2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE
MUNICH, GERMANY
OCTOBER 14-15, 2025

Early Chip-Level Power Estimation using Digital Mixed-Signal Simulations

An Estimation Method for Top-Level Power Consumption Prior to Analog Mixed-Signal (AMS) Simulation Availability

Dantas, Ricardo, ams OSRAM, Munich, Germany, Ricardo.Dantas@ams-osram.com
Zaidi, Akib Mohammad Azam, ams OSRAM, Munich, Germany, Akib.Zaidi@ams-osram.com
Alecu, Gabriela, ams OSRAM, Bucharest, Romania, CorneliaGabriela.Alecu@ams-osram.com

***Abstract*— This paper proposes a methodology for early power estimation during chip design phase using digital mixed-signal (DMS) simulations. It is not a replacement for the more accurate chip-level analog mixed-signal (AMS) simulations, which are typically considered the sign-off criterion for pre-Silicon power consumption estimation. However, AMS simulations are time-consuming and usually executed late in the development phase; if implementation or architectural issues are found, it is often too late for major design updates, thus delaying the tape-out process (i.e., pre-Silicon phase completion). The proposed method is an auxiliary technique aiming at identifying power consumption related problems early in the design phase, which can later be confirmed through AMS simulations. It relies on analog model-based digital simulations to estimate dynamic power consumption. Individual contributions of all analog IPs are consolidated according to the activation sequence of each corresponding analog model during digital simulations, thus allowing early identification of potential power violations. Although it is not intended to produce accurate power figures, especially over transient states, this method offers a preliminary check for power consumption across different operational modes and configurations.**

Keywords—Power consumption, power estimation, power state machine, mixed-signal, DMS, AMS

I. INTRODUCTION

One important aspect of functional verification, besides ensuring correct chip functionalities, is the estimation of its power consumption across all operating modes. It becomes increasingly complex when multiple analog blocks (intellectual properties - IPs) have different power modes (also known as power states) that are activated independently throughout the system operation. The maximum power consumption of individual IPs does not reflect the temporal system behavior across all operational scenarios, particularly in battery-powered devices, which predominantly operate in low power modes for extended periods. During pre-Silicon verification, a reliable and accurate method for chip-level power analysis, commonly used as a sign-off criterion, is based on analog mixed-signal (AMS) simulations. However, these are generally executed later in the pre-Silicon phase and have extensive simulation times. In cases where power violations due to unexpected IP activations are observed, required re-design and re-simulations can significantly impact project schedule. Several methods of system-level power estimation can be found in the literature, such as [1] and [2], but they aim at high accuracy, rather than faster, low-effort results. Particularly for devices targeting the consumer market, where the full pre-Silicon development phase (concept, design, verification and layout) lasts less than a year, it is crucial to ensure that the chip will remain within the specified power budget.

This paper proposes an easy-to-implement method for early power estimation during chip design using digital mixed-signal (DMS) simulations, i.e. a system-level simulation with analog behavioral models. Its merit relies on the simplicity of the implementation (thus avoiding increased effort during pre-Silicon phase) and on the adoption of fast event-driven digital simulators, as opposed to continuous-time analog solvers. The dynamic activation of a model during simulation defines its current power mode (e.g., OFF, WARMUP, ON). This is then correlated with the actual power consumption of each individual IP, which can be derived from the specification (e.g., power budget), from analog block level simulations or estimation methods, as proposed in [3]. The power contribution of all individual models is consolidated, resulting in the total estimated power consumption.

Although not intended to replace AMS simulations, this method offers early identification of potential power violations across different configurations and modes of operation, which could have been caused by architectural problems during concept phase or implementation issues during design phase. For instance, a given analog block is expected to be in low power mode or even disabled in a certain operating state but is left inadvertently activated, e.g. a voltage regulator left enabled unnecessarily when all circuitry that it supplies is not in operation. It may not affect the functionality but could exceed the maximum allowed power consumption for that particular mode of operation. System-level AMS simulation with schematic views is still required at the end of the pre-Silicon phase for signing off but power violations are then less likely to be discovered.

One aspect not covered by this method is the power consumption of digital parts. Major EDA vendors provide tools for digital power analysis, which relies on gate-level simulations (GLS) of post-layout digital netlist and annotated delays from an SDF (standard delay format) file. The whole process (GLS simulation and subsequent digital power estimation) does not require much time, and the same stimulus or test pattern used for analog power estimation can be executed in GLS to simulate the same scenarios. The digital power contribution can then be added to the analog estimation to achieve total system-level power consumption.

