -input_supply_port { Pin VDD } -control_port { cntrl pso } -on_state { ST_on Pin { ! cntrl } }
set_domain_supply_net PD_def -primary_power_net VDD -primary_ground_net VSS

set_retention ret_0 -domain PD_r0 \ 
   -save_signal {save posedge} -restore_signal {restore posedge} \ 
   -retention_power_net VDD -retention_ground_net VSS

set_isolation iso_2 -domain PD_i0 -elements i0/in \ 
   -isolation_signal iso -isolation_sense high -clamp_value 1 \ 
   -isolation_power_net VDD -isolation_ground_net VSS

add_port_state VDD_sw -state {on 1.0} -state {off off}
create_pst pst_VDD_sw -supplies {VDD_sw}
add_pst_state on -pst pst_VDD_sw -state {on}
add_power_state SS_sw \ 
   -state off {logic_expr {ps0} -simstate CORRUPT -legal} \ 
   -state on {logic_expr {!ps0} -simstate NORMAL -legal}
describe_state_transition ON -object SS_sw -from {off} -to {on} -legal
```

Diagram:
- Supply network
- Power domain
- Retention
- Isolation
- PST state
- Power state
Power Aware Functionality Modeling with IEEE1801

Figure 7. Isolation gate and power-down switch

Figure 8. State retention power gating

Figure 9. Power-up/down sequence
SoC Power Analysis Requires “Deep” Cycles

Component-level

FLASH
DDR3
Memory Controller
Graphics Controller
10G/40G Ethernet I/F
Scheduler

System-level

Embedded Software
OS, Driver, Applications
FLASHTDDR3 GDRAM
Memory Controller
Encrypted IP
Graphics Controller
Switch
IP
IP
10G/40G Ethernet I/F
Scheduler
Wireless Controller
General Purpose I/O

Deep cycles:
Dynamic power profiling calculates average power over long run w/ “real” stimulus & SW interactions

Local Max Case:
Simulation captures narrow window

Explore the ‘What if’s to avoid ‘What now

Additional cycles are needed for system-level power analysis

Identify and analyze peak and average power at system level
Low Power Functional Verification and Power Analysis at System Level

- Leverage tools which supports verification and analysis at system level
  - Allow PMS and application software to run with design to create real scenario
  - Controllability – let users specify run condition/environment for low power operations
  - Observability – provide full visibility of system behavior (design states, control signal values, etc.)

- For functional verification
  - Mimic low power control and circuitry behavior at system level
    - Be able to mimic power shutoff, retention, and isolation condition
    - flip-flop and memory randomization from PSO to power back-on
  - Be able to dynamically configure ‘trigger’ condition with low power objects
    - Allow to check impact of various low power conditions with quick turnaround time
  - Self-checking of power intents, and report errors when there are violations
  - Low power event log – record and report power control activity during a test
Low Power Functional Verification and Power Analysis at System Level (cont’d)

- For power analysis
  - Power analysis takes time
    - Need special strategy
  - Measure power consumption in hardware with software impact factored in
  - Progressive power analysis techniques (time domain)
    - Power analysis at coarse grain level at high speed for long period of time
    - Identify window of interest
    - Detailed power analysis at smaller set of runtime window
    - Repeat the detailed analysis without needing to re-run test
  - ‘Hot spot’ analysis – which design module consumes more power (space domain) in the SoC
  - Analysis tool needs to generate various files for different analysis purpose
    - saif file for long time window for average power calculation
    - Vcd/fsdb/sst2 for short (peak power) window for IR drop analysis
  - Be able to support both RTL level and gate level power analysis