II. METHODOLOGY

A. Analog modelling and overview of power mode estimation

Analog behavioral models are developed to mimic analog functionality and are then validated against the design schematic to ensure that desired features are properly implemented. They are later integrated with the digital counterpart (typically in RTL – register transfer level – of gate-level netlist) for top-level simulations. In this paper, we have used Real Number Models (RNM) written in System Verilog, as they can be re-used in DMS simulations and fully support the Universal Verification Methodology (UVM [4]) constructs. The models themselves have no visibility into the system-level configuration or current mode of operation; they can only observe their input pins and how they are driven over time, which is enough to infer their current power states. Analog IPs do not use the digital concept of states, as seen in finite state machines (FSM). Therefore, a power state of a model can be interpreted as the power-aware operating mode, which could include basic modes like OFF (disabled, leakage only) and ON (enabled), or additional modes such as low power (LP) and high power (HP). This is illustrated in Figure 1. For reference, a digital FSM state and clock are shown in Figure 1.a, whilst the power modes of a few generic analog blocks are shown in Figure 1.b. PwnMng stands for power management (e.g., references or voltage regulators), Osc stands for oscillator (internal to the chip), Func stands for functional IP (e.g. ADC, DAC) and Prot stands for protection (e.g. under-voltage or over-current protection). This is just an example and does not represent any specific real product. Simulation results are presented in the Results section. Figure 1.c illustrates the estimated analog power consumption, which will be covered later in the text.

As briefly stated previously, a power mode is not necessarily an externally driven configuration mode but often a combination of inputs and internal states. For instance, if a linear regulator is properly supplied, its references are at appropriate levels and it is enabled, its power mode could be inferred to be ON (enabled). In some cases, the ON power mode can be split into low power (ON_LP) and high power (ON_HP), which could be defined directly by an input (e.g. high-power enable pin) or by an internal delay after its enable pin (EN) is asserted. The important fact is that power modes are solely calculated based on the stimuli applied to the model and do not use any information that is not available internally. For instance, the digital FSM state shown in Figure 1.a is only meant to illustrate the current system-level operating mode, whilst the models themselves do not have any information regarding the current FSM state, ACTIVE, SLEEP or DEEP_SLEEP.

Once the actual power is known for each power mode, either from the specification or from analog IP-level simulations, the dynamic power mode of all models can be converted to power figures, i.e., quiescent current (Iq). These power figures can be then added together to provide the total power for a given rail. Figure 1.c shows this step: analog blocks depicted in Figure 1.b are repeated but with power figures instead (arbitrary unitless values were used). The bottom part of Figure 1.c depicts the total power in numeric and graphical format. As can be

observed, total power consumption reaches its peak during ACTIVE state and its lowest values during DEEP_SLEEP state.

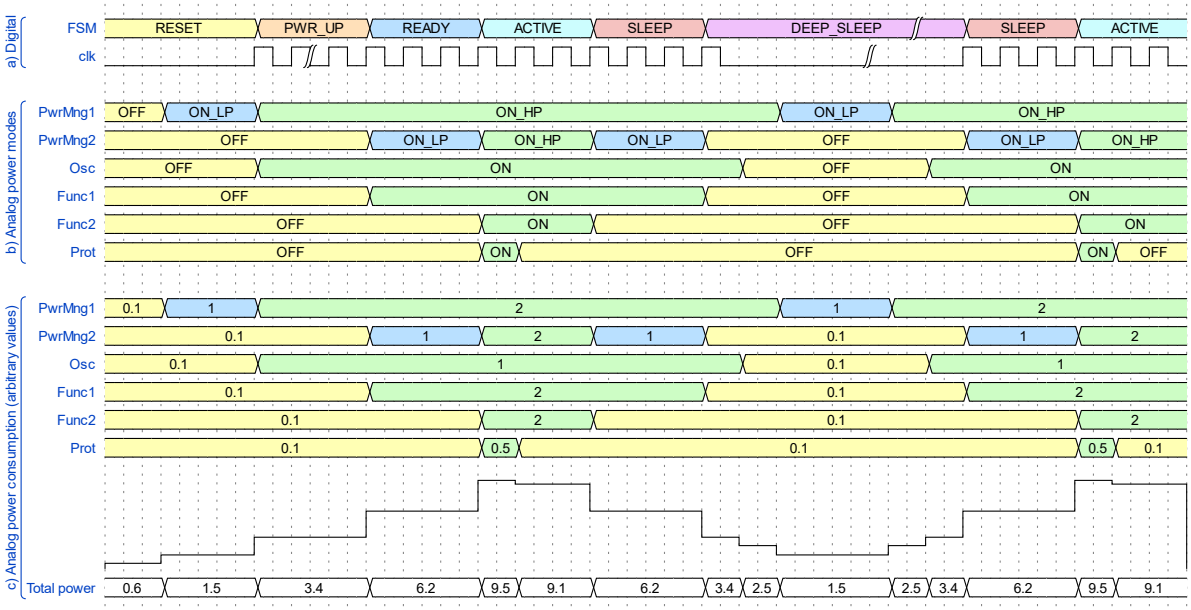

Figure 1. a) Digital FSM state and clock; b) power modes of individual analog blocks; c) analog power consumption per block and total (arbitrary values used).

B. Power modes realization and Iq estimation

As previously introduced, power modes can be implemented by observing how a model is driven. The code snippet in Figure 2 illustrates how a generic block with three power modes, {OFF, ON_LP, ON_HP}, is implemented. It takes into consideration the supplies (supply_ok signal), references (ref_ok), the enable (en signal) and its delayed version (signal en_dly, asserted after en). Provided the supplies and references are at valid range and the enabled pin is asserted, it will enter low-power mode (ON_LP), reaching high-power mode (ON_HP) after a certain time (after internally generated signal en_dly is asserted). This is a simplistic and combinatorial way of modelling power modes. More complex IPs may require sequential logic or even a state machine, as discussed in [2] and in [5]. The advantage of using power modes instead of directly using numeric power values is that one can map a power mode to different Iq values, such as those from the specification or simulation results over different corners.

Power modes need to be made available outside the model's scope through out-of-module-reference (OOMR). We have chosen to use virtual interfaces, which are made accessible to other testbench components via uvm_config_db, a UVM method to store and retrieve references. In this way, each model 'publishes' its power modes, which can even be visualized in a simulation waveform viewer.

```
blk_if pwr_if();              // Interface to "publish" the current power mode

initial begin                 // Make interface accessible to other components
  uvm_config_db #(virtual interface blk_if)::set(null,"*",blk_vif,pwr_if);
end

always_comb begin             // Definition of power modes
  if      (supply_ok & ref_ok & en & !en_dly) pwr_if.pwr_state = ON_LP; // low-power
  else if (supply_ok & ref_ok & en &  en_dly) pwr_if.pwr_state = ON_HP; // high-power
  else                                        pwr_if.pwr_state = OFF;   // disabled
end
```

Figure 2. Example of power modes implementation of a model.

Once the power states of all models are implemented, they are consolidated in a central testbench component. A DMS top-level testbench component connects itself to all virtual interfaces and performs the mapping between power modes and actual power consumption (Iq). Figure 3 illustrates an example of this mapping using a look-up-table (LUT); with figures are in µA. For each power mode there are three Iq values available: two based on analog block-level simulation results, fast corner (FF) at 35C and 75C, as well as the maximum values defined in the specification. A default value of 1A is used to help identify mistakes in the LUT. If analog block-level simulation results are not available, this LUT may initially contain only specified values. Once analog simulation results become available, the LUT can be expanded to include any required corner.

```
forever begin
  @(blk_vif.pwr_state);
  blk_iq =                          // FF @ 35C      // FF @ 75C      // spec value
    (blk_vif.pwr_state == OFF  ) ?  ff35 ?  75e-3 :  ff75 ? 447e-3 :    1 :
    (blk_vif.pwr_state == ON_LP) ?  ff35 ? 305    :  ff75 ? 321    :  350 :
    (blk_vif.pwr_state == ON_HP) ?  ff35 ? 407    :  ff75 ? 415    :  450 : 1e6;
end
```

Figure 3. Example of power modes mapped to Iq values: a) analog simulation at FF corner and 35C, b) analog simulation at FF corner and 75C and c) specified values.

The power modes of all models will comprise a chip-level power map, which can be visualized graphically in a waveform viewer tool, as exemplified in Figure 1.c. The power map does not provide absolute power figures, but it serves as a valuable source of information that chip architects can use to validate the initial concept and identify issues, e.g., a high frequency oscillator mistakenly left enabled when the chip is supposed to be in ultra-low power mode. It is also useful for verification engineers, who can simulate complex or random configurations and observe if the specification holds.

In DMS simulations, the estimated power is not an electrical discipline but just a floating-point variable (real number in System Verilog), which can be used in traditional digital verification checkers, such as assertions or inside scoreboards, or even in functional coverage. The implementation of checkers or coverage using estimated power is beyond the scope of this paper but nothing prevents them from being implemented.

III. RESULTS

This methodology was successfully applied to a mixed-signal design with multiple power rails (power domains) and several analog IPs. Figure 4 shows DMS simulation results of the power states of six models (BLK1 … BLK6) along with the overall estimated Iq for a particular rail. The names of the models were altered for confidentiality reasons and clarity. The simulation was done using Cadence® XCELIUM [6] and waveforms observed with Cadence® Simvision [6].

As seen in Figure 4, power consumption is constant during power states, and the transitions show no glitches or overshoots. This is expected, as each power mode is mapped to a single value. As stated earlier, the goal of this method is not to replace accurate AMS simulations but provide a fast and early way of identifying analog blocks that are wrongly activated. If required, power modes can be further refined by adding transitory and steady state power modes. The question is whether the increasing modelling complexity effort proves to be effective. The advantage of DMS simulations over AMS is that the former uses a fast event-driven simulator, whilst the latter uses a much slower continuous-time solver. Simulation time difference between DMS and AMS is typically of a few orders of magnitude (minutes or hours in DMS, compared to days or weeks of full schematic AMS). DMS also allows random configurations that would otherwise not be tested in AMS. If a particular random DMS simulation raises doubt concerning a certain scenario, that random seed can be re-simulated in AMS, making more effective use of computing power and tool licenses.

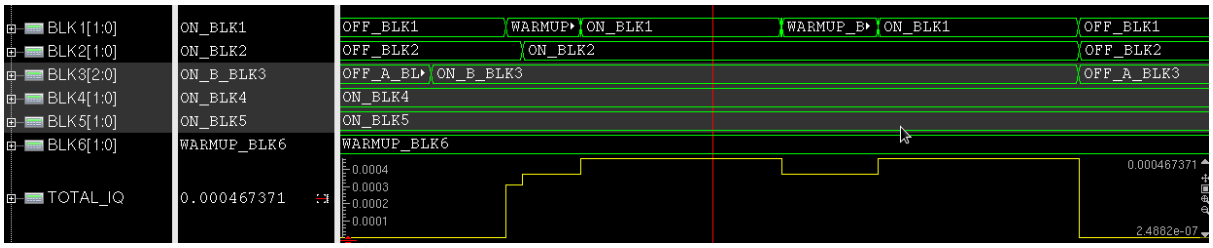

Figure 4. DMS simulation results showing the power states of models and total estimated Iq.

DMS results from Figure 4 were re-simulated in AMS with analog schematics replacing all models. Both results are shown in Figure 5 for two different rails (Rail 1 and Rail 2). Besides a few current spikes observed in AMS, the overall shapes of both DMS and AMS results are similar, and transitions between operating modes could be clearly identified.

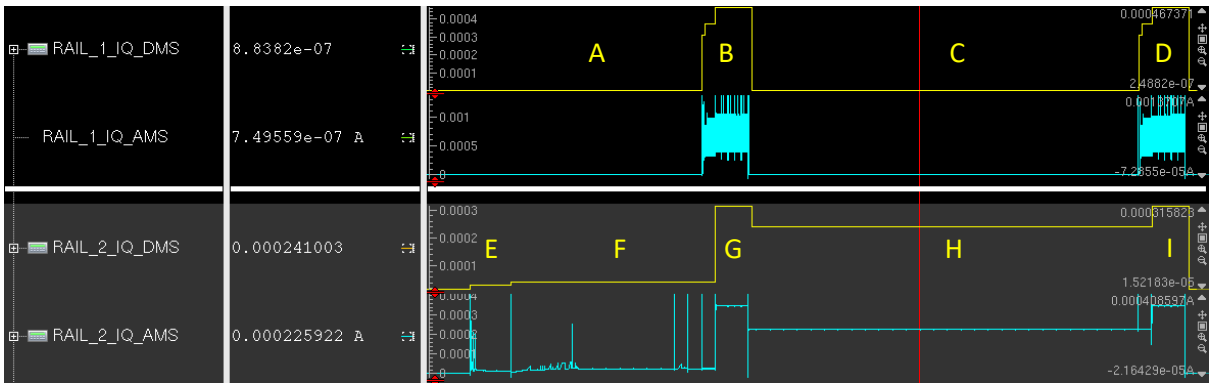

Figure 5: Results of same test simulated in DMS and AMS, showing the Iq of two different rails (Rail 1 and Rail 2) and 9 different regions of interest (A … I).

Iq results from Figure 5 were averaged for the nine regions of interest of Rail 1 (A … D) and Rail 2 (E … I) and shown in Table 1. In addition to the average Iq, the logarithmic gain (in decibels) was calculated for each transition between two consecutive regions, e.g. from A to B or from B to C.

The absolute Iq values are closely related in a few regions, such as 0.25 µA for DMS and 0.29 µA for AMS in region A, as well as 24.1 µA for DMS and 22.6 µA for AMS in region H. In other regions they deviate a bit more, for instance, in region B, where Iq is 441 µA for DMS and 344 µA for AMS. However, the gains between two consecutive regions are highly consistent. For example, the transition A → B → C for both DMS and AMS show, respectively, 64.4 dB and 61.5 dB for transition A → B, and as -53.9 and -53.2 dB for B → C. If a particular IP with dominating power consumption is incorrectly enabled or disabled in a given scenario, it is very likely that it will be noticed with this method.

Table 1. Comparison of Iq estimation (absolute values and logarithmic gain) from DMS and AMS simulations of rails 1 and 2.

	Rail 1				**Rail 2**				
Average Iq (µA)	***A***	***B***	***C***	***D***	***E***	***F***	***G***	***H***	***I***
DMS	0.25	441	0.84	414	29.7	41.1	315	24.1	315
AMS	0.29	344	0.75	341	19.3	27.1	349	22.6	349
Gain (dB)		***A → B***	***B → C***	***C → D***		***E → F***	***F → G***	***G → H***	***H → I***
DMS		64,4	-53,9	53,9		2,8	17,7	-2,3	2,3
AMS		61,5	-53,2	53,2		2,9	22,2	-3,8	3,8

IV. CONCLUSIONS

This flow offers the ability to find power intent issues in the early stages of the design phase, where adjustments and even architectural changes can be done with less impact on the project schedule. This is particularly relevant in projects with shorter time cycles, where the entire pre-Silicon phase is counted in months, rather than years. Although it does not aim at mimicking exact analog current values, especially in transient periods, it does provide a current profile that gives a good indication of the power consumption across various modes of operation. The fact that it relies entirely on model-based digital simulations (with no analog solver required) allows complex scenarios or even random configurations to be exercised. This method can also be used to identify specific scenarios and optimize test patterns in DMS before launching extensive AMS simulation close to the sign-off phase of a project.

REFERENCES

[1] X. Pan, C. Grimm, "Dynamic Simulation of Power Consumption for Model Based Design of Mixed-Signal Systems", Frontiers in Analog CAD, July 21.22, 2017

[2] B. Fischer, C. Cech, H. Muhr, "Power Modeling and Analysis in Early Design Phases", Design, Automation & Test in Europe Conference & Exhibition, 2014

[3] E. Lauwers, G. Gielen, "Power Estimation Methods for Analog Circuits for Architectural Exploration of Integrated Systems", IEEE Trans. VLSI Systems, vol. 10, no. 2, April 2002

[4] Universal Verification Methodology (UVM) 1.2 User's Guide, accellera Systems Initiative, October 2015, https://www.accellera.org/images/downloads/standards/uvm/uvm_users_guide_1.2.pdf

[5] A. Suissa, O. Romain, J. Denoulet, K. Hachicha, P. Garda, "Temperature and Supply Voltage Aware Power Modeling of Analog Functions at System Level", International Conference on Design & Technology of Integrated Systems in Nanoscale Era, 2010

[6] Cadence Design Systems Inc., www.cadence.